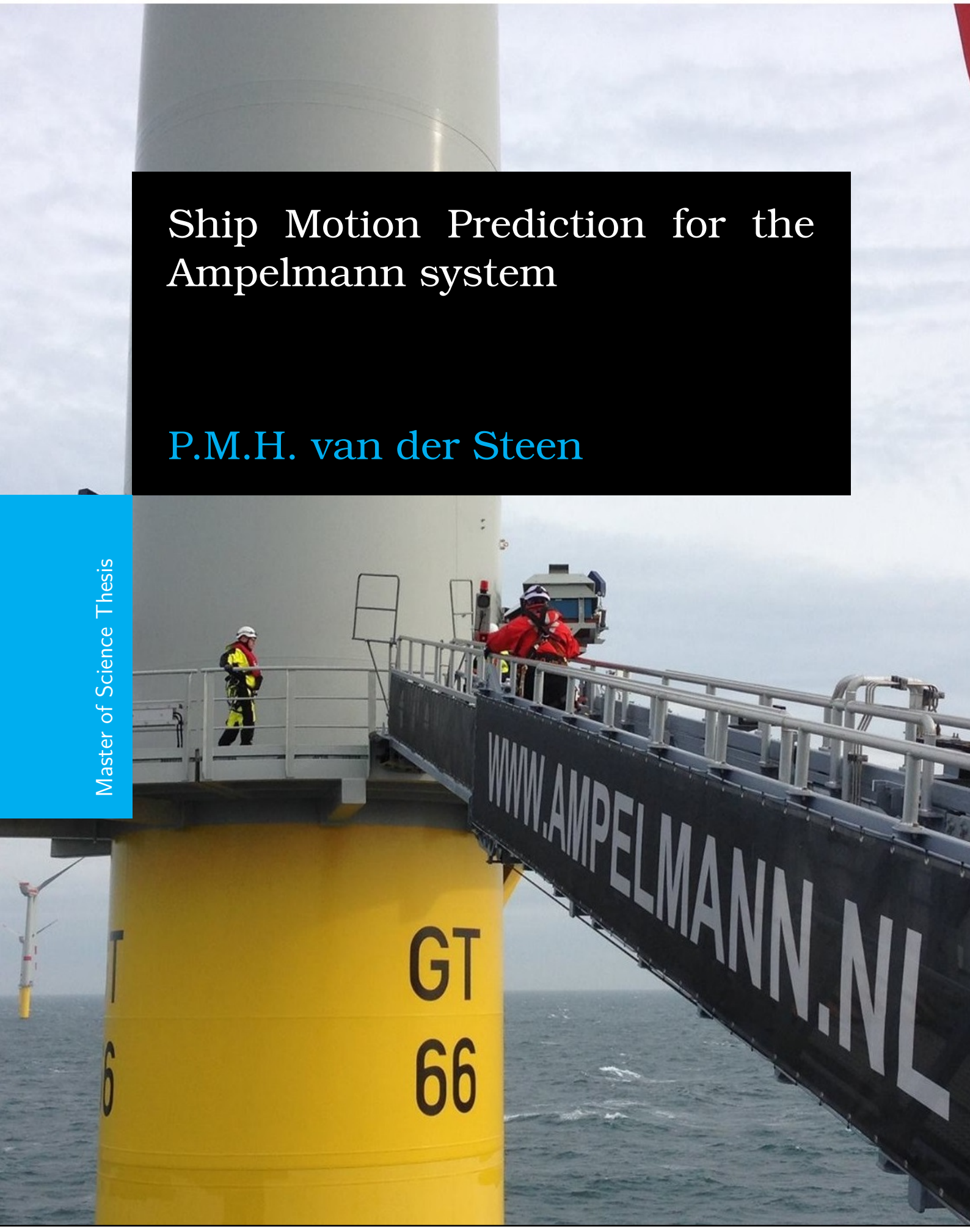
Ship Motion Prediction for the Ampelmann system

P.M.H. van der Steen

Master of Science Thesis



TU Delft Delft University of Technology

Delft Center for Systems and Control

Ship Motion Prediction for the Ampelmann system

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

P.M.H. van der Steen

February 9, 2016

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

AMPELMANN

The work in this thesis was supported by Ampelmann Operations. Their cooperation is hereby gratefully acknowledged.

TU Delft Delft University of Technology

DCSC

Delft University of Technology
Department of
Delft Center for Systems and Control (dcsc)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Ship Motion Prediction for the Ampelmann system

by

P.M.H. van der Steen

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: February 9, 2016

Supervisor(s):

Prof.Dr. R. Babuška

Ir. A. van Leer

Ir. J.W. de Vriend

Reader(s):

Dr.Ir. J.W. van Wingerden

Ir. P. Naaijen

Abstract

During rough wave climates, vessel motions prevent people to be transferred safely from and to an offshore structure. The Ampelmann system is a ship-based, self stabilizing platform that actively compensates all vessel motions to make access to offshore structures safe, easy and fast.

The Ampelmann system uses a Steward platform to compensate ship motions. The platform consists of a rigid base frame and a rigid top frame connected by six hydraulic actuators in parallel. A vessel has six degrees of freedom (DoF) (surge (x), sway (y), heave (z), roll (ϕ), pitch (θ) and yaw (ψ)). The system actively compensates for five DoF. Yaw is not considered in the control system, since a vessel has its own Dynamic Positioning (DP) system which controls the heading of the ship.
An Octans motion sensor placed at the bottom frame and measures the motions with a frequency of 50 [Hz]. These signals are then filtered, limited and smoothed. With Inverse Kinematics (IK) the reference lengths of the cylinders are determined. The actual lengths are measured and fed back to the controller.

Still some residual motions can occur in the Ampelmann system. This has two main reasons:

- For safety reasons the system does not use its complete workspace. The signal is limited so the cylinder wont abruptly stop at the end of its stroke. In previous research [1] model predictive control (MPC) has been applied, to increase the workability and reduce the accelerations, which occur when the Ampelmann reaches its workspace boundaries.
- The filters in the control loop of the Ampelmann system cause a significant delay. The difference between the input signal and the filtered signal could increase up to 13% of the amplitude of the wave. This problem can be solved by applying an anti-causal filter, which uses knowledge of future data to filter without creating a phase lag.

A forecast of the vessel motions is necessary to implement MPC and an anti-causal filter. Therefore the aim of this master thesis research is to find an algorithm which predicts real-time short-term vessel motions.

A pure time series forecasting approach was followed and both linear as non-linear models are proposed. Real vessel motion data is used to compare and test the models. The time series is (wide-sense) stationary, therefore the linear Auto Regressive (AR) model and Auto Regressive Moving-Average (ARMA) model are chosen for describing the signals. By comparing the multi-step ahead prediction error of the different models, it is found that the AR(7) model it the most suitable model to describe the vessel motions.
To be able to implement the prediction real-time, the parameters could be recursively determined or batch wise. Also resampling the data to 5 [Hz] results in an improvement of the forecast and decreases the computational demand.
Wavelets are a powerful tool for analyzing time series. The wavelet is applied in two different manners to make a forecast. First the the wavelet transform is used to decompose the time series into varying scales of resolutions. Each resolution scale can be used for modeling and prediction. Since there are no clear levels of details and trends observed in the data, this approach did not benefit the prediction.
The second method uses a Wavelet Neural Network (WNN). This is an artificial neural network with one hidden layer of nodes, whose activation function are drawn from a family of nonlinear wavelets. The prediction error of the rotational signals (ϕ and θ) decreases by using a wavenet for prediction, compared to the AR model. The AR model is more suitable for modeling the translations (x,y and z). A multivariate wavenet model combines the signals in two models, instead of five separate models. Coupling the translations in one model reduces the prediction error. A model with three inputs (z, ϕ and θ) and two outputs (ϕ and θ) is better capable of modeling the rotations, compared to the separate models.
The difference between the prediction errors of all models are small. Therefore more data is required to give a more exact estimate of the prediction error and to be able to determine which model the most suitable for describing the signals.

The prediction can be used in combination with an anti-causal filter, which is able to filter with zero-phase shift and creating a delay. The AR-model is able to predict accurate enough to implement the anti-causal filter. The Root mean squared error (RMSE) between the raw data and the filtered data reduces by 88% compared to the original Octans filter.

Table of Contents

Chapter 1

Introduction

Accessibility of offshore structures is strongly affected by local weather and wave conditions. During rough wave climates, vessel motions prevent people to be transferred safely from and to an offshore structure. To increase accessibility, the Ampelmann system has been developed. The Ampelmann system is a ship-based, self stabilizing platform that actively compensates the vessel motions to make access to offshore structures safe, easy and fast.

The Ampelmann system uses a Stewart platform to compensate ship motions. This Stewart platform consists of a rigid base frame and a rigid top frame connected by six hydraulic actuators in parallel. By calculating the required lengths of these actuators, a motionless transfer deck is created. On top of the Stewart platform, a transfer deck is mounted. Here, people wait to enter the offshore structure or to go back to the deck of the ship. Furthermore, an operator is standing on this deck to control the Ampelmann system. A gangway or Telescopic Access Bridge (TAB) is attached to the transfer deck to enable people to walk safely to and from the offshore structure. The TAB can move in three Degrees of Freedom (DoF): telescoping (in and out), luffing (up and down) and slewing (rotating left and right). Figure 1-1 shows the Ampelmann system during operation.

Although the system is able to stabilize the transfer deck, there are still undesirable residual motions present during operation. This means that the top frame is not completely motionless, which causes difficulties in positioning the gangway and reduces comfort and safety when standing on the transfer deck. This has two main reasons:

- For safety reasons the system does not use its complete workspace. The signal is limited so the cylinder wont abruptly stop at the end of its stroke. In previous research [1] model predictive control (MPC) has been applied, to increase the workability and reduce the accelerations, which occur when the Ampelmann reaches its workspace boundaries.
- The filters in the control loop of the Ampelmann system cause a significant delay. The difference between the input signal and the filtered signal could increase up to 13% of the amplitude of the wave. This problem can be solved by applying an anti-causal filter, which uses knowledge of future data to filter without creating a phase lag.

Figure 1-1: The Ampelmann system during operations

A forecast of the vessel motions is necessary to implement MPC and an anti-causal filter. Therefore the aim of this research is to find an algorithm which predicts real-time short-term vessel motions.

In order to reach this goal first the Ampelmann system needs to be further explained. In Chapter 2 an overview of the Ampelmann system is given. The kinematics and workspace of the Stewart platform are explained. The reference generator as well as the motion control loop are presented, which gives an insight in the limiting, filtering and smoothing of the signals. The filters are analyzed in order to see the impact of the delays.
In Chapter 3 the data set, which is used during this research is analyzed. The properties of the waves give information for the prediction model.
Next in Chapter 4 linear prediction models are discussed. The model is trained, validated and tested. Also real-time implementation is taken into account.
In Chapter 5 the mathematical tool 'Wavelet' will be explained. The data is analyzed with the wavelet transform. Then the wavelet is used to construct a non-linear Wavelet Neural Network (WNN). The performance of this network is compared to the linear model.
In Chapter 6 an example is shown of an implementation of the obtained prediction method, by applying a non-causal filter which removes the phase shift caused by causal filters.
Finally, Chapter 7 gives an overview of the conclusions that have been drawn. Furthermore, recommendations for future work are given.

Chapter 2

The Ampelmann System

2-1 Introduction

The Ampelmann system is a ship-based, self stabilizing platform that actively compensates all vessel motions to make access to offshore structures safe, easy and fast. The largest Ampelmann system, the E-type, is shown in Figure 2-2.
A ship has six Degrees of Freedom (DoF), three translations and three rotations, see Figure 2-1. The Steward Platform as applied in the Ampelmann system can move in all six degrees of freedom to keep the transfer deck motionless on a moving vessel [2]. It consists of a top frame, attached to the transfer deck, bottom frame, attached to the vessel, six hydraulic cylinders and the gangway. The vessel motions are actively compensated by six hydraulic actuators. The workspace is limited by the maximum stroke of the cylinders. The motion range of the Stewart platform is large enough to compensate for a significant wave height of 2.5m and a mean wave period of 4.5 [sec]. The significant wave height (H_s) is defined as the mean wave height of the highest one-third of the waves. These sea state limits are determined to obtain a 90% accessibility in the Dutch North Sea [2].
To have a better insight in the Ampelmann system, first the kinematics are discussed. Kinematics are used for calculating the cylinder lengths, which are needed for compensation. Then the workspace of the Ampelmann is explained. From the kinematics and the workspace a reference signal can be determined. The reference signal generator is further explained in the next section, which covers the whole control loop. The delays of the filters are also calculated. Finally a simulation of the compensation is shown.

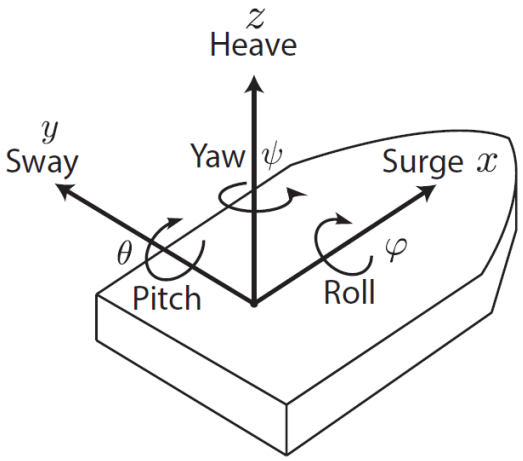


Figure 2-1: Six degrees of freedom

Figure 2-2: The Ampelmann E-type

2-2 Kinematics

The Stewart platform has a rigid base frame and a rigid top frame, connected by six linear actuators that are connected by gimbals. The pose (or configuration) of a Stewart platform in any arbitrary point in time can be defined by the position and orientation of the top and base frame. Given the relative position and orientation of both frames, all six actuator lengths can be determined; this is called *inverse kinematics*. The derivation of the pose of a Stewart platform with given actuator lengths, is referred to as *forward kinematics* [3]. The mathematical forward kinematics problem has more than one solution whereas the inverse kinematics problem has a single solution, though there is only one solution feasible in reality.

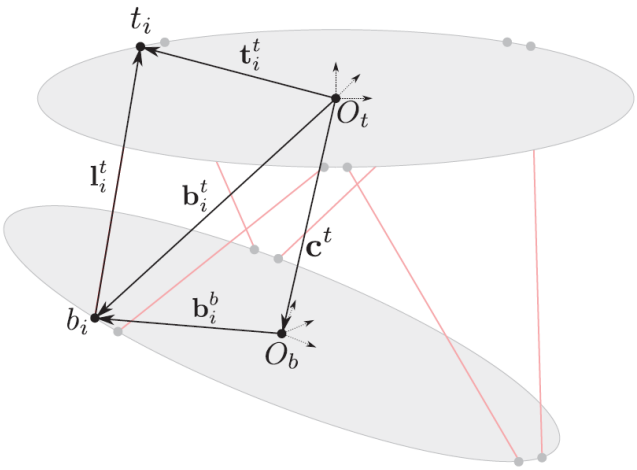


Figure 2-3: Kinematics of the Stewart platform

2-2-1 Inverse kinematics

The pose of the top frame is assumed to be horizontal and motionless. The position of the centre of the top frame can be expressed in a vector c^t, see Figure 2-3. The coordinate system is attached to the centre of the base frame O_b. The Octans motion sensor is placed at the centre of the the bottom frame. So c^t can be expressed as:

$$c^t = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad \text{with respect to } O_b \tag{2-1}$$

Where x_c, y_c and z_c are the coordinates of the centre of the top frame with O_b as origin. The three rotations of the top frame relative to the base frame are to be taken into account, the angles are called the *Euler angles*. The Euler transformation matrix can be derived by

considering the three separate rotations about the principle axes. The rotation around the x-axis is roll (φ), around the y-axis is pitch (θ) and around the z-axis is yaw (ψ). The following three rotation matrices can be derived, a rotation matrix describes in three dimensions the rotation of an angle over an axis.

$$R_{x\varphi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & \sin\varphi \\ 0 & -\sin\varphi & \cos\varphi \end{bmatrix} R_{y\theta} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} R_{z\psi} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2-2}$$

The bottom frame point is described by a vector b_i^b (see Figure 2-3), from this the top frame point vector b_i^t can be derived [4].

$$\begin{aligned} b_i^t &= (R_t^b)^{-1} b_i^b + c^t \\ b_i^t &= (R_b^t) b_i^b + c^t \end{aligned} \tag{2-3}$$

The transformation matrix R_b^t transforms the vector b_i^b of the bottom point expressed in the bottom frame coordinates into the vector of the bottom point expressed in the top frame coordinates. The matrix results R_b^t from the Euler rotation sequence. The rotation sequence used is z-y-x, first yaw, then pitch, then roll. It is formed by multiplying the three rotation matrices.

$$R_b^t = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi\cos\varphi + \cos\psi\sin\theta\sin\varphi & \sin\psi\sin\varphi + \cos\psi\cos\varphi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\varphi + \sin\varphi\sin\theta\sin\psi & -\cos\psi\sin\varphi + \sin\psi\cos\varphi\sin\theta \\ -\sin\theta & \cos\theta\sin\varphi & \cos\theta\cos\varphi \end{bmatrix} \tag{2-4}$$

The top frame point expressed in top frame coordinate system is a fixed vector t_i^t. Then the length of each actuator is found by subtracting the bottom frame point (expressed in the top frame coordinates) b_i^t from the top frame point (also expressed in the top frame coordinates) t_i^t.

$$l_i = t_i^t - b_i^t = t_i^t - (R_b^t) b_i^b + c^t \tag{2-5}$$

The reference length can be calculated by subtracting the initial length of the actuator (fixed). The initial length is the length of the actuator when the platform is in its neutral position.

$$r_i = \|l_i\| - l_0 \tag{2-6}$$

This completes the inverse kinematics sequence, computing six actuator lengths from a given top frame position and orientation.

2-2-2 Forward kinematics

Forward kinematics (FK) determines the position and orientation of the top frame relative to the bottom frame [3],[1]. In geometrical sense, it is equivalent to the problem of placing a rigid body in such a way that six of its given points lie on six given spheres. These spheres have a diameter l_i around the bottom frame gimbals (b_i). FK will not have an unique solution, the maximum number of solutions is 40, also containing solutions in the complex domain. The

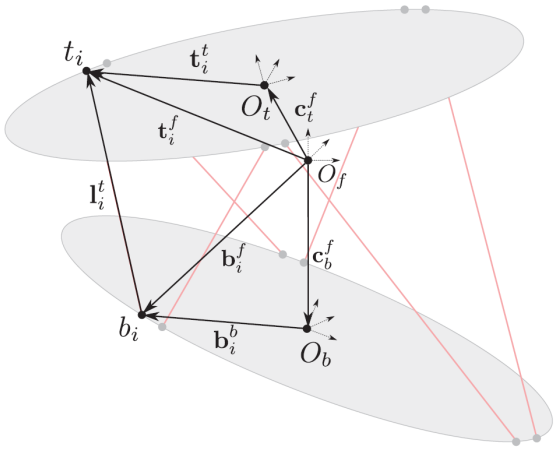


Figure 2-4: Forward Kinematics

solution to FK can be found numerically by the iterative Newton-Rhapson method.
The numerical method approximates the roots of the following function:

$$f_i(\mathbf{q}^t) = \mathbf{l}_i^T \mathbf{l}_i - l_i^2 \tag{2-7}$$
$$\mathbf{q}^t = \begin{bmatrix} z^t & y^t & z^t & \varphi^t & \theta^t & \psi^t \end{bmatrix}^T$$

The actuator lengths (l_i) are known, but the length vector ($\mathbf{l}_i$) has to be calculated. An extra coordinate system is added (see Figure 2-4) with superscript f. The top and bottom gimbal coordinates can be expressed in O_f. Since the top frame does not remain horizontal and motionless O_t cannot be used.
The length vector $\mathbf{l}_i$ can analytically be described as:

$$\mathbf{l}_i = \mathbf{t}_i^f - \mathbf{b}_i^f = (\mathbf{R}_t^f \mathbf{t}_i^t + \mathbf{c}_t^f) - (\mathbf{R}_t^f \mathbf{b}_i^t + \mathbf{c}_i^f) \tag{2-8}$$

Eq. (2-8) can be inserted in Eq. (2-7). The Newton-Rhapson method approximates the roots with the following iterative process:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{2-9}$$
$$\mathbf{q}_{n+1}^t = \mathbf{q}_n^t - \left[\frac{\partial \mathbf{f}(\mathbf{q}_n^t)}{\partial \mathbf{q}_n^t}\right]^{-1} \mathbf{f}(\mathbf{q}_n^t) \tag{2-10}$$

This iteration will quadratically converge to $\mathbf{q}^{t*}$, satisfying $f_i(\mathbf{q}^{t*}) \approx 0$. The function $\mathbf{f}(\mathbf{q})$ is defined as:

$$\mathbf{f}(\mathbf{q}) = \begin{bmatrix} f_1(\mathbf{q}) & f_2(\mathbf{q}) & f_3(\mathbf{q}) & f_4(\mathbf{q}) & f_5(\mathbf{q}) & f_6(\mathbf{q}) \end{bmatrix} \tag{2-11}$$

The function $\mathbf{f}(\mathbf{q})$ represent the difference between the squared measured and calculated length (see Eq. (2-7)). For $f_i(\mathbf{q}^{t*}) \approx 0$ these are equal and the position and orientation of the top frame can be found.

2-3 Workspace

The Ampelmann workspace is defined as the total of six DoF motion range limited by the minimum or the maximum stroke of the cylinders. The workspace relations are highly non-linear and the physical interpretation is not straight forward. The motion of one DoF is limited by the other five DoF. For each DoF a part of the cylinder stroke needs to be reserved.
The workspace can be estimated through a numerical approach [4]. By repeatedly searching for boundary point (cylinder at its minimum or maximum) in a number of directions the local workspace boundaries can be determined. There are fifteen cross sectional planes (x-y, x-z, x-φ, x-θ, x-ψ, y-z, y-φ, y-θ, y-ψ, z-φ, z-θ, z-ψ, φ-θ, φ-ψ and θ-ψ).
These local workspace boundaries can analytically be approximated by a circle until it touches the actual boundary. Since the width and height of each local boundary are not equal an ellipse will be used. This will result into a six dimensional hyper ellipsoid, which is given by:

$$\left(\frac{x}{\rho_x}\right)^2 + \left(\frac{y}{\rho_y}\right)^2 + \left(\frac{z}{\rho_z}\right)^2 + \left(\frac{\varphi}{\rho_\varphi}\right)^2 + \left(\frac{\theta}{\rho_\theta}\right)^2 + \left(\frac{\psi}{\rho_\psi}\right)^2 \leq R_{\max}^2 \tag{2-12}$$

Where $\rho_{x,\ldots,\psi}$ are the weighting factors of each DoF and $R_{\max}$ is the maximum radius of the ellipsoid. If the Steward platform would be rotated 90 degrees around the z-axis, the x- and y-axis are interchanged. This also works for φ and θ. So the workspace boundary in the x-y and φ-θ plane should be defined by a circle. This can be done by setting $\rho_x = \rho_y = \rho_{xy}$ and $\rho_\varphi = \rho_\theta = \rho_\varphi\theta$. Then Eq. (2-12) can be reduced to:

$$r_{xy}^2 + \left(\frac{z}{\rho_z}\right)^2 + r_{\varphi\theta}^2 + \left(\frac{\psi}{\rho_\psi}\right)^2 \leq R_{\max}^2 \tag{2-13}$$

$$r_{xy}^2 = \frac{x^2 + y^2}{\rho_{xy}}$$

$$r_{\varphi\theta}^2 = \frac{\varphi^2 + \theta^2}{\rho_{\varphi\theta}}$$

The hyper ellipsoids are defined in [4] and shown in Figure 2-5. Now an analytical description of the Stewart platform is available, which can be used to limit six DoF in a specific order.

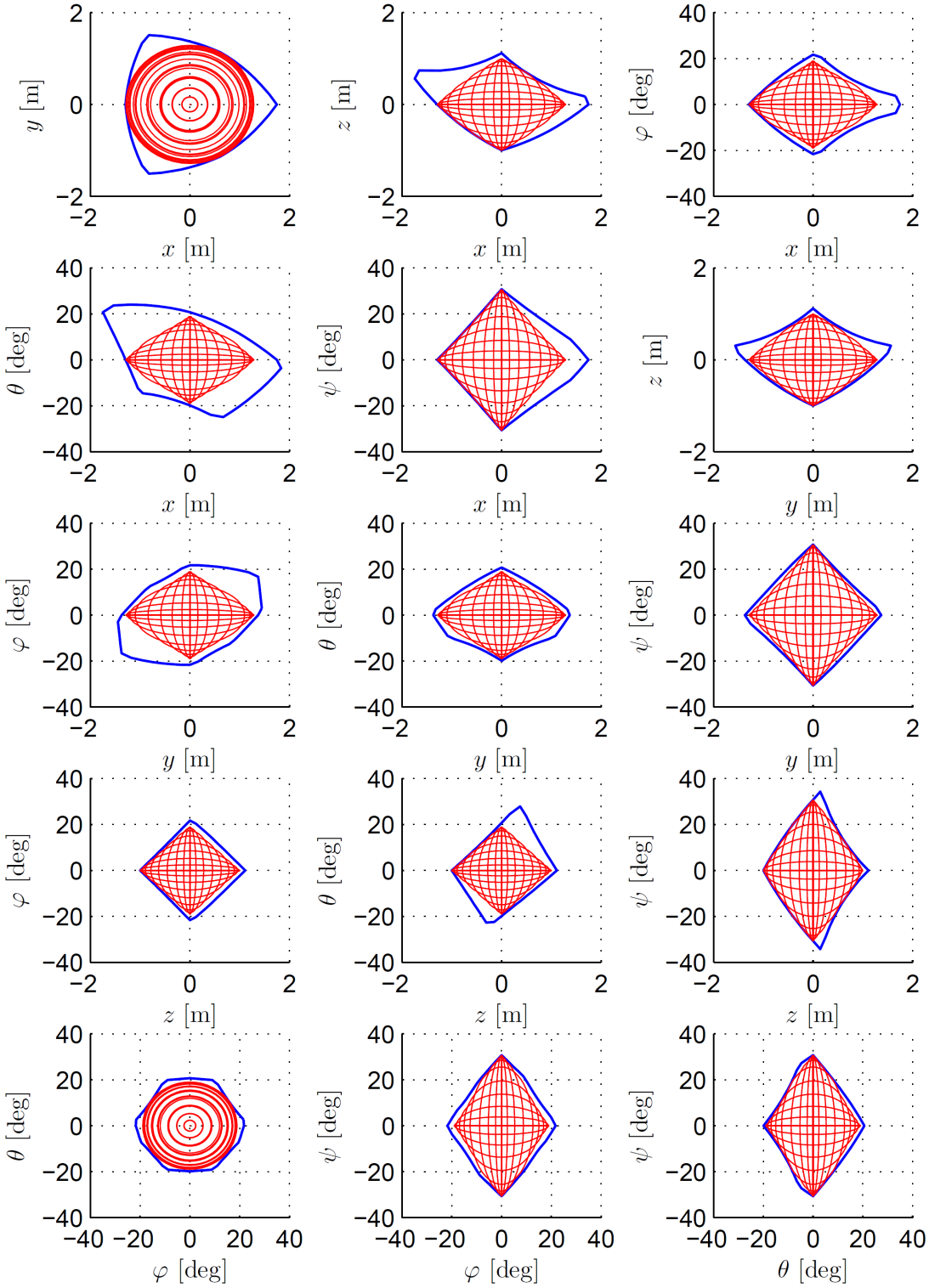


Figure 2-5: Local workspace boundaries of the Ampelmann Stewart platform (blue) and optimized hyper ellipsoids (red) [4]

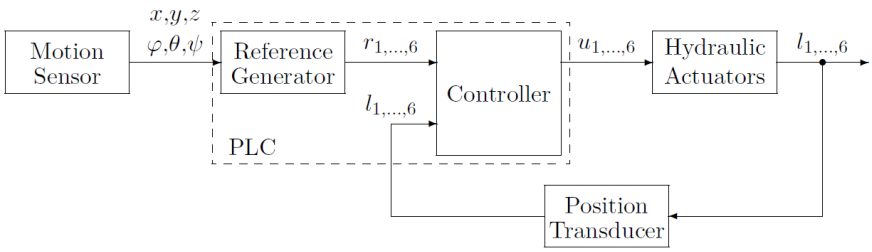


Figure 2-6: Block diagram of the Ampelmann motion control loop [4]

2-4 Motion control loop

The task of the Ampelmann system is to keep the top platform horizontal and motionless, in order to provide a safe access. This is accomplished by the control loop. It is schematically shown in Figure 2-6. An extended block scheme is shown in Figure 2-7.

The control loop will be described by explaining all the blocks in Figure 2-7 in more detail. First the vessel motions (the motions of the bottom frame) are measured by an Octans motion sensor. These measurements are sent to a Programmable Logic Controller (PLC), which is responsible for controlling the whole system. The reference generator filters out unwanted measurements and limits the vessel motions. It generates reference lengths for the actuators by Inverse Kinematics (IK), which are sent to the controller. The controller transforms the reference lengths to a input signal for the actuators. The lengths of the cylinders are measured with a position transducer and used for feedback in the controller.

2-4-1 Motion sensors

The ship motions are measured by the Octans motion sensor. This is done at a sampling rate of $f_m = 50$ [Hz]. Ship motion measurement are expected to contain frequencies in a range 0.1-0.5 [Hz]. It is located at the base frame and it contains three Fibre Optic Gyroscopes (FOG), three accelerometers and a real-time Digital Signal Processing (DSP) computer.

The FOGs measure rotations and time shifts of pulses, from which the three rotational velocities can be determined ($\dot{\varphi}$, $\dot{\theta}$ and $\dot{\psi}$). And integrating them results in φ, θ and ψ. The roll ψ is measured relative to the north direction, which is determined by projecting the earth's rotation vector on the plane orthogonal to the gravity vector. ψ, θ are measured relative to this plane. The North direction and gravity vector are found during a five minute alignment phase.

The three accelerometers measure the sum of the accelerations and apparent gravity. The DSP computer filters out the gravitational acceleration and the earth's rotation rate, which results in $\ddot{x}$, $\ddot{y}$ and $\ddot{z}$. Integrating twice results in x, y and z. But due to a small bias the double integration goes to infinity. And even when standing still an acceleration will be measured. To remove this bias a high pass filter with a cut-off frequency close to zero will be applied.

The Octans needs 10 minutes to start up after which x, y and z reach full accuracy.

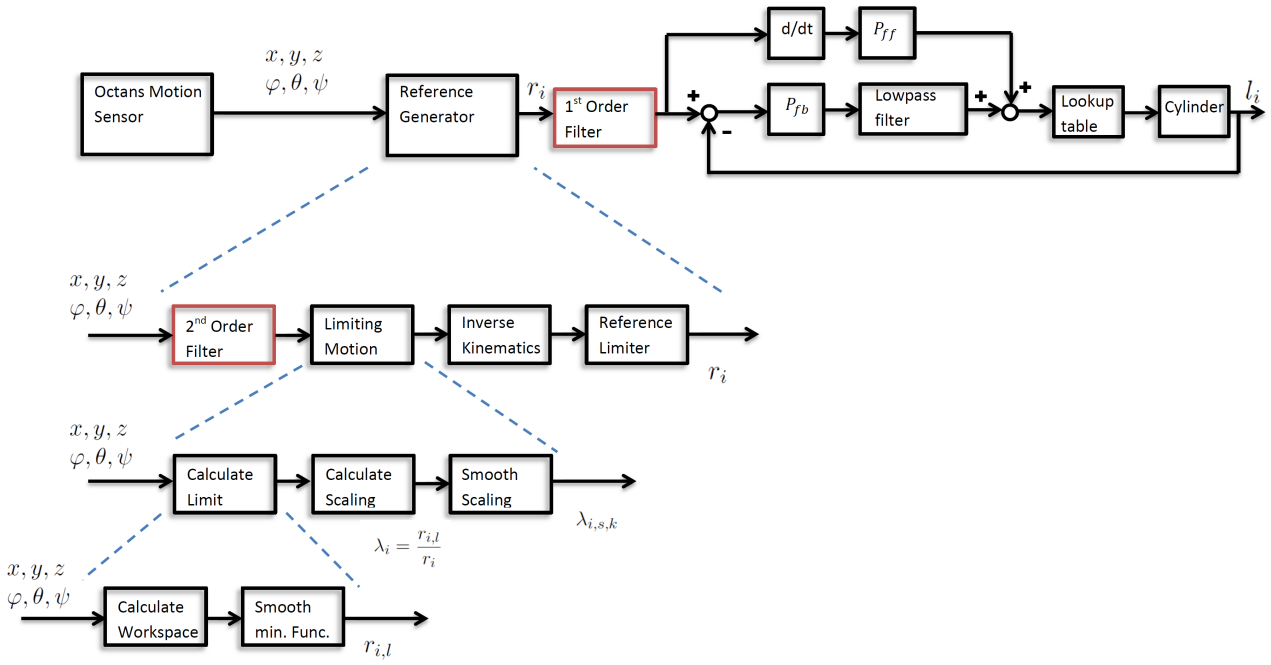


Figure 2-7: Extended block diagram

2-4-2 Reference generator

The reference generator needs to calculate a reference length for each actuator such that the transfer deck remain horizontal and motionless. It is schematically shown in Figure 2-7. The unwanted signals are filtered out, after which the vessel motion data is limited. The lengths are calculated with IK, see Section 2-2-1. Finally the actuator motions are limited and sent to the controller.

Filtering

The Octans motion measurement contain some unwanted motions. A second-order low-pass filter with cut-off frequency of 2 [Hz] is applied which filters out the high frequencies, it is highlighted in Figure 2-7. These frequencies are generated by machinery on board. The filter causes a noticeable delay in compensation, but it can not be removed since the system will vibrate heavily. The filter is explained in further detail in Section 2-5, also the lag is calculated.

Motion limiting

The measured vessel motions need to be limited in order to stay within the Ampelmann workspace. Residual motion of the top platform are undesired, but can sometimes be inevitable. Therefore residual motion are only allowed in three directions surge (x) , sway (y)

and heave (z). Since the gangway can compensate for those. The gangway will stay connected to a platform or an other vessel, by applying pressure.
The vessel motions are limited by the analytic workspace described in Section 2-3. The signals are not allowed to exceed the Ampelmann workspace, since this would lead to an abrupt stop at the end of its cylinder lengths which leads to high accelerations. The ship motions will be limited in four steps:

- Starting with the main purpose of the Ampelmann system, creating a stabilizing transfer deck, roll and pitch ($r_{\varphi\theta}$) are limited. In this way the top frame will remain horizontal.
- Secondly the compensation for rotations will be limited (r_{ψ}).
- Next the translational movements in the x and y plane are limited (r_{xy}).
- Residual motion in the heave (z) direction can most easily be dealt with by the gangway, so r_z will be compensated for last.

In practice most vessels contain a Dynamic Positioning (DP) system, which keeps the ship from drifting. In this case the yaw does not need to be compensated for and step two can be neglected.
It can be written in a general expression:

$$\begin{aligned} r_{i,l} &= \min(r_i, r_{i,m}), \qquad i = 1, \ldots, 4 \\ r_{i,m} &= 1 - \sum_{j=1}^{i-1} r_{j,l} \end{aligned} \tag{2-14}$$

Where:

$$r_{xy} + r_z + r_{\phi\theta} + r_{\psi} \leq 1 \tag{2-15}$$

or

$$r_3 + r_4 + r_1 + r_2 \leq 1 \tag{2-16}$$

Where r_i, m is the available work space for each direction. Where i is the order of limiting, so i is equal to 1 for $\varphi\theta$, 2 for ψ, 3 for xy and 4 for z. The limited value is denoted as r_i, l.
In order to provide a smooth transition towards the workspace boundary (to ensure no abrupt stopping which leads to high accelerations) the limited value $r_{i,l}$ is smoothed by a smooth minimum function. The function determines the minimum value between two values x_1 and x_2.

$$\begin{aligned} \operatorname{smin}(x_1, x_2, \gamma) &= x_1 - u(x_1 - x_2, \gamma) \\ u(x, \gamma) &= \begin{cases} 0 & \text{if } x < y \\ -\frac{x^4}{16\gamma^3} + \frac{3x^3}{8\gamma} + \frac{x}{2} + \frac{3\gamma}{16} & \text{if } -\gamma \leq x \geq \gamma \\ x & \text{if } x > y \end{cases} \end{aligned} \tag{2-17}$$

This can be rewritten to:

$$r_{i,l} = \operatorname{smin}(r_i, r_{i,m}, \gamma) = x_1 - u(x_1 - x_2, \gamma) \tag{2-18}$$

The value of γ can be interpreted as the percentage of the maximum value at which the function will be smoothed. It means that taking $\gamma = 0.5$ results in limitation if the DOF becomes

larger than 50% of its maximum value. Due to the smoothing factor the amount of residual motions will be increased but the top frame acceleration will decrease. Still this smoothing is not enough since it only creates a smooth transition between a motionless and a moving transfer deck. There is no smoothing to the motion trajectory once the transfer deck is moving.

Each DOF needs to be scaled with a scaling factor λ_i. Which ensures that the system operates within its workspace. This scaling factor is defined by:

$$\lambda_i = \frac{r_{i,l}}{r_i} \tag{2-19}$$

The DOF are limited in the following way:

$$\begin{aligned} \varphi_l &= \lambda_1 \varphi \\ \theta_l &= \lambda_1 \theta \\ \psi_l &= \lambda_2 \psi \\ x_l &= \lambda_3 x \\ y_l &= \lambda_3 y \\ z_l &= \lambda_4 z \end{aligned} \tag{2-20}$$

Further smoothing can be achieved by smoothing of λ_i. The rate of change of λ_i is limited. Equation 2-19 shows that λ_i increases close to the center and decreases close to the boundary, which results in top frame accelerations. These accelerations can be reduced by limiting the maximum rising rate:

$$\alpha_k = \frac{\lambda_{i,l,k-1} - \lambda_{i,k}}{T_s} \tag{2-21}$$

$$\lambda_{i,l,k-1} = \begin{cases} \lambda_{i,k} & \text{if } \alpha_k \leq \alpha_m \\ \lambda_{i,l,k-1} + \alpha_m T_s & \text{if } \alpha_k > \alpha_m \end{cases}$$

The rate of change for time step k is given by α_k, α_m is the maximum rate of changing and $\lambda_{i,l}$ is the limited λ_i value. In order to smooth the behavior of λ_l over time the recursive moving average (MA)-filter is used. In this way λ_s will respond less aggressive to changes in λ_m, the smoothing factor used is β.

$$\lambda_{i,s,k} = \beta \lambda_{i,l,k} + (1 - \beta) \lambda_{i,s,k_1} \tag{2-22}$$

The MA-filter has one major drawback, it creates a phase-lag on the λ_l signal. λ_s lags behind λ_l. The DOF are not scaled properly and can lead to a violation of the workspace when $\lambda_s > \lambda_l$. It does not necessarily have to become a problem, since the actual workspace is larger than the approximated. But still this is not guaranteed.
Increasing β reduces the lag but increases top frame accelerations. Now a buffer in the maximum value for each DOF is added, which reduces the chance of exceeding the boundary. Here ϵ is the buffer width.

$$r_{i,m} = 1 - \sum_{j=1}^{i-1} r_{j,l} - \epsilon \tag{2-23}$$

Inverse kinematics

After the signal is filtered and the motions is limited, the actuator lengths are calculated using IK. It can be calculated now the orientation of the top and base frame are known. This technique was explained in Section 2-2-1.

Reference limiter and filter

Since the estimated workspace described by the hyper-ellipsoids (Section 2-3) is an approximation, the reference lengths are limited to ensure that the stroke will not exceed the available stroke of the cylinders. This is done using an asymptote function. Furthermore the maximum velocity of an actuator is limited to 1.5 m/s^2. Last, before sending the reference signal to the controller a 1st order low-pass filter with a cut-off frequency of 5.0 [Hz] is applied. This filter is also highlighted in Figure 2-7.

2-4-3 Controller

The control input u_i to each actuator is determined by a velocity feed forward and a proportional feedback control, see Figure 2-7 and 2-8. In Figure 2-8 $F(s)$ represents the second order low-pass filter, F_l the look-up table and $G(s)$ is the actuator. The control inputs are the reference length $R(s)$ and the actual length $L(s)$ of the cylinder. A second order low pass filter is used to stabilize the control loop. Because of the non-linear characteristics of the actuators a look up table is used. Which maps the velocity $\dot{l}_i$ to the valve position.

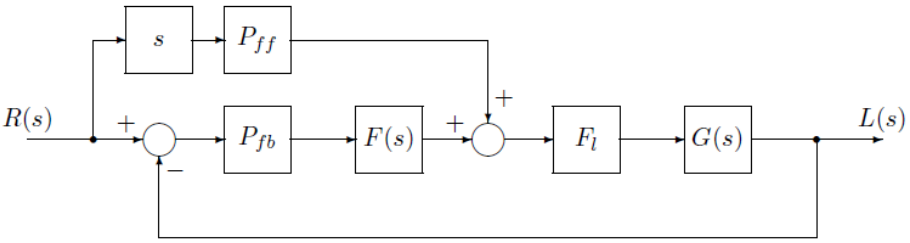


Figure 2-8: Control loop of an single actuator

2-4-4 Hydraulic actuator

The hydraulic actuator consists of a three-stage servo valve and an symmetrical hydraulic cylinder. The pressure supplied by a Hydraulic Pressure Unit (HPU) is kept constant at $P_s = 250$ [bar]. When a current i_i is applied to the valve this will result in a displacement l_i of the hydraulic actuator.

2-4-5 Position transducer

The last step is the position transducer, Figure 2-6. The position transducers measures the lengths from inside. These need to fed back to the controller. It consist of a magnetic rod (waveguide), which is placed at the bottom of the actuator such that the piston moves around this waveguide. A magnetic ring is attached to the piston. A current pulse is sent through the waveguide which causes the magnetic ring to elastically deform. The time between sending of the current pulse and the moment the mechanical wave reaches the waveguide can be used to determine the position of the ring.

2-5 Filter analysis

The filters in the control loop introduce delays in the Ampelmann system, which results in residual motions of the top platform, since the control signal will always lag the actual motions. The filters will be analyzed in order determine the amount of effect on the Ampelmann system. The working limits of the Ampelmann system are set to a significant wave height (H_s) of 2.5 [m] (which resembles to an amplitude of 1.25 [m]) and a mean wave period (T_z) of 4.5 [sec] (frequency of 0.2222 [Hz]) [2].
The filter which are present in the Ampelmann system are:

Octans filter Filters incoming data from the Octans motions sensor. It is a 2nd order low-pass filter with a cut-off frequency of 2.0 [Hz].

Cylinder filter Filters the set-point of the cylinder. It is a 1st order low-pass filter with a cut-off frequency of 5.0 [Hz].

Controller filter The 2nd order low-pass filter, which is placed after proportional gain in the controller. The main control action is done by the feed-forward gain, the low pass filter is placed on the P-gain for stability. so this filter does not have a large influence of the signal. Therefore this filter will not be taken in to account for this section.

The octans and cylinder filter are analyzed in the next subsections.

2-5-1 Octans filter

The transfer function of the Octans filter is the following:

$$H(s) = \frac{(2\pi F_c)^2}{s^2 + 2(2\pi F_c)s + (2\pi F_c)^2} \tag{2-24}$$

$$F_c = \frac{f \cdot t}{2} \tag{2-25}$$

Where $f = 2.0$ [Hz] is the cut-off frequency and $t = 0.02$ [sec] is the sampling time.
The bode plot is shown in Figure 2-9.

As seen in the phase-shift plot the phase shift increases as the frequency increases, so the maximum error will be found at the maximum frequency. For the Ampelmann system this

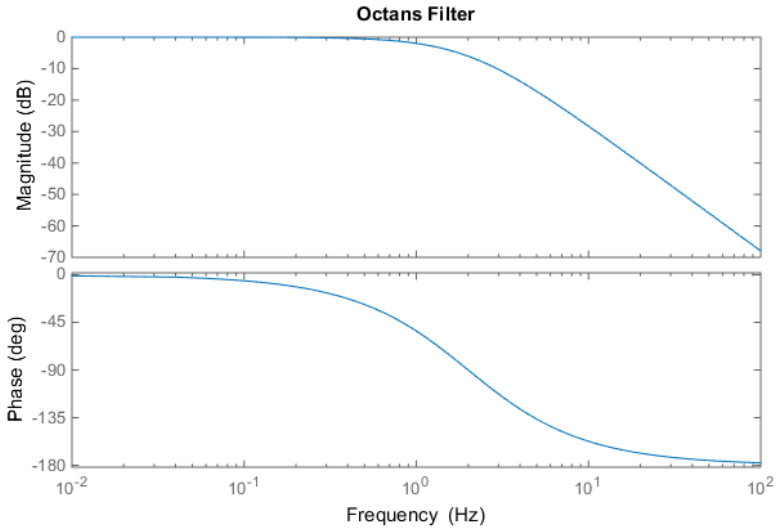


Figure 2-9: Bode plot of the Octans filter

workability limit is set to $\frac{1}{4.5}$ [Hz].
From Figure 2-9 the frequency of $\frac{1}{4.5}$ [Hz] resembles to a phase shift of -12.7°. This translates to a time delay of:

$$\frac{12.7^\circ}{360^\circ} \cdot 4.5 \approx 0.16 \text{ [sec]} \tag{2-26}$$

In order to gain knowledge about the error caused by the Octans filter a sine-wave is used as input. The maximum amplitude and frequency are used in order to find the maximum error. The following input is used:

$$x(t) = 1.25 \sin(\frac{1}{4.5} 2\pi t) \tag{2-27}$$

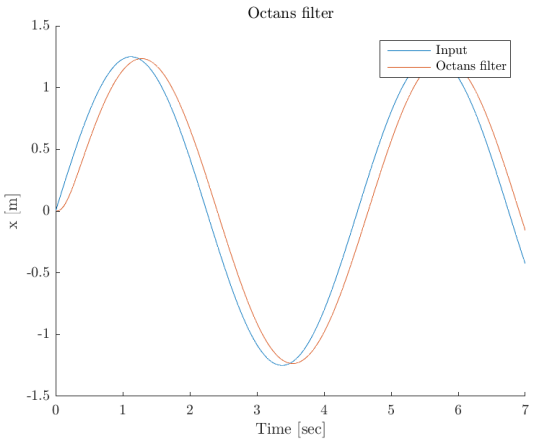


Figure 2-10: Filtered and unfiltered signal

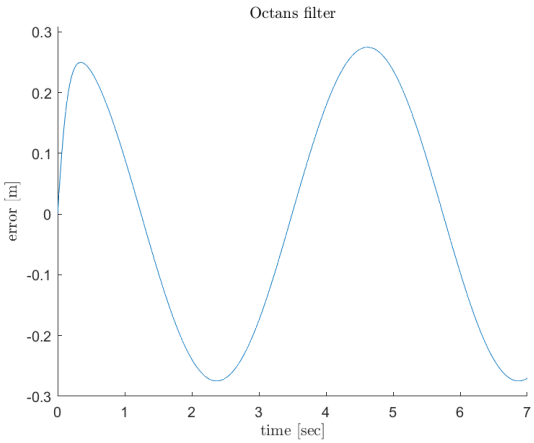


Figure 2-11: Error

In Figure 2-10 the input and the filtered signal are shown and in Figure 2-11 the error is shown.
The maximum error is:

$$\max |x(t) - x_f(t)| = 0.2748[m] \tag{2-28}$$

Where $x(t)$ is the input and $x_f(t)$ is the filtered signal. Then the error percentage is:

$$\frac{0.2748}{2.5} \approx 10.99\% \tag{2-29}$$

The time delay is approximately 0.16 [sec], which is also shown in Figure 2-10.

2-5-2 Cylinder filter

The transfer function of the cylinder filter is the following:

$$H(s) = \frac{(2\pi F_c)}{s + (2\pi F_c)} \tag{2-30}$$

$$F_c = \frac{f \cdot t}{2} \tag{2-31}$$

Where $f = 5.0$ [Hz] is the cut-off frequency and $t = 0.02$ [sec] is the sampling time.
The bode plot is shown in Figure 2-12.
The maximum phase shift for the Ampelmann system is found at the workability limit of $\frac{1}{4.5}$ [Hz]. From Figure 2-12 the frequency of $\frac{1}{4.5}$ [Hz] resembles to a phase shift of -2.54°. This translates to a time delay of:

$$\frac{2.54^\circ}{360^\circ} \cdot 4.5 \approx 0.032 \text{ [sec]} \tag{2-32}$$

The input is the filtered output of the Octans filter, the input for the Octans filter is described in Eq. (2-27).

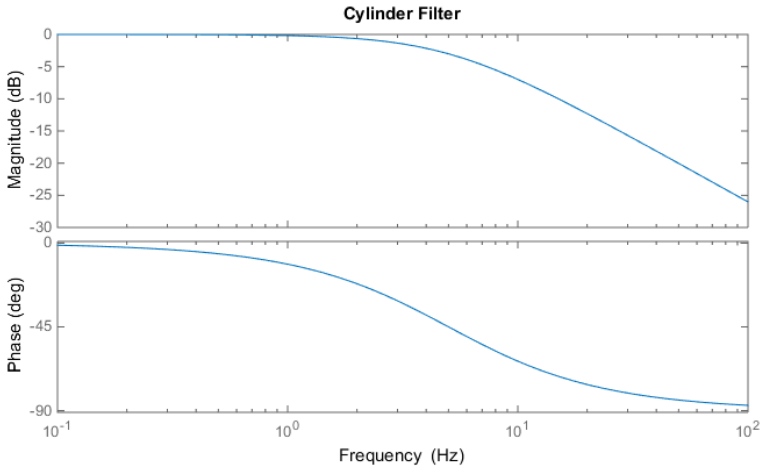


Figure 2-12: Bode plot of the Cylinder filter

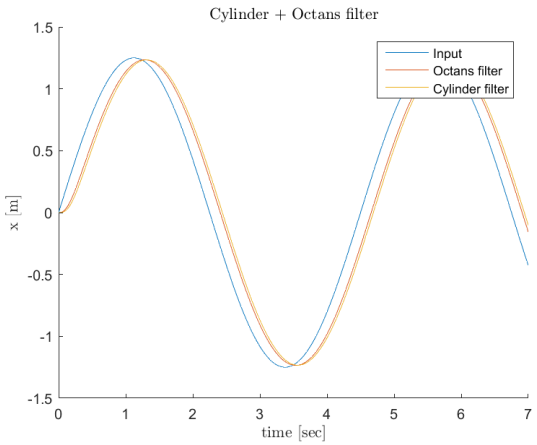


Figure 2-13: Filtered and unfiltered signal

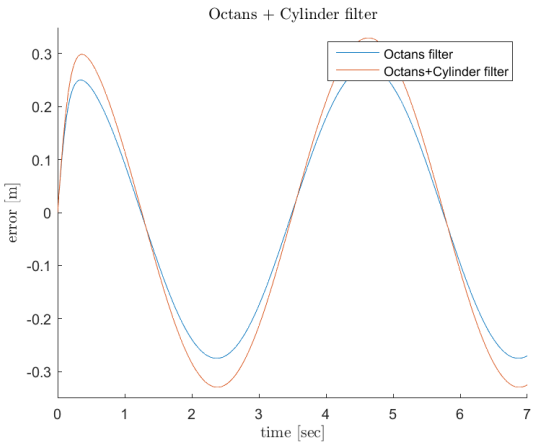


Figure 2-14: Error

The results are shown in Figure 2-13 and Figure 2-14. The following values are obtained: The maximum error after the Octans and Derivative filter is:

$$\max |x(t) - x_f(t)| = 0.3294[m] \tag{2-33}$$

Where $x(t)$ is the input and $x_f(t)$ is the filtered signal. Then the error percentage becomes:

$$\frac{0.3294}{2.5} \approx 13.18\% \tag{2-34}$$

The time delay, introduced by both filters, is approximately 0.192 [sec] and is also shown in Figure 2-13.

2-6 Simulations

The effects of the delays caused by the filters are discussed in the previous section. In this sections the influence of the limiting of the reference signal is shown.
In [4] a data set of vessel data is implemented to the motion control loop. The data set is of a sea state with $T_z = 5.0$ [s] and $H_s = 2.5$ [m]. The order for compensation is respectively $r_{\phi\theta}$, r_ψ, r_{xy} , r_z, which is explained in Section 2-4-2. The reference signal of heave ($r_4 = r_z$) is chosen for the simulation, since heave is compensated for last and will be the first DoF that will result residual motions.
The output of the reference generator is shown in Figure 2-15. Where:

- r_4 is the necessary workspace to fully compensate for the heave motions of the vessel.
- $r_{4,m}$ represents the maximum available workspace of the Ampelmann for the heave motions. This is the workspace that remains after the other DoF are compensated for:

$$r_{4,m} = 1 - r_{1,l} - r_{2,l} - r_{3,l} \tag{2-35}$$

- $r_{4,l}$ represents the limited reference signal, see Eq. (2-18).

$$r_{4,l} = \text{smin}(r_4, r_{4,m}, \gamma) \tag{2-36}$$

In this simulation the necessary workspace is always greater than the limited signal which results in residual motions in the heave direction. There is a gap between the limited workspace and the maximum workspace of the Ampelmann, which leaves room for improvement. In previous research [1] a solution is presented for this problem by the use of model predictive control (MPC).

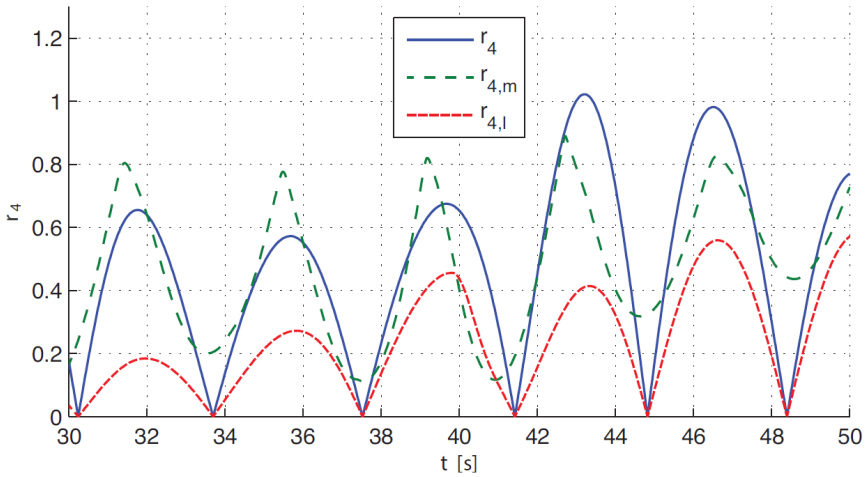


Figure 2-15: r_4, $r_{4,m}$ and $r_{4,l}$ during compensation of motion [4]

2-7 Conclusion

The Ampelmann can compensate for vessel motions in a sea state with a significant wave height of 2.5 [m]. In order to compensate for the wave motions, first an Octans motion sensor is used to measure the motions of the base frame. These incoming signals are filtered.
The workspace of the Ampelmann is highly non-linear and therefore is analytically described by hyper-ellipsoids. The approximation of the workspace is used to limit the signals. The Ampelmann system will for the described sea states fully be able to compensate for the three rotational components (φ, θ and ψ), since those are limited first. The translational directions (first x, y, next z) are limited, so residual motions will first occur in the Heave direction (z). But these motions can be compensated for by the gangway. After limiting the reference length to the cylinder is generated by smoothing the signal and using inverse kinematics. Inverse kinematic uses the relative position of the top and base frame to determine the lengths of the cylinders.
The filters which are present in the control loop cause delays in the system, which creates a noticeable lag during compensation. The delay can cause an error of approximately 13% of the amplitude of the wave.

Chapter 3

Data Analysis

3-1 Introduction

During this research real vessel data is used. Before a prediction model is derived the available data is analyzed. First, the data of the Octans motions sensor is shown and from the accuracy of the sensor the desired accuracy of the prediction can be determined. Next, the data set is considered as a discrete-time stochastic process and its probabilistic behavior is discussed. Last the interdependencies between the signals are tested.

3-2 Data

The data gathered by the Octans motion sensor consists of six separate signals. The data, which will be used for this research, is shown in Figure 3-1 for 100 seconds and in Figure 3-2 the whole data set is shown. The data is vessel motion data measured at the North Sea near Aberdeen, United Kingdom, in July 2015. The incoming information by the Octans motion sensor is logged. The sampling time is 0.02 [sec]. The resolution of the Octans motion sensor is 0.001 [m] or 0.01 [deg], that is why the data is filtered first to receive a smooth signal. The filter which is used is a 2nd-order low pass filter, with a cut-off frequency of 2 [Hz], the same filter as in the actual control loop.

Yaw is not compensated for in the Ampelmann system, since it is controlled by the dynamic positioning (DP) system of the vessel. The DP system tries to keep the yaw constant. Since the yaw signal is controlled the signal will not be natural and would be difficult to predict. For these reasons the yaw is not taken into account during this research.

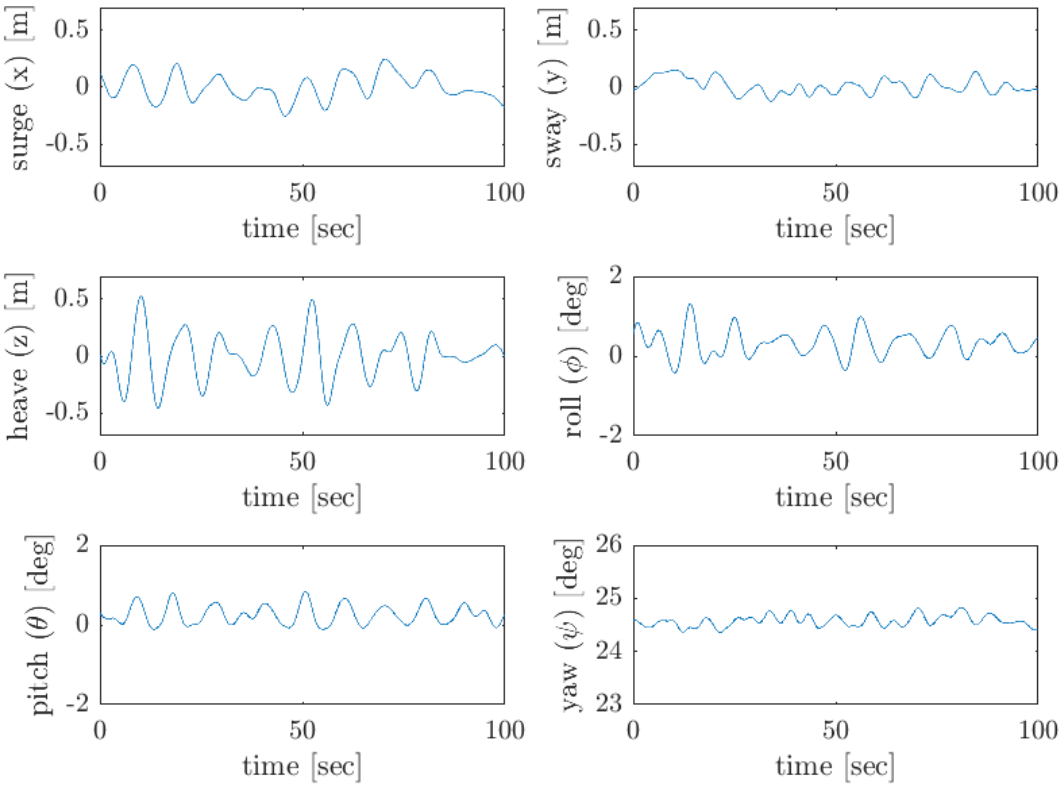


Figure 3-1: Wavedata of 100 seconds

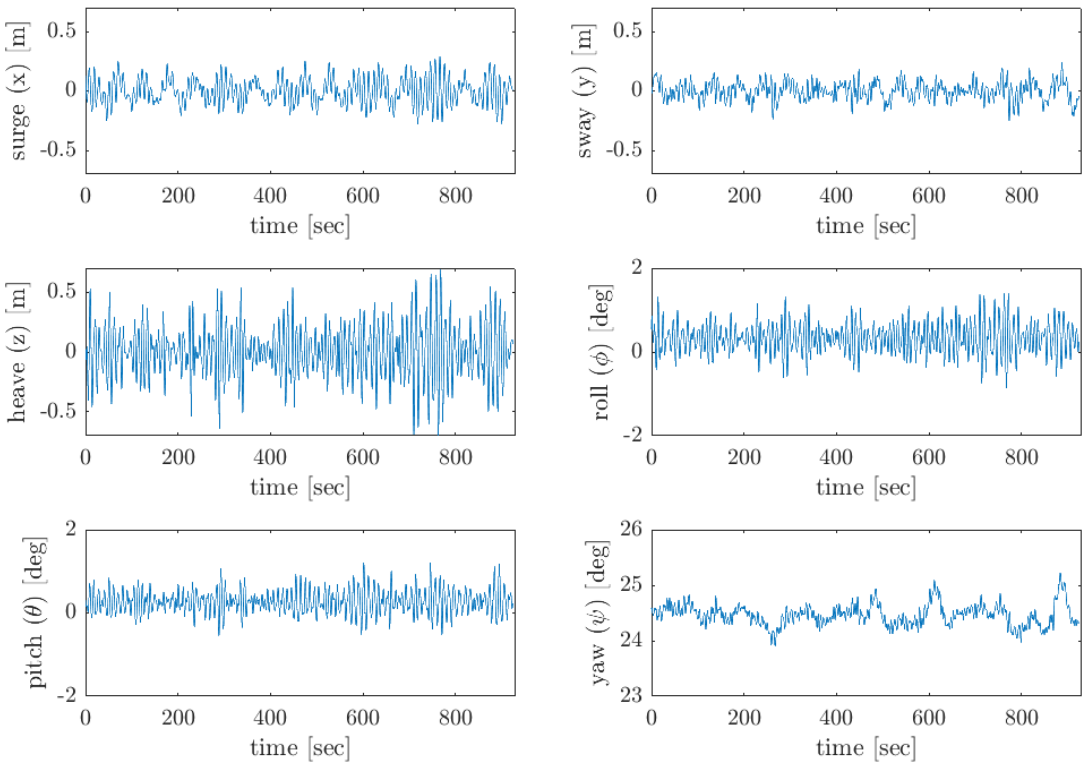


Figure 3-2: The total data set

The complete data set, as seen in Figure 3-2, contains approximately 50.000 data points. The data set will be split into 10 smaller data sets, these will be considered as 10 separate datasets. Then each of these 10 datasets will be divided in three parts; a training set, a validation set and a test set.
The parameters are determined (trained) with the first part of the data. The size of the model can be determined with the second set, the validation set. This set will prevent the model from over-fitting, when the prediction errors of both set are taken into consideration. Last the chosen model will be tested with the third data set, this is used to test the performance of the model compared to other models. This process is repeated 10 times. Two of the data sets are shown in Figure 3-3.

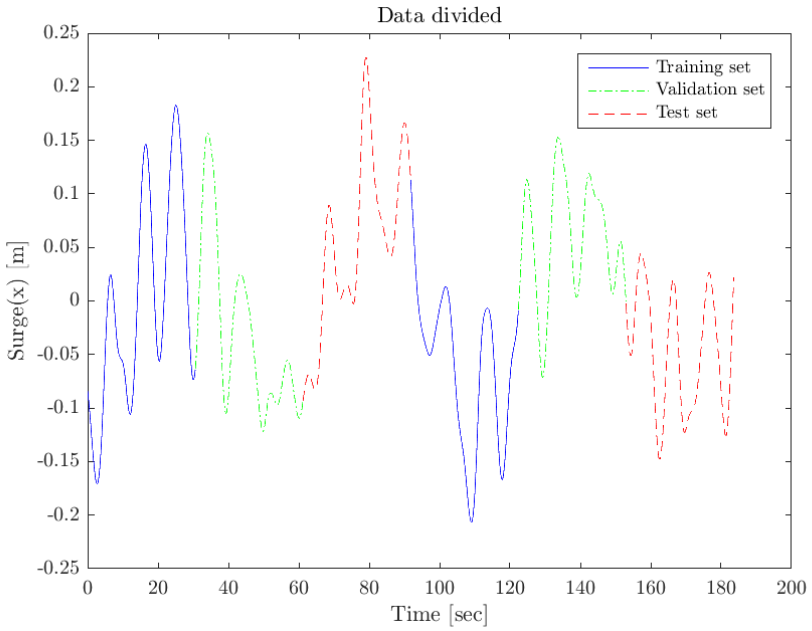


Figure 3-3: The data divided for first two smaller data sets

3-3 Accuracy

The Octans motion sensor has a resolution of 0.001 [m] or 0.01 [deg]. The accuracy of the translational movements (surge (x), sway (y) and heave (z)) is 0.05 [m] or 5 %, whichever is greater. The dynamic accuracy of the rotational movements (roll (ϕ, pitch θ and yaw ψ) is 0.01 [deg] [5].
In order to set a goal for the desired accuracy of the prediction, the accuracy of the Octans motion sensor is taken. Since it would be inefficient to predict more accurate than the sensor. So this means that the accuracy of the prediction is in the same range as the accuracy of the measurements. For surge (x), sway (y) and heave (z) this results in an accuracy of 0.05 [m]. For roll(ϕ) and pitch (θ) it resembles an accuracy of 0.01 [deg], which is a very tight constraint. Therefore for the rotations an accuracy of 5 % is also taken, similar to the translation signals.

3-4 Stationarity

The vessel motion data can be referred to as a discrete-time stochastic process or a time series process. A time series process is a set of random variable $\{X_t, t \in T\}$ where T is the set of time. Assumed is that each random variable X_t is distributed according some univariate distribution function F_t.
Furthermore a process can be stationary, which means that the probabilistic character of the series must not change over time. A discrete-time random signal $x(k)$ is stationary if the joint cumulative function of any finite number of samples does not depend on the placement of the time-origin, that is [6],

$$F_{x(k_0),x(k_1),\ldots,x(k_{N-1})}(\alpha_0, \alpha_1, \ldots, \alpha_{N-1}) = F_{x(k_{0+\tau}),x(k_{1+\tau}),\ldots,x(k_{N-1+\tau})}(\alpha_0, \alpha_1, \ldots, \alpha_{N-1}) \quad (3\text{-}1)$$

for all time-shifts $\tau \in \mathbb{Z}$.
In practice however there is no explicit knowledge of the latent time series process. But it can be verified if the time series process shows the following three properties:

- Its mean is constant: $m_x(k) = E[x(k)] = m_x$
- Its auto-correlation function $R_x(k, l)$ depends only on the lag $k - l$
- Its variance is finite $\text{var}[x(k)] = E[(x(k) - m_x)^2] < \infty$

Then a time series is called wide sense stationary (WSS) or weak stationary. Where the auto-correlation function of a random variable X_t is defined as:

$$R_x(k, l) = \frac{E[(X_k - \mu_k)(X_l - \mu_l)]}{\sigma_k \sigma_l} \quad (3\text{-}2)$$

This is a dimensionless measure for the linear association between two random variables. Since for stationary series, it is required the moments to be non-changing over time, the auto-correlation is only a function of the lag . Where the lag is defined as $\tau = k - l$.
The auto-correlation function $R_x(\tau)$ of a WSS random signal x(k) has the following basic properties. It is symmetric in its own argument τ, the auto-correlation function at $\tau = 0$ is greater than or equal to zero ($R_x(0) \geq 0$) and the maximum of the auto-correlation function occurs at $\tau = 0$.

The stationary property is important since the linear models: Auto Regressive (AR)(p) and Auto Regressive Moving-Average (ARMA)(p,q) model, which will be discussed in Chapter 4, must only be fitted to stationary time series. The AR(p) and ARMA(p,q) models are based on the assumption that the time series can be written as:

$$X_t = \mu_t + E_t \quad (3\text{-}3)$$

Hereby, μ_t is the conditional mean of the series ($\mu_t = E[X_t | X_{t-1}, X_{t-2}, \ldots]$), which is a function of past instances of the series and/or the past innovations. And E_t is a disturbance term. For the Wavelet Neural Network (WNN), which will be discussed in Chapter 5, stationarity is not a requirement.
For the (relatively) short period of time, which is considered, there is no reason to assume

that the probabilistic character of the process will change. As expected the data shows cyclic behavior. No trends or seasonal effect are visible, this would be different if wave data with a time length of a day or a year would be used. In Figure 3-4 the auto correlation function are shown. Clearly the cyclic behavior is visible, but the auto-correlation decays approximately exponentially, which is an indication a stationary process [7]. Furthermore it satisfies the basic properties mentioned earlier.

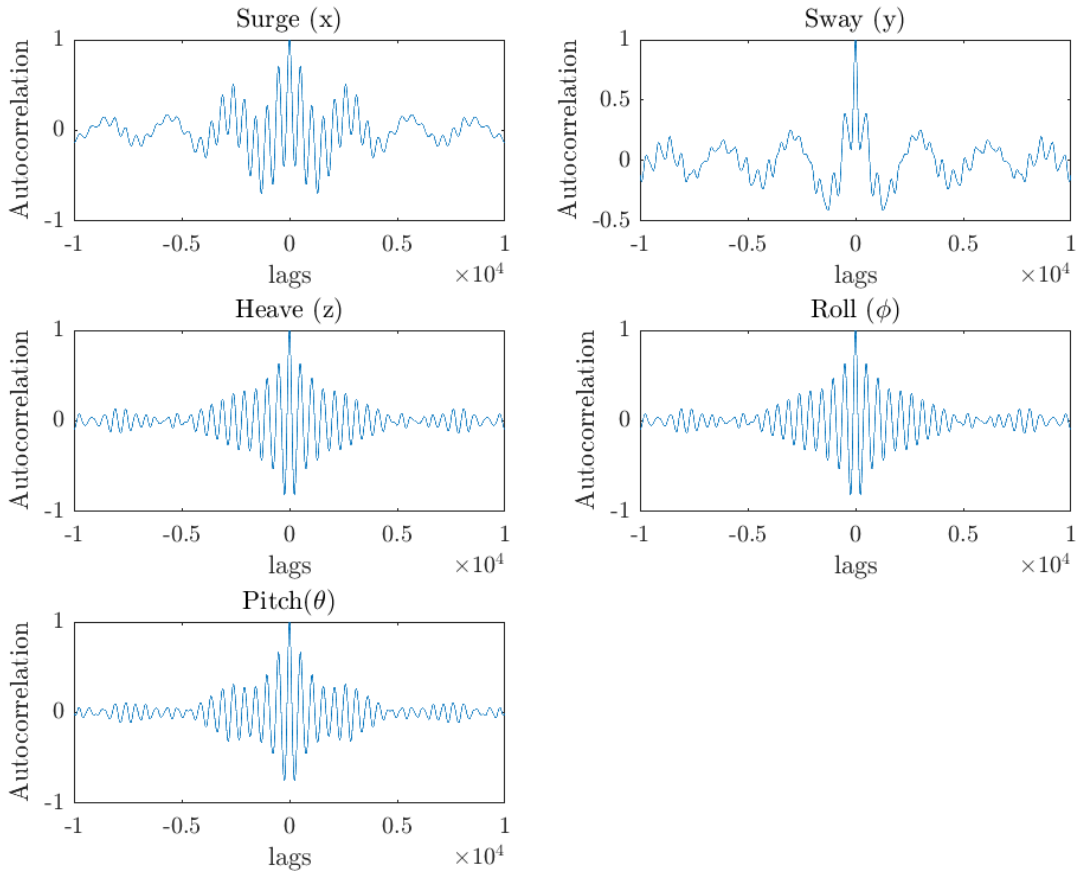


Figure 3-4: Autocorrelation functions

3-5 Multivariate time series

In Figure 3-5 the cross correlations of the 5 signals are shown. The cross correlation is the measure that describes the amount of linear dependence between two time series.
The highest correlation can be seen between roll (ϕ) and pitch (θ). This can be caused by the fact that these are both rotational signals. Also a high correlation is seen between heave (z) and roll (ϕ) and heave (z) and pitch (θ). From which we could assume that the signals and therefore the prediction models can be interdependent. For example, the next value for heave (z) could not only depend on its own previous values but also on those from roll (ϕ). These properties can be used in the prediction model.

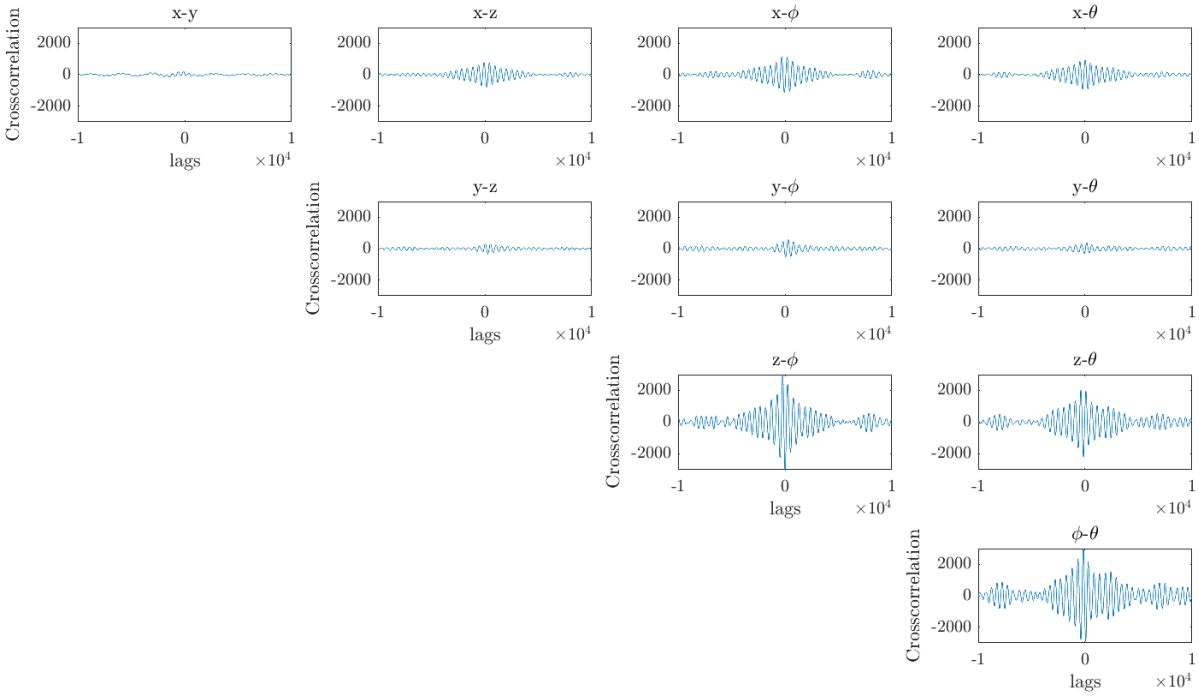


Figure 3-5: Cross correlations

3-6 Conclusion

The data set which is used during this research is measured vessel motions at the North Sea near Aberdeen, United Kingdom in July 2015. The sampling time is 0.02 [sec] and the resolution of the Octans motions sensor is 0.001 [m] or 0.01 [deg]. The data set is split up in 10 smaller data set which are considered as 10 separate data sets. Each data set is dived in a training set, validation set and a testing set.
The accuracy of the Octans motions sensor for the translational movements is 0.05 [m] or 5%, whichever is greater. For the rotational movements the accuracy is 0.01 [deg]. The goal for the accuracy of the prediction is therefore also set to 5% of the amplitude.

The vessel motion data can be referred to as a discrete-time stochastic process. From the auto-correlation functions of the data it can be assumed that the time series is a stationary process, which means that the probabilistic behavior of the series does not change over time. This property is important for the AR and ARMA model, which will be explained in the next chapter. The stationary property is not required for the WNN, which is explained in Chapter 5. The rotational signals (ϕ and θ) have a high correlation, which indicates that a multivariate model could be beneficial.

Chapter 4

Linear Model

4-1 Introduction

In this chapter the linear predictor models are use as a representation of the dynamical system. The problem can be seen as a pure time-series problem and no *a priori* knowledge is taken in to account.
In the first section an overview will be given of several black box model structures and the candidate functions for the process to be modeled are chosen.
Next the choice of an appropriate model model order performed through the application of the classic criteria, Akaike information criterion (AIC), which is slightly modified for a multi-step ahead prediction error.
The main motivation of the time series analysis is to predict future observations in the series, therefore in the next section it is discussed how to generate forecasts with the chosen models. After this, the achieved results of the models are shown and the validation is done.
Last three methods are explained to reduce the computational power of the prediction method in order to be able to predict real time.

4-2 Model structures

The general discrete-time polynomial model to represent a system takes the following form:

$$A(q)y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}e(k) \tag{4-1}$$

With input vector $u(k)$, output vector $y(k)$ and the disturbance $e(k)$. Where $e(k) \in \mathbb{R}$ is a zero-mean white-noise sequence that is independent from $u(k) \in \mathbb{R}$. The polynomials $A(q)$, $B(q)$, $C(q)$, $D(q)$ and $F(q)$ are polynomials, which are expressed in the time-shift operator q with respective degrees n_a,n_b,n_c,n_d and n_f. The coefficients of the polynomials are collected in a parameter vector θ. In Figure 4-1 the polynomial model is schematically shown
From Eq. (4-1) different model structures can be defined:

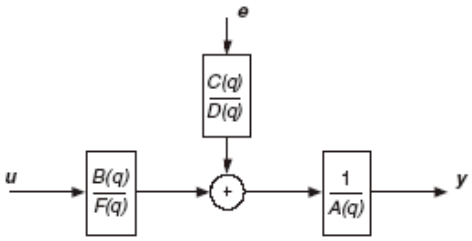


Figure 4-1: Schematic overview

AR The Auto Regressive (AR) model is based on a linear combination of the past observations and a disturbance term $e(k)$, according to the following equation:

$$A(q)y(k) = e(k) \tag{4-2}$$

ARX The Auto Regressive with Exogenous inputs (ARX) input model. The model is similar to the AR model, but it includes an exogenous input in the model.

$$A(q)y(k) = B(q)u(k) + e(k) \tag{4-3}$$

FIR The Finite Impulse Response (FIR) model structure is a special case of an ARX model structure with A(q)=1.

$$y(k) = B(q)u(k) + e(k) \tag{4-4}$$

ARMA The Auto Regressive Moving-Average (ARMA) model. The model is based on a linear combination of the past observations and a linear combination of the current and past innovation terms, $e(k)$.

$$A(q)y(k) = C(q)e(k) \tag{4-5}$$

ARIMA The Auto-Regressive Integrated Moving Average (ARIMA) model. This model can be used if the data is non-stationary, for example when seasonal trends are visible. The 'integrated' part can reduce the non-stationarity.

$$A(q)(1-q)y(k) = C(q)e(k) \tag{4-6}$$

ARMAX The Auto Regressive Moving-Average with Exogenous inputs (ARMAX) model. This model is similar to the ARMA model, but it includes an exogenous input in the model.

$$A(q)y(k) = B(q)u(k) + C(q)e(k) \tag{4-7}$$

OE The Output-Error (OE) model. No parameters are used for modeling the disturbance characteristics.

$$y(k) = \frac{B(q)}{F(q)}u(k) + e(k) \tag{4-8}$$

BJ The Box-Jenkins (BJ) structure is a complete model with disturbance properties modeled separately from system dynamics.

$$y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}e(k) \tag{4-9}$$

The choice for applying a specific model structure in an identification problem can be a very important issue. Choosing the wrong structure, may lead to identified models that are "bad" [8]. As explained in Section 2-4 the waves can be seen as a (wide sense) stationary time series process, which means that the probabilistic character of the series must not change over time. For pure time series problem the system has no input. For this reason the AR and ARMA model seem the best fit for describing the signals, if the signals are (wide sense) stationary. When the data shows signs of non-stationarity the ARIMA model can be used.

The AR(p) model assumes that $y(k)$ depends on past values of $y(k)$ and the disturbance term $e(k)$, which comes from a White Noise process. It is required to be an innovation, i.e. that it is stochastically independent of $y(k-1), y(k-2), \ldots$ [7]. Here p is the model order.

$$A(q)y(k) = e(k) \tag{4-10}$$
$$y(k) + a_1 y(k-1) + \ldots + a_p y(k-p) = e(k) \tag{4-11}$$
$$y(k) = -a_1 y(k-1) - \ldots - a_p y(k-p) + e(k) \tag{4-12}$$

The ARMA(p,q) model assumes that the $y(k)$ depends on the past values of $y(k)$ as well as the current and previous innovation terms of $e(k)$. Here p and q determine the model order.

$$A(q)y(k) = C(q)e(k) \tag{4-13}$$
$$y(k) + a_1 y(k-1) + \ldots + a_p y(k-p) = e(k) + c_1 e(k-1) + \ldots + c_q e(k-q) \tag{4-14}$$
$$y(k) = -a_1 y(k-1) - \ldots - a_p y(k-p) + e(k) + c_1 e(k-1) + \ldots + c_q e(k-q) \tag{4-15}$$

4-2-1 One-step ahead prediction

The general one-step ahead prediction error $e(k)$ can be denoted as:

$$e(k) = y(k) - \hat{y}(k|k-1) \tag{4-16}$$

Which results in the following one-step ahead prediction for the AR and ARMA model [6].

$$\text{AR:} \quad e(k) = y(k) - \hat{y}(k|k-1) = A(q)y(k) \tag{4-17}$$

$$\hat{y}(k|k-1) = [1 - A(q)]y(k) \tag{4-18}$$

$$\tag{4-19}$$

$$\text{ARMA:} \quad e(k) = y(k) - \hat{y}(k|k-1) = \frac{A(q)}{C(q)}y(k) \tag{4-20}$$

$$\hat{y}(k|k-1) = \frac{C(q) - A(q)}{C(q)}y(k) \tag{4-21}$$

4-3 Model order

The order of the model, p and q, determine the size of the model, i.e. the number of previous observations in the time series which are taken into account. In general a larger model results in better performance, up to a certain point when the model will overfit. But a larger model also increase the computational power necessary to determine the parameters, which is an important aspect for real-time prediction.
As mentioned in Section 2-2, each of the ten data sets is divided in three parts. The first part, training set, is used to determine the parameters θ. The second part, validation set, is used to choose an appropriate model order. First the order p of the AR(p) model is determined, next the orders p and q of the ARMA(p,q) model are chosen.

Multiple criteria can be used to determine the model order. Prediction-error methods minimize a cost function given by

$$J_N(\theta) = \frac{1}{N-p-1} \sum_{k=p+1}^{N} (y(k) - \hat{y}(k|k-1,\theta))^2 \tag{4-22}$$

Where $y(k)$ is the real output and $\hat{y}(k)$ is the one-step ahead prediction. N is the length of the data set, and p is the model order. Since the main motivation behind the model is to make a multi-step ahead prediction, the cost function is slightly modified and the model will now be evaluated for a 50-step ahead prediction, which resembles 1 second. This value is chosen for comparison.

$$J_N(\theta) = \frac{1}{N-p-50} \sum_{k=p+50}^{N} (y(k) - \hat{y}(k|k-50,\theta))^2 \tag{4-23}$$

A scaled version of the cost function $J_N(\theta)$ that is often used for assessing the quality of a model is the variance accounted for (VAF). The VAF is defined as:

$$\mathrm{VAF}(y(k),\hat{y}(k,\theta)) = \max\left(0, \left(1 - \frac{\frac{1}{N-p-50}\sum_{k=p+50}^{N} \mathrm{var}(y(k) - \hat{y}(k|k-50,\theta))}{\frac{1}{N-p-50}\sum_{k=p+50}^{N} \mathrm{var}(y(k))}\right) \cdot 100\%\right) \tag{4-24}$$

The VAF has a value between 0% and 100%, the higher the VAF, the lower the prediction error and the better the model.

Although, focusing on the value of the cost function $J_N(\theta)$ alone can be misleading. The cost-function will decrease with a higher model order m since the model is more flexible. Therefore the complexity of the model should be penalized in addition to the value of the prediction error. One often-used modification of the cost function is the classical criteria Akaike's Information criterion (AIC) [9]:

$$J_N^{\mathrm{AIC}} = (N-\theta-50)\log\left(\frac{\sum_{k=\theta+50}^{N}(y(k) - \hat{y}(k|k-50,\theta))^2}{N-\theta-50}\right) + 2\theta \tag{4-25}$$

Where N is the number of elements in the data set and θ is the number of free parameters, so $\theta = p + q$.
Another method to prevent overfitting, which is choosing a model order to high, is the cross-validation test [8]. When a model order is chosen too high it will model the specific realization of the noise disturbance. So the the first data set will be used for identification of the parameters $\hat{\theta}_N$, and the other part will be used to calculate the loss function:

$$Y^N = Y^{(1)} \quad Y^{(2)} \tag{4-26}$$

$$\hat{\theta}_N^{(1)} = \arg\min_{\theta} J_N(\theta, Y^{(1)}) \tag{4-27}$$

$$J_N(\hat{\theta}_N^{(1)}, Y^{(2)}) = \frac{1}{N^{(2)}} \sum_{k=1}^{N^{(2)}} e(k, \hat{\theta}_N^{(1)})^2 \tag{4-28}$$

In this case when the model order $\hat{\theta}_N^{(1)}$ has been chosen too high, this will result in an increase of the cost-function $J_N(\hat{\theta}_N^{(1)}, Y^{(2)})$. This technique is applied because a separate training and validation set are used. So the parameters are trained with a different data set than the data set which is used for validation.

4-3-1 AR

The auto regressive model (Eq. (4-29)) will be evaluated first. The parameter polynomial $A(q)$ will be estimated by the least squares method. Which minimizes the cost function in Eq. (4-22).

$$A(q)y(k) = e(k) \tag{4-29}$$

$$y(k) + a_1 y(k-1) + \ldots + a_p y(k-p) = e(k) \tag{4-30}$$

Where $y(k)$ is the output of the model, which can be any of the 5 Degrees of Freedom (DoF) (yaw neglected).
For all of the 10 data sets the parameters are trained (with the training set) and the AIC and VAF value are determined by using the validation set. In Figure 4-2 the AIC values are shown for surge (x), for $p = 1, \ldots, 10$ and for all 10 data sets. These are comparative values, the value says nothing about the goodness of the fit. The model with the smallest AIC value is the best model compared to the other models. The VAF-values for $p = 1, \ldots, 10$ are determined and shown in Figure 4-3 for surge (x). The AIC- and VAF-values for the other DoF are shown in Appendix A-1.

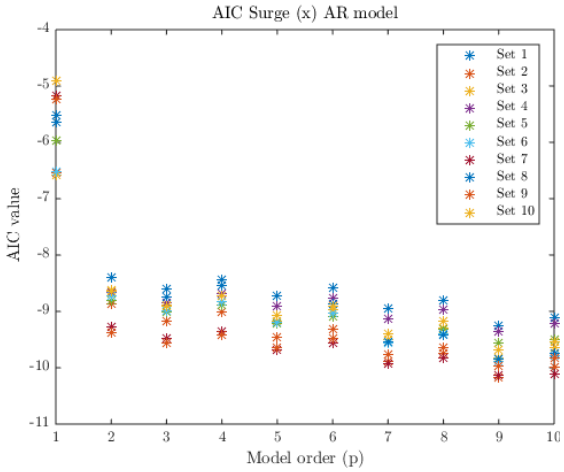


Figure 4-2: AIC, 50-step ahead

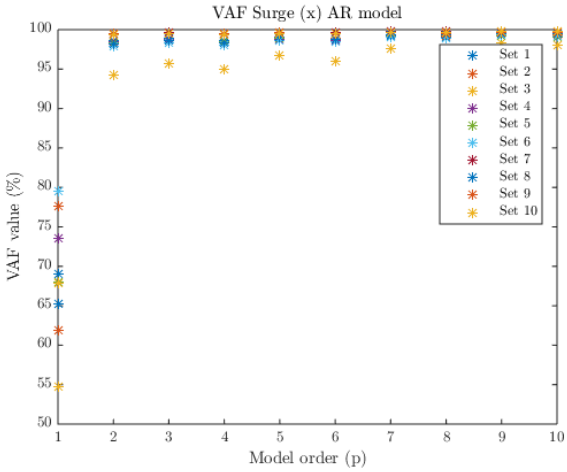


Figure 4-3: VAF, 50-step ahead

Both figures do not show a clear best choice. Clearly for all signals model order 1 is not a good choice. For all signals, the VAF and AIC values do (approximately) not become better after model order $p = 7$. So it is most reasonable to choose model order $p = 7$, since a smaller model order is the less computational demanding.

4-3-2 ARMA

An ARMA(p,q) model is represented as:

$$A(q)y(k) = C(q)e(k) \tag{4-31}$$

$$y(k) + a_1 y(k-1) + \ldots + a_p y(k-p) = e(k) + c_1 e(k-1) + \ldots + c_q e(k-q) \tag{4-32}$$

The parameters of $A(q)$ and $C(q)$ are determined by minimizing the cost function of Eq. (4-22).
Using the validation data set, the VAF and AIC values are calculated for $p = 1, \ldots, 5$ and $q = 1, \ldots, 5$. The ARMA model requires generally fewer parameters than a pure AR or MA model. In Figure 4-4 and Figure 4-5 the AIC and VAF values of surge (x) for a 50-step ahead prediction for all values of p and q are shown. The AIC and VAF values which are shown are the average value of the 10 data sets.
This is also done for the other 4 signal for a 50-step ahead prediction. The results are shown in Appendix A-2. The results of surge are taken as a representation of all signals, since the behavior is similar.

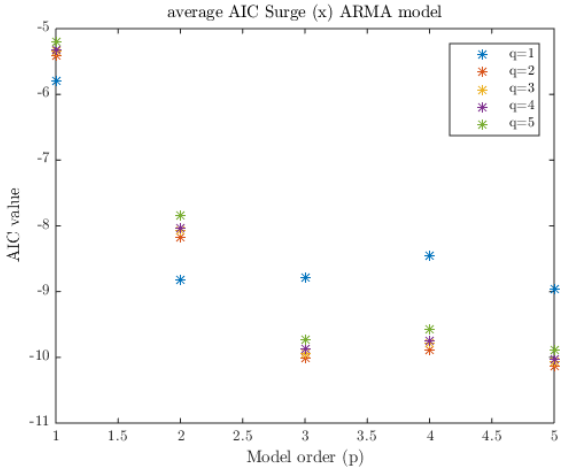


Figure 4-4: AIC values, 50 step ahead

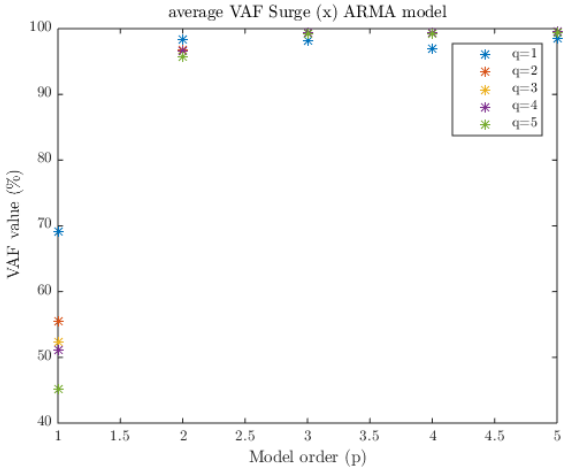


Figure 4-5: VAF values, 50 step ahead

The ARMA(3,2) model and ARMA(5,2) model have the smallest AIC value, therefore these models are chosen for testing and comparing its performance.
The ARMA model does not present significantly different modeling capabilities than pure AR-models. In practice however, to obtain a parsimonious parametrisation, it is sometimes necessary to include both autoregressive and moving average terms [10]. The ARMA(3,2) model reduces the number of parameters compared to the AR(7) model.

4-4 Forecasting

The main motivation of the previously found stochastic model is to predict future observations in the time series. Forecasting is an extrapolation in the time domain, which can lead to false conclusion. Three main factors can contribute to an unreliable forecast. First, the data generating process should not change over time. Second, there is no guarantee if the correct model is chosen. Last, an additional uncertainty arises from the parameter estimation method. However if the signal is stronger than the noise and a short prediction horizon is used, accurate forecasting is possible [7].

4-4-1 AR

The AR(p) model is defined as:

$$y(k) + a_1 y(k-1) + \ldots + a_p y(k-p) = e(k) \tag{4-33}$$

Where $y(k)$ is the data, $a_1, \ldots, a_p$ are the parameters and $e(k)$ is the innovation term. The forecasted value of the AR model is based on the conditional expectation of y_{k+m} as seen in the following equation:

$$\hat{y}(k+m) = E[y_{k+m}|y(1)\ldots y(k)] \tag{4-34}$$

$$= -\sum_{i=1}^{p} a_i y_{k+m-i} \tag{4-35}$$

Where m is the prediction step. Since $e(k)$ is assumed to be white noise, the expected value of $e(k+m)$ is zero. The one step ahead prediction the reads as:

$$\hat{y}(k+1) = -a_1 y(k) - \ldots - a_p y(k-p+1) \tag{4-36}$$

To obtain the next prediction, $\hat{y}(k+2)$, the forecasted value, $\hat{y}(k+1)$, is plugged in:

$$\hat{y}(k+2) = -a_1 \hat{y}(k+1) - \ldots - a_p y(k-p+2) \tag{4-37}$$

This procedure is repeated until the forecasting horizon is reached.

4-4-2 ARMA

The ARMA(p,q) model is defined as:

$$y(k) + a_1 y(k-1) + \ldots + a_p y(k-p) = e(k) + c_1 e(k-1) + \ldots + c_q e(k-q) \tag{4-38}$$

Where $y(k)$ is the data, $a_1, \ldots, a_p$ and $c_1, \ldots, c_q$ are the parameters and $e(k)$ is the innovation term. The forecasted value of the ARMA model is based on the conditional expectation of y_{k+m} as well as the expectation of the innovation terms as seen in the following equation:

$$\hat{y}(k+m) = -\sum_{i=1}^{p} a_i E[y(k+m-i)|y(1)\ldots y(k)] + \sum_{j=1}^{q} c_j E[e(k+m-j)|y(1)\ldots y(k)] \tag{4-39}$$

The expectation of the future values of $e(k)$ are zero, because of the white noise assumption. Which leads to the one-step ahead prediction for the ARMA model:

$$\hat{y}(k+1) = -a_1 y(k) - \ldots - a_p y(k-p+1) + c_1 e(k) + \ldots + c_q e(k-q+1) \tag{4-40}$$

The next prediction is again obtained by plugging in $\hat{y}(k+1)$.

$$\hat{y}(k+2) = -a_1 \hat{y}(k+1) - \ldots - a_p y(k-p+2) + c_2 e(k) + \ldots + c_q e(k-q+2) \tag{4-41}$$

The initial values of $e(k)$ need to be initialized and are found using:

$$e(k) = y(k) + a_1 y(k) + \ldots + a_p y(k-p) - c_1 e(k-1) + \ldots + c_q e(k-q) \tag{4-42}$$

4-5 Results

In the previous section the forecasting principle is explained. The AR(7), ARMA(5,2) and ARMA(3,2) models were selected in Section 4-3. In order to compare the models, the models are tested with the third part of each data set, the test set. The parameters are determined (trained) with the first part of each data set, the training set. At every time-step a 50-step ahead prediction is made, see Figure 4-6.

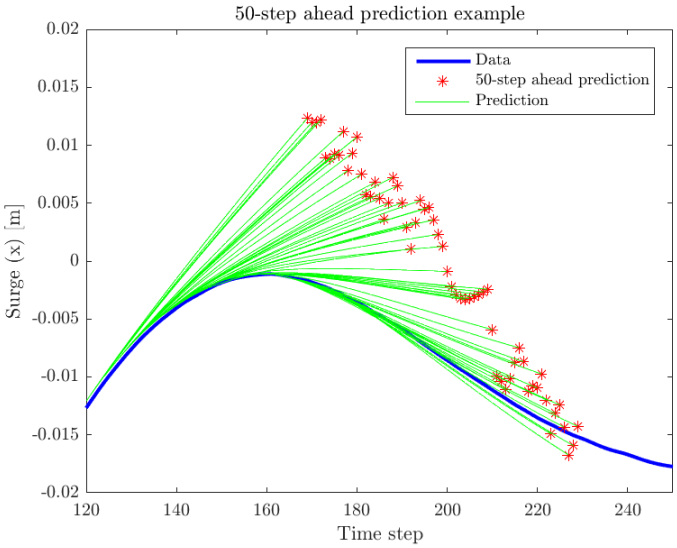


Figure 4-6: Example prediction

For every 50-step ahead prediction the average Root mean squared error (RMSE) over 10 data sets is determined:

$$\text{RMSE}_{50} = \sqrt{\frac{1}{n-1-50-p}\sum_{i=1+p+50}^{n}(y(k)-\hat{y}(k|k-50)^2} \tag{4-43}$$

Where $y(k)$ is the actual data, $\hat{y}(k|k-50)$ is the 50-step ahead prediction, n is the length of the data set and p is the model order.
In Table 4-1 the RMSE-values for the three models are shown. The AR(7) model has the smallest prediction error for all 5 signals. From these results it can be concluded that the AR(7) model is the best model to describe the wave data. An example of a 50-step ahead prediction for the three models is shown in Figure 4-7.

DoF	AR(7) 50-step	ARMA(5,2) 50-step	ARMA(3,2) 50-step
Surge (x) [m]	0.0098	0.0180	0.0185
Sway (y) [m]	0.0097	0.0115	0.0115
Heave (z) [m]	0.0142	0.0369	0.0368
Roll (ϕ) [deg]	0.0699	0.1042	0.1106
Pitch (θ) [deg]	0.0531	0.0897	0.0936

Table 4-1: average RMSE over 10 data sets

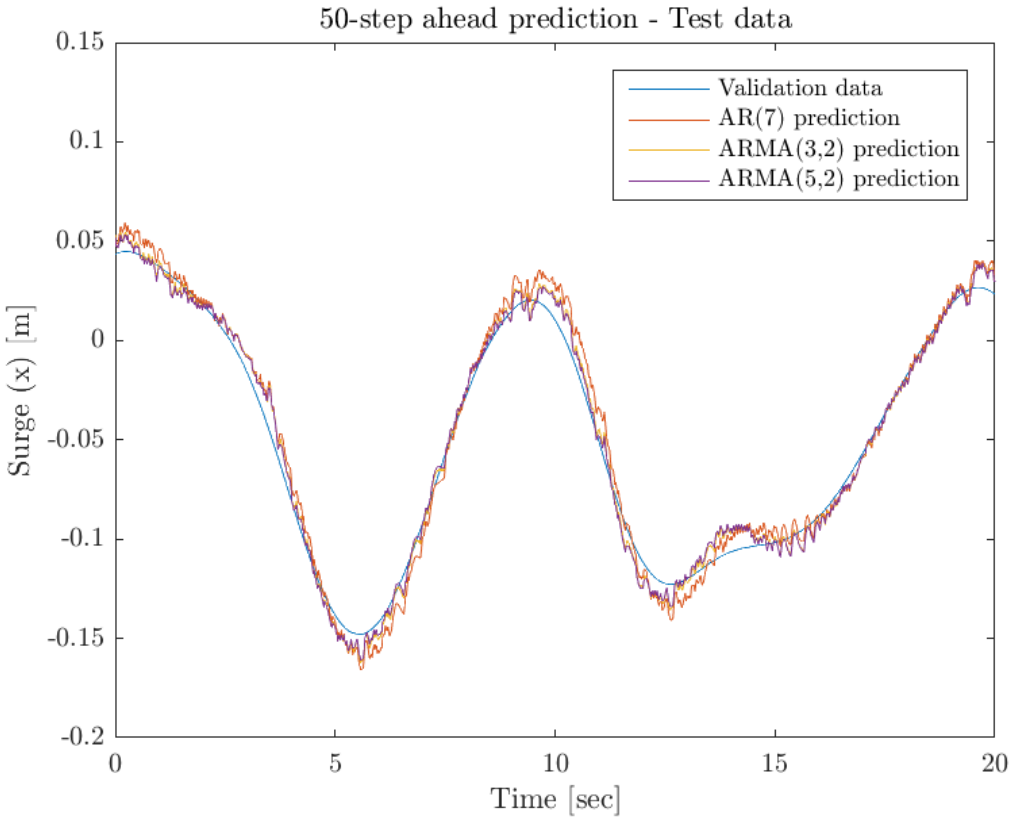


Figure 4-7: 50 step ahead prediction

4-6 Real-time parameter estimation

The results shown so far are obtained considering only a *static* AR model estimation, through an offline batch least squares parameter estimation on a fixed training data set. The problem could be that the AR model becomes unrepresentative for the current waves. But solving the least squares problem every times step is too computational costly.
Therefore in this section three solutions are proposed to solve this problem. First the basic principle of least squares will be explained. Next the recursive estimation algorithm to adapt the AR to each observation is discussed. The batch-wise training is explained in more detail. Finally a re-sampling method is shown.

4-6-1 Least Squares

The AR model assumes the signal can be described as follow:

$$A(q)y(k) = e(k) \tag{4-44}$$

$$y(k) + a_1 y(k-1) + \ldots + a_p y(k-p) = e(k) \tag{4-45}$$

This can be rewritten to:

$$y(k) = a_1 y(k-1) + a_2 y(k-2) + \ldots + a_p y(k-p) + e(k) \tag{4-46}$$

Where $y(k)$ the data set, $e(k)$ is the error term, p is the model order and k is the time step. This can be rewritten to a matrix notation:

$$Y = \begin{bmatrix} y(k+1) & y(k+2) & \ldots & y(k+n) \end{bmatrix} \tag{4-47}$$

$$\Phi = \begin{bmatrix} y(k) & y(k-1) & \ldots & y(1) \\ y(k+1) & y(k) & \ldots & y(2) \\ \vdots & \vdots & \ddots & \vdots \\ y(n-1) & y(n-2) & \ldots & y(n-p) \end{bmatrix} \tag{4-48}$$

$$\theta = \begin{bmatrix} a_1 & a_2 & \ldots & a_p \end{bmatrix}^T \tag{4-49}$$

$$e = \begin{bmatrix} e(k+1) & e(k+2) & \ldots & e(k+n) \end{bmatrix} \tag{4-50}$$

$$Y = \Phi\theta + e \tag{4-51}$$

Where n is the length of the training's data set, so the number of data point which are used for the estimation of β.

The least squares solution for parameter estimation satisfies the following minimization:

$$\hat{\theta} = \arg\min_{\theta} ||Y - \Phi\theta||_2^2 \tag{4-52}$$

Which can be seen as the minimization of the difference between the actual value and the predicted value. The solution is found by solving the quadratic minimization.

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T Y \tag{4-53}$$

The solution is computational intensive. The matrix Φ grows rapidly when the number of available data increases. Also numerical matrix inversion is very challenging for large matrices.

4-6-2 Adaptive estimation

In Section 4-6-1 the least squares solution is described. When this method is implemented real-time the number of measurements is constantly increasing and may lead to very large matrices, since it is successively supplemented by new rows of data obtained by new measurements. One solution to this problem might be to simply implement a recursive algorithm to adapt the AR model to each new observation, through recursive least squares or through the Kalman filter.

Recursive Least Squares

The Recursive Least Squares (RLS) approach minimizes the following cost-function:

$$J(\theta(k)) = \sum_{\tau=k-T}^{k} \lambda^{k-\tau} \left[y(\tau) - \phi(\tau)^T \hat{\theta}(k) \right]^2 \tag{4-54}$$

Here $\hat{\theta}(k)$ is a time-varying parameter vector, k is the time step and T is the length of the data set. λ is the weighting factor, which can vary between 0 and 1. When $\lambda = 1$ it is a regular RLS estimation and a large weight is given to the old data. With a small λ recent data is more important than old data.

The minimization results in the following algorithm. The recursive estimation of the time-varying parameter vector $\theta(k)$ is given as:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + g(k)e(k) \tag{4-55}$$

Where $e(k)$ can be seen as the innovation term and $g(k)$ as a gain term. These are expressed as:

$$g(k) = \frac{\lambda^{-1} P(k-1)\phi(k)}{1 + \lambda^{-1}\phi^T(k)P(k-1)\phi(k)} \tag{4-56}$$

$$e(k) = y(k) - \phi(k)^T \hat{\theta}(k-1) \tag{4-57}$$

$$\tag{4-58}$$

The matrix $P(k) \in \mathbb{R}^{n \times n}$ represents the covariance matrix of the estimate $\hat{\theta}(k)$ and is updated as follows:

$$P(k) = \lambda^{-1} P(k-1) - \lambda^{-1} g(k)\phi^T(k)P(k-1) \tag{4-59}$$

Kalman filtering

The Kalman filter assumes that the parameter vector of the AR model is a random walk model:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + v(k) \tag{4-60}$$

The process noise $v(k)$, is zero-mean with a covariance matrix R_1. The observation equation is the following:

$$y(k) = \hat{\theta}(k)\phi^T(k) + \epsilon(k) \tag{4-61}$$

Where $\epsilon(k)$ is the error with zero mean and covariance matrix R_2. Now the parameter vector can be updated according to the recurrence formula of Kalman:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + K(k)\alpha(k) \tag{4-62}$$

$$\tag{4-63}$$

Where $K(k)$ is the Kalman gain and $\alpha(k)$ the interest rate or innovation term.

$$K(k) = Q(k)\phi(k) \tag{4-64}$$

$$Q(k) = P(k-1)\phi(k)\left[\phi^T(k)P(k-1)\phi(k) + R_2\right]^{-1} \tag{4-65}$$

$$\alpha(k) = y(k) - \phi^T(k)\hat{\theta}(k-1) \tag{4-66}$$

Where the covariance matrix $P(k)$, which indicate the parameter errors, is expressed as:

$$P(k) = P(k-1) - Q(k)\phi^T(k)P(k-1) + R_1 \tag{4-67}$$

The recursive Kalman filtering equations are very similar to the recursive least squares, but in the Kalman filter estimation the variances of ϵ, R_2, and of $v(k)$, R_1, are incorporated.

Stability

The recursive filters are a powerful tool for adaptive filtering, prediction and identification. However, when implemented in finite precision the RLS filters can suddenly become unstable [11]. This form of sudden or explosive divergence is a big problem. For the conventional RLS is linked to the loss of positive definiteness of the information or autocorrelation matrix $M(k)$ [12]:

$$M(k) = \sum_{\tau=1}^{k} \phi(\tau-1)\phi(\tau-1)^T \tag{4-68}$$

$$M(k)^{-1} \approx P(t) \tag{4-69}$$

During poor excitations old information is continuously forgotten while there is very little new dynamic information coming in. Which leads to the exponential growth of the covariance matrix $P(k)$ matrix. This loss of positive definiteness has also been observed in Kalman filtering. When examining a single error in an infinite precision algorithm and assuming that positive definiteness is preserved, it is proved that RLS is stable. Unfortunately we cannot assume that.
In [11] a fixed point analysis approach is developed for analysing the RLS filters. This is used to develop a scenario for explosive divergence and determine which factors make explosive divergence more likely. It turns out that the information matrix $M(k)$ multiplies with the round off error, so the larger the maximum eigenvalue of $M(k)$ becomes the larger the error term can become. This is equivalent to the loss of positive definiteness of the inverse information $P(k) = M(k)^{-1}$. Secondly, if the finite precision wordlength is small the roundoff term will

be large which increases the possibility to diverge. Finally, the smaller the weighting factor λ is, the smaller the effective data window is and will also make loss of positive definiteness more likely.

Some solutions have been proposed to cope with this problem. In [11] a method is developed to bias the fixed point filter towards stabilization.
Another possibility is to monitor the matrix $P(k)$, and reset its values to *acceptable* ones when its eigenvalues assume too large values [13]. An aprroach could also be to use multiple or variable forgetting factors [14]. Last a damping factor can be introduced which will prevent the parameter vector $\hat{\theta}$ from rapid changes [15].

4-6-3 Batch-wise

To overcome the stability problem of the adaptive estimation one practical possibility might be to run a batch estimation of the AR model, through least-squares, every certain amount of time to be sure that the model is always valid for the current sea states.
Batch wise updating can be seen as a more safe method than the recursive least squares, due to the stability problem of the method. In order to determine the minimum necessary batch-size a prediction is made with a training set with a length between $n = 14$ and $n = 100$. The parameters are determined over all the different training sets. A 50-step ahead prediction is made for 200 time-steps. The RMSE is determined for the different predictions. The ten different surge (x) data sets are used and the average RMSE error is calculated. In Figure 4-8 the average RMSE is shown. It can be seen that a minimum required length of the training set is approximately 70 data points. The RMSE is small for all data sets at that point and so a stable prediction can be guaranteed. In Figure 4-9 a 50-step ahead prediction is shown for different sizes of the training set. Clearly a larger training set gives a better result up to a certain point.

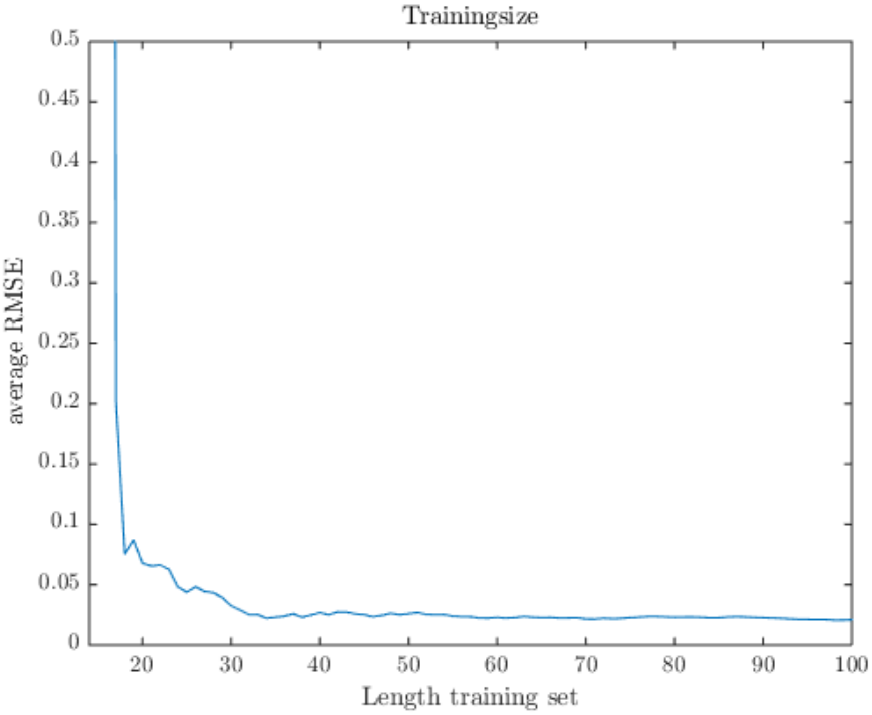


Figure 4-8: RMSE for different lengths

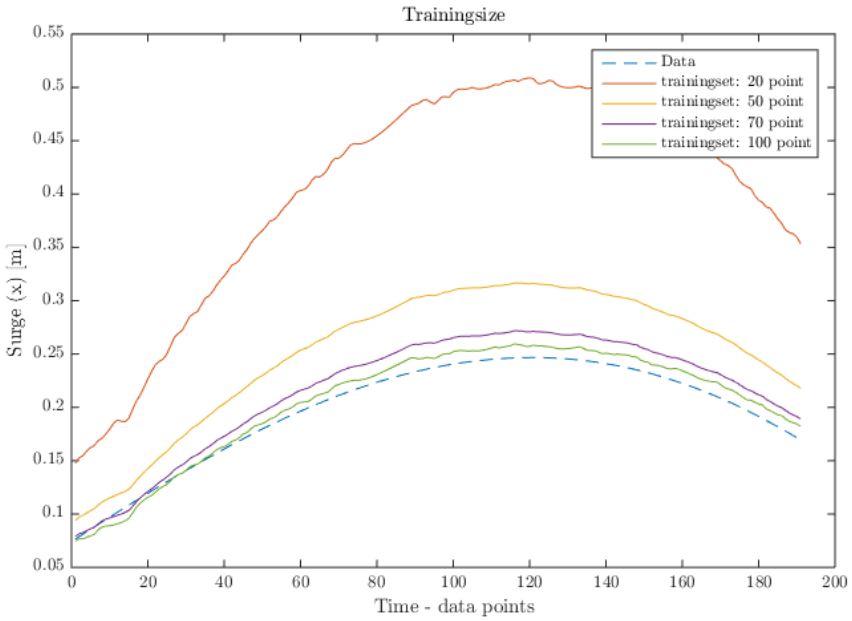


Figure 4-9: Example of different training set lengths

4-6-4 Resample

In order to make the prediction less computational costly, re-sampling can be considered. The data is gathered at a very high frequency (50 [Hz]). By down sampling to 5 [Hz] is only a 5-step ahead prediction is necessary to predict 1 second ahead in time. For all 10 data set of the 5 degrees of freedom are down-sampled and a 5-step ahead prediction is made. The RMSE are determined in order to compare this with the 'regular' prediction at 50 Hz. The results are shown in Table 4-2. The RMSE values are smaller for the down sampled prediction, therefore re-sampling is a good option. An example of a 5 step-ahead prediction at a sample rate of 5 Hz is shown in Figure 4-10. With resampling the model can capture more of the dynamics which results in a better performance.

DoF	AR(7) 50 Hz 50-step	AR(7) 5 Hz 5-step
Surge (x) [m]	0.0098	0.0062
Sway (y) [m]	0.0097	0.0067
Heave (z) [m]	0.0142	0.0077
Roll (ϕ) [deg]	0.0699	0.0378
Pitch (θ) [deg]	0.0531	0.0354

Table 4-2: Average RMSE over 10 data sets

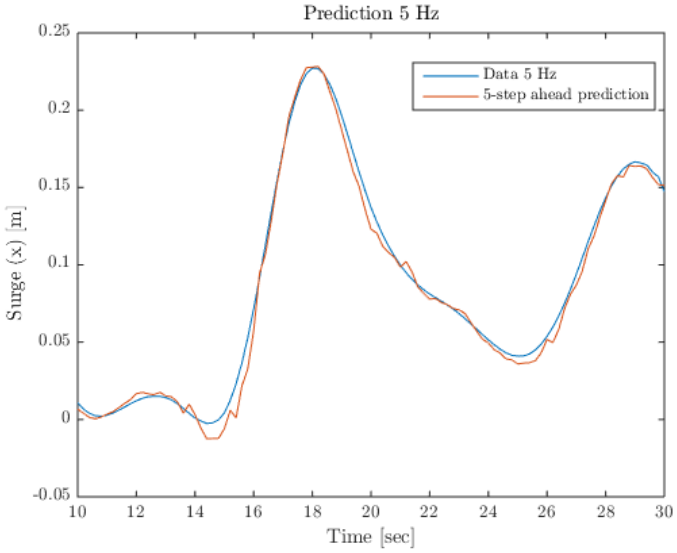


Figure 4-10: Example of 5-step ahead prediction at 5 Hz

Interpolate

To implement this in the Ampelmann system, the data should be resampled to 50 [Hz] again. Two interpolation methods are used:

Linear Interpolation: This is the simplest form of interpolation, it takes two data points, say (x_a, y_a) and (x_b, y_b) and the interpolant (x, y) is given as:

$$y = y_a + (y_b - y_a)\frac{x - x_a}{x_b - x_a} \tag{4-70}$$

Spline: The spline interpolation fits a low-order polynomial through the data points. A spline will take the shape that minimizes the bending. Spline produce an interpolated function that is continuous through the second derivative. Which is more stable than fitting a polynomial, because of a smaller possibility of wild oscillations.

The result is shown in Figure 4-11. The interpolation method does not make a large difference. The spline method is slightly better than the linear interpolation, but the linear-interpolation is more simple.

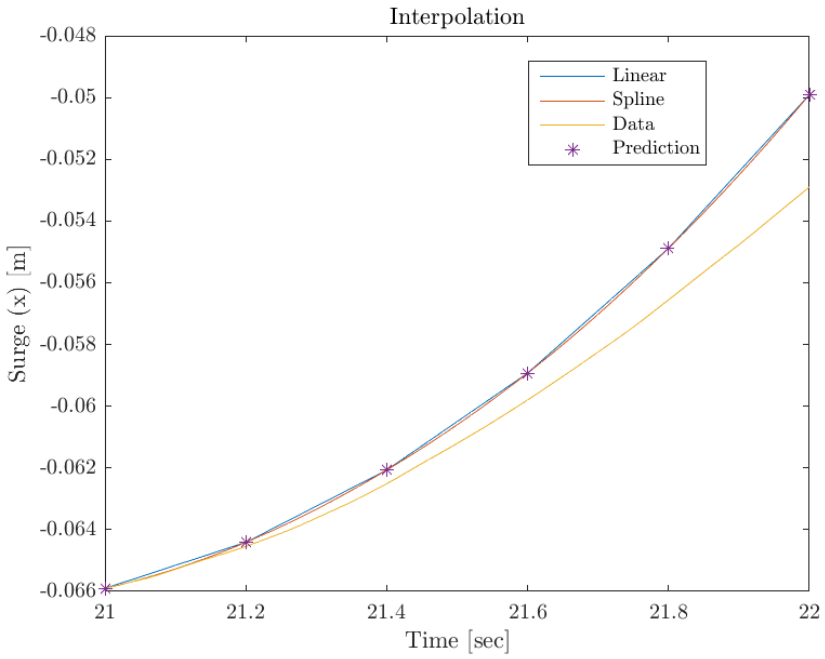


Figure 4-11: Interpolation between the 5 [Hz] prediction

4-7 Conclusion

Different linear model structures are presented based on a general discrete time polynomial. For pure time series problems the AR and ARMA model are the best fit. The AR model assumes that the one-step ahead prediction depends linearly on the previous data points and an error term. The ARMA model also assumes that the one-step ahead prediction depends on the past data points as well as the previous innovation terms.
The optimal number of regressors, i.e. the size of the model, is determined with the VAF value, AIC criterion and the cross validation test. These tests are modified for a multiple step ahead prediction. From these results the AR(7), ARMA(3,2) and ARMA(5,2) models are chosen to be validated and tested.
It is explained how to generate the forecast from the chosen models and a 50-step ahead prediction is made. The average RMSE over the 10 data sets is calculated. The AR model gives the smallest prediction error.
To be able to implement the AR-model real-time the parameters cannot be estimated by the least squares approach every time-step. A solution is to recursively estimate the parameters by using RLS or Kalman, but these methods have some stability problems which could make them unreliable estimators. Another option is to train the parameters batch-wise every certain amount of time to be sure that the model is always valid for the current sea state. A minimum batch size of 70 data-points is necessary. This is the more safe option.
Last, the data is re-sampled to 5 [Hz]. The resampled data is used to train the AR model and make a prediction. This down sampling method reduces the computational demand for training and prediction. On top of that the prediction error decreased. The resampled model is able to capture more of the dynamics of the signals.

To summarize:

- The AR(7) model is the best-linear model to represent the wave data.
- To model should be trained batch-wise, with a batch size of a least 70 data-points.
- Down-sampling the data, decreases the prediction error and the computational demand.

Chapter 5

Wavelets

5-1 Introduction

A linear Auto Regressive (AR) model is derived in the previous chapter. It is however interesting to compare this results to other (non-linear) models and those modeling capability's. A great number of tools for time series analysis and forecasting exists, but for this research the scope has to be narrowed. From literature the mathematical tool 'wavelets' are chosen to be futher researched, since this technique seems to be very promising. Wavelets are seen as a powerful tool for analyzing time series [16]. Also wavelets offer an adequate framework for the representation of *natural* signals [17]. On top of that the technique is frequently used for time series prediction [18], [19], [20], [21].
First the mathematical properties of wavelets will be explained. Then the wavelets will be applied in two different manners make a time series prediction:

Multiresolution analysis A wavelet transform is used to decompose the time series into varying scales of temporal resolution. Then an AR model is trained on each resolution scale.

Wavenet A Wavenet is an artificial neural network with one hidden layer of nodes, whose basis functions are drawn from a family of nonlinear wavelet basis functions.

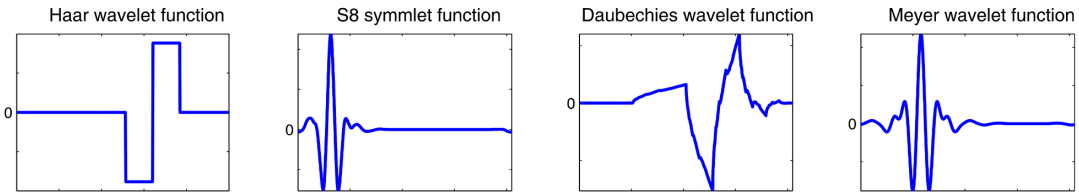


Figure 5-1: Wavelet examples

5-2 Wavelet

Wavelets can be considered topic of pure mathematics, however in only a few years of existence as a theory of their own, they have shown great potential and applicability in many fields [22]. Wavelets are mathematical tools for analyzing time series and images [16]. There are two main types of wavelet transforms, the continuous wavelet transform (CWT) and the discrete wavelet transform (DWT). A wavelet is a 'small wave', usually denoted as $\psi(\cdot)$. A small wave grows and decays essentially in a finite time period. The opposite is a 'big wave', an example is a sine-function which keeps oscillating. (The sine wave is used for the Fourier transform). The function $\psi(t)$ defined over the real axis $(-\infty, \infty)$ is to be classed as a wavelet if:

1. The integral of $\phi(t)$ is zero:
$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \tag{5-1}$$

2. The square of $\phi(t)$ integrates to unity:
$$\int_{-\infty}^{\infty} \psi^2(t)dt = 1 \tag{5-2}$$

   (For a sine-function the above integral would be infite)

The length of the interval of a wavelet is vanishingly small compared to the infite length of the real axis $(-\infty, \infty)$.
There are many kinds of wavelets. The most simple is the *Haar wavelet*. Some examples of wavelets are shown in Figure 5-1.

A wavelet transformation is similar to that of a Fourier transformation. Fourier analysis can tell us the composition of a give function in terms of sinusoidal waves of different frequencies and amplitudes. However there are some important differences between Fourier analysis and wavelets. Fourier basis functions are localized in frequency but not in time, it will give the average of changing frequencies over the whole function. Wavelets are local in both frequency/scale (via dilations) and in time (via translations). Furthermore many functions can be represented by wavelets in a more compact way. Functions with a lot of spikes and discontinuities usually take substantially fewer wavelet basis functions than sine-cosine based functions to achieve a comparable approximation. Also the computational complexity of the fast Fourier transform is $O(n \cdot \log_2(n))$ and for the fast Wavelet transform $O(n)$.

5-2-1 Continuous Wavelet Transform

The wavelets are generated from a single basic wavelet $\psi(t)$, the *mother wavelet*, by scaling and translation. An infinite number of *daughter* wavelets can be derived.

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \tag{5-3}$$

The variables s and τ are scale or dilation and translation respectively. Which are the new dimension after the wavelet transform. The wavelet satisfies the following condition

$$||\psi_{s,\tau}|| = ||\psi|| \tag{5-4}$$

Because of the normalization at the right-side of (5-3). The Continuous Wavelet transform (CWT) is used to transform a signal over continuous time. The CWT of function $f(t)$ is written as:

$$\gamma(s,\tau) = \int_{\infty}^{\infty} f(t)\psi^*_{s,\tau}(t)dt \tag{5-5}$$

Where $*$ indicates the complex conjugate of the wavelet.

5-2-2 Discrete Wavelet Transform

The signal analysis with the CWT yields to a large amount of information, since the signal is analyzed over infinitely many dilations and translations of the *mother* wavelet. So clearly there will be a lot of redundancy in the CWT. The key features of the transform can also be retained considering only subsamples of the CWT. This lead to the Discrete Wavelet transform (DWT).
Discrete wavelets are not continuously scalable and translatable but can only be scaled and translated in discrete steps. Generally, 'dyadic' scales are used, so s has the form of $2^{j-1}, j = 1, 2, 3, \ldots$ and then τ is sampled at 2^j. This is achieved by modifying the wavelet representation:

$$\psi_{j,k}(t) = \frac{1}{s_0^j}\psi\left(\frac{t - k\tau_0 s_0^j}{s_0^j}\right) \tag{5-6}$$

The values of s and τ are now restricted to discrete values. Usually these are chosen that the sampling corresponds to dyadic sampling, which is $s_0 = 2$ and $\tau_0 = 1$ and hence:

$$\psi_{j,k}(t) = \frac{1}{2^j}\psi\left(\frac{t - k2^j}{2^j}\right) \tag{5-7}$$

when discrete wavelets are used to transform a continuous signal the results will be a series of wavelet coefficients, and it is also called the wavelet series decomposition.

5-2-3 Results

As explained in the previous sections the Wavelet and Fourier transformation are tools for data analysis. The techniques will show the wave spectrum, which shows how the energy is distributed across different frequency components of the wave. A visualization is made of the Fourier and Wavelet transform, the difference is clear now. Both transformations are performed over the vessel motion data set. See Figure 5-2, Figure 5-3, Figure 5-4, Figure 5-5 and Figure 5-6, the data signal is shown on top, below the signal the wavelet transform is shown. The wavelet plot presents the frequencies and its power versus time. Next the Fourier transform plot is rotated 90 degrees in order to connect the frequencies to the wavelet transform plot. The Fourier transform can clearly be seen as the average of changing frequencies over the whole data set. From these results is can be seen that the data has a narrow bandwidth form $0 - 0.3[Hz]$. The wave spectra of x and y are similar, two peaks around 0.01 [Hz] and 0.1 [Hz]. The wave spectra of z, ϕ and θ also show resemblance, all of them have a peak around 0.1 [Hz].

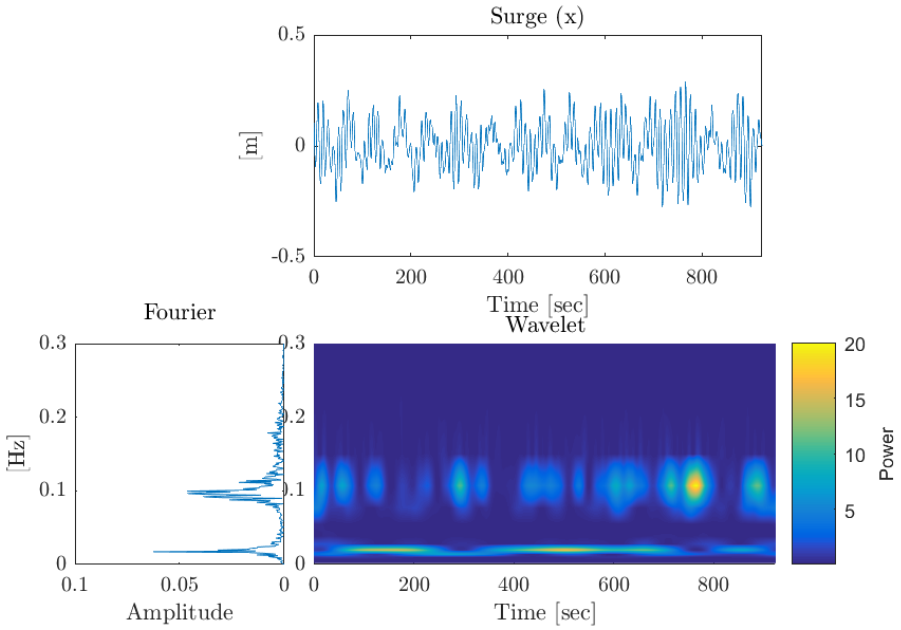


Figure 5-2: Fourier and Wavelet transform for x

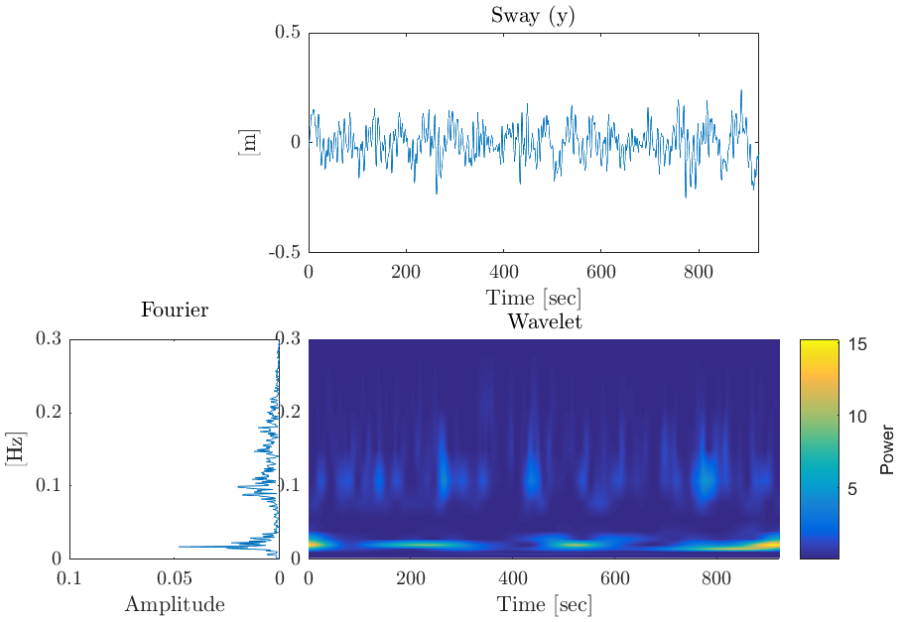


Figure 5-3: Fourier and Wavelet transform for y

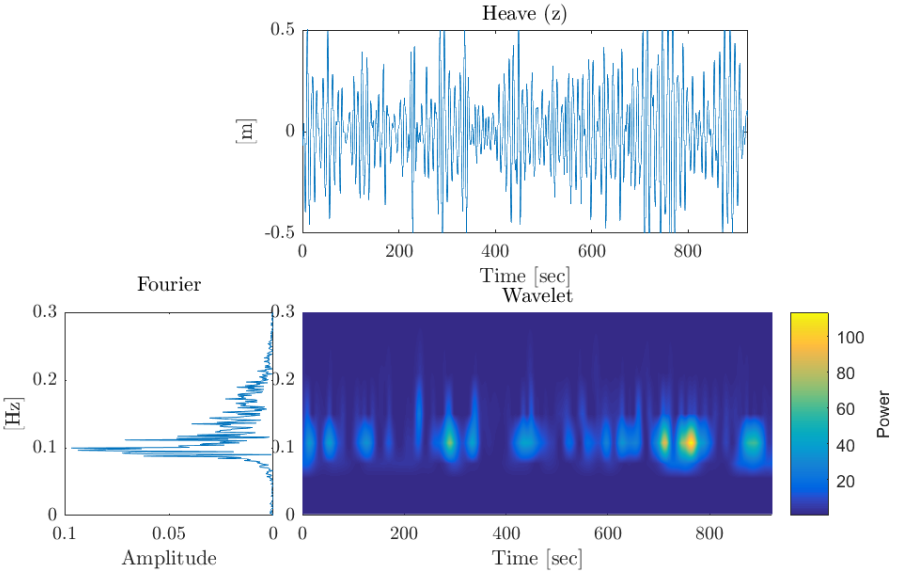


Figure 5-4: Fourier and Wavelet transform for z

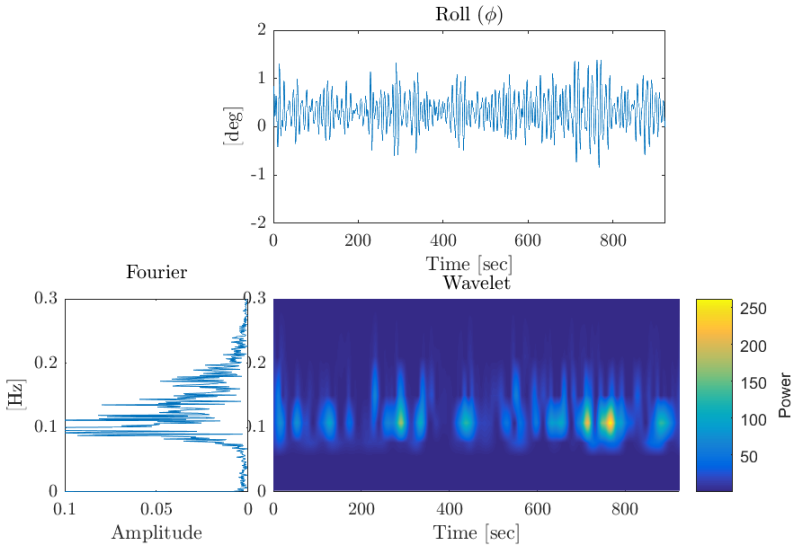


Figure 5-5: Fourier and Wavelet transform for ϕ

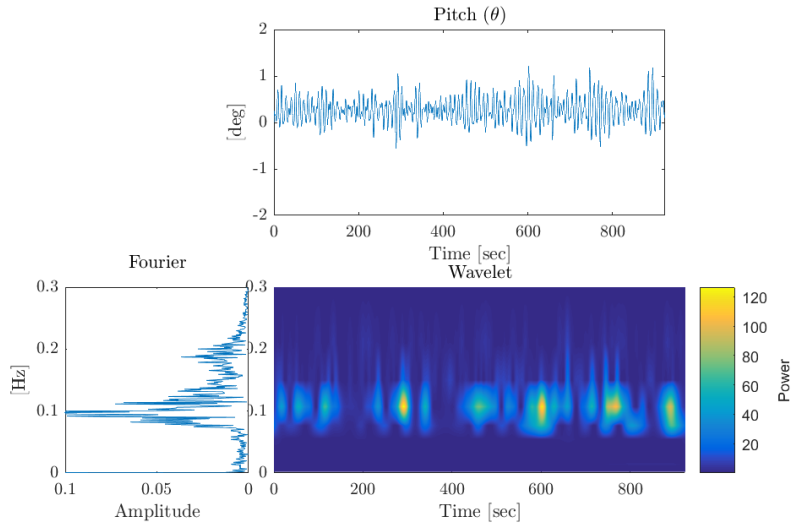


Figure 5-6: Fourier and Wavelet transform for θ

5-3 Wavelet decomposition

The discrete wavelet transform is already described in section 5-2-2. Here the wavelet decomposition is explained with more detail.
The discrete wavelet transform of a signal x is calculated by passing through a series of wavelet filters. The decomposition uses low and high pass filters. Let h be the impulse response of the low pass filter and g the impulse response of the high-pass filter. This decomposition scheme may be carried out for several iterations to increase frequency resolution, see Figure 5-7, where $\downarrow$ denotes subsampling . Then the trend (c) and the different detail series (d) of a signal x are described as:

$$x_k = c_{0,k} = c_{N,k} + \sum_{j=1}^{N} d_{j,k} \tag{5-8}$$

$$c_{N,k} = (h_1 \cdot h_2 \cdot \ldots \cdot h_N \cdot x)_k \tag{5-9}$$

$$d_{j,k} = (h_1 \cdot h_2 \cdot \ldots \cdot h_{j-1} \cdot g_N \cdot x)_k \tag{5-10}$$

Where $k = 1, \ldots, n$, which is the number of measurements and $j = 1, \ldots, N$ where N is the number of iterations. The main motivation of using the wavelet decomposition is the easy

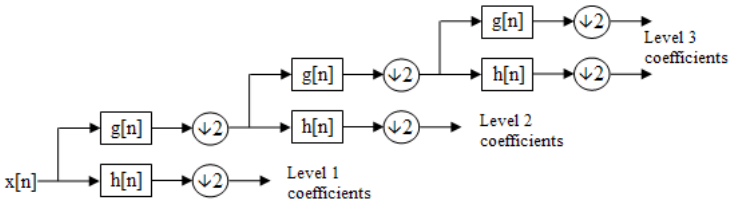


Figure 5-7: Filter bank [23]

analysis of the obtained series [18]. In fact, the trend $c_{N,k}$ contains the slowest dynamics. It is practically noisy free. The details series d_j, k contains the dynamics at a certain immediate scale. The low detail series may be corrupted by noise, one can neglect those in the corresponding prediction. A consequence of this property, training an estimator on these time series is simpler than on the original data [18].
Now the obtained series will be predicted separately and added to obtain the final prediction of x:

$$\hat{c}_{N,k+p} = \hat{f}_0(c_{N,k}, c_{N,k-1}, \ldots, c_{N,k-r_0}) \tag{5-11}$$

$$\hat{d}_{j,k+p} = \hat{f}_j(c_{j,k}, c_{j,k-1}, \ldots, c_{N,j-r_j}) \quad j = 1, \ldots, N \tag{5-12}$$

$$\hat{x}_{k+p} = \hat{c}_{N,k+p} + \hat{d}_{j,k+p} + \ldots + \hat{d}_{1,k+p} \tag{5-13}$$

The predictors $\hat{f}_0, \hat{f}_1, \ldots, \hat{f}_N$, can be any of prediction method, for example a linear AR model.

5-4 Results

The *mother-wavelet* function has to be chosen. Unfortunately there is no straight forward way of choosing. Trial-and-error is a method of finding one. The shape of the function should have some resemblance to the features present in the time series. So for the wave data, the Haar-wavelet would not be a fit. The choice of the wavelet function is not critical and one function can give the same qualitative results as another [24].
The Daubecies wavelet is chosen, the shape is shown in Figure 5-8. The Daubechies wavelets are a family of orthogonal wavelet defining a discrete wavelet transform and characterized by a maximal number of vanishing moments.

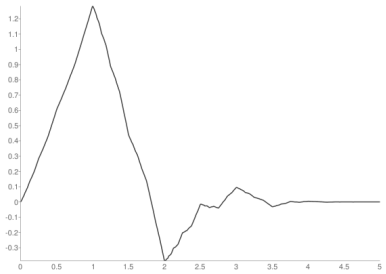


Figure 5-8: Daubechies 3

The training data set is decomposed and the result is shown Figure 5-9. From the decomposition it can be seen that level 2 and higher almost completely consists of noise. So the AR model is only trained for the first level of wavelet resolution.
The testing data set is also decomposed and a prediction is made with the decomposed data. This is compared to the actual data by calculating the RMSE over a 50-step ahead prediction. The results are shown in table Table 5-1. Also a comparison is shown in figure Figure 5-10.

DoF	AR(7) 50-step	Wavelet dec. 50-step
Surge (x) [m]	0.0098	0.0128
Sway (y) [m]	0.0097	0.0123
Heave (z) [m]	0.0142	0.0238
Roll (ϕ) [deg]	0.0699	0.0742
Pitch (θ) [deg]	0.0531	0.0550

Table 5-1: Average RMSE over 10 data sets

From these results it is seen that including the wavelet decomposition does not improve forecasting. The AR-model trained with the original data gives a smaller prediction error. This results is expected since no clear levels of details and trends could be distinguished, as seen in Figure 5-9 . In Section 5-2-3 it is shown that the data has a narrow bandwidth.

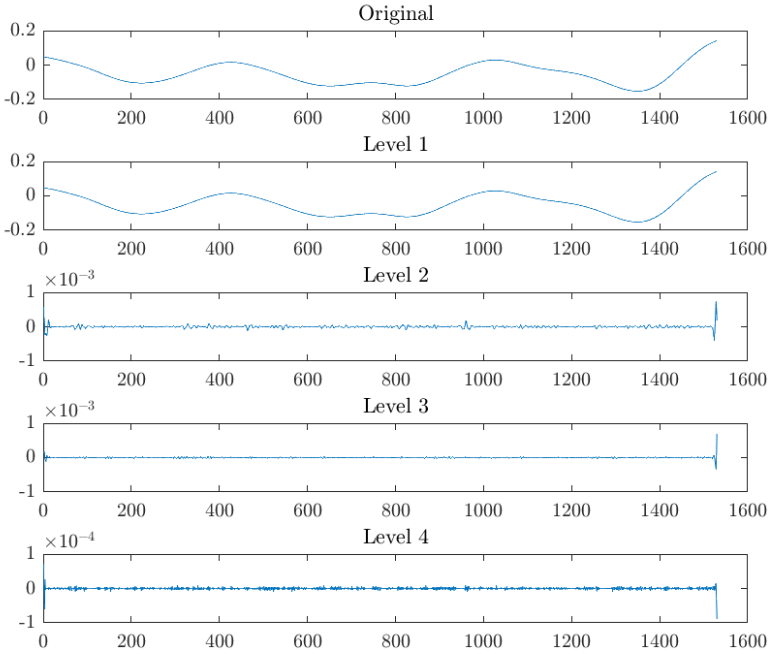


Figure 5-9: Wavelet decomposition for Surge (x)

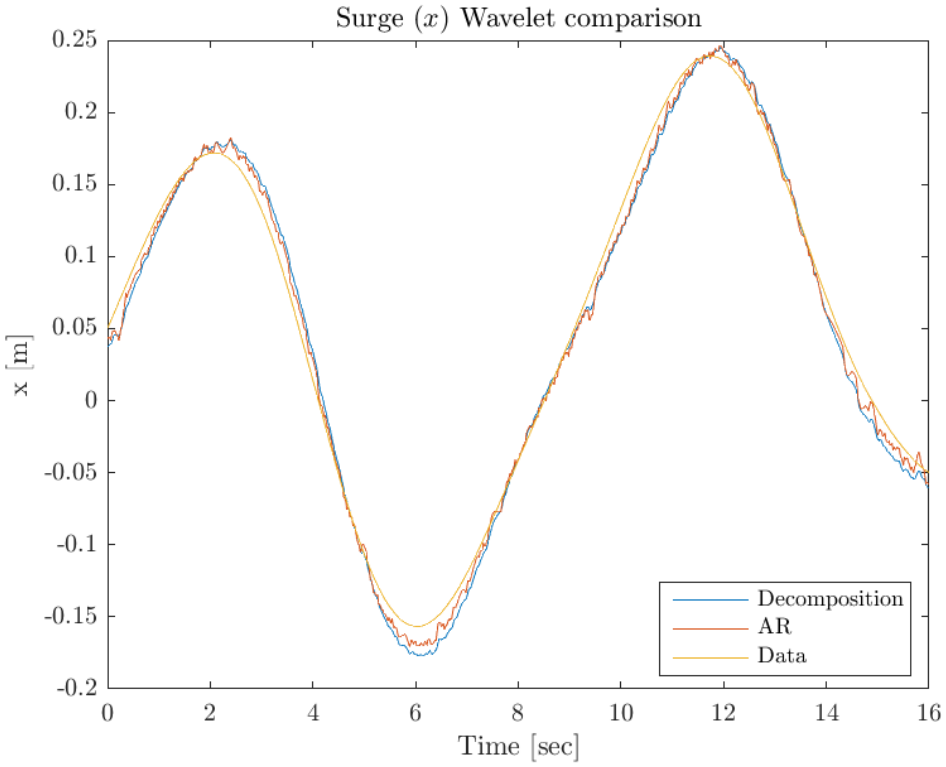


Figure 5-10: Comparison of wavelet prediction

5-5 Wavenet

Wavenet is short for Wavelet Neural Network (WNN). Artificial Neural Network (ANN) are powerful approximants of non-linear functions and adequate for model-free estimation of systems. Combining the advantages offered by neural network processing, on one hand, with wavelet representation, on the other had is an attractive idea [17].

5-5-1 Structure

An artificial neural network is a highly parallel distributed network of connected processing units called neurons. The structure of a neuron is shown in Figure 5-11, which forms the basis of the neural network. Mathematically the neuron can be described as follows:

$$v = \sum_{i=1}^{m} w_1 x_i - \theta \tag{5-14}$$

$$y = \psi(v) \tag{5-15}$$

Where x are the inputs, y is the output, w are the weights and ψ is the activation function. The activation function defines the output of the neurons. The structure of a wavelet neural network is shown in Figure 5-12. It is a feed-forward neural network, with one or more inputs, with one hidden layer and a linear combiner output layer. The hidden layer consists of neurons, whose activation functions are drawn from a wavelet basis.

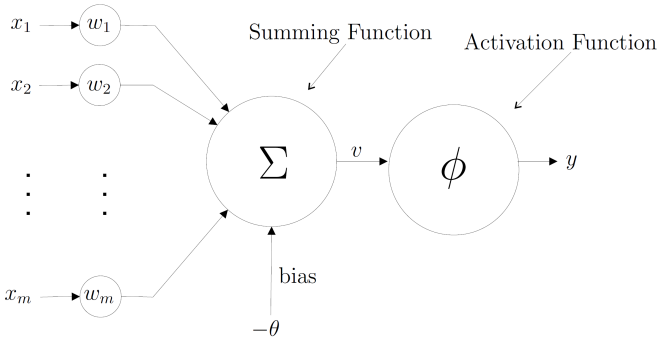


Figure 5-11: Mathematical model of a Nonlinear Neuron [25]

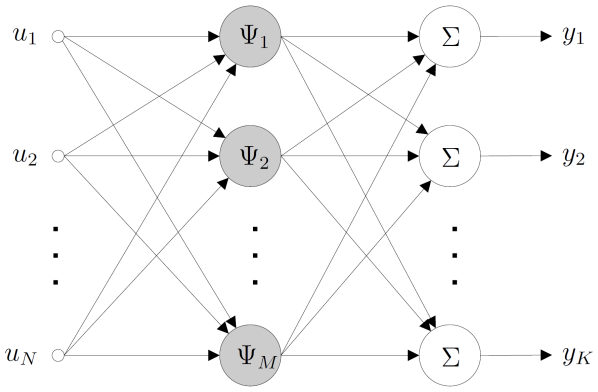


Figure 5-12: Structure of a wavelet neural network [25]

Activation function

The most common form of activation function used in artificial neural networks is the Sigmoid function. The sigmoidal wavelet is proven to be useful for representing real-world bandlimited signals in a wavelet neural network [26]. The sigmoid function is typically defined as the following:

$$s(t) = \frac{1}{1 + e^{-qt}} \tag{5-16}$$

Where q is the slope parameter and $s(t)$ has the property of being continuously differentiable. A wavelet can be generated by combining sigmoid functions in the proper manner. Four sigmoidal functions ($s(t)$) combined result in a sigmoidal wavelet ($\psi(t)$).

$$\phi(t) = s(t+d) - s(t-d) \tag{5-17}$$

$$= \frac{1}{1 + e^{-q(t+d)}} - \frac{1}{1 + e^{-q(t-d)}} \tag{5-18}$$

$$\psi(t) = \phi(t+p) - \phi(t-p) \tag{5-19}$$

$$= \frac{1}{1 + e^{-q(t-d-p)}} - \frac{1}{1 + e^{-q(t-d+p)}} - \frac{1}{1 + e^{-q(t+d-p)}} - \frac{1}{1 + e^{-q(t+d+p)}} \tag{5-20}$$

Letting $q = 2$, $d = 1$ and $p = 1$ gives three terms

$$\psi(t) = \frac{1}{1 + e^{-2(t-2)}} - \frac{2}{1 + e^{-2(t)}} - \frac{1}{1 + e^{-2(t+2)}} \tag{5-21}$$

The wavelet activation function is shown in figure Figure 5-13. It satisfies the requirement of having zero average energy.

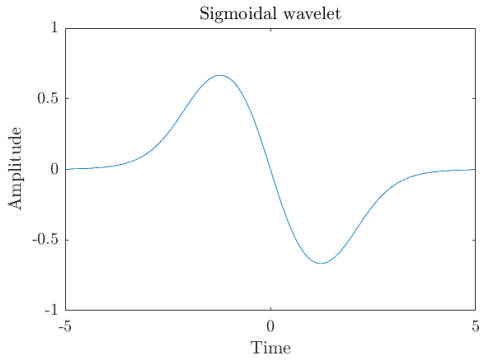


Figure 5-13: Sigmoidal wavelet

5-5-2 Training

The parameters of the model must be estimated, which is generally called 'training' for neural networks. The parameters will be estimated by minimizing the error between the output of the model and the actual data.

$$\min_{\theta} V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} ||y(t) - g(\varphi(t), \theta)||^2 \tag{5-22}$$

Where $g()$ is the non-linear mapping of the model, φ are the regressors and θ are the parameters. The actual data is $y(t)$. The parameter estimate is defined as the minimizing argument:

$$\hat{\theta}_N = \arg\min V_N(\theta) \tag{5-23}$$

In general the minimum of $V_N(\theta)$ cannot be computed analytically , so the minimization has to be done in by some numerical search procedure [27]. The non linear optimization has a basic search algorithm, which is based on iterative local search in a 'downhill' direction from the current point. Then an iterative scheme of the following kind is derived:

$$\hat{\theta}^{i+1} = \hat{\theta}^i - \mu_i R_i^{-1} \nabla \hat{f}_i \tag{5-24}$$

Where:

$$\begin{aligned} \hat{\theta}^i &= \text{Parameter estimate after iteration number } i \\ \mu_i &= \text{Step size} \\ \nabla \hat{f}_i &= \text{estimate of gradient } V_N'(\hat{\theta}^i) \\ R_i &= \text{Matrix that modifies the search direction} \end{aligned}$$

The basis for the local search is the gradient:

$$V_N'(\theta) = -\frac{1}{N}\sum_{t=1}^{N}(y(t) - g(\varphi(t)\theta))h(\varphi(t), \theta) \tag{5-25}$$

where

$$h(\varphi(t), \theta) = \frac{\partial}{\partial\theta} g(\varphi(t), \theta) \tag{5-26}$$

The gradient search for the minimum can be inefficient, especially for ill-conditioned problems close to the minimum. For the Wavenet the *Leverberg-Marquant direction* is used:

$$R^{-1}(\theta)V_N'(\theta) \tag{5-27}$$

Where

$$R_i = H_i + \delta I \tag{5-28}$$

$$H_i = \frac{1}{N}\sum_{t=1}^{N} h(\varphi(t), \hat{\theta}^i)h^T(\varphi(t), \hat{\theta}^i) \tag{5-29}$$

And δ is the step size.

5-5-3 Data

For the linear model the data set was split up in ten smaller sets, which were each used as separate data sets. The wavenet requires more trainings data in order to train a prediction model. Therefore the data set is now split up in two parts or two separate data sets. Again each data set is split in three parts, the trainings set (60%), validation set (20%) and testing set (20%). The trainingset, which is used for computing the gradient and updating the network weights and biases. The validation set is used to monitor the error, the validation error normally decreases during the initial phase of training, as does the training error. When the network begins to over-fit the data, the error on the validation set typically begins to rise. The third part will be used to compare the model to other models.

5-5-4 Stability

Atrifical neural networks can be unstable predictors. It means that a small change in the trainings data used to build the model may result in very different models. The instability problem (specifically feedforward ANN) stems from the gradient descent training process settling in local minima [28]. Different training conditions, such as small differences in training data or differences in network parameters, will result in the network stopping at different points in the weight space with resulting variations in performance.
This problem occurs when making a 50-step ahead prediction at 50 [Hz], the prediction very likely grows out of bounds after training with different initializations of the weights and biases over the same data set. As for the linear-model the Wavenet can also be trained on a 5 [Hz] data set and making a 5-step ahead prediction. The resampled prediction gives very stable results and will be used for prediction with the Wavenet.

5-5-5 Results

The number of regressors (previous data points taken into account for the prediction) and the number of neurons should be determined.
The number of regressors is chosen as 7, the same as for the linear model, in order to make a comparison.

For different numbers of neurons a network is trained, validated and tested. For this test the first (re-sampled) data set is used. The models of different sizes are used to make a 5-step ahead prediction, then the root mean squared error over a 5-step ahead prediction is determined, over the testing set. The results are shown in Table 5-2.

Neurons	RMSE x [m]	RMSE y [m]	RMSE z [m]	RMSE ϕ [deg]	RMSE θ [deg]
5	0.0053	0.0061	0.0065	0.0304	0.0279
10	0.0052	0.0065	0.0065	0.0297	0.0281
20	0.0054	0.0170	0.0134	0.0300	0.0308
50	0.0054	0.0126	0.0110	0.0349	0.0452
80	0.0056	0.0193	0.0506	0.0422	0.0400

Table 5-2: RMSE for different model sizes

The error will differ every time the model is trained. During every training the gradient descent method could find another minimum based on the initial conditions. To be able to reproduce the results the random setting for initialization of the weights and biases is set to default at the beginning of each training. Still a general trend can be derived from this data. A size above 20 neurons leads to a too complex model. Using a model with ten neurons in its hidden layer is the best choice.

The Wavenet with 10 neuron in the hidden layer is trained, validated and tested over both data sets. It is difficult to compare the AR model to the Wavenet. The AR model will always have the same model parameters when using the same training data, for the Wavenet this will be different every time the model is trained with different initial conditions, which results in a different Root mean squared error (RMSE).
On top of that the average RMSE of the AR model was determined over 10 validation sets, the RMSE of the Wavenet was determined over 2 validations sets. To be able to compare, the AR-model will be evaluated again with the same division in training and testing set as which is used for the WNN. The average RMSE over a 5 step ahead prediction is determined and result is shown below:

DoF	AR(7) 5-step 10 sets	Wavenet(7) 5-step 2 sets	AR(7) 5-step 2 sets
Surge (x) [m]	0.0062	0.0055	0.0053
Sway (y) [m]	0.0067	0.0066	0.0062
Heave (z) [m]	0.0077	0.0072	0.0068
Roll (ϕ) [deg]	0.0378	0.0293	0.0339
Pitch (θ) [deg]	0.0354	0.0304	0.0332

Table 5-3: Results of the wavenet

The results are interesting, using the exact same training and testing data the AR model gives a smaller prediction error for x, y and z. The rotations ϕ and θ are better modeled by the wavenet. But the differences are very small and the wavenet with different initial conditions

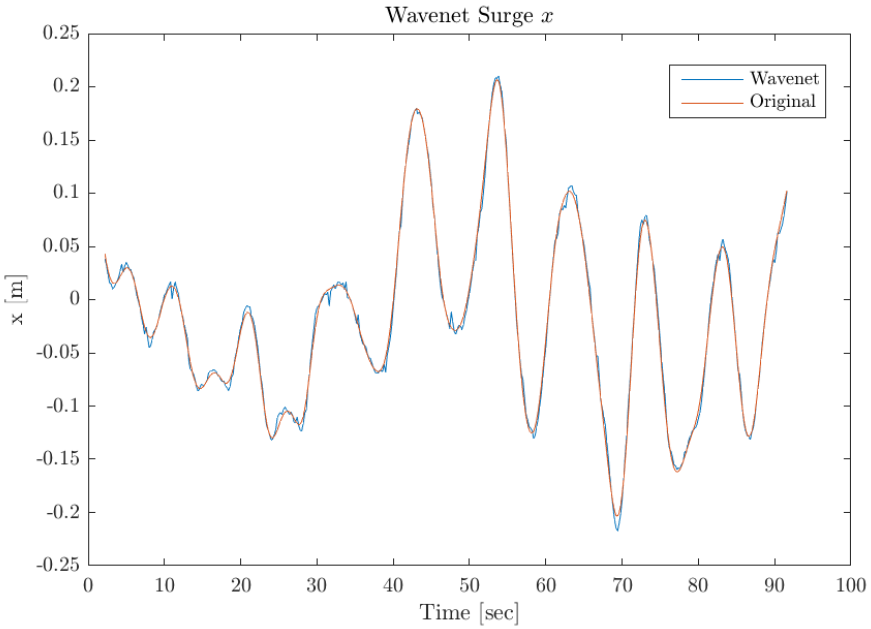


Figure 5-14: Wavenet prediction

could give better results. Therefore a larger data set is required to give a more exact estimate of the prediction error and to be able to determine which model is better.

5-5-6 Multivariate

In Section 3-5, it was shown that the signals have a high correlation between each other. Therefore it could be useful to combine the signals into one model. In this way the prediction of, for example, surge ($x(t)$) will not only depend on past values of surge ($x(t-1)\dots x(t-p)$) but also on those of the other signals. It can be used to reduce the computational power, since the signals will not have to be modeled separately.
Three different multivariate model are considered:

- x, y, z, ϕ, θ-coupled, The model will have 5 input and 5 outputs. Only one model needs to be trained.
- x, y, z and ϕ, θ coupled. Two models are used, one with 3 inputs and outputs for the translation signals, one with 2 inputs and outputs for the rotational signals. These signals are coupled because the translation and rotations have similar dynamics.
- x, y and z, ϕ, θ coupled. Two models are used, one with 2 inputs and outputs for x, y, one with 3 inputs and outputs for z, ϕ, θ . In Section 3-5 it is shown that z, ϕ and θ have a high correlation and the wavelet transforms in Section 5-2-3 of z, ϕ and θ show similar behavior.

The results are shown in Table 5-4.

DoF	Separate	x, y, z, ϕ, θ	x, y, z and ϕ, θ	x, y and z, ϕ, θ
Surge (x) [m]	0.0055	0.0090	0.0049	0.0051
Sway (y) [m]	0.0066	0.0160	0.0061	0.0065
Heave (z) [m]	0.0075	0.0149	0.0071	0.0101
Roll (ϕ) [deg]	0.0294	0.0239	0.0351	0.0231
Pitch (θ) [deg]	0.0309	0.0279	0.0276	0.0264

Table 5-4: 5-step ahead RMSE for multivariate wavenet

The model with all 5 signals coupled gives a smaller prediction error for the rotational signals, but a larger error for the translation signals, compared to the the 5 separate models. The model where the translation and rotation signals are separately coupled gives the smallest prediction error for x, y and z. The smallest prediction error for rotations ϕ and θ is achieved when z,ϕ and θ are coupled. Including the heave (z) signal in the model improves the prediction of the rotations, unfortunately it does not benefit the prediction of heave. The best method is combining the third and fourth method and use two models. The first model combines x, y and z and has three input and three outputs. This model will be used for the prediction of the translations. The second model combines z, ϕ and θ and has three inputs, but only two output. This model will predict the rotations.

As mentioned before, the Wavenet will not always find the same parameters during training, so the result will vary every time the model is trained. A combination of the third and fourth method gives the best results, but the improvements are small. More data should be used to confirm that this multivariate model is the best. But it shows promises for further research.

5-6 Prediction time

During the previous tests, the 50-step ahead prediction is chosen for comparison. For practical implementation it is necessary to know up to which prediction step the model can give an accurate prediction. As explained in Section 3-3 the accuracy goal is set to 0.05 [m] for x, y and z. For the rotations ϕ and θ the goal is determined as 5% of the amplitude of the wave. The maximum amplitude resembles to approximately 2 [deg], so the accuracy should be 0.01 [deg].
A prediction is made with the AR model and the Wavenet Neural Network up to 20 prediction steps ahead at 5 [Hz] (4 seconds). The results for one translation (x) and one rotation (ϕ) are shown in Figure 5-15 and Figure 5-16, the results of the other signals can be found in Appendix B. The models are able to predict surge (x) accurately for approximately 15 steps, 3 seconds, and roll (ϕ) for approximately 9 time steps, 1.8 seconds.
For the other signals the maximum prediction step is stated in Table 5-5

The AR model has a larger maximum prediction step for translational signals, the wavenet is able to make a better prediction for the rotations. These results are similar to the results observed in Section 5-5-5.

DoF	AR	WNN
Surge (x)	15	15
Sway (y)	15	14
Heave (z)	13	13
Roll (ϕ)	8	9
Pitch (θ)	9	10

Table 5-5: Maximum prediction step at 5 [Hz]

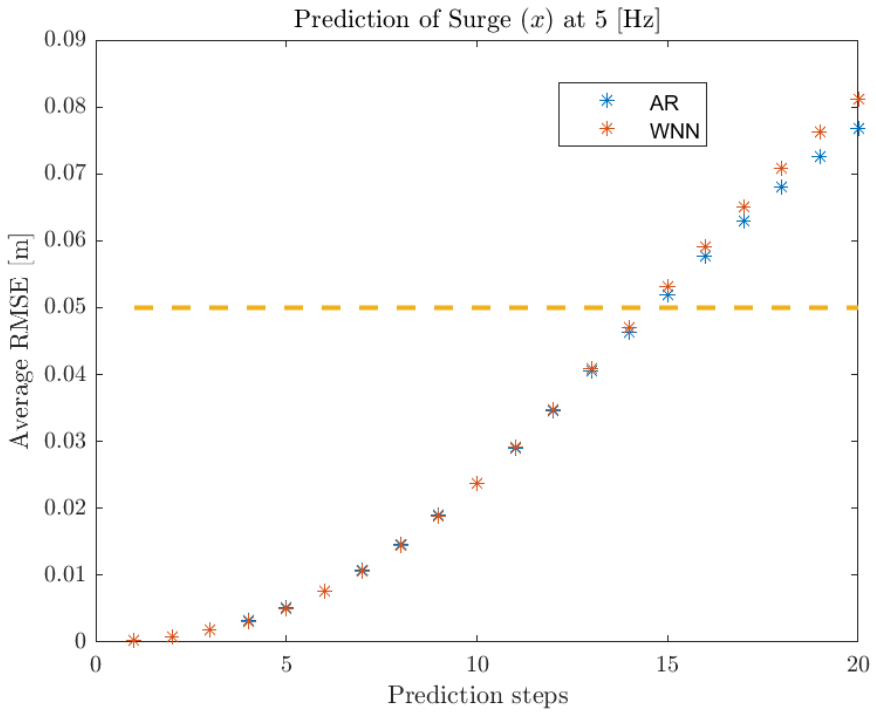


Figure 5-15: Prediction multiple steps ahead for x

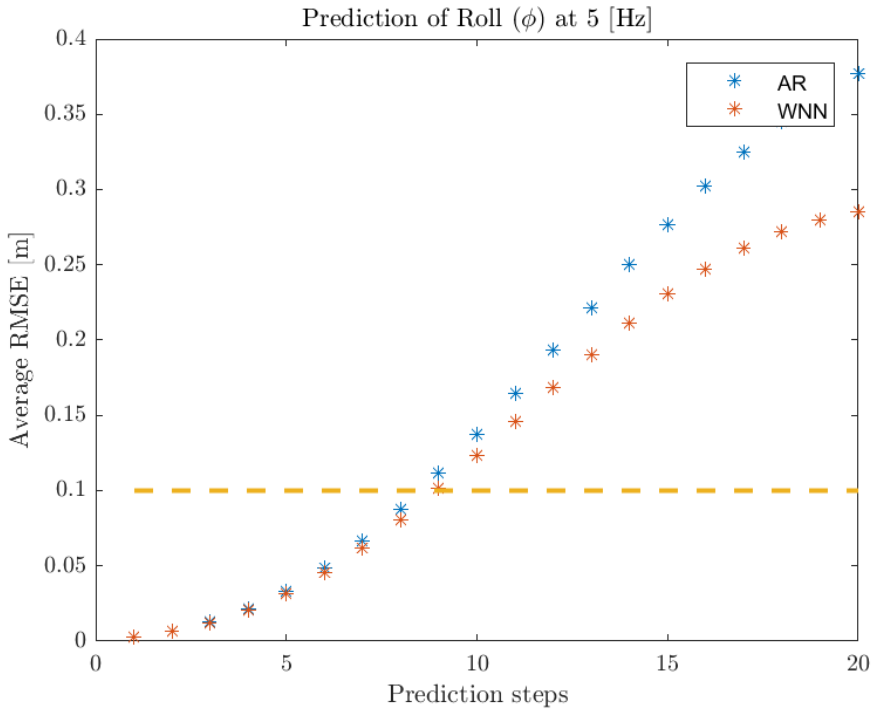


Figure 5-16: Prediction multiple steps ahead for ϕ

5-7 Conclusion

The wavelet transform is a mathematical tool for analyzing time series. A wavelet is a 'small' wave, which grows and decays in a finite time period and had zero average energy. It is similar to the Fourier transform. However, the wavelet is expressed in both frequency (via dilations) and in time (via translations). There are two main types of wavelet transforms: the continuous wavelet transform (CWT) and the discrete wavelet transform (DWT).
First, the wavelet transform is used for multiresolution analysis. The main motivation of using wavelet decomposition is the easy analysis of the obtained series, different levels of trends and details can be distinguished. These level can be modeled separately. Unfortunately, no clear levels of detail and trends could be defined for this data set, the data has a small bandwidth. So the wavelet decomposition does not improve the forecast.

The wavelet can also be used as an activation function for an artificial neural network (ANN), which is called a wavelet neural network (WNN). The WNN is a feed-forward neural network, with one or more inputs, with one hidden layer of ten neurons and a linear combiner output layer. The wavelet activation function is constructed from a Sigmoid function, since this is the most often used activation function for ANN. The parameters are trained using the *Levenberg-Marquant* search direction.
The training of a wavelet neural network requires more data than the data that was used for training the AR model, so the data set is split up in two data sets, instead of ten. For stability reason the WNN is only trained, validated and tested for a resampled data set of 5 [Hz]. To be able to compare the AR model with the WNN, the AR model is also trained and tested over the exact same data sets.
The values of the prediction error of the AR model and the WNN are very close. The AR model gives a smaller error for x, y and z, the rotations however are better modeled by the wavenet. The same results are observed when the maximum prediction step within the accuracy bounds is determined. Coupling the translational (x, y and z) signals in one model reduces the prediction error. A model with three inputs (z, ϕ and θ) and two outputs (ϕ and θ) is better capable of modeling the rotations, due to the high correlations between z, ϕ and θ .
The differences are small and the wavenet will give a different results every time the model is trained, since the initial conditions are randomly initialized. More data is required to give a more exact estimate of the prediction error and to be able to determine which model should be chosen.

Chapter 6

Anti-Causal filter

6-1 Introduction

In Section 2-5 the filters which are present in the Ampelmann system have been discussed and the problem of the delay which arise. Since a causal filter cannot be constructed such that the signal at the output of the filter is not phase shifted with respect to the input signal. The filters create a maximum phase shift of approximately 15° which resembles to a maximum error of approximately 13% of the amplitude of the wave. To solve this problem an anti-causal filter can be applied, which requires knowledge of future data. Although this thesis is focused on the prediction method, it is interesting to see what the obtained prediction can contribute to the Ampelmann control loop.

6-2 Anti-causal filter

Anti-causal filters are also called 'zero phase filters'. There are several methods to construct an anti-causal filter $H_{NC}(z)$ given a causal filter $H_C(z)$ [29].
One common method is is to construct the new anti-causal filter from, for example the Finite Impulse Response (FIR) $H_C(z)$ by:

$$H_{NC}(z) = H_C(z^{-1}) \cdot H_C(z) \tag{6-1}$$

Clearly $H_C(z^{-1})$ is non causal because if:

$$H_C(z) = \sum_{k=0}^{N} a_k z^{-k} \qquad \text{is causal} \tag{6-2}$$

$$H_C(z^{-1}) = \sum_{k=0}^{N} a_k z^{k} \qquad \text{is non causal} \tag{6-3}$$

In Fourier space $z \rightarrow e^{j\omega}$:

$$H_{NC}(e^{j\omega}) = H_C(e^{-j\omega}) \cdot H_C(e^{j\omega}) \tag{6-4}$$
$$= H_C^*(e^{j\omega}) \cdot H_C(e^{j\omega}) \tag{6-5}$$
$$= |H_C(e^{j\omega})|^2 \tag{6-6}$$

Here $H_C^*(e^{j\omega})$ represents the complex conjugate. Here H_{NC} is real and greater than zero in Fourier space, therefore the phase shift will be zero.
The filter is also shown in the following block diagram:

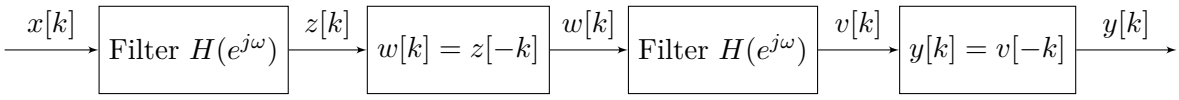


Figure 6-1: Non-causal filter

Where:

$$Z(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega}) \tag{6-7}$$
$$W(e^{j\omega}) = Z^*(e^{j\omega}) = H^*(e^{j\omega})X^*(e^{j\omega}) \tag{6-8}$$
$$V(e^{j\omega}) = H(e^{j\omega})W(e^{j\omega}) = |H(e^{j\omega})|^2 X^*(e^{j\omega}) \tag{6-9}$$
$$Y(e^{j\omega}) = V^*(e^{j\omega}) = |H(e^{j\omega})|^2 X(e^{j\omega}) \tag{6-10}$$
$$H_{eff}(e^{j\omega}) = |H(e^{j\omega})|^2 \tag{6-11}$$

So the effective frequency response caused by the non-causal filter is the magnitude of that in one direction squared. The time-reversal step requires that future data needs to be available.

6-3 Results

The AR-model gives the fastest and most stable prediction, therefore this will be used for testing the anti-causal filter. At every time step a 50-step ahead prediction will be made with the AR(7)-model. The past data and the 50-step ahead prediction are filtered by the anti-causal filter. For this test the Octans filter, 2nd order low-pass 2.0 [Hz], is used, since this filter causes the largest phase shift.
The result is shown in Figure 6-2. The (smooth) data is filtered with both the anti-causal filter and the Octans filter. The differences is clearly visible. The 50-step ahead prediction of the AR-model is accurate enough to be able to implement a zero-phase filter. The Root mean squared error (RMSE) between the data and both filter outputs are shown in Table 6-1.

The Octans motion sensor has a resolution of 0.001 [m], so the incoming data is a 'bumpy' signal. The AR-model is not able to predict this raw data signal, since it will try to model those non-linearities. Which are obviously not present in the actual motion of the ship. Therefore here an example is shown with smooth data.

	RMSE[m]
Octans filter	0.0082
AC filter	0.0005

Table 6-1: RMSE between data and filter output

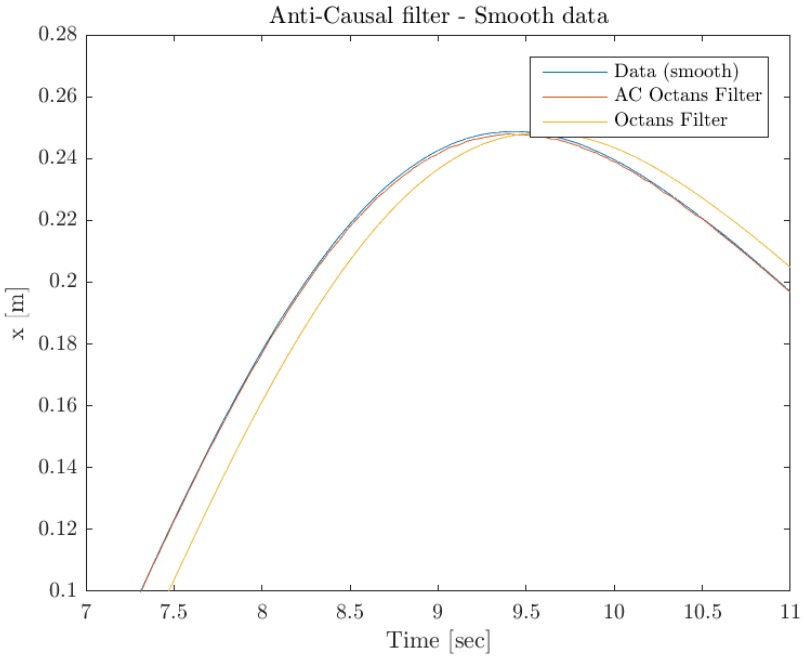


Figure 6-2: Non-causal filter and Octans filter

To be able to filter the raw data with the anti-causal filter, the method is slightly adjusted. The raw data is filtered by the Octans filter, with the filtered data the prediction is made. The past data and the prediction is time-inverse and passed through the second filter. The prediction part is now only filtered backwards and has a large influence on the filter output. The 50-step ahead prediction based on 7th order model was not satisfactory for this type of filtering. When using a model with an order of 70, which resembles to resampling to 5 [Hz], the prediction was is more accurate and is able to capture more of the dynamics of the waves. The result is shown in Figure 6-3. The RMSE is shown below:

	RMSE[m]
Octans filter	0.0094
AC filter	0.0012

Table 6-2: RMSE between raw data and filter output

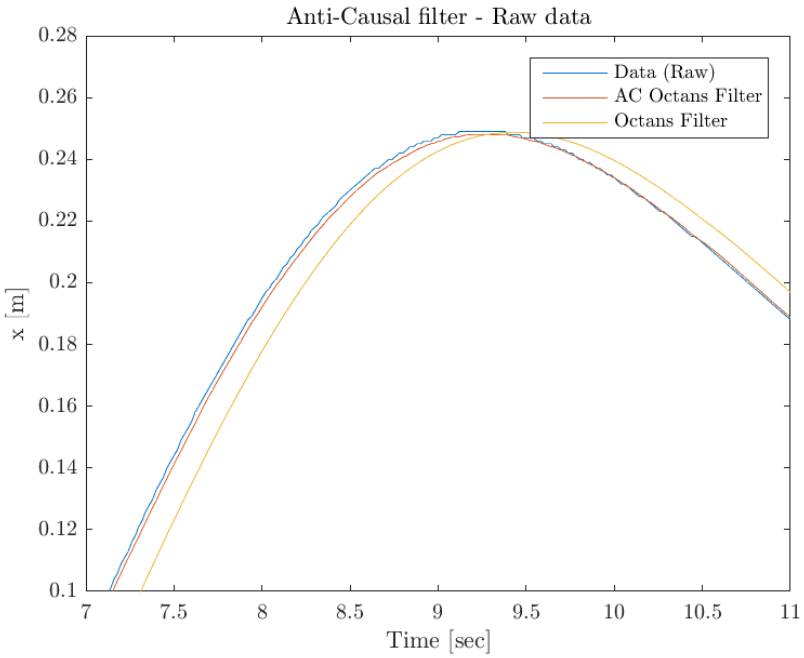


Figure 6-3: Non-causal filter and Octans filter

6-4 Conclusion

Although this research focuses on the prediction method, in this final chapter the possible contribution of the prediction to the Ampelmann system is shown. With knowledge of future data an anti-causal filter can be applied. This anti-causal filter can filter with zero phase shift and in this way the delays in the control loop can be removed.
The anti-causal is constructed by filtering the signal (including the prediction) through a causal filter, then the signal is time-inversed and passed through the same filter again. By time-inversing the twice filtered signal again, the zero-phase filtered signal is obtained.
The 50-step ahead prediction is accurate enough to be able to implement a non-causal filter. The RMSE between the filtered signal and the original decreased by approximately 95%. The data from the Octans motions sensor has a resolution of 0.001 [m] , so the signal is not smooth. The AR model is not able to model and predict those non-linearities. To be able to implement the non-causal filter on the raw data, the filter is adjusted. A more accurate prediction is necessary, so a larger model is used. The non-causal filter reduces the RMSE of the filtered raw data with 88%.

Chapter 7

Conclusion and Recommendations

In the previous chapters, a number of conclusions have been drawn. These conclusions are summarized in this chapter. And last, the final conclusion regarding the goal of this thesis is given. Furthermore, a number of recommendations are made for the future work.

7-1 Conclusions

- The workspace of the Ampelmann is defined as the total of six Degrees of Freedom (DoF) motion range limited by the minimum or the maximum stroke of the cylinders. The workspace relations are highly non-linear, therefore only a numerical estimation of the workspace can be found. In order to analytically describe the workspace, hyper ellipsoids are used to approximate the workspace boundaries. This analytical description can be used to limit the DoF, to make sure that the system will not exceed its workspace.

- The Ampelmann system gives a high priority to compensating the rotational components (ϕ, θ and ψ), because this will generate a horizontal stabilizing transfer deck. Second the translation directions (first x, y, next z) are limited. Residual motions are only allowed to occur in those three directions, since the gangway can compensate for those. Two main reason for residual motions are the difference in the limited reference signal and the actual available workspace. In previous research [1] this problem is solved by applying model predictive control (MPC). The second cause for residual motions are the delays introduced by the filters, which are present in the control loop. The delay can cause an error up to approximately 13% of the amplitude of the wave. This problem can be solved by applying an anti-causal filter. For both solution (MPC and an anti-causal filter) a prediction of the vessel motions is required.

- The Octans motion sensor has a resolution of 0.001 [m] or 0.01 [deg]. The accuracy of the translational movement (surge, sway and heave) is 0.05 [m] or 5% whichever is greater. The dynamic accuracy of the rotational movements (roll, pitch and yaw) is 0.01 [deg]. The goal for the accuracy of the prediction is set to the accuracy of the Octans

motion sensor. Since it is inefficient to predict more accurate than the sensor. So for surge (x), sway (y) and heave (z) a prediction accuracy of 0.05 [m] is desired. The Octans motions sensor is able to measure the rotational signal more accurately, which leads to a tight constraint. Therefore the accuracy goal for the rotational movements is also set to 5%, similar to the translations.

- During this research vessel motion data is considered which is measured at the North Sea near Aberdeen, United Kingdom, in July 2015. The data set is considered as a time series process and it is assumed to be stationary. Stationary means that the probabilistic character of the series must not change over time. This properties can not be proven since there is no explicit knowledge of the latent time series process. But considering the data set and the auto-correlation plots there is no reason to assume that the probabilistic character will change and therefore (wide-sense) stationarity can be concluded. The stationarity principle is important for using the linear auto regressive models.

- A linear predictor model can be used as a representative of the dynamical system. The problem is seen as a pure (wide-sense) stationary time-series problem, therefore the Auto Regressive (AR) and the Auto Regressive Moving-Average (ARMA) model are seen as the best fit for describing the signals. The AR model is based on linear combination of the past obeservations and a disturbance term. The ARMA model is based on a linear combination of the past observations and a linear combination of the current and past innovation terms. An adequate model order (i.e. size) is determined by multiple criteria; variance accounted for (VAF), Akaike information criterion (AIC) and cross validation. By comparing the multiple step-ahead prediction error of the different models, the AR(7) model is the most suitable model to describe the wave data. The prediction error generated by the AR(7)-model for surge (x) for a 1 second ahead prediction is 0.0062 [m], with a wave amplitude of approximately 0.5 [m].

- The parameters of the AR model can be found by the least squares approach, but this method is computational intensive, due to large matrix inversions. To be able to predict real-time three solutions for this problem are found. The first solution is adaptive estimation, where the parameters are continuously updated by a recursive algorithm to adapt the AR model to each new observation. Unfortunately this method does not guarantee stability. The second method is batch-wise training. A batch estimation of the AR model, through least-squares, every certain amount of time, to be sure that the model is always valid for the current sea state. The minimum required batch size is 70 data points (50 [Hz]). Last re-sampling is considered; using down-sampling less data points need to be considered for training and less prediction steps have to be made. A combination of batch-wise training with down-sampling gives a less computational costly, stable and accurate prediction.

- Wavelets are mathematical tools for analyzing time series. The wavelet transform will show the wave spectrum, which show how the energy is distributed across different frequency components over time. The wavelet transform of the data set shows that the waves have a narrow bandwidth 0 [Hz]- 0.3 [Hz], with similar behavior over time. The wavelet transform can be used to decompose the time series into varying scales of

temporal resolution. The AR model is trained on each resolution scale. No clear levels of detail and trends are distinguished, so this method did not benefit the prediction.

- A Wavenet is a feed forward Artificial Neural Network (ANN) with one-hidden layer which uses wavelets as activation functions. It combines the the advantages offered by neural network processing and wavelet representation. The wavenet uses a sigmoidal wavelet activation function and is trained using the *Levenberg-Marquant* direction. For training of the Neural network a larger data set is required compared to training of the AR model. The prediction error of the rotational signals decreases using a wavenet for prediction, compared to the AR-model. The AR model is more suitable for modeling the translations (x, y and z). The difference between the prediction errors of both models is small. More data is required to give a more exact estimate of the prediction error and to be able to determine which model is better.

- A multivariate wavenet model combines the signals in one model. It reduces the computational power necessary for training, since the signals will not have to be modeled separately. Compared to the five separate models, coupling the translations (x, y and z) in one model reduces the prediction error. Also a model with three inputs (z, ϕ and θ) and two outputs (ϕ and θ) is better capable of modeling the rotations. The prediction error generated by the multivariate-wavenet for surge (x) for a 1 second ahead prediction reduces to 0.0049 [m], with a wave amplitude of approximately 0.5 [m]. Again it must be noted that the differences are small and more data should be used to confirm the results.

- The maximum prediction step within the accuracy limits (0.05 [m] or 0.1 [deg]) is determined for both models. Both AR-model and the wavenet are able to predict the surge (x) signal for three seconds ahead accurately. But the maximum prediction step of the complete model depends on the signal with the smallest maximum prediction within the accuracy limits. Both models have the smallest maximum prediction step for roll (ϕ). So the AR-model is able to predict all signals accurately for 1.6 [sec] and the wavenet for 1.8 [sec].

- The prediction can be used in combination with an anti-causal filter. An anti-causal filter can filter with zero-phase shift, therefore does not introduce a delay in the system. The AR-model is able to predict accurate enough to implement the non-causal filter for filtering the raw data coming in from the Octans motion sensor. The RMSE between the raw data and the filtered data reduced by 88% compared to the original Octans filter.

Finally, regarding the goal of this research as described in Chapter 1, the following conclusion can be drawn:

Two prediction models, the AR model and the Wavenet, are derived which are able give an accurate short term prediction. The prediction is satisfactory to implement an anti-causal filter, which reduces the residual motions. The AR-model is stable, need less training data and is more simple compared to the Wavenet, therefore would be the best choice to implement on the Amplemann system with the current control system. When a more advanced control system is used in the Ampelmann system, the Wavenet can be applied. The results indicate that the Wavenet will most likely improve the prediction

7-2 Recommendations

- The Ampelmann system uses a Programmable Logic Controller (PLC) to control the system. This controller has very limited computational power. In further research it should be tested whether both models, with adjustments for real-time prediction, are able to be implemented on the real system. Also implementing another controller or adding an extra computer could be solutions which should be researched.
- When continuing with a more advanced ANN, Long Short-Term Memory (LSTM) should be considered. LSTM is a recurrent neural network based on a special network architecture that is able to capture very long time dependencies. Which can be interesting for ship motion prediction where long time dependencies can be present [30].
- In Chapter 6 an example is shown of an anti-causal filter. In further research other methods for zero-phase filtering should be researched. The non-causal filter should also be tested to its properties for disturbance rejection.
- The wavenet is able to give an accurate 1,8 [sec] ahead prediction for all signals. In previous research [1] it is stated that a prediction of 3 [sec] is required to implement MPC for the Ampelmann system. It should be tested whether the accuracy bound could be less strict and if the prediction is satisfactory for MPC.

Appendix A

Model order

A-1 AR

The Akaike information criterion (AIC) and variance accounted for (VAF) values are shown for all signals, for different models orders of the Auto Regressive (AR) model.

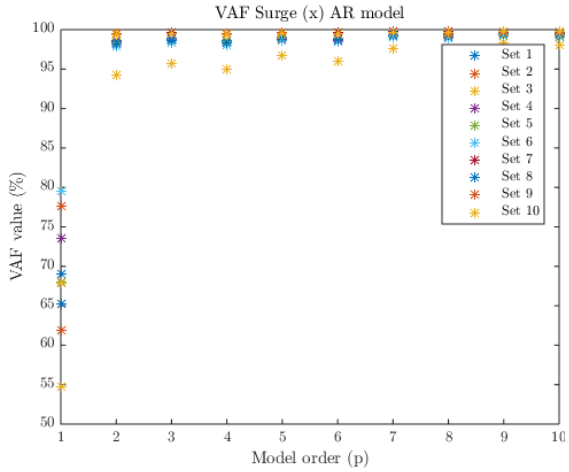


Figure A-1: VAF values, 50 step ahead

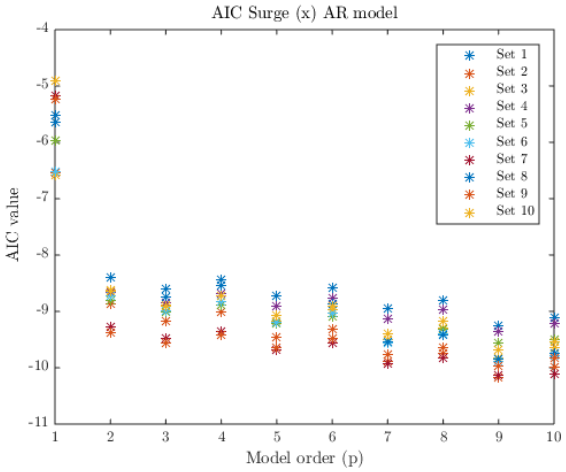


Figure A-2: AIC values, 50 step ahead

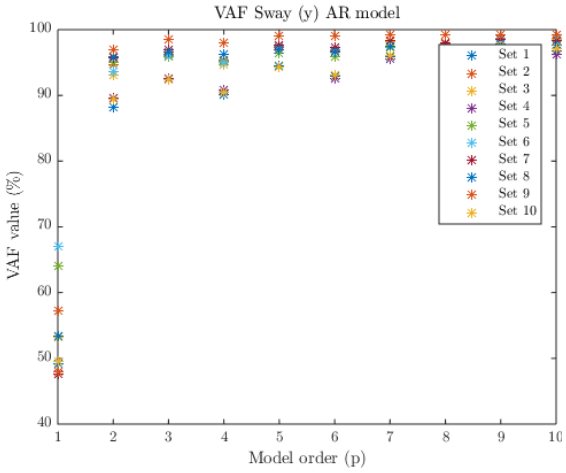


Figure A-3: VAF values, 50 step ahead

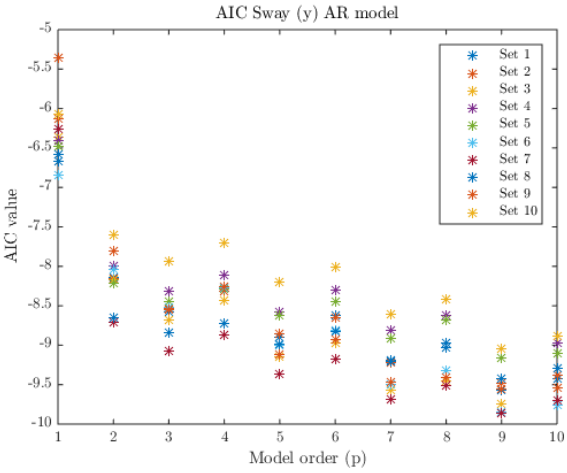


Figure A-4: AIC values, 50 step ahead

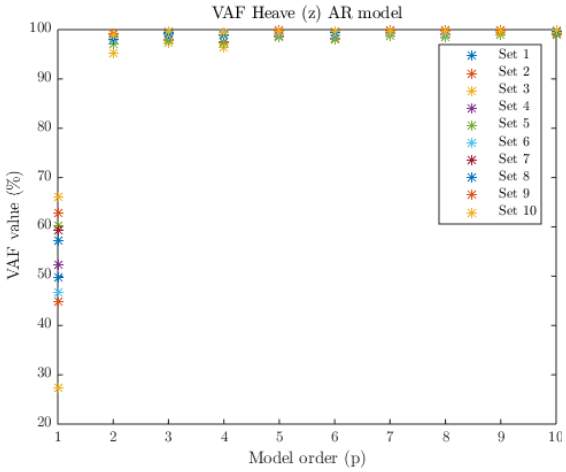


Figure A-5: VAF values, 50 step ahead

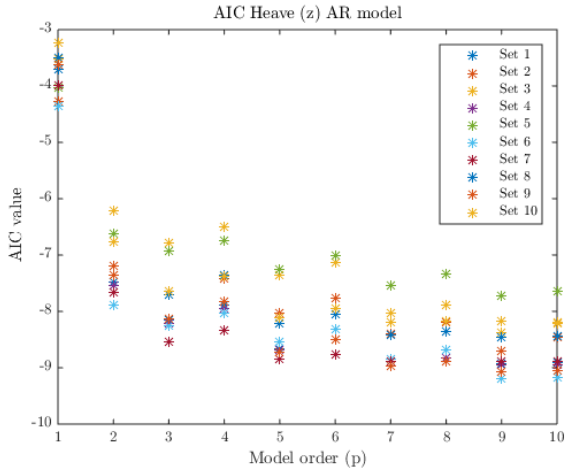


Figure A-6: AIC values, 50 step ahead

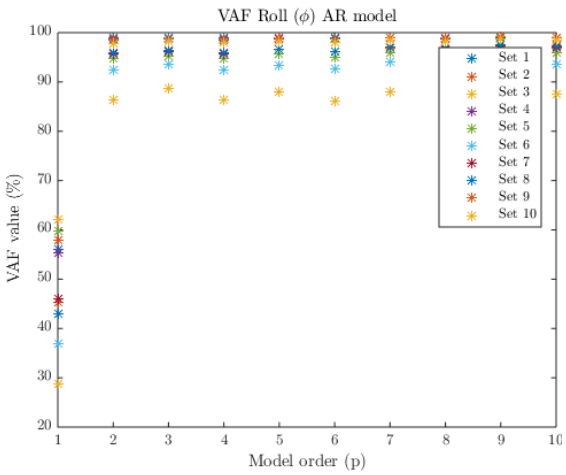


Figure A-7: VAF values, 50 step ahead

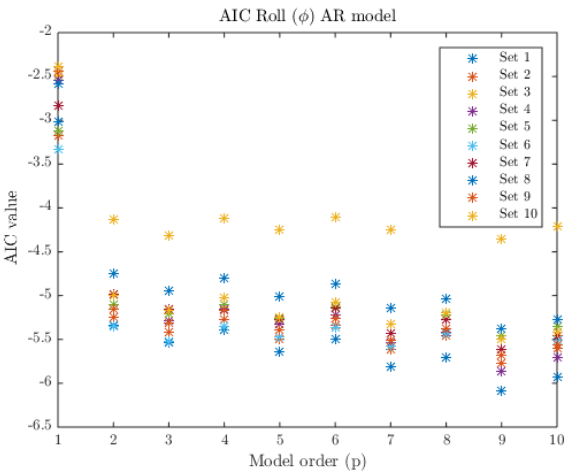


Figure A-8: AIC values, 50 step ahead

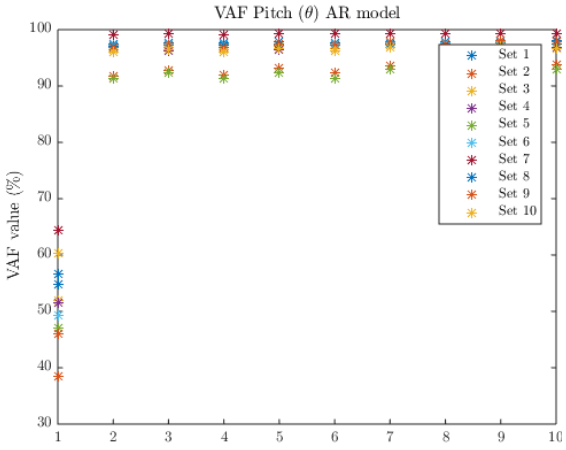


Figure A-9: VAF values, 50 step ahead

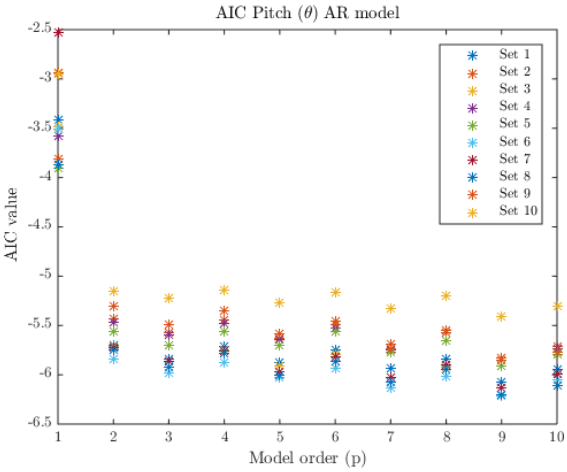


Figure A-10: AIC values, 50 step ahead

A-2 ARMA

The AIC and VAF values are shown for all signals, for different models orders of the Auto Regressive Moving-Average (ARMA) model.

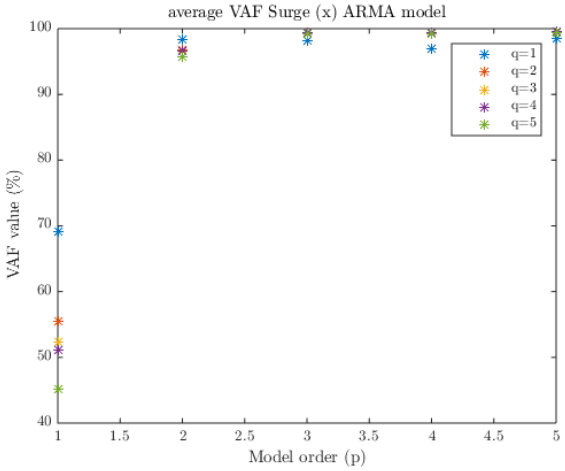


Figure A-11: VAF values, 50 step ahead

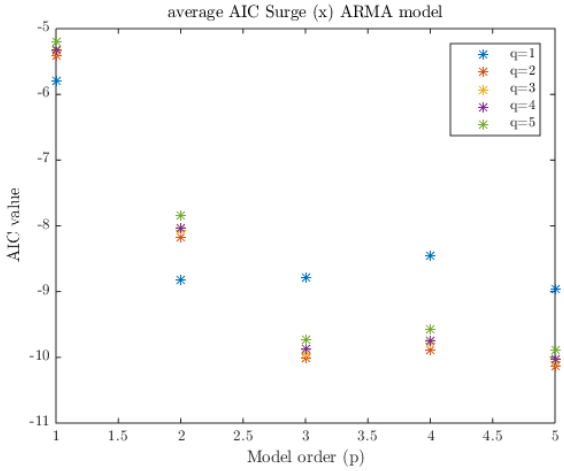


Figure A-12: AIC values, 50 step ahead

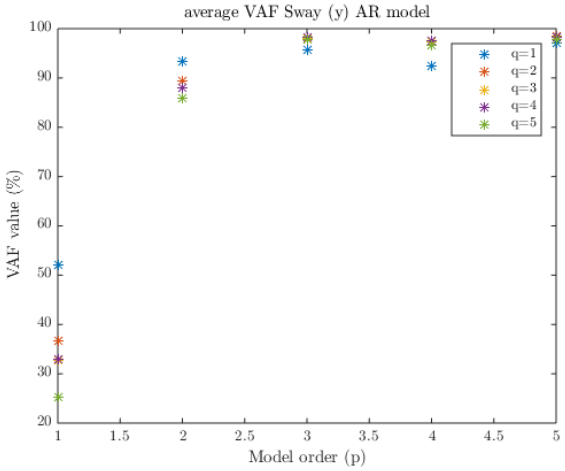


Figure A-13: VAF values, 50 step ahead

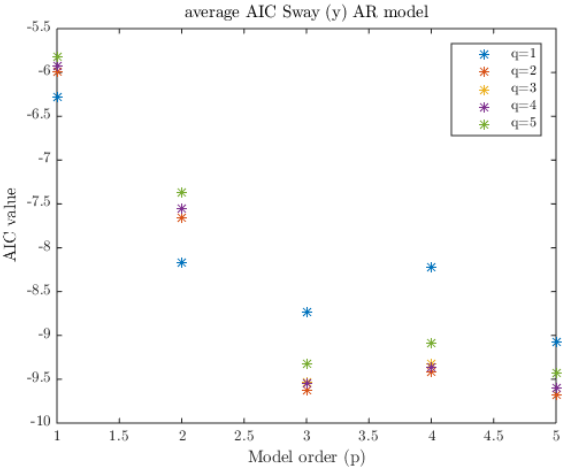


Figure A-14: AIC values, 50 step ahead

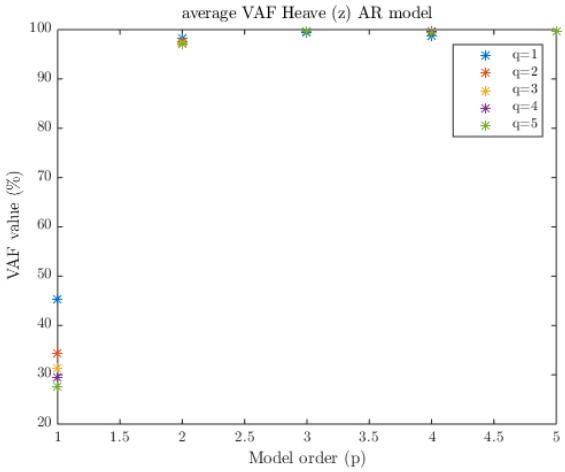


Figure A-15: VAF values, 50 step ahead

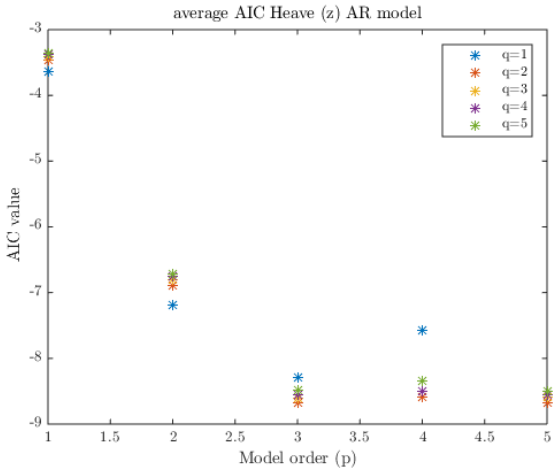


Figure A-16: AIC values, 50 step ahead

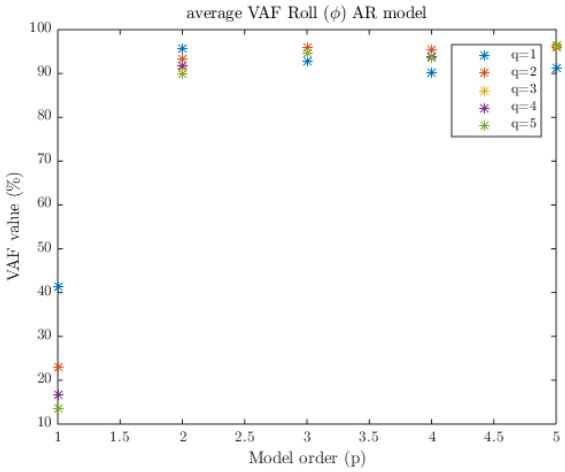


Figure A-17: VAF values, 50 step ahead

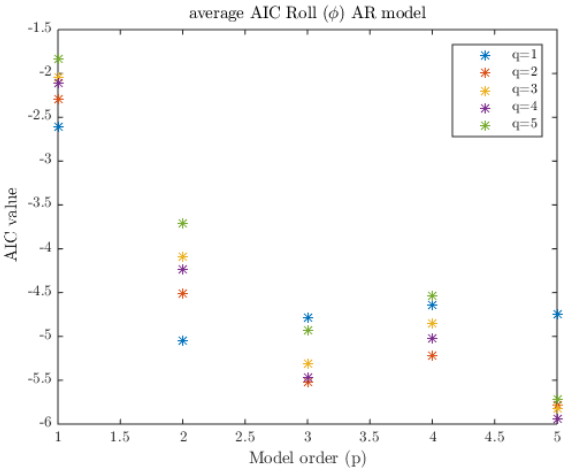


Figure A-18: AIC values, 50 step ahead

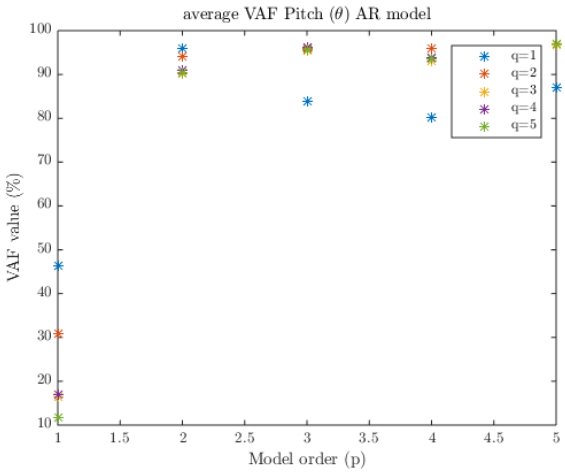


Figure A-19: VAF values, 50 step ahead

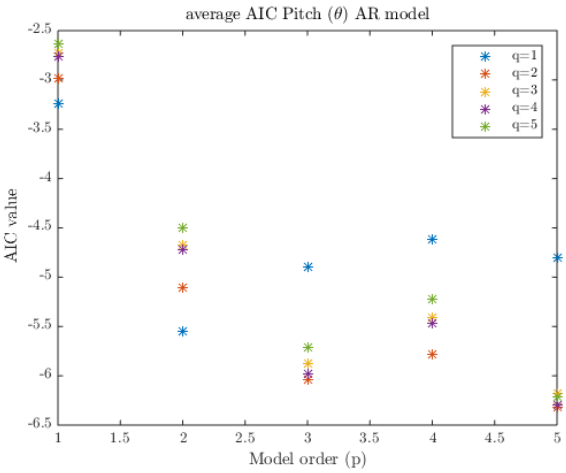


Figure A-20: AIC values, 50 step ahead

Appendix B

Maximum prediction step

A prediction is made up to 4 seconds with the Auto Regressive (AR) model and the Wavelet Neural Network (WNN). The prediction errors for all signals are shown.

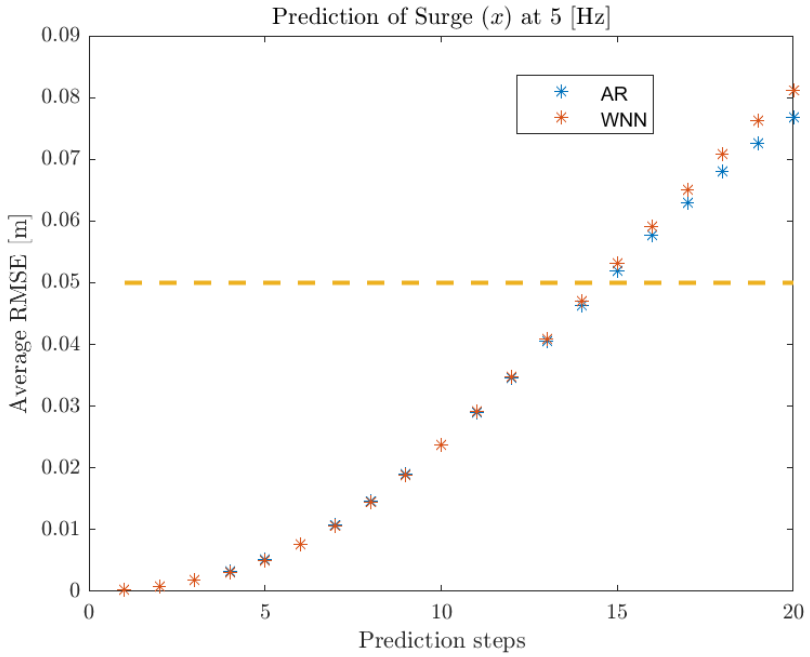


Figure B-1: Prediction multiple steps ahead of x

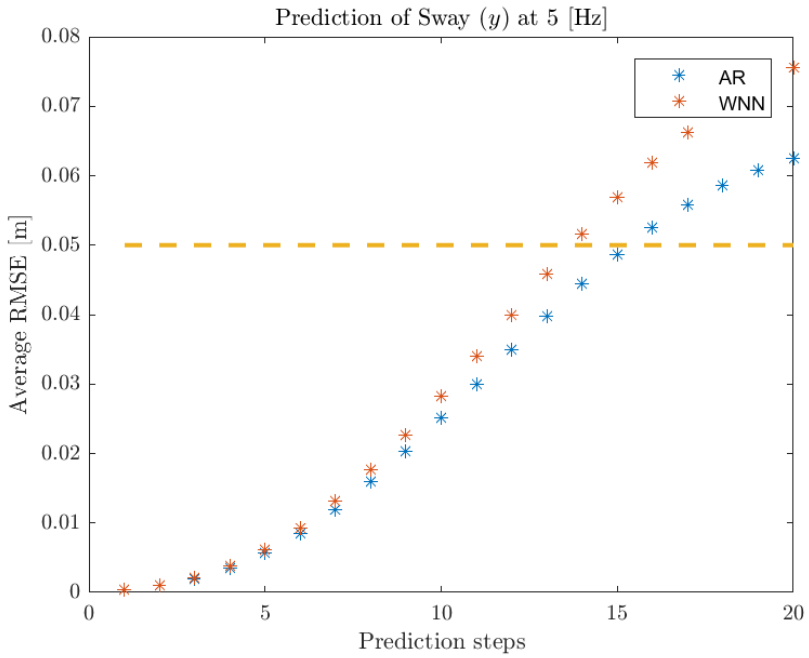


Figure B-2: Prediction multiple steps ahead of y

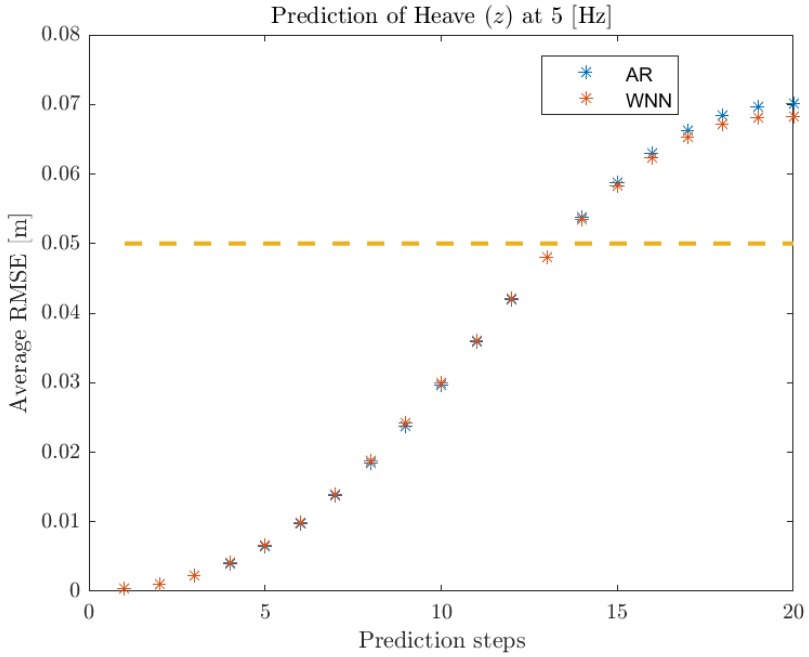


Figure B-3: Prediction multiple steps ahead of z

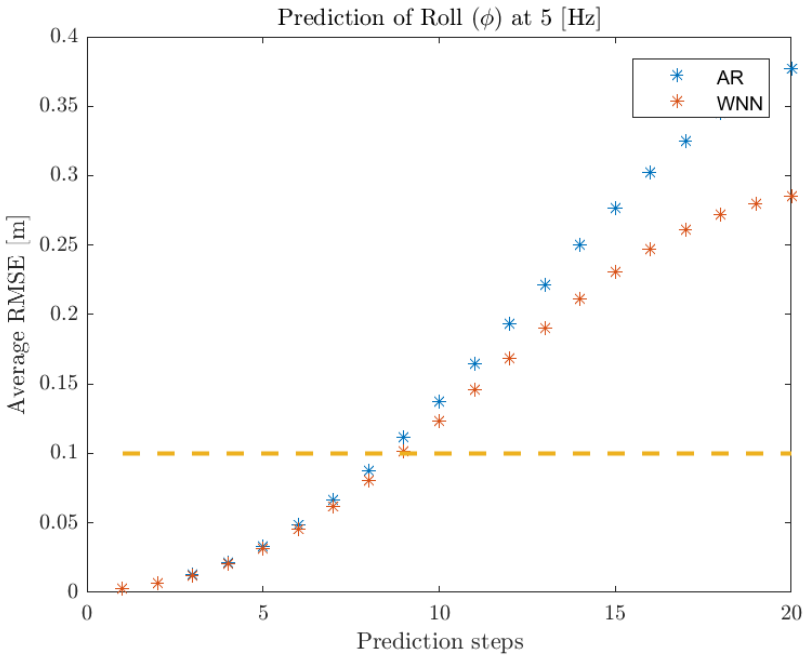


Figure B-4: Prediction multiple steps ahead of ϕ

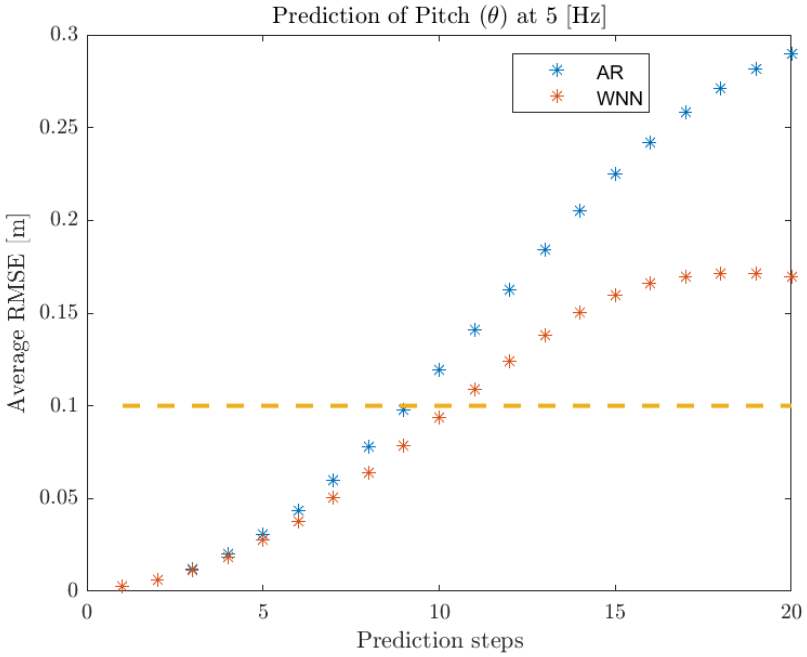


Figure B-5: Prediction multiple steps ahead of θ

Bibliography

[1] P. van Tricht, "Constrained model predictive control for the ampelmann system," Master's thesis, TU Delft, 2014.

[2] D. Cerda Salzmann, *Ampelmann: Development of the access system for offshore wind turbines*. TU Delft, Delft University of Technology, 2010.

[3] D. Jakobović and L. Budin, "Forward kinematics of a stewart platform mechanism," *Faculty of Electrical Engineering and Computing Faculty of Electrical Engineering and Computing. Unska*, vol. 3, 2002.

[4] A. van Leer, "Motion control enhancements for the ampelmann system," Master's thesis, TU Delft, 2012.

[5] iXBlue, "Fifth generation survey-grade surface gyrocompass and motion sensor," 2015.

[6] M. Verhaegen and V. Verdult, *Filtering and system identification: a least squares approach*. Cambridge university press, 2007.

[7] D. M. Dettling, "Applied time series analysis." Course Notes, 2014. Institute for Data Analysis and Process Design, Zurich University of Applied Sciences.

[8] P. M. van den Hof, "System identification." Lecture Notes, 2006.

[9] H. Akaike, "A new look at the statistical model identification," *Automatic Control, IEEE Transactions on*, vol. 19, no. 6, pp. 716–723, 1974.

[10] F. Fusco, "Short-term wave forecasting as a univariate time series problem.(ee/2009/jvr/03)," 2009.

[11] G. E. Bottomley and S. T. Alexander, "A novel approach for stabilizing recursive least squares filters," *Signal Processing, IEEE Transactions on*, vol. 39, no. 8, pp. 1770–1779, 1991.

[12] S. Bittanti and M. Campi, "Least squares based self-tuning control systems," *NATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES*, vol. 153, 1996.

[13] G. W. Colman and J. W. Wells, "On the use of rls with covariance reset in tracking scenarios with discontinuities," in *Electrical and Computer Engineering, 2006. CCECE'06. Canadian Conference on*, pp. 693–696, IEEE, 2006.

[14] A. Vahidi, A. Stefanopoulou, and H. Peng, "Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments," *Vehicle System Dynamics*, vol. 43, no. 1, pp. 31–55, 2005.

[15] C. Zengqiang, L. Maoqiong, and Y. Zhuzhi, "Convergence and stability of recursive damped least square algorithm," *Applied mathematics and mechanics*, vol. 21, no. 2, pp. 237–242, 2000.

[16] D. B. Percival and A. T. Walden, *Wavelet methods for time series analysis*, vol. 4. Cambridge University Press, 2006.

[17] P. Cristea, R. Tuduce, and A. Cristea, "Time series prediction with wavelet neural networks," in *Neural Network Applications in Electrical Engineering, 2000. NEUREL 2000. Proceedings of the 5th Seminar on*, pp. 5–10, IEEE, 2000.

[18] S. Soltani, "On the use of the wavelet decomposition for time series prediction," *Neurocomputing*, vol. 48, no. 1, pp. 267–277, 2002.

[19] S. Osowski and K. Garanty, "Forecasting of the daily meteorological pollution using wavelets and support vector machine," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 6, pp. 745–755, 2007.

[20] Y. Chen, B. Yang, and J. Dong, "Time-series prediction using a local linear wavelet neural network," *Neurocomputing*, vol. 69, no. 4, pp. 449–465, 2006.

[21] A. Aussem and F. Murtagh, "Combining neural network forecasts on wavelet-transformed time series," *Connection Science*, vol. 9, no. 1, pp. 113–122, 1997.

[22] B. Vidakovic and P. Mueller, "Wavelets for kids," *Instituto de Estadística, Universidad de Duke*, 1994.

[23] Wikipedia, "Discrete wavelet transform," 2015.

[24] C. Torrence and G. P. Compo, "A practical guide to wavelet analysis," *Bulletin of the American Meteorological society*, vol. 79, no. 1, pp. 61–78, 1998.

[25] D. Veitch, "Wavelet neural networks and their application in the study of dynamical systems," *Department of Mathematics university of York UK*, 2005.

[26] C. M. Westphal, "Signal approximation with a wavelet neural network," tech. rep., DTIC Document, 1992.

[27] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[28] P. Cunningham, J. Carney, and S. Jacob, "Stability problems with artificial neural networks and the ensemble solution," *Artificial Intelligence in Medicine*, vol. 20, no. 3, pp. 217–225, 2000.

[29] C.-Y. Tan, "Non-causal zero phase fir filter with examples," *Accelerator Division Tevatron.*

[30] T. de Bruin, "Neural network based condition monitoring for track circuits," Master's thesis, TU Delft, 2015.

Glossary

List of Acronyms

MPC	model predictive control
DoF	Degrees of Freedom
TAB	Telescopic Access Bridge
IK	Inverse Kinematics
PLC	Programmable Logic Controller
DSP	Digital Signal Processing
FOG	Fibre Optic Gyroscopes
DP	Dynamic Positioning
HPU	Hydraulic Pressure Unit
AR	Auto Regressive
ARX	Auto Regressive with Exogenous inputs
ARMA	Auto Regressive Moving-Average
ARMAX	Auto Regressive Moving-Average with Exogenous inputs
ARIMA	Auto-Regressive Integrated Moving Average
FIR	Finite Impulse Response
OE	Output-Error
BJ	Box-Jenkins
RLS	Recursive Least Squares
ANN	Artificial Neural Network

VAF	variance accounted for
AIC	Akaike information criterion
RMSE	Root mean squared error
CWT	Continuous Wavelet transform
DWT	Discrete Wavelet transform
WNN	Wavelet Neural Network

List of Symbols

λ	Weighting factor
ψ	Yaw
$\psi(\cdot)$	Wavelet
$\rho_{x,\ldots,\psi}$	Weighting factors of each Degrees of Freedom (DoF)
τ	Time-shift
τ	Translation factor
θ	Parameter vector
θ	Pitch
φ	Regressor vector
φ	Roll
$A(q)$	Polynomial of the linear model
$B(q)$	Polynomial of the linear model
b_i	Bottom gimbal coordinates
$C(q)$	Polynomial of the linear model
c^t	Position vector of the centre
$D(q)$	Polynomial of the linear model
$e(k)$	Disturbance
$F(q)$	Polynomial of the linear model
$F(s)$	Second order low pass filter
F_l	Look-up table
F_t	Univariate distribution function
$G(s)$	Actuator
H_s	Significant wave height
$J_N(\theta)$	Cost-function
$K(k)$	Kalman gain
l_i	Length of cylinder
$m_x(k)$	Mean

O_b	Origin of base frame
p	Model order of $A(q)$
$P(k)$	Covariance matrix
q	Model order of $C(q)$
R_1	Covariance matrix
R_2	Covariance matrix
R_b^t	Rotation matrix
$R_x(k,l)$	Auto-correlation
r_z	Weighted radius of Heave
r_ψ	Weighted radius of Yaw
$r_{\varphi\theta}$	Weighted radius of Roll and Pitch
r_{xy}	Weighted radius of Surge and Sway
s	Dilation factor
$s(t)$	Sigmoid function
T	Set of time
T_z	Mean wave period
$u(k)$	Input vector
x	Surge
x_c	X-coordinate of the top frame
X_t	Set of random variable
y	Sway
$y(k)$	Output vector
y_c	Y-coordinate of the top frame
z	Heave
z_c	Z-coordinate of the top frame