

EE3L11

Motion Reference Unit Testing Platform

MRU assessment

Bachelor Thesis

in partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical Engineering

at Delft University of Technology
Faculty of Electrical Engineering, Mathematics, and Computer Science

by

Group B:

Casper van der Horst, 4472136

Martijn van der Marel, 4464397

Supervisors:

dr. ir. Rob Remis - TU Delft

dr. ir. Frank Nieuwenhuizen - Ampelmann Operations B.V.

Abstract

In order to make offshore working conditions safer, Ampelmann B.V. builds platforms that stay stable by compensating for a ship's motion. These platforms are controlled via motion reference units (MRUs). In order to test the performances of these MRUs, a linear motion MRU test setup was developed. The system was divided into three parts: hardware, software and MRU assessment. This thesis focuses on the design and implementation of the MRU assessment part and on testing MRUs with the complete prototype setup. Ampelmann supplied the MRU with which the test were conducted. Validation tests have been done with a simulator and with the setup itself. Included in the assessment of the MRUs was the latency, signal-to-noise ratio and low-frequency behaviour, compared to a ground truth. Concluded from the tests was that the designed test setup has potential to accurately assess the performance of MRUs. There are some minor improvements to be made, but the overall system works as intended.

Preface

This thesis is written as part of the graduation project of the bachelor Electrical Engineering at Delft University of Technology. For this project, a cooperation with Ampelmann Operations B.V. ("Ampelmann") was set up to provide the project to be worked on, as well as the necessary assistance and supervision.

Ampelmann is a company developing offshore access systems for people and cargo. Due to sea motion, transfer from ships to offshore structures and back is usually a difficult and sometimes dangerous task. Ampelmann has developed a motion-compensating platform to make this process easier and safer.

The project is divided into three parts: hardware, software and MRU assessment. A separate thesis is written for each of these parts. This thesis will focus on the MRU assessment.

We would like to thank Frank Nieuwenhuizen for being our project coordinator at Ampelmann, and Rob Remis as coordinator from TU Delft. We would also like to thank Alexander Verweij, Suzanne Weller, and Niels van der Geld who provided guidance for the three parts of the project. Further thanks go to the other Ampelmann employees who kindly answered our questions.

Contents

1	Introduction	4
1.1	State of the Art	4
1.2	Problem definition	4
1.3	Thesis outline	5
2	Program of requirements	6
2.1	Functional requirements	6
2.2	Non-functional requirements and limits	6
3	Design and Implementation	8
3.1	Wave point generator	9
3.2	Data processing	11
3.2.1	Latency	12
3.2.2	Error	12
3.2.3	Signal to Noise Ratio	13
3.2.4	Drift power percentage	13
3.3	Simulator	14
3.4	User Interface	14
4	Validation	16
4.1	Simulator tests	16
4.1.1	Latency	16
4.1.2	Signal to Noise Ratio	16
4.1.3	Drift percentage	16
4.1.4	Errors	17
4.2	Prototype tests	17
5	Results	18
5.1	Simulator tests	18
5.1.1	Latency estimation	18
5.1.2	SNR estimation	20
5.1.3	Drift power estimation	21
5.1.4	Error estimation	21
5.2	Prototype tests	23
6	Discussion	25
6.1	Simulator tests	25
6.2	Prototype tests	25
7	Conclusion	27
7.1	Function block evaluation	27
7.2	Prototype test	27
7.3	Future work	27
7.4	Recommendations	27
7.5	Ethical considerations	28
	References	30
	Appendices	32
A	MRU test data	32

B WPG flow diagram

33

1 Introduction

This section introduces the part of the project related to the assessment of motion reference units (MRUs). First, the state-of-the-art of MRUs will be briefly covered. Then, the problem definition will be introduced, followed by an outline of the rest of the thesis.

1.1 State of the Art

The main use for high-end motion reference units is in the (sub)marine navigation or marine motion compensation industries. MRUs measure six degrees of freedom (6DOF) relative to a ship: three translational (surge, heave and sway) and three rotational (yaw, pitch and roll), as shown in Figure 1.

Since the 1990's, the accuracy of motion reference units has gone up to approximately 0.01° pitch and roll angles and 1 cm for heave motion [2].

The main challenge for modern MRUs is in accurately measuring low-frequency motions in real-time [3]. This is caused by the fact that most MRUs use accelerometers for measuring translational motion. Low-frequency motion also has low accelerations, which are difficult to measure for accelerometers. Furthermore, the integration from acceleration to position also integrates measurement errors, leading to an unwanted effect known as integration drift [4].

MRUs usually have internal filters that filter out unwanted frequencies, including drift. These filters are often adapted based on the application: For a wave compensation application, other filters are used than for a submarine application. One MRU usually has different operating modes, each of which applies different filters.

1.2 Problem definition

Ship motion compensation requires accurate motion sensors. Due to ethical aspects regarding the safety of users and the high cost of commercial motion reference units, it is desirable to test these MRUs under controlled circumstances before using them in applications. In that way their performance can be compared against certain benchmarks or other MRUs.

Within the context of designing a complete test platform for evaluating MRUs, this thesis focuses on MRU assessment. This includes determining and generating input wave motions to the test system and processing data the test system produces. Eventually this will provide numerical results signifying the performance of an MRU. In addition, both the generation of the input and the processing of the output need to be validated independently of the rest of the system. For this purpose a simulator is needed which simulates the rest of the system. Lastly a user interface (UI) is required to provide an easy human-machine interface for running tests. This allows the user to both determine the input of the system and view the results.

The main purpose of the thesis can therefore be defined as follows: To develop software that can be used to evaluate the performance of motion reference units by allowing a user to choose inputs to a linear motion

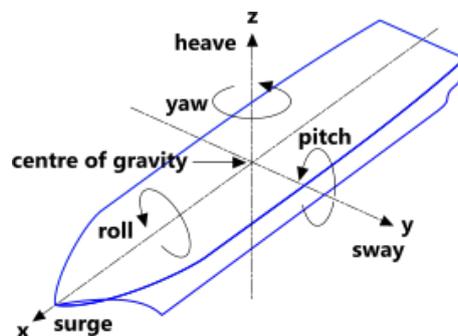


Figure 1: The six degrees of freedom of a ship on sea. [1]

test system and returning relevant performance indicators of the motion reference units.

1.3 Thesis outline

This thesis is structured as follows: First the requirements are specified in the program of requirements (2). Then the system design and implementation are explained (3). This implementation is validated with both simulator and prototype tests with the test system and an MRU (4). After that, the MRU assessment results (5) are discussed (6), followed by the conclusion and recommendations for further work (7).

2 Program of requirements

This chapter outlines the requirements for the assessment of MRUs. First, the functional requirements are listed. These will be followed by the non-functional requirements and limitations.

2.1 Functional requirements

- The wave motions used as input to the test system should be within the range of typical North Sea conditions. Typical in this context is defined as at least 90% of all waves, with 5% or less outliers on each side of the given range.
 - Typical wave periods for the North Sea: $T_s = 3-7$ seconds [5] [6].
 - Typical significant wave heights for the North Sea: $H_s = 0-2.5$ meters (trough to crest) [5] [6].
- Sway, surge and heave motions should be able to be tested.
- MRU performance should be tested against a ground truth. This ground truth is the MRU position as indicated by a Tamagawa H48 incremental rotary encoder [7].
- The MRU performance quantities that need to be estimated are the following:
 - Latency: The latency is the time difference between the ground truth and the MRU output. The resolution of the latency measurement should be in the order of milliseconds.
 - Error: After compensating for latency, the MRU output should be compared to the ground truth to find four error indicators (in meters):
 - * Root mean square error e_{rms}
 - * Maximum absolute error e_{max}
 - * Standard deviation of the error σ_e
 - * Mean of the error μ_e

These indicators are chosen because of their relevance for the eventual use of MRUs in the Ampelmann system. It should also be noted that the same indicators have been used in other academic papers [8] [9] to assess MRU performance for sea motion compensation applications.
 - SNR: The signal-to-noise ratio of the MRU output needs to be given in dB.
 - Drift: As indicated, MRUs have difficulty detecting waves with very low frequencies. The error caused by these low-frequency motions is called drift. The drift power will be defined as power in the frequency band $0 - 0.01Hz$. The ratio of drift power between the MRU and the encoder is used to indicate the drift.

2.2 Non-functional requirements and limits

- Tests should be able to run continuously for 24 hours at most. These long tests may consist of multiple repetitions of shorter wave motions.
- The system has to be usable with different types and brands of MRUs.
- Any filter that is used during the data processing should be implemented as a real-time filter, to simulate how filtering the actual MRU on an Ampelmann system would work.
- The complete test system should be able to be started and stopped at any time via a pc, or when an external emergency button is pressed.
- For safety reasons, there are several limitations on the wave signal sent to the test system:
 - No wave motions with an amplitude above 78 cm may be used. This is based on the length of the rail (186 cm) with a safety zone of 15 cm on each side.

2 PROGRAM OF REQUIREMENTS 2.2 Non-functional requirements and limits

- When sampling the wave points the Nyquist rate has to be satisfied. With the typical wave periods this would result in a minimal sampling frequency of at least 0.7 Hz.
 - The maximum velocity of the cart may not exceed 4.4 m/s.
 - The motor power output or input (in the case of decelerating) may not exceed 250 W.
- The maximum weight of the cart + MRU is 5 kg.
- The digital storage required to save the input wave signal should be kept to a minimum.
- Concerning the software environment:
 - MATLAB is preferred, but only the standard MATLAB toolboxes can be used.
 - Octave is an alternative to MATLAB which supports most functions that are limited to a non-standard toolbox in MATLAB. However, Octave is limited in documentation and generally has fewer functions available compared to MATLAB.
 - Python is also an option, but it is not particularly built for mathematical applications and has a steeper learning curve.
- All written software should be well documented, so future work can be done on the project.
- When replacing the MRU, the user should not have to change the source code except when absolutely necessary.

3 Design and Implementation

The MRU assessment part of the complete MRU test system can be divided into three function blocks: Wave point generation, data processing and the UI. Additionally, for testing purposes, a fourth block is developed: A simulator to work as a stand in for the software and hardware parts of the test system. The block diagram can be seen in Figure 2.

The functions called by the UI use a set way of communicating with the UI and with other functions. The inputs of the functions are values/settings that the UI asks from the user. As an output, the functions will return the string 'success' if the function completes successfully or an error message if something went wrong. In order to send data to other functions, each function will save relevant variables in a .mat file and other functions will be able to load these files if they require the data. By transmitting data in this manner, it is easy to change which variables are sent without having to change higher level functions. Additionally, functions can be replaced easily without influencing the rest of the system.

- Wave point generator: This function block generates the wave sequence that will be executed by the test system. It saves a time array with corresponding locations, as well as the set-point frequency. The set-point frequency of this sequence is partly determined by the rest of the test system. The motor is controlled at 1 kHz. However a set-point frequency this high is probably not necessary, especially since the wave motions are of a very low frequency (0.1-0.3 Hz). Aliasing therefore will not occur for any reasonable set-point frequency. A high set-point frequency will however mean the PC spends more time transmitting the data, which could result in more possible conflicts while transmitting and logging in real-time.
- Data processing: This function block uses the time series of the MRU and the motor encoder. It processes these sequences to produce quantities that signify the performance of the MRU. These quantities are mentioned in the technical requirements. The data processing is not done in real-time, so there are no strict limitations on computation time. This block should be able to take as input time series from the MRU and the motor encoder that do not have the same sampling frequency. Additionally, the time series do not necessarily have uniform time intervals.
- UI: This is the interface with the user. Therefore, it should be easy to understand and simple to work with. It should both be able to take user inputs for the system and receive and display the results of the data processing block. It acts as the top-most function for the entire system, controlling all other functions. The UI should have the functionality to stop the system safely at any time.
- Simulator: The main function of this block is to allow testing of the previously described blocks. When the complete test setup is not available, this simulator can take its place and provide a similar response as the real system would. This means introducing some latency, noise, possibly bias or other distortion. Also, the signals may need to be resampled. Most likely, the real system needs to be studied to find typical values for latency, noise etc.

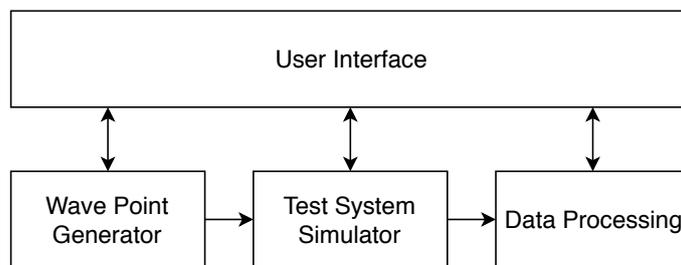


Figure 2: Block diagram of the MRU assessment part of the test system. Vertical arrows represent control signals, and horizontal arrows represent data signals.

3.1 Wave point generator

The wave point generator (WPG) generates the set-points that will control the motor driver of the MRU test setup. This signal consist of a time sequence t and a position sequence $s(t)$, used to control the cart. The WPG also generates velocity $v(t)$ and acceleration $a(t)$ sequences, these are used to run safety checks. The $s(t)$, $v(t)$ and $a(t)$ sequences that the WPG generates are periodic, this is done in order to decrease the storage that is required to save the wave points. Learning from the experience of Ampelmann, the sampling frequency of the WPG was set to 50Hz, as this would allow for smooth motion control of the motor. The WPG works in three stages: generation, verification and saving. A flow diagram of the wave point generator can be found in Appendix B.

In the first stage the base signal is loaded or generated. This stage can operate in one of three modes, dependent on the inputs. Firstly, this stage can create a signal out of a series of sine waves that are superimposed onto one another. The user specifies the desired amplitude and period for each sine wave. The superposition of these waves results in the position array $s(t)$. Using the sum rule of differentiation, stating that the derivative of a sum is the sum of the derivatives, and the known differential forms for a sine, the velocity and acceleration arrays are calculated. In order to decide the length of one (periodic) interval, the least common multiple of the wave periods was computed, since that is equal to the fundamental period of the signal.

The second mode uses data from other software. Ampelmann uses a MATLAB program called AWESOME to simulate their platform's performances. This program is also able to simulate the ship motions that are of interest for the MRU tests. The WPG is able to read the files created by AWESOME to use in an MRU test. The user specifies the interval that will be used during the test and the WPG ensures the periodicity of the signal by adding a three second window in which the cart moves from the end position back to the starting position. To avoid discontinuities at the borders of signals when generating the $v(t)$ and $a(t)$ arrays, the $s(t)$ signal was elongated to include three periods. By calculating the velocity and acceleration for this elongated signal and then cropping it back to only the middle period, the periodicity of the velocity and acceleration arrays was assured. Before cropping, the signal is also resampled to the desired 50Hz, this is done at this stage because the non-zero edges of the signal tended to oscillate when resampled.

When in the third mode the WPG reads the data from a .mat file. This file could either contain a signal that was previously made by the WPG or it could contain a custom signal. These signals contain the position and time arrays and the sampling frequency. The WPG will then make the signal periodic and generate the velocity and accelerations arrays in a similar fashion as in the second mode. If the sampling frequency of the signal is not equal to 50Hz, then it will be resampled.

The second stage is used to verify that the generated wave points don't exceed the boundaries of the test rail and motor, as stated in the program or requirements. This stage tests the signal against three parameters: the maximum deviation from the centre of the rail, the maximum velocity and the maximum power output of the motor. If any of these limits is exceeded, the WPG will return an error and the test cannot be started. The cart is not allowed to move off the track. Therefore if the cart gets within a safety region of 15 cm from the edge, it will come to a standstill. The WPG checks the largest deviation from the centre, to prevent the user from sending a signal that would send the cart past the border. If the maximum absolute value of $s(t)$ reaches a value above 0.78m, then the signal will be rejected.

In order to calculate the maximal power output, $F = m \cdot a$ is substituted into $P = F \cdot v$, this results in formula 3.1. In order to verify this condition the velocity v and the acceleration a for the same time instance are multiplied and compared to the maximum power output divided by the maximum mass of the payload. If the track is positioned vertically, earth's gravitational acceleration g has to be added to the acceleration of the cart. These conditions result in the following relations, 3.2 (horizontal) and 3.3 (vertical), that are used to test that the maximal power output is not exceeded. When decelerating, the power would be negative. This negative power is not allowed to exceed $-P_{max}$, hence the absolute value of $a \cdot v$ is used.

$$P = m \cdot a \cdot v \quad (3.1)$$

$$\frac{P_{max}}{m_{max}} > |v(t)a(t)| \quad (3.2)$$

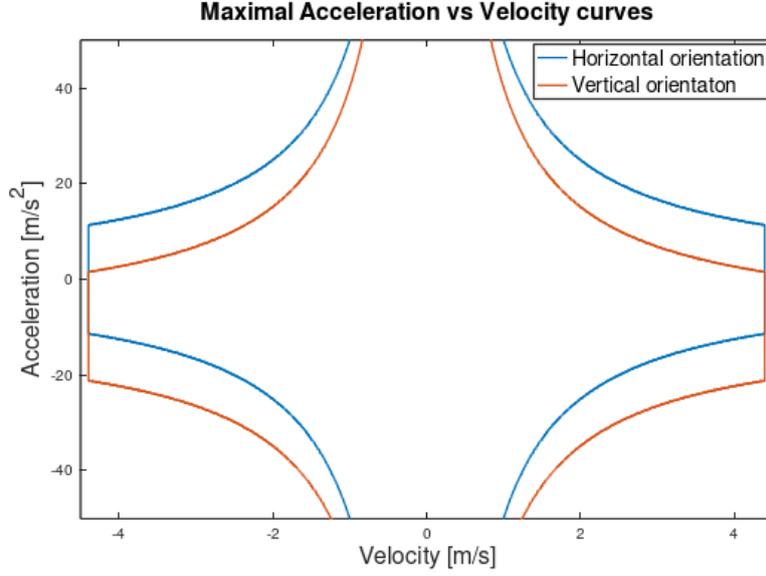


Figure 3: Plots of the maximal acceleration and deceleration for different velocities for both the horizontal and vertical orientations of the track.

$$\frac{P_{max}}{m_{max}} > |v(t)(a(t) + g)| \quad (3.3)$$

Equations 3.2 and 3.3 can be used to graphically show the region the velocity and acceleration are allowed to be in during the test, see Figure 3.

Since most of the waves that will be tested are sinusoids, and because the position, velocity and acceleration equations are known (3.4a - 3.4c), it is possible to calculate the maximum frequency for the setup.

$$s(t) = A \sin(2\pi ft) \quad (3.4a)$$

$$v(t) = A \cdot 2\pi f \cos(2\pi ft) \quad (3.4b)$$

$$a(t) = -A(2\pi f)^2 \sin(2\pi ft) \quad (3.4c)$$

Substituting the velocity and acceleration functions (3.4b and 3.4c) into 3.2 results in function 3.5. The maximal value (for t) is reached when $\sin(2\pi ft) = 1$. Using this, the maximal frequency can be determined with equation 3.6. Filling in the track requirements from 2.2 ($P_{max} = 250$ W, $m_{max} = 5$ kg and $A = 0.78$ m) gives a maximal wave frequency of 0.87 Hz or a minimal wave period of 1.15 seconds. Since the system is meant to simulate typical wave periods of 3-7 seconds, there shouldn't be any problems with the power output of the motor.

$$\begin{aligned} \frac{P_{max}}{m_{max}} &> | -A^2(2\pi f)^3 \cos(2\pi ft) \sin(2\pi ft) | \\ &> | \frac{-1}{2} A^2(2\pi f)^3 \sin(4\pi ft) | \end{aligned} \quad (3.5)$$

$$f_{max} = \frac{1}{2\pi} \sqrt[3]{\frac{2P_{max}}{A^2 m_{max}}} \quad (3.6)$$

The maximal velocity depends on both the maximal velocity of the motor and on the distance the cart is from the end of the track. Following the hardware limits of 2.2, the maximal speed of the track is set to 4.4 m/s. The maximal power equation 3.1 shows that the motor can't decelerate at an infinite rate. This

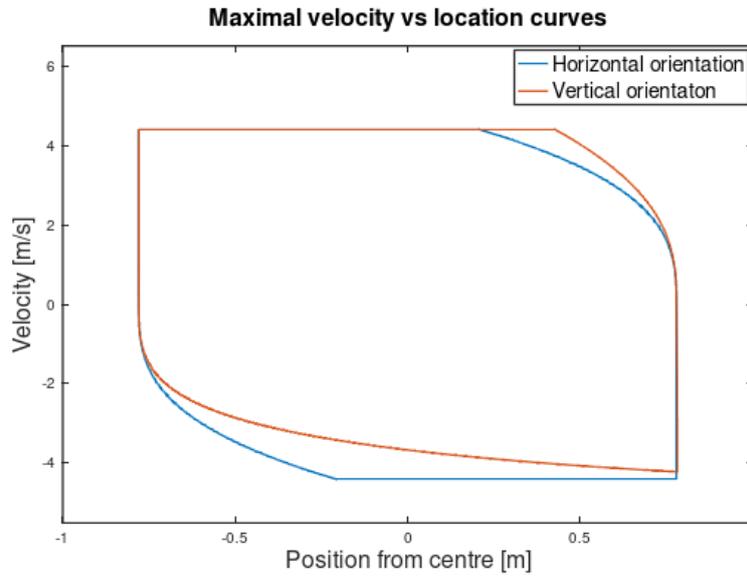


Figure 4: Plots of the maximal velocity for different positions on the track for both the horizontal and vertical orientations.

means that the maximal velocity towards the end of the track is limited due to the maximal deceleration 3.7.

$$a = -\frac{P_{max}}{m_{max} \cdot v} \quad (3.7)$$

Because the cart needs a certain breaking distance, the maximum velocity can be plotted against its position on the track, Figure 4. These curves are generated by starting at the maximal velocity of 4.4 m/s and calculating the maximal deceleration, distance travelled and end velocity during a short time interval. This process is repeated until the cart's velocity has become zero. The total distance travelled is then placed at the end of the track and the side of the curve with the maximal velocity is extended to the other end of the track. This procedure is repeated for a negative maximal velocity. When set to a vertical orientation, a positive velocity/position is defined to be pointed in a upward direction, changing equation 3.7 into .

$$a = -\frac{P_{max}}{m_{max} \cdot v} - g \quad (3.8)$$

Before going to the saving stage the WPG displays to the user how long a single period lasts and requests the user to input how often the period should be repeated. This number of repetitions is an integer larger than zero and is saved in the next stage.

During the last stage the signal is saved. In order to make interfacing with data transmitting (part of another thesis) easy, the signals are saved as a .mat file. Data transmitting can simply load the file. The saved file contains the position and time arrays for a single wave period, as well as the sampling frequency F_s at which the arrays are sampled and an integer n which states how many times the period has to be repeated during the test.

3.2 Data processing

The program of requirements lists the quantified performance indicators that need to be distilled from the incoming time-stamped sequences of the MRU position (mr_u) and the motor rotary encoder (me) as the ground truth. What is the strategy to find these indicators? The errors are computed using signals that are compensated for latency (synchronized). To synchronise these signals, the latency first needs to be

estimated. The magnitude plot obtained from the DFT is independent of the delay in the signal, so finding indicators such as the drift power percentage via the DFT can be done independently from the latency estimation.

3.2.1 Latency

There are several methods of measuring delay or latency between two signals. [10] The most useful method for this problem is by means of cross-correlation. The cross-correlation shifts two signals past each other and measures the similarity of the signals. Therefore to find the latency the strategy is to find how much the signal needs to be shifted until the cross-correlation is at its maximum, and translate this shift in samples to a latency in seconds.

Cross-correlation is the best method to find the latency in this setup because it is very robust: it takes into account all data points to find the similarity between two signals. Other methods of latency estimation use one specific event in both signals to find the delay. Outliers or noise can disturb these latency measurements. The cross-correlation method is furthermore independent of amplitude or units, making it even more robust. The resolution at which the latency needs to be determined is in the order of magnitude of milliseconds, as indicated in the program of requirements. Unfortunately the sampling frequency of the incoming signals may be smaller. The MRU has a sampling frequency of 50 to 200 Hz, and the motor encoder 1000 Hz. To get a resolution of 1 ms, the sampling frequency of both signals needs to be $1/t_{res} = 1/0.001 = 1000$ Hz. This means the MRU signal needs to be upsampled. This is done using linear interpolation.

After upsampling, the cross-correlation between the MRU signal and the motor encoder signal is computed and the latency is determined from the cross-correlation. The discrete cross-correlation that is used is defined as follows:

$$R_{mru,me}[k] = \sum_{m=-\infty}^{\infty} mru[m]me[m-k] \quad (3.9)$$

From this equation it can be seen that the cross-correlation shifts the two signals over each other and for each shifted version, the similarity is determined. The cross-correlation result $R_{mru,me}[k]$ thus peaks where the shifted signals are most similar. The amount of shifted samples k_{peak} therefore estimates the delay of mru compared to me . To find the delay in seconds, k_{peak} is divided by the sampling frequency $F_s = 1000$ Hz:

$$\text{Delay} = k_{peak}/F_s \text{ [s]} \quad (3.10)$$

3.2.2 Error

After the latency is found, the MRU signal is shifted to synchronise the MRU and motor encoder signals. The amount of samples shifted is the product of the latency (in seconds) and the sampling frequency (in Hz).

From the synchronised signals, four indicators are estimated (in meters):

- e_{rms} , the root mean square error. This gives an indication of the the average deviation of the MRU signal from the ground truth. The rms error is calculated using Equation 3.11.
- $|e|_{max}$, the maximum absolute error. This gives an indication of the worst outlier of the MRU signal compared to the ground truth over the entire signal. The maximum absolute error is calculated using Equation 3.12.
- σ_e , the standard deviation of the error. This gives an indication of how consistent the error is. The standard deviation is calculated using Equation 3.13.
- μ_e , the mean error. This indicates the constant offset of the MRU signal compared to the ground truth. If the mean error is zero, σ_e and e_{rms} are equal. The mean error is calculated using Equation 3.14.

$$e_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N (mru_i - me_i)^2} \quad (3.11)$$

$$|e|_{max} = \max(|mru - me|) \quad (3.12)$$

$$\sigma_e = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (mru_i - me_i - \mu_e)^2} \quad (3.13)$$

$$\mu_e = \frac{1}{N} \sum_{i=1}^N (mru_i - me_i) \quad (3.14)$$

In Equations 3.11-3.14, N is the length of the signals mru (the MRU output) and me (the motor encoder) in samples.

3.2.3 Signal to Noise Ratio

The signal to noise ratio is a very common indicator of the quality of a signal. It is defined as the ratio of the signal power to the noise power. In equation form:

$$SNR = \frac{P_s}{P_n} = \frac{(A_s)^2}{(A_n)^2} \quad (3.15)$$

With A_s and A_n the rms amplitudes of the signal and noise signals. To find the SNR, the signal and the noise need to be separated. Fortunately, a ground truth signal is available, and can be considered as the signal without noise. The noise signal can be found as the difference between the (latency-compensated) MRU signal and the ground truth.

With this strategy, the rms amplitude is easily calculated using Equation 3.16:

$$A_s = \sqrt{\frac{1}{N} \sum_{i=1}^N (me_i)^2} \quad (3.16)$$

For the noise amplitude, the equation becomes exactly Equation 3.11, giving $A_n = e_{rms}$. Substituting this into Equation 3.15 gives the following equation for the signal-to-noise ratio:

$$SNR = \frac{\frac{1}{N} \sum_{i=1}^N (me_i)^2}{\frac{1}{N} \sum_{i=1}^N (mru_i - me_i)^2} \quad (3.17)$$

3.2.4 Drift power percentage

A known problem of MRUs is their inability to measure very low frequencies. This is also known as drift. The cause is that accelerometers are used in MRUs. Very low frequencies also have very small accelerations, and accelerometers have trouble measuring these small accelerations.

To quantify the ability of the MRU to measure low frequencies, the energy in the very-low-frequency bands of the MRU signal and the ground truth signal is compared, and translated to a percentage.

This percentage is calculated by calculating the power spectral density (PSD) of both the MRU and rotary encoder signals. Then, the integral of the PSD is calculated over the frequency band of 0 to 0.01 Hz. The result is a measure of the power in this frequency band of the signal, denoted P_d . Since the discrete Fourier Transform is used, the integral is implemented as a summation.

The last step is to calculate the percentage of the MRU drift power compared to the ground truth ME drift power:

$$\text{Drift percentage} = \frac{P_{d,mru}}{P_{d,me}} \cdot 100\% \quad (3.18)$$

A percentage greater than 100% would mean that the MRU measures more low frequencies than there actually are, whereas a percentage between zero and 100 means the MRU does not measure drift frequencies that are actually there. Note that this indicator does not indicate anything relevant when no frequencies below 0.01 Hz are part of the input signal.

3.3 Simulator

The simulator should resemble the test system as accurately as possible. Ideally, a transfer function $h(t)$ would be known for both the motor encoder and the MRU. Unfortunately, this information is not available or retrievable via tests until the complete system is functional. Therefore, the simulator is built using information from data sheets and company experience at Ampelmann. It should be noted that the simulator does not just simulate the MRU and the motor encoder, but rather the entire test system from the moment the wave is generated to the moment the data is provided to the data processing functional block.

The MRU time series needs to be altered as follows:

- Resample to 50 Hz (the rate at which most MRUs can return values [11] [12] [2], and also the rate at which Ampelmann's system works).
- Add normally distributed noise to both the time sequence with $\sigma_t = 0.02$ ms (to simulate timing inconsistencies) [11] and the position sequence with $\sigma_N = 5$ cm (to simulate MRU and system noise) [11] [12] [2].
- Filtering the signal in order to simulate the inability to measure very low frequencies and the internal filters that remove high frequencies (like vibrations caused by the ships engines)
- Introduce a delay in the position signal in the order of milliseconds (to simulate latency of the MRU) [11].

The motor encoder time series needs to be altered as follows:

- Resample to 50 Hz (the rate at which the controller passes back motor encoder samples).
- Quantize the position sequence with a resolution of $44 \mu\text{m}$ (to simulate the discrete nature of the motor encoder) [7].

The transfer functions for the MRUs are not available, but based on data sheets [11], [12] and [2] and on research data [3], it became clear that a bandpass filter with cutoff frequencies at 0.01 and 10 Hz would be an appropriate estimation. Based on the experiences of Ampelmann, a second order Butterworth filter [13] was designed in order to simulate the MRUs.

The resampling of the signals is not used in the current setup of the system since both the input and output signals use a sampling frequency of 50 Hz, but remains in place to allow Ampelmann to simulate with different sampling frequencies for the set-points, the MRU or the encoder.

3.4 User Interface

In order to control the test system, a user interface was made. The first task of the UI function is to gather the necessary inputs from the user, most of which are used by the WPG. Secondly the UI acts as a control function, calling the other functions of the system. First the WPG is called with the gathered inputs. After which the system will start with the test. While running the test it is possible to stop the system by exiting the function. In that case the function will send a command to the motor controller to safely stop and it will save the data it has gathered. Lastly the UI will call for the data processing function discussed in the previous section. The data between the different main functions is handled by using saved workspaces. By predefining the location and names for the workspaces of the functions, the interfaces between these functions are both rigid enough to ensure the correct transfer of data and flexible enough to change the data

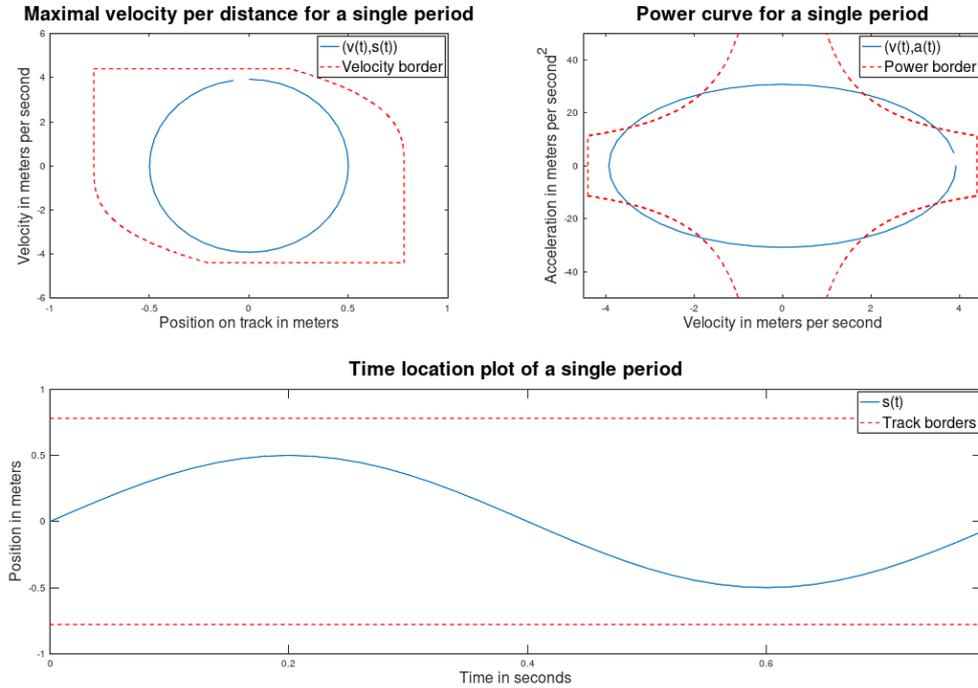


Figure 5: Plotted results for a sine wave that fails the check ($s(t) = 0.5\sin(2\pi/0.8 \cdot t)$)

that is interfaced between functions without the need to alter all the higher level functions (only the 'load' function on the sending function has to be altered). After the data is processed, the data from the WPG, the motor encoder and from the MRU together with the results from data processing are saved into a .mat file (using a format recognised by MATLAB). This is to allow the employees of Ampelmann to work on the raw data themselves. Lastly the UI handles errors from the lower level functions. When a function returns an error the UI tells the user where thing went wrong and is able to give other information to the user. When the generated signal does not meet the safety checks from the WPG, the UI will show a plot like in figure 5 (in this example breaking and accelerating would generate/use more power than the motor can handle).

4 Validation

To evaluate the performance of all function blocks under controlled circumstances, some validation tests were set up. The goal of these tests is to find the accuracy of the performance estimators knowing how exactly the signals were altered in the simulator. Additionally, the limits of the estimators is sought: At what noise level can the latency not be estimated accurately any more? And how long does a simulation have to take to estimate that latency? These are examples of questions that should be answered with these tests.

For all validation tests, the estimation resulting from the data processing function block is compared to what is added to the signals by the simulator. Since latency and noise are explicitly added by the simulator, the latency and SNR estimators can be tested without problem. For drift power and errors this is a bit more difficult because their effects are mostly caused by the bandpass filter in the simulator and/or indirect noise. Still, an effort is made to check the basic functionality of the drift power estimator.

After the simulator validation tests, tests are done using the prototype test system to be able to analyse real MRU and encoder data. These can not be properly used for the validation of the data processing function block as it is unknown how the signal is altered exactly in the MRU and the rest of the test system. However it is still useful to validate the design of the simulator, and to an extent the design of the test system. Finally, assuming that the wave point generator, the test system and the performance estimators function properly, these tests evaluate the MRU itself, which was the original purpose of setting up the system.

4.1 Simulator tests

4.1.1 Latency

What are the parameters that can influence the performance of the latency estimation? Firstly, the magnitude of the the latency itself may have influence. Secondly, a high noise level may disrupt the cross-correlation function and cause inaccuracies in the latency estimation. Lastly, the signal length also has influence. More data points means the cross-correlation becomes more accurate and therefore the latency estimation also becomes better. This could therefore mean that inaccuracies due to noise can be solved by using a longer signal.

The latency magnitudes that were tested for are in the range of 1 ms to 100 ms. For the noise levels a standard deviation range of 0 to 0.1 m was tested. Lastly, three signal durations were used for testing: 1 minute, 10 minutes and 100 minutes.

The input wave motion for the latency tests were sinusoids with an amplitude of 0.5 meters and a wave period of 5 seconds. These motions were chosen over other wave-model-generated motions for reproducibility reasons.

Due to the inherent randomness of noise, multiple tests need to be done when varying noise levels to obtain a sample sets from which probability distributions can be calculated. These distributions are then visualised through box plots. The minimum sample size for each test was set to 50 samples. This allows for a high certainty that the results obtained are valid.

4.1.2 Signal to Noise Ratio

The system simulator takes the standard deviation of the noise σ_N as input and adds normally distributed noise with this standard deviation to the MRU signal. Using the same sinusoid as in the latency tests, the 'real' signal-to-noise ratio can be found as $SNR = A_{rms}^2 / \sigma_N^2$ with $A_{rms} = 0.5m$ the amplitude of the sinusoid. This SNR is compared to the SNR found by the data processing function block. Because the bandpass filter in the simulator might influence the SNR, the test is done both with and without the filter.

4.1.3 Drift percentage

To test the basic functionality of the drift percentage estimator and the bandpass filter in the simulator, sinusoids with frequencies < 0.01 Hz were used as input to the simulator. The test was done both with and without the simulator filter, and the resulting drift percentages were compared. Ideally, with the filter the drift percentage estimator should return zero percent, and without the filter it should return 100%.

4.1.4 Errors

Like drift percentage, there is no direct parameter in the simulator that introduces error. Rather, the error estimators form a sort of summary of all the filter, noise, drift and distortion effects that may be present in the MRU. Still, to verify these estimators some simple tests need to be done. The most natural parameter to vary in the system simulator to check the workings of the error estimators is the noise. It is expected that the rms error, the error standard deviation and the maximum absolute error scale linearly with the noise standard deviation, in the case that filter, drift and distortion effects are not dominant. Under the same assumption, the mean error is expected to be zero, similarly to the mean noise value.

Besides the test with the standard 0.2 Hz 0.5 m sinusoid, another test is done using the same sinusoid with an additional DC offset of 0.1 m, to validate the performance of the mean error estimator. The rms error, maximum absolute error and the error standard deviation are also expected to be different in this second test.

4.2 Prototype tests

For the tests with the prototype test system including the MRU, similar wave motions are used as for the simulator validation tests: sinusoids with frequencies in the range of 0.2-0.5 Hz and amplitude 0.3 m. For the prototype tests, the motor controller is not yet fully in operation, and as such wave motions are generated in the software on the microcontroller, thus removing the wave point generator and data transmitting from the test loop. Because the motor controller is not yet functional, the test system can not compensate for drift yet. To prevent the cart with the MRU from drifting off the test rail, the duration of the test is still limited for this prototype. As a consequence, the amplitude of the wave motion was brought down from 0.5 m (used for most of the simulator tests) to 0.3 m, since this would allow for a longer test duration. At this amplitude, approximately 20 wave periods could be measured in a single test.

The MRU is placed on the test rail with the rail along the sway axis. The MRU is set to 'marine mode', similar to how it is used on Ampelmann systems. The incoming data of the MRU and the encoder (both at 50 Hz) are logged, time-stamped and saved in Octave for post-processing.

5 Results

The results of the validation tests as described in the previous section are presented here in the form of graphs and figures with some additional remarks commenting on what is visible in the graphs. Possible causes and/or consequences of interesting aspects of the graphs will be covered in the discussion.

First the simulator validation results will be covered, followed by the results of the tests with the prototype test system and the MRU.

5.1 Simulator tests

5.1.1 Latency estimation

As seen in Figures 6-9, a consistent positive error in the latency estimation is present which diminishes as the signal duration becomes longer. For $T = 60$ seconds, the mean latency estimation error is 7 milliseconds, whereas for $T = 600$ s it is 1 ms and for $T = 6000$ s the mean error is zero.

From Figure 6 it is also evident that this error does not scale with latency magnitude, but in fact remains constant. Because of the logarithmic scale of the figure, this constant error seems to be diminishing for greater latencies.

Figures 7,8 and 9 show that noise does have an influence on the latency estimation. It can also be seen that this influence decreases for longer signals. It should also be noted that the influence of noise only starts to show an effect at noise with $\sigma_N = 0.01$ m and higher.

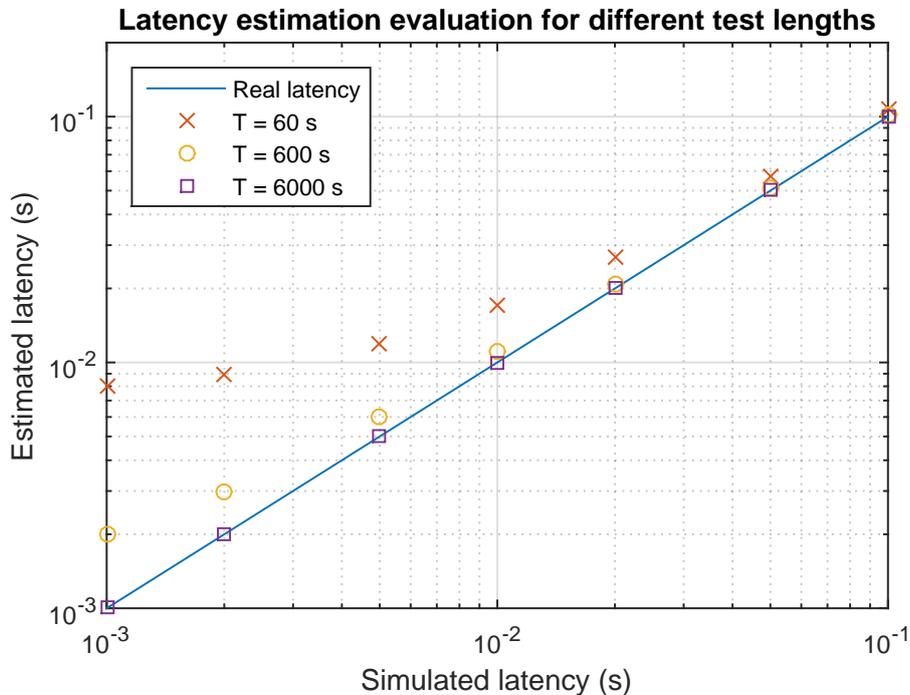


Figure 6: Latency estimation performance as a function of latency introduced by the simulator, using sinusoids ($f = 0.2$ Hz, $A = 0.5$ m) with a duration of 1, 10 and 100 minutes.

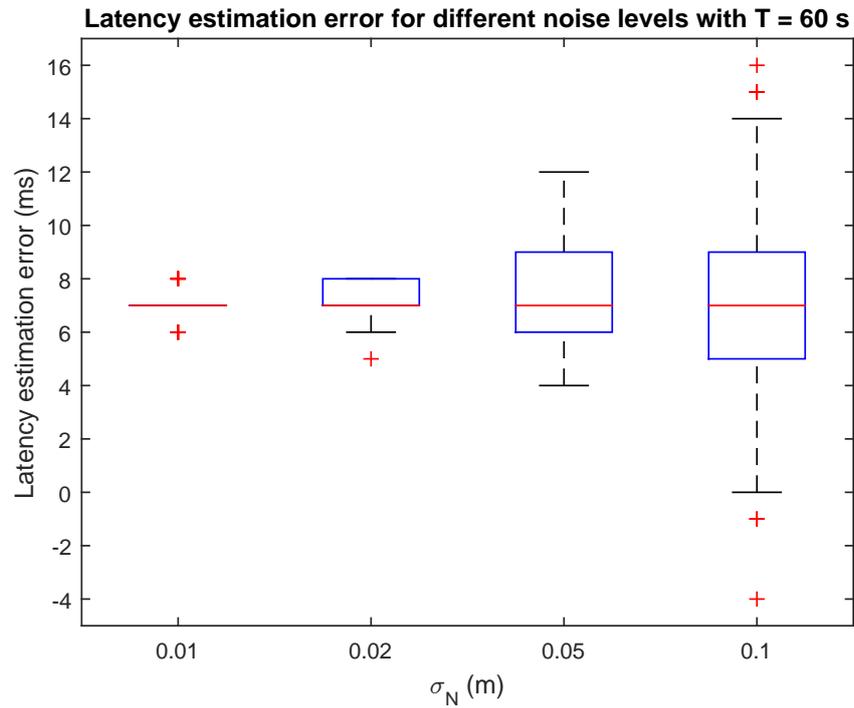


Figure 7: Effect of noise magnitude on the distribution of latency estimation errors, for a sinusoid ($f = 0.2$ Hz, $A = 0.5$ m) with a duration of one minute.

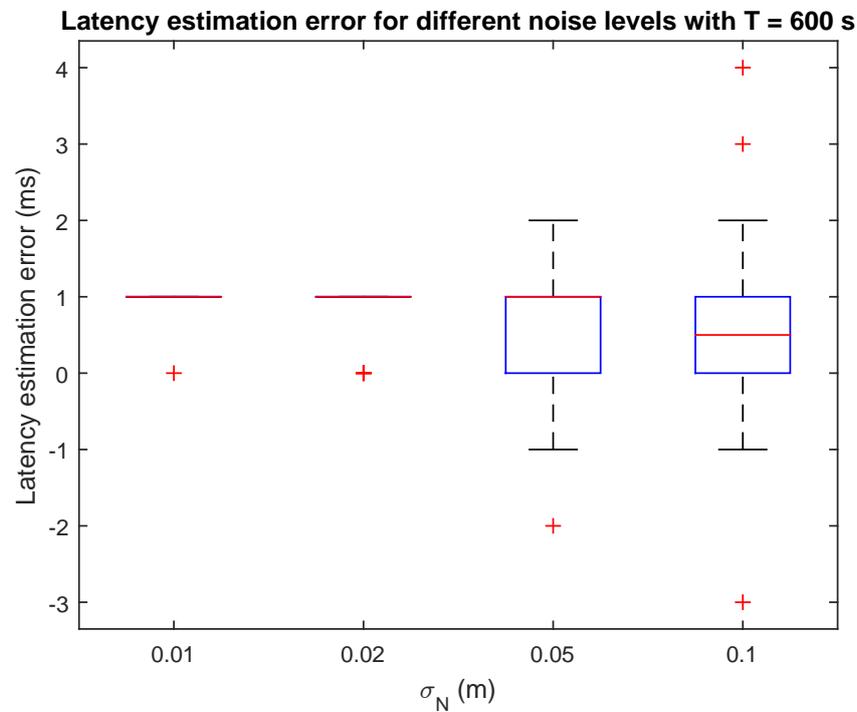


Figure 8: Effect of noise magnitude on the distribution of latency estimation errors, for a sinusoid ($f = 0.2$ Hz, $A = 0.5$ m) with a duration of 10 minutes.

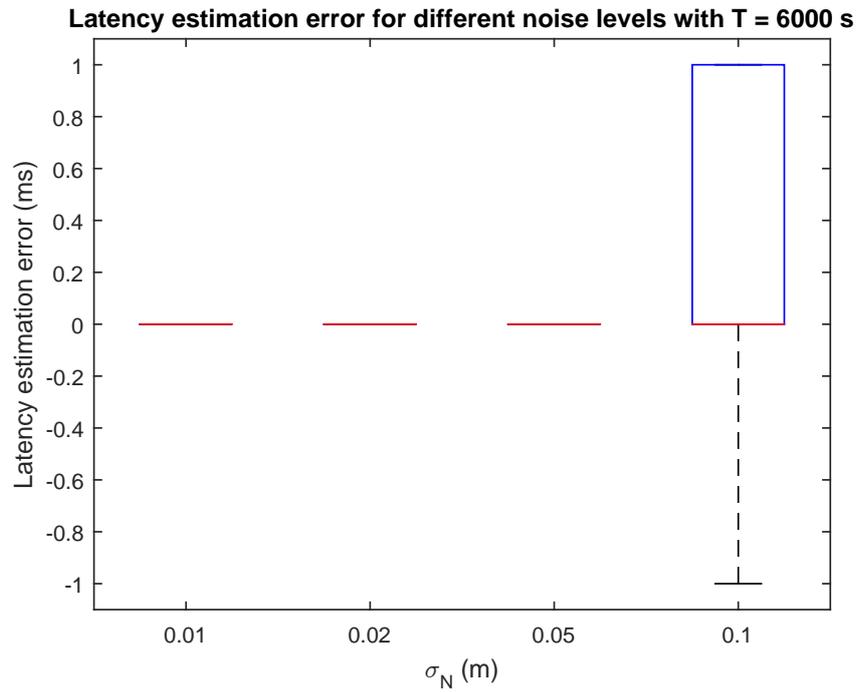


Figure 9: Effect of noise magnitude on the distribution of latency estimation errors, for a sinusoid ($f = 0.2$ Hz, $A = 0.5$ m) with a duration of 100 minutes.

5.1.2 SNR estimation

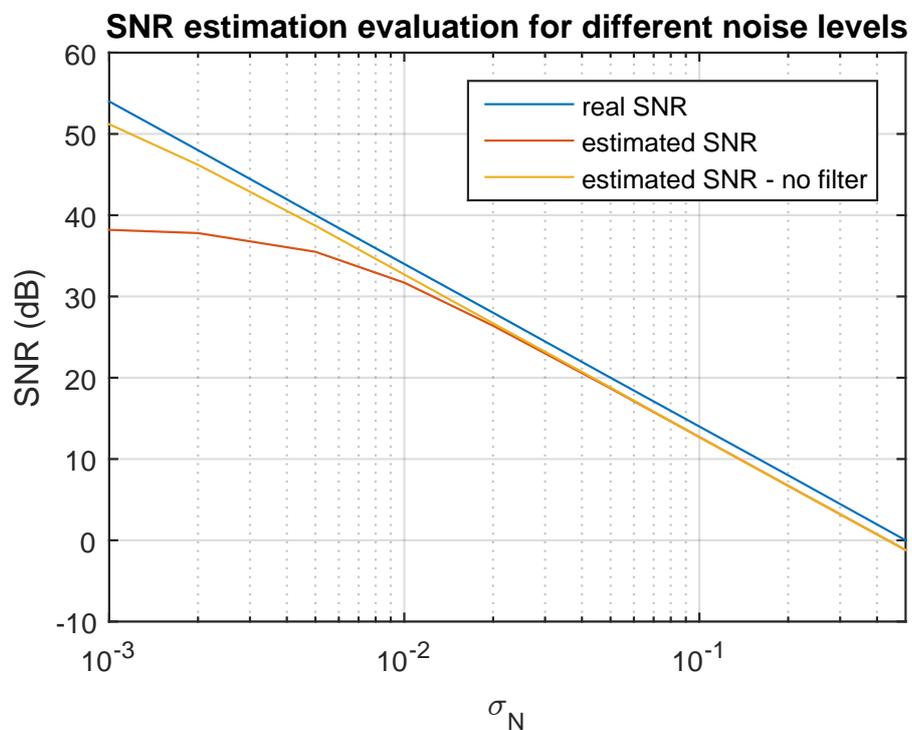


Figure 10: The real and estimated signal-to-noise ratio as a function of the noise level introduced by the simulator. Also shown is the estimated SNR when the bandpass filter in the simulator was turned off.

Figure 10 shows the estimated signal-to-noise ratio and the real SNR calculated from the signal power and the noise level introduced in the simulator. Interesting to note is that the estimated SNR is consistently approximately 1.3 dB lower than the real SNR.

Additionally, the estimated SNR does not exceed 40 dB, even when no noise is added to the signal. When the bandpass filter in the simulator is turned off, the estimated SNR does increase past 40 dB, but even so at very low noise levels the difference between the estimated and real SNR grows slightly from 1.3 dB to approximately 3 dB at $\sigma_N = 10^{-3}$ m.

5.1.3 Drift power estimation

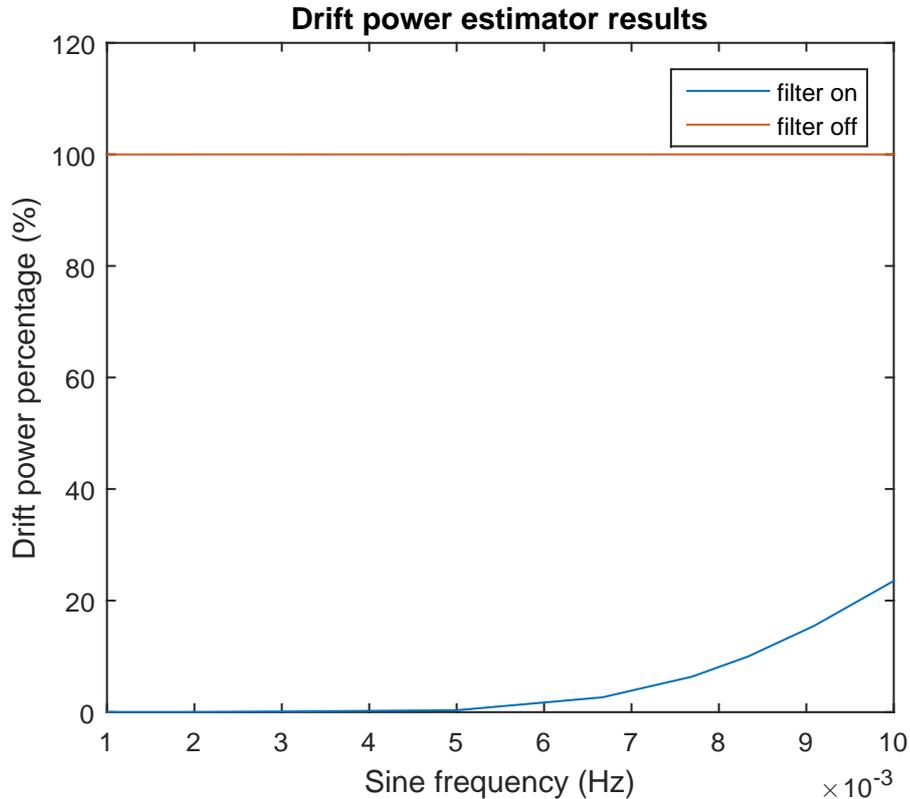


Figure 11: Drift power percentage estimation with and without the bandpass filter in the simulator, for very-low-frequency sinusoids.

With the filter off, the drift power estimation is consistently at 100%. With the filter, the frequencies below 0.01 Hz are filtered out and the estimator returns 0%, except for near the 0.01 Hz cutoff frequency of the filter, where the estimation goes as far up as 23%.

5.1.4 Error estimation

As can be seen in Figures 12 and 13, the mean error remains relatively constant for any noise level, and approaches the introduced DC offset of the input signal. With a DC offset of 0 (Figure 12), the rms error and the error standard deviation are equal, and for $\sigma_N > 10^{-2}$ they both increase linearly. The maximum absolute follows the same pattern but is approximately a factor 6-10 greater. For inputs with a DC offset however (Figure 13), the rms error and error standard deviation are not equal for low noise. The rms error instead seems to approach the maximum of the mean error and the noise level.

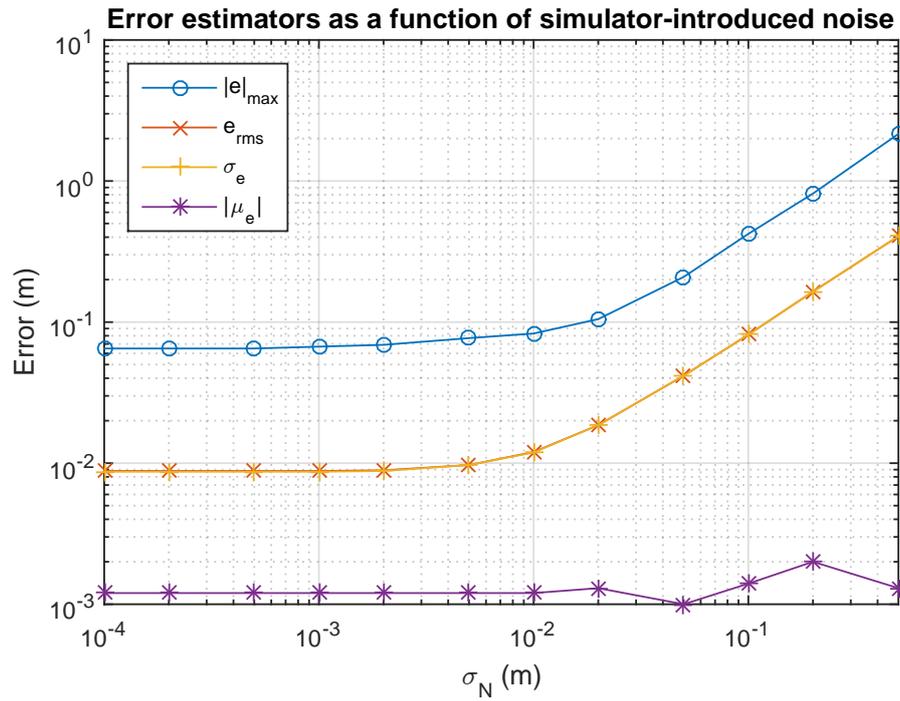


Figure 12: Effect of noise on error estimators, for a sinusoid ($f = 0.2$ Hz, $A = 0.5$ m) with a duration of 500 seconds.

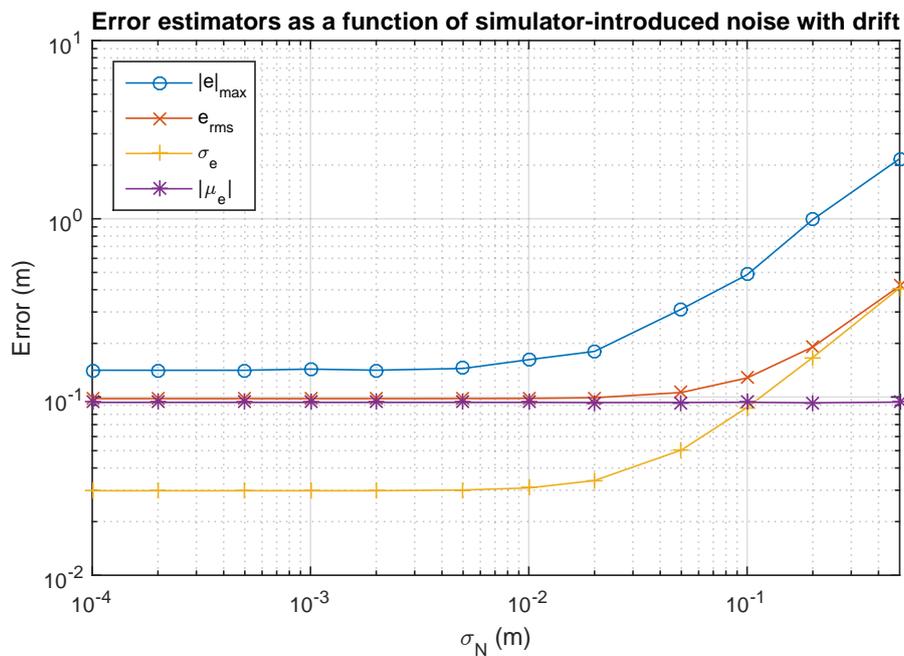


Figure 13: Effect of noise on error estimators, for a sinusoid ($f = 0.2$ Hz, $A = 0.5$ m) with a 0.1 m DC offset and a duration of 500 seconds.

5.2 Prototype tests

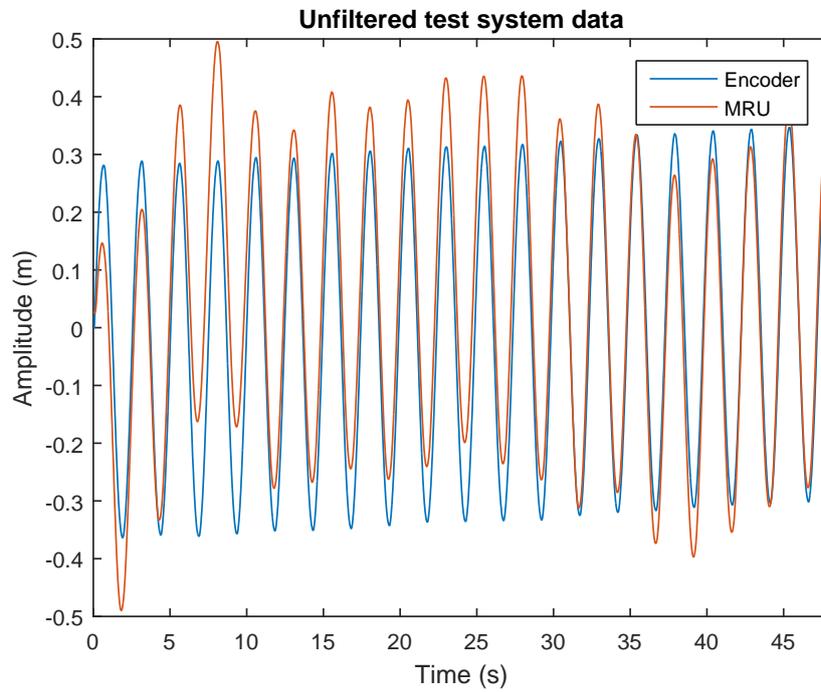


Figure 14: Raw MRU sway data and rotary encoder data for a prototype test using as input a sine with frequency 0.4 Hz and amplitude 0.3 m.

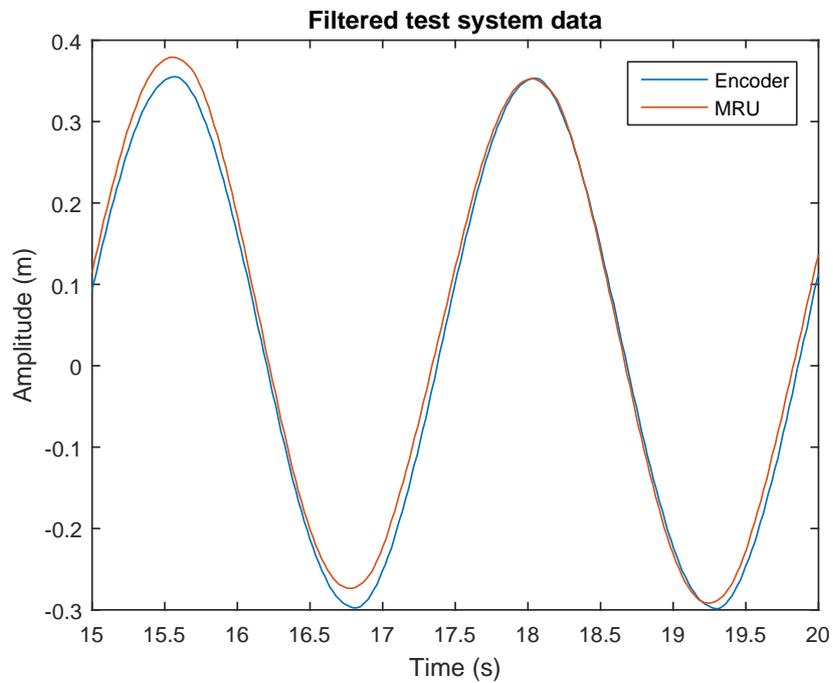


Figure 15: Two periods of the MRU sway data and rotary encoder data bandpass-filtered around the main wave frequency (0.4 Hz).

Table 5.1: Data processing results for the raw and filtered MRU and encoder data, for a test using a sinusoid with frequency 0.4 Hz and amplitude 0.3 m as input.

Estimator	Unfiltered	Filtered
delay (s)	-0.026	-0.025
e_{rms} (m)	0.117	0.037
$ e _{max}$ (m)	0.266	0.167
σ_e (m)	0.107	0.037
μ_e (m)	0.049	-0.004
SNR (dB)	5.90	15.86
Drift (%)	389	73.1

Results of the tests with the prototype test system with the MRU and the rotary encoder are shown in Figures 14 and 15. The results obtained by analysing the data from a single test are shown in Table 5.1. Results for other tests with different frequencies can be found in Appendix A. Besides the raw data, data processing is also performed on bandpass-filtered sets of data, thereby removing drift and only keeping the original frequency component of the signal.

The constant velocity drift of the test setup can be clearly seen by looking at the raw encoder data in Figure 14. The MRU, unable to measure such low frequencies, does not register this constant drift. The MRU signal does however contain drift frequencies itself. According to the drift percentage estimator, the MRU signal contains approximately 4 times more power in the DC drift frequency band (0 – 0.01 Hz) than the encoder, and also has quite some drift power in the frequency band between the DC drift frequencies and the main signal frequency, as seen in the Fourier spectrum shown in Figure 16.

The drift of both the encoder and the MRU has a significant effect on both the error estimators and the SNR estimator. When the drift is filtered out, all four error estimators drop significantly. The SNR increases with almost 10 dB.

The delay, on the other hand, remains constant regardless of any filtering. The MRU signal leads the encoder signal, as can also be seen in Figure 15. A repeated test also gave the exact same delay of -25/-26 ms, confirming the consistency of both the delay and the estimator.

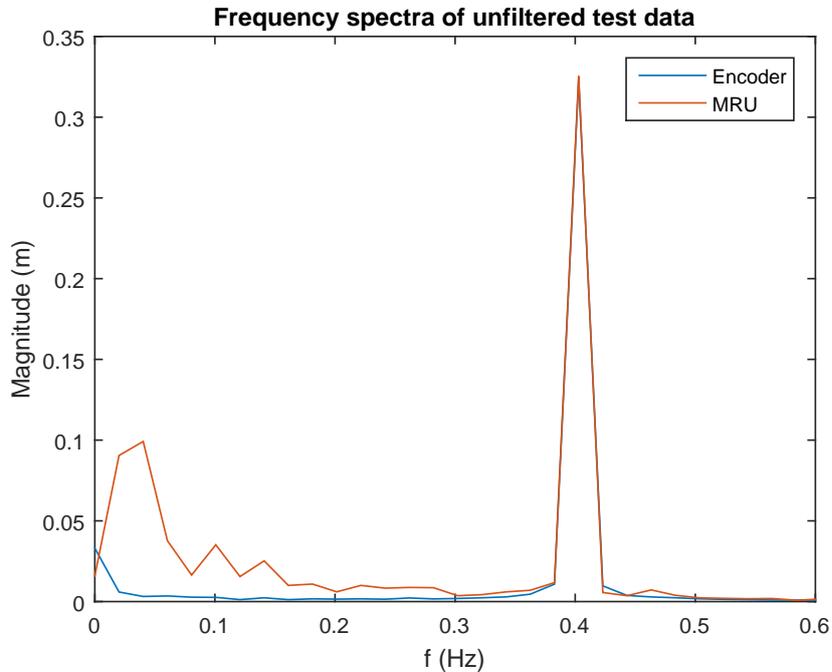


Figure 16: Fourier spectra of the unfiltered data sets plotted in Figure 14

6 Discussion

The results of the simulator and MRU tests shown in the previous section will be interpreted here. Causes and consequences of the results will be discussed. The results will also be used to validate both the performance estimators and the simulator, by comparing the simulator tests with the prototype tests.

6.1 Simulator tests

The constant positive latency estimation error as seen in Figure 6 is notable, especially because it diminishes for longer test durations. Whereas the effect of noise seen in Figures 7 to 9 is expected in that the latency estimation is less accurate for more noise, the mean value of 7 ms and 1 ms for signals of 60 and 600 seconds respectively is more striking. One of more likely explanations is that the cause lies with boundary effects of the signal: due to the filter in the simulator, the first and last few samples may be off from the original signal. This would also explain why the effect diminishes for longer signals, as there are more samples in longer signals that can cancel out these boundary effects.

The estimated signal-to-noise ratio in Figure 10 shows two effects compared to the real SNR: it levels off at around 38 dB and it is 1.3 dB lower than the real SNR for any noise level. The first effect can be partly explained through the effects of the filter in the simulator: indeed, with the filter turned off, the estimated SNR follows the real SNR much better for low noise levels. The constant 1.3 dB offset is more difficult to find an explanation for. It is probably not caused by boundary effects, since without the simulator filter the effect is still present. Because the effect is constant for all noise levels, it is likely also independent of the noise estimator, leaving the signal power estimator as the most likely culprit.

The only effect of note seen in Figure 11 is that of the drift percentage estimator measuring drift in the MRU simulator signal even when the bandpass filter is used. This is easily explained as the filter used is not an ideal brick-wall filter but a Butterworth filter which has a non-vertical slope, leading to part of the low-frequency components persisting in the MRU output signal, which the drift power estimator then measures.

As expected, most error estimators in Figures 12 and 13 increase linearly with noise, but level off at a certain noise level, just like the signal-to-noise ratio in Figure 10. Also as expected is that in the linear part of the plots, σ_e is almost equal to the input σ_N . The rms error approximately equals the greatest source of error, either μ_e or σ_e , as would be expected. The lack of unexpected effects for this test indicates that the error estimators function as intended.

6.2 Prototype tests

One of the first things to note about the MRU test is that the lack of a proper motor controller with feedback limits the tests in a major way. The limited test duration reduces the ability of the data processing function block to accurately estimate e.g. latency (as seen in the previous section). Shorter tests also means a lower resolution for the FFT used in the drift estimation, leading to a loss of accuracy for this estimator. In addition, the velocity drift of the rotary encoder hinders the functionality of the SNR and error estimators. The irregular drift in the MRU signal is also worrisome, although expected given the low-frequency behaviour of accelerometers. Perhaps this drift might diminish for longer tests, as the internal MRU filters then have more data points to filter out the long-term trends.

In the case that the drift is still present for longer tests, the SNR estimator should probably be revised so that the drift is first filtered out using a high-pass filter. The current estimator uses the rms error as the noise amplitude. When drift is present, however, the rms error is dominated by this drift instead of noise. With a high-pass filter applied to the incoming encoder and MRU data, the rms error is dominated by noise again.

Concerning the delay estimator, the results confirm that cross-correlation is a robust method of measuring the delay, even though the results of the prototype test were somewhat unexpected. There are multiple hypotheses to explain the fact that the MRU is leading the encoder instead of vice versa. Firstly, the internal

high-pass filter in the MRU could cause a phase lead in the output signal of the MRU. This hypothesis is reinforced by the fact that the latencies are dependent on the frequency of the sine (see Appendix A), with the latency becoming more positive as the frequency increases. This corresponds with the phase response of a bandpass filter.

The second possibility is that the encoder has a greater latency than the MRU. This does not explain the different latencies for different frequencies, but the delay introduced by the microcontroller should be tested nonetheless to confirm that this is not the cause of the negative delay.

The rotary encoder retrospectively seems to be sufficiently accurate to use as a ground truth. It experiences no drift (the drift in the encoder signal for the prototype test was caused by unwanted effects in the motor and mechanical system, which the encoder proceeded to accurately measure), or other unexpected behaviour.

Looking back at the simulated encoder signal, the lack of any unexpected behaviour in the encoder also means the simulated encoder signal is valid to use, as it thus closely resembles the real encoder.

The simulated MRU on the other hand might need some improvements. The introduced noise in the simulator is most likely too high compared to what has been seen in the measured data. The drift on the other hand is more prominent in the real MRU compared to the simulator, and comprises a wide frequency band, as per Figure 16.

Another difference between the results of the simulator and the real MRU appears at the beginning of the signal. When following a sine wave for the position, the speed acts as a cosine (90 degree phase shift), resulting in a maximal acceleration during the first few samples of the test. This sudden 'kick' that the MRU experiences, is the cause of the low frequency oscillations in the beginning of Figure 14. Because the simulator only accounts for the position and not for the acceleration, these effects are not present in the simulated MRU signal. These effects diminished towards the end of the test, where the simulator and the MRU approached each other. Due to the drifting of the cart it was not possible to do longer tests. In the final system longer test can be done and the control loop will prevent the 'kick' at the beginning of the test sequence.

Lastly, the latency of the MRU doesn't seem to be a constant positive value, but rather dependent on frequency and it can be either positive or negative. The simulator may be adapted to include this.

7 Conclusion

This section reflects on the obtained results and on the project as a whole. First, the the performance of the developed function blocks is reflected upon. Then, a conclusion will be drawn on the prototype tests, followed by any work to be done in the immediate future to improve the test system. The block diagram of the complete system as designed is included as well, in Figure 17. Lastly, some recommendations are given regarding the use of MRUs on the Ampelmann system, and how the performance of the sensing part of the test system might be improved.

7.1 Function block evaluation

The wave point generator has no problems with generating test sequences and checking these for the boundaries set by the program of requirements. These safety checks were thoroughly tested to ensure that they work as intended. While testing with the system it appeared that the simulator gave a good estimation of the MRU after settling, but a large acceleration at the start of the sequence could cause frequencies to appear in the MRU data that were not present in the simulator. In a possible future version of the simulator, the acceleration data could be used (over the positional data) since the MRU is based on accelerometers. The data processing block gave good results for the error measurements and determining the drift. On the other hand, when determining the latency it was heavily influenced by phase delays from the internal filters of the MRU. A possible way of determining the latency of the system is by designing a test that test the events based latency [10], this could be implemented in a future version of the system

7.2 Prototype test

While testing the encoder worked as expected, showing the cart position in an accurate manner. The MRU on the other hand did display some unexpected behaviour. It was not expected to have the negative latency visible in Figure 15. Based on the inclusion of bandpass filters in MRUs and on the frequency dependant behaviour of the latency, it was concluded that a big part of the latency could be caused by phase delays. These frequency depended delays are introduced by the internal filters. Since the tests that were run consisted of a single constant frequency, this phase delay was extra prominent. When the control loop is complete, tests with more irregular signals as an input can be run. These tests should give a more conclusive measurements concerning the latency of the overall system, as this latency should be constant over the frequency range.

7.3 Future work

There are a number of minor tasks that are planned for the coming weeks before the system is delivered to Ampelmann. Firstly a test manual has to be made to allow the employees of Ampelmann to use the system safely. This also includes documentation on the Octave scripts and functions that were written for the system.

Another point of improvement is in the way the simulator works. The MRU reacts quite strongly on a sudden change in acceleration (as seen in Figure 14) with low frequency oscillations. A possible point where the simulator can be improved is by having it start at a stand still (position, velocity and acceleration are zero) and by using the acceleration data to calculate the simulated location of the MRU.

Currently the test rail has only been tested in a horizontal orientation. In this orientation it is only possible to test the surge and sway measurements of the MRUs. When measuring the heave motion, the MRU takes the gravitational acceleration into account. Simply placing the MRU side ways (to align the heave axis with the test rail), would cause the MRU to counteract a gravitational pull that isn't there. In the future it should be possible to place the track vertically, making it possible to test the heave axis of the MRU.

7.4 Recommendations

A drawback of using accelerometer-based MRUs to collect motion data is that they have a hard time detecting slow movements. A possible solution for this drift is to add a separate sensor that is more

accurate at low frequencies. This two sensor approach has been explored extensively, MRUs have already been combined with laser- [14] and image-based [15] [16] sensors. A prediction model can also be used alongside an MRU [17] [18] by employing, for example, a Kalman filter. Many of these filters also include real-time phase compensation [19] [20] to counteract the phase shift effect of the bandpass filter in the MRU.

7.5 Ethical considerations

Looking back, it is important to keep in mind the ethical aspects of the project. The test system that is developed may be used to make decisions regarding the use of MRUs on the Ampelmann system. As a result, the information obtained through this test system may be partially responsible for the safety of users of the Ampelmann systems.

For this reason, it is of utmost importance to keep the results produced by the test setup unbiased. By nature the developed test setup functions exactly the same regardless of the type of MRU used, thereby remaining unbiased.

Furthermore those who will interpret the results should have a sufficiently good idea of what they mean. In that light it is good to emphasise once again that results should be interpreted relatively rather than absolutely. Of course it is the shared responsibility of the developers of the test system to inform users of this and other information regarding the use of the system.

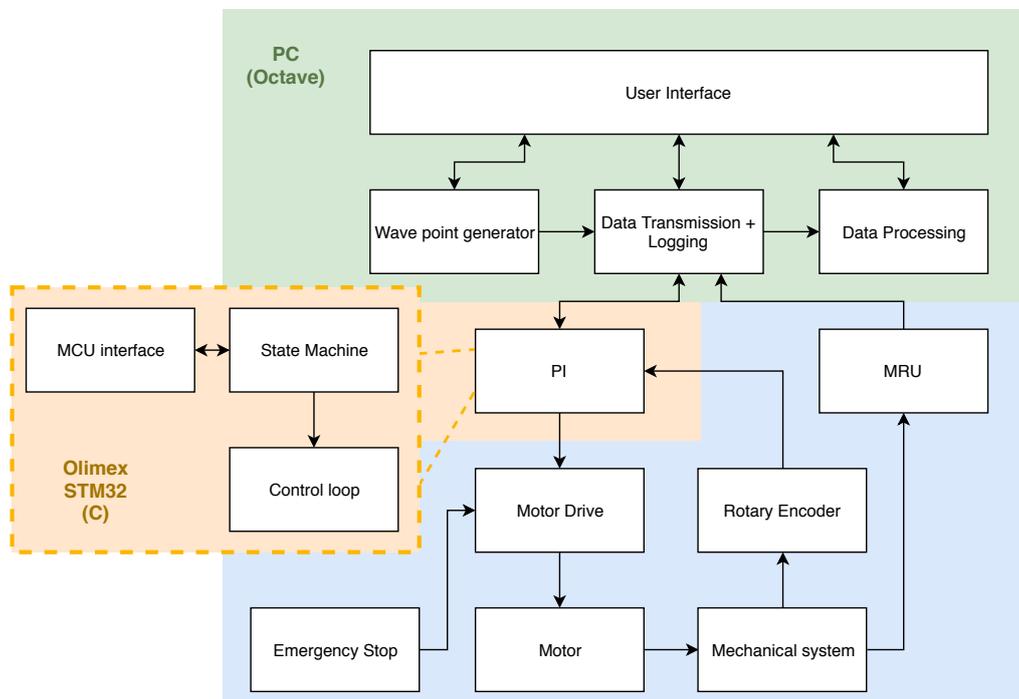


Figure 17: Final block diagram of the complete system.

References

- [1] CalQlata, “Vessel 6-degrees of movement.” [Online]. Available: <http://www.calqlata.com/productpages/00059-help.html>
- [2] Kongsberg Seatex, “Motion Reference Unit.” [Online]. Available: <https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/61118F926CD5A6E5C125700B0033CF55>
- [3] D. Bishop, “Qualifying heave performance at sea,” *Hydro international*, 2009. [Online]. Available: <https://www.hydro-international.com/content/article/qualifying-heave-performance-at-sea>
- [4] M. Kok, J. D. Hol, and T. B. Schon, “Using Inertial Sensors for Position and Orientation Estimation,” *Foundations and Trends in Signal Processing*, vol. 11, no. 1-2, pp. 1–153, 2017.
- [5] D. J. C. Salzmann, “Development of the Access System for Offshore Wind Turbines,” Ph.D. dissertation, Delft University of Technology, November 2010.
- [6] W. Bierbooms, “Wind and Wave conditions,” Dutch Offshore Wind Energy Converter project, Tech. Rep., 2002.
- [7] Tamagawa Seiki Co., “Incremental Encoders,” 2017. [Online]. Available: <https://www.tamagawa-seiki.com/products/rotaryencoder/incremental.html>
- [8] S. S. Tørdal and G. Hovland, “Inverse kinematic control of an industrial robot used in vessel-to-vessel motion compensation,” in *2017 25th Mediterranean Conference on Control and Automation (MED)*, July 2017, pp. 1392–1397.
- [9] S. S. Tørdal, P. O. Løvslund, and G. Hovland, “Testing of wireless sensor performance in vessel-to-vessel motion compensation,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2016, pp. 654–659.
- [10] K. Dudacek, “Short Time Delay Measurements,” University of West Bohemia, Tech. Rep. DCSE/TR-2015-03, July 2015.
- [11] iXblue, “High grade gyrocompass for navigation & platform stabilization.” [Online]. Available: <https://www.ixblue.com/products/octans>
- [12] SBG Systems, “Ekinox2-A: Attitude and Heading Reference System.” [Online]. Available: <https://www.sbg-systems.com/products/ekinox-high-performance-mems-ahrs>
- [13] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*, 4th ed. Pearson, July 2013.
- [14] S. S. Tørdal, W. Pawlus, and G. Hovland, “Real-time 6-dof vessel-to-vessel motion compensation using laser tracker,” in *OCEANS 2017 - Aberdeen*, June 2017, pp. 1–9.
- [15] S. S. Tørdal and G. Hovland, “Relative Vessel Motion Tracking using Sensor Fusion, Aruco Markers, and MRU Sensors,” *Modeling, Identification and Control*, vol. 38, no. 2, pp. 79–93, 2017.
- [16] J.-H. Yi, J.-H. Kim, W. M. Jeong, and J.-W. Chae, “Field evaluation of optical-based three-dimensional dynamic motion measurement system with multiple targets for a floating structure,” *Ocean Engineering*, vol. 62, pp. 140 – 151, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0029801813000073>
- [17] R. Kannan and S. Jain, “Adaptive recalibration algorithm for removing sensor errors and its applications in motion tracking,” *IEEE Sensors Journal*, vol. 18, no. 7, pp. 2916–2924, April 2018.
- [18] J. M. Godhaven, “Adaptive tuning of heave filter in motion sensor,” in *OCEANS '98 Conference Proceedings*, vol. 1, Sep 1998, pp. 174–178 vol.1.

-
- [19] M. Richter, K. Schneider, D. Walser, and O. Sawodny, "Real-time heave motion estimation using adaptive filtering techniques," vol. 19, pp. 10 119–10 125, 01 2014.
- [20] H. Yongpan and T. Limin, "Real-time zero phase filtering for heave measurement," in *2013 IEEE 11th International Conference on Electronic Measurement Instruments*, vol. 1, Aug 2013, pp. 321–326.

Appendices

A MRU test data

Table A.1: Data processing results for prototype MRU tests

Wave frequency (Hz)	Delay (s)		SNR (dB)		Drift (%)			
	Unfiltered	Filtered	Unfiltered	Filtered	Unfiltered	Filtered		
0.2	-0.381	-0.384	3.81	20.81	0.91	3.67		
0.3	-0.117	-0.116	10.72	18.58	268	0.15		
0.4	-0.026	-0.025	5.90	15.86	389	20.2		
0.5	0.014	0.015	1.96	14.94	1316	73.1		
Wave frequency (Hz)	e_{rms} (m)		$ e _{max}$ (m)		σ_e (m)		μ_e (m)	
	Unfiltered	Filtered	Unfiltered	Filtered	Unfiltered	Filtered	Unfiltered	Filtered
0.2	0.177	0.020	0.282	0.109	0.094	0.020	-0.150	-0.003
0.3	0.066	0.027	0.170	0.145	0.064	0.027	-0.019	-0.003
0.4	0.117	0.037	0.266	0.167	0.107	0.037	0.049	-0.004
0.5	0.184	0.041	0.404	0.165	0.180	0.041	0.039	-0.005

B WPG flow diagram

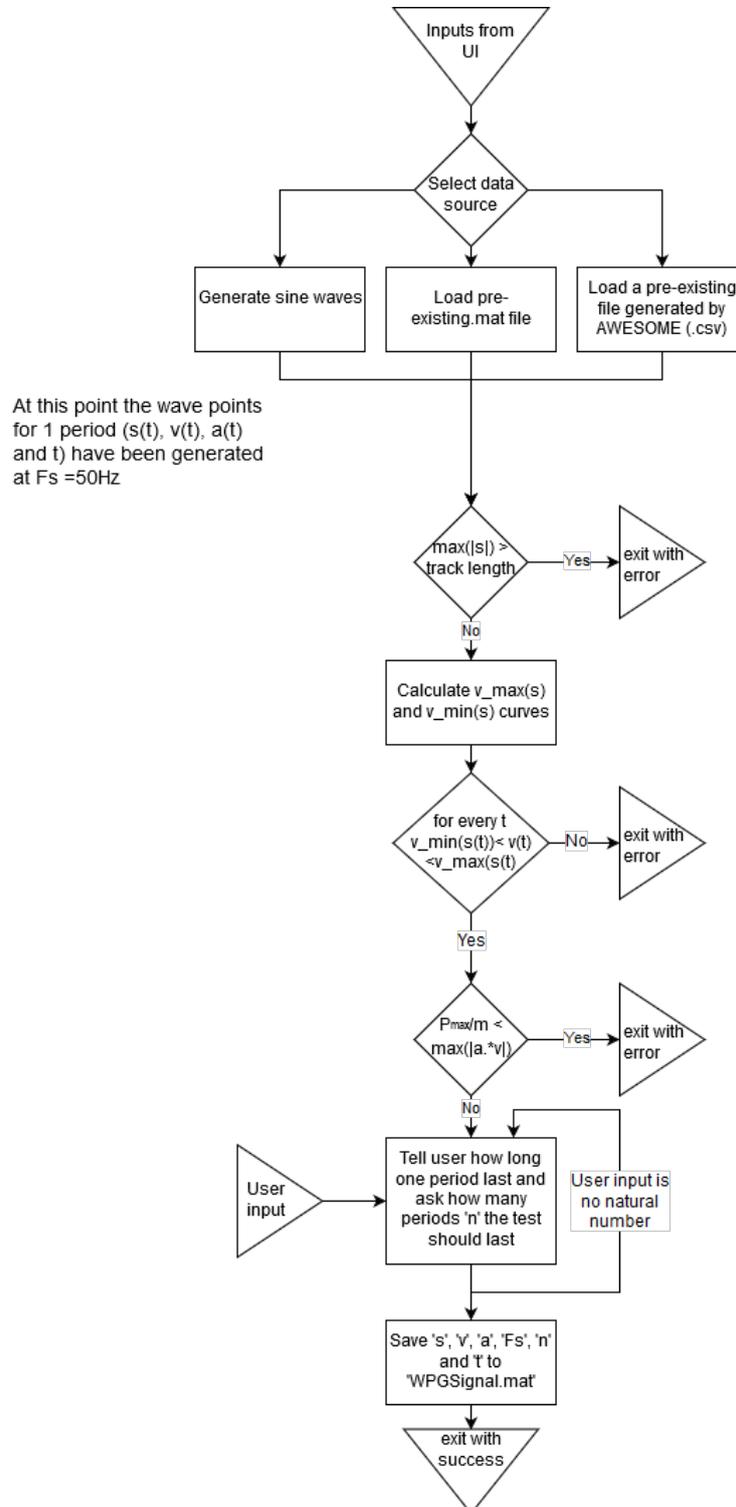


Figure 18: Flow diagram of the wave point generator.