Design of a Propeller Load Observer and Speed Predictor using Extended Kalman Filter and Neural Network

Ioannis Tzimis

Diploma Thesis

School of Naval Architecture and Marine Engineering
National Technical University of Athens

Supervisor: Assistant Prof. George Papalambrou

Committee Member : Prof. N. Kyrtatos
Committee Member : Associate Prof. C. Papadopoulos

July 2020

Acknowledgements

This work has been carried out at the Laboratory of Marine Engineering (LME) at the School of Naval Architecture and Marine Engineering of the National Technical University of Athens, under the supervision of Assistant Professor George Papalambrou.

I would first like to thank my thesis supervisor Assistant Professor George Papalambrou for giving me the chance and motivation to work on this topic. I would also like to thank him for his patience, continuous support and immense knowledge. His guidance helped me in all the time working on this thesis.

I would like to thank Professor Nikolaos Kyrtatos for providing the opportunity to work with the full-scale hybrid diesel-electric marine propulsion powertrain of LME. I also thank him for being a member of my supervisors committee.

I would also like to thank Associate Professor Christos Papadopoulos for evaluating my work and being a member of my supervisors committee.

I would like to express my sincere gratitude to Mr. Nikolaos Planakis, PhD candidate of School of Naval Architecture and Marine Engineering for his contribution to the experimental validation of observer system performance. I am extremely thankful to him for sharing expertise, and sincere and valuable guidance and encouragement extended to me.

I take this opportunity to express gratitude to all LME fellow members for their help and support.

I am also sincerely grateful to my family, for the unceasing encouragement, support, motivation and attention throughout my studies.

I also place on record, my sense of gratitude to one and all, who directly or indirectly, have lent their hand in this venture. This accomplishment would not have been possible without them. Thank you.

Abstract

In this thesis, the optimal design of an observer for the determination of the propeller's power demand is investigated, during the charging of the propulsion system of the ship. The main purpose is to use this observer, for the operation of an optimal system controller.

The design of the observer is based on the comparison of two proposed models, the first of which calculates the torque of the propeller, using basic propulsion principles, and the second also gives an estimate of the propeller torque coefficient. The equations governing the two systems are used to construct the corresponding Extended Kalman Filter observers. These observers are able to apply to real-time sensors and filter their measurements, thus removing unwanted disturbances-sounds.

In the next stage of this work, the feasibility of using a shaft speed prediction system in a short future horizon is investigated. The purpose is to ensure the best possible operation of the propulsion control system. It is found that for this problem, the method of neural networks gives accurate and reliable predictions.

The complete modeling of the propulsion system is carried out, based on the technical characteristics of the available experimental testbed of LME. The implementation of the simulations takes place in MATLAB/Simulink programming environment and both the use of a duct propeller and the use of a controllable pitch propeller are tested.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Problem Framework

The growing global environmental concern and the intensification of the effort to reduce pollutant emissions have rapidly changed the operating conditions of the shipping industry. The need for fleets to increase their efficiency and adapt quickly to new regulations, makes it as urgent as possible to research and use advanced control technologies on ship propulsion energy management systems. Many recent researches and works have focused on the most accurate possible modeling of the ship dynamics but also on the provision of some basic system states required for automatic control. Designing a Model Predictive Controller for the ship's energy management needs, also becomes necessary for a number of additional issues related to the operation of ships, such as: performance monitoring, navigation control, thrust loss estimation, ventilation detection, power fluctuations and fuel reduction, wear and tear reduction in high sea state, hull fouling detection and maintenance.

The basic principles and mathematical formulas for ship resistance and propulsion should be applied, to model the ship's propulsion plant and obtain feedback control information in sea going, transient, port or other ship operating conditions. In addition to performing simulations, a key advantage of ship modeling is saving time adjusting the pre-installation propulsion control systems during sea trials, as this can be done during shop tests.

In order to achieve high vessel control performanc all propulsion system states should be available. In practise, this is often difficult due to reasons such as the inability to place a sufficient number of sensors, the high cost of specialized sensors installation or the uncertainties regarding the propulsion plant, propeller and ship resistance characteristics over time. Therefore, propeller torque (and its coefficient), along with propeller law constant could be estimated using observers. The observer signals can be more accurate, less expensive to produce, and more reliable than sensed signals, reducing the phase lag inherent in the sensor. In general, observers offer designers an inviting alternative to adding new sensors or upgrading existing ones. In some cases, the observer can be used to enhance system performance. The adaptive propeller observation scheme can be upgrated to the corresponding Extended Kalman Filter (EKF) observer so as to filter and reconstract noisy sensors' measurements during actual operating conditions. EKF has been recently the subject of extensive research and application, particularly in the area of autonomous or assisted navigation.

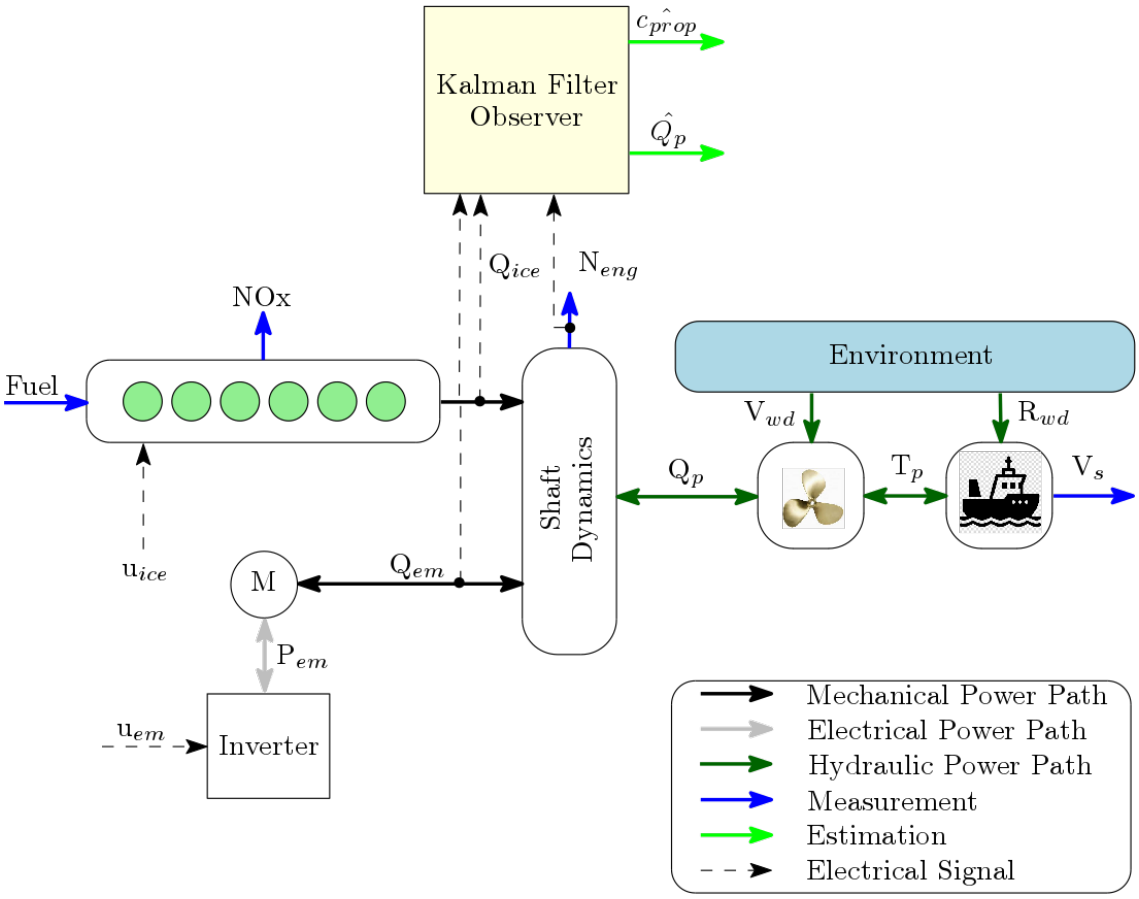

Figure 1.1: System's Layout

To maximize the efficiency of the propulsion control system, it is advisable to predict shaft rotational speed within a sufficient future horizon. Many methods have been proposed for various system states prediction. In this work, a Neural Network (NN) model is described as a well performed deterministic prediction approach to this problem. Although neural networks were firstly inspired by the goal to model a biological neuron, they have since diverged and become a matter of engineering and machine learning. For certain types of problems artificial NNs are among the most effective learning methods currently known.

Literature Review

As mentioned above, the basic initial requirement was the appropriate modeling of the ship's propulsion plant. Several bibliographic references focus on the ship's basic modeling principles, such as [2] or [3]. To perform simulations, a type of ship with similar horsepower levels was chosen with those of the internal combustion engine of the available experimental testbed at LME and the complete modeling of the ship was based on [4]. In order to simulate realistic sea states, it was deemed necessary to develope a model for the production of irregular waves. The toolbox presented in [4] (based on [1]) was used for this purpose.

In recent years several observer models have been proposed for marine applications such as those described in [5], [6]. For the appropriate observer choise, two adaptive propeller estimation schemes were tested in this work, based on the models described in [7]. The first one arises from the main ship propulsion dynamics equations and the second depends on the torque coefficient estimation. Both of them were used to compose the corresponding Extended Kalman Filter observers, for noisy measurements and disturbance estimation. The main EKF mathematical formulas and state estimation theory are described extensively in [8], [9] and its implementation can be done using the corresponding EKF algorithms in MATLAB/Simulink. The basic idea is that given the output measurement (*shaft rotational speed*) and the control input (*engine torque*) the EKF observer filters, reconstrucs and estimates the states in interest of the propulsion system (*filtered shaft rotational speed* and *propeller torque as disturbance*). In [7] also a thrust estimation scheme is proposed but with some application restrictions mentioned in fourth Chapter.

The propeller load characteristics can be used by the control system to estimate future load demand, if the future rotational speed is known/predicted. For this purpose a Markov Chain and a Neural Network model were tested. In [10] a Markov Chain that models the power request dynamics is proposed ,to improve the prediction capabilities of model predictive control, while in [11], [12] Markov Chain performance is compared to other stohastic models. The architecture of Neural Networks is described and formulated in detail in a number of bibliographic references such as [13], [14].

Thesis Objective

In this thesis observer models were investigated in order to provide an accurate propeller load torque estimation to the controller of the modeled propulsion system. In particular, two main observation schemes were implemented both in adaptive and Kalman Filter form. Initially, the basic characteristics of the modeling system were listed, through the mathematical principles of ship resistance and propulsion. The type of vessel, the main propulsion engine and the propeller were selected. In order to obtain the most complete and accurate results possible, simulations were performed for both ducted propeller and controllable pitch propeller and the appropriate observer gains were selected. The simulations were performed considering sea wave disturbance, in order to approach real operating conditions, both during acceleration and during towing. The observed results were compared with the actual ones of the modeling, in order to confirm observervs' efficiency. To predict shaft rotational speed, it has been shown that a feedforward neural network makes it possible to give highly accurate estimates in a 10-second horizon. To select the appropriate structure and architecture of the neural network, a number of simulations were performed and the training times, regression were evaluated.

The rest structure of this thesis is organized as follows: In Chapter 2, a brief review of state estimation theory and Kalman Filter formula is illustrated. In Chapter A, the propulsion plant of the ship is modeled. The propeller load observer models are designed in Chapter 3 and the shaft speed prediction models are described in Chapter 4. The evaluation of the performance of the tested models takes place in Chapter 5 alongth with their corresponding simulation results. In Chapter **??** the experimental facility is presented, along with the installed engine sensors and the data acquisition system. Finally,

the conclusions of this work are presented in Chapter 6.

Chapter 2

State Observers in Control Theory

In this chapter, the state estimation and filtering theory are presented. After the importance and the applications of the observers are mentioned, an extensive description of their mathematical formulas is given. In the first section the main parts of the state estimation theory are recorded, through the description of a general observer model. This is the Generalized Extended State Observer, which is used in Chapter 3, for the design of the adaptive propeller load observer. The Kalman filter and its uses are then described. In the last section an extension of the Kalman filter for nonlinear systems is introduced. This filter, called the Extended Kalman Filter, composes a realistic version of the proposed propeller load observer.

2.1 State Estimation Theory

In order to present state estimation theory in the most clear way possible it is necessary to define what the term state means. The states of a system are those variables that provide a complete representation of the internal condition or status of the system at a given instant of time [9]. To achieve feedback control, a basic requirement is that all system states are available. In fact, this is often difficult due to the following reasons:

- The inability to place a sufficient number of sensors.
- The high cost of specialized sensors installation. Sensor cost can substantially raise the total cost of a control system. In many cases, the sensors and their associated cabling are among the most expensive components in the system.
- The inability to measure some quantities due to their position. The objects being measured may be inaccessible for such reasons as harsh environments and relative motion between the controller and the sensor.
- Sensors usually induce significant errors such as stochastic noise, cyclical errors, and limited responsiveness.

Therefore, some or all system states are estimated by *observers*. The observer signals can be more accurate, less expensive to produce, and more reliable than sensed signals. Observers offer designers an inviting alternative to adding new sensors or upgrading existing ones. In some cases, the observer can be used to enhance system performance. It can be more accurate than sensors or can reduce the phase lag inherent in the sensor. Observers can also provide observed disturbance signals, which can be used to improve

disturbance response. In other cases, observers can reduce system cost by augmenting the performance of a low-cost sensor so that the two together can provide performance equivalent to a higher cost sensor [15].

The general concept of estimating the states of a system by observers is discribed ,according to [16], [17]. State observers consist of a mathematical model, which is similar to the system's model. Given the output variables y and the control variables u, the observer estimates the state variables $\hat{x}$. The state vector contains these variables. It describes the state of the dynamic system and represents its degrees of freedom. As mentioned above, the variables in the state vector cannot be measured directly but they can be inferred from values that are measurable. A gain matrix K is also chosen so that the error between the system states and the estimated states converges rapidly to zero. If the observer estimates all state variables of the system, regardless of whether some state variables are available for direct measurement, it is called a "full-order observer". There are times when this is not necessary, when it is needed observation of only the unmeasurable state variables, but not of those that are directly measurable as well. An observer that estimates fewer state variables, is called a "reduced-order observer". If the order of the reduced-order state observer is the minimum possible, the observer is called a "minimum-order observer".

A prerequisite for state estimation is that the system is observable. The system is said to be completely observable if every state $x(t_0)$ can be determined from the observation of $y(t)$ over a finite time interval $t_0 \leq t \leq t_1$. The system is, therefore, completely observable if every transition of the state eventually affects every element of the output vector.

Similar state estimation formulas are proposed in [18], [19], [20] ,while in [21] different models of observers are examined. In this section the Extended State Observer (ESO) formula is proposed ,based on [22]. Generalized form of ESO (for non integral systems) is presented in the next subsection.

An uncertain system with the order of n under the standard consideration is usually an integral chain system, described as follows:

$$\left\{\begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \quad \vdots \\ \dot{x}_n = f(x_1, ..., x_n, w(t), t) + bu \\ \quad y = x_1 \end{array}\right\} \tag{2.1.1}$$

where $x_1, ..., x_n$ are the states, u is the control input, y is the output, $w(t)$ is the external disturbance, b is a system parameter and $f(x_1, ..., x_n, w(t), t)$ represents the uncertain function.

In the framework of ESO, an augmented variable

$$x_{n+1} = f(x_1, ..., x_n, w(t), t) \tag{2.1.2}$$

is introduced to linearize system 2.1.1. Combining 2.1.1 with 2.1.2, the extended state equation is given by:

$$\left\{\begin{array}{c} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \vdots \\ \dot{x}_n = x_{n+1} + bu \\ \dot{x}_{n+1} = z(t) \\ y = x_1 \end{array}\right\} \tag{2.1.3}$$

with $z(t) = \dot{f}(x_1, ..., x_n, w(t), t)$.

In order to estimate the states, a linear ESO is designed as:

$$\left\{\begin{array}{c} \dot{\hat{x}}_1 = \hat{x}_2 - g_1(\hat{x}_1 - y) \\ \dot{\hat{x}}_2 = \hat{x}_3 - g_2(\hat{x}_1 - y) \\ \vdots \\ \dot{\hat{x}}_n = \hat{x}_{n+1} - g_n(\hat{x}_1 - y) + bu \\ \dot{\hat{x}}_{n+1} = -g_{n+1}(\hat{x}_1 - y) \end{array}\right\} \tag{2.1.4}$$

where $\hat{x_1}, \hat{x_2}, ..., \hat{x_n}$ and $\hat{x_{n+1}}$ are estimates of states $x_1, x_2, ..., x_n$ and x_{n+1} , respectively, and $g_1, g_2, ..., g_{n+1}$ are the observer gains to be designed.

Substracting 2.1.3 from 2.1.4, the error system is written as:

$$\left\{\begin{array}{c} \dot{e}_1 = e_2 - g_1 e_1 \\ \dot{e}_2 = e_3 - g_2 e_1 \\ \vdots \\ \dot{e}_n = e_{n+1} - g_n e_1 \\ \dot{e}_{n+1} = -g_{n+1} e_1 - z(t) \end{array}\right\} \tag{2.1.5}$$

where $e_i = \hat{x_i} - x_i (i = 1, 2, ..., n+1)$ represents the estimation error. By properly choosing the observer gains $g_1, g_2, ..., g_{n+1}$, the bounded stability of 2.1.5 is guaranteed under the assumption that $z(t)$ is bounded.

The standard ESO control law is usually designed as:

$$u = K_x x - \frac{s_{n+1}}{b} \tag{2.1.6}$$

where K_x is the feedback control gain.

Generalized Extended State Observer

The standerd ESO method ,presented above, is possibly not available for the following simple second-order system:

$$\begin{aligned} \dot{x}_1 &= x_1 - 2x_2 + f(x_1, x_2, w(t), t) \\ \dot{x}_2 &= x_1 + x_2 + u \end{aligned} \tag{2.1.7}$$

System 2.1.7 does not satisfy the standard formulation as 2.1.1 in the following two aspects. On the one hand 2.1.7, does not satisfy the integral chain form. On the other hand, the uncertainties $f(x_1, x_2, w(t), t)$ enter the system with a different channel from the control input u, i.e., the so-called matching condition is not satisfied. For the aforementioned case, the standard ESO law is no longer available. Thus, it is imperative to develop GESO for general systems which do not satisfy the standard formulation of system 2.1.1.

For the sake of simplicity, the following single-input-single-output system with mismatched uncertainties is considered (as the proposed GESO method is able to extend to multiple-input-multiple-output system with almost no modification).

$$\begin{aligned} \dot{x} &= Ax + B_u u + B_d f(x, w(t), t) \\ y_m &= C_m x \\ y_0 &= c_0 x \end{aligned} \tag{2.1.8}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}$, $v \in \mathbb{R}$, $y_m \in \mathbb{R}^r$ and $y_0 \in \mathbb{R}$ are the state vector, input, external disturbance, measurable outputs and controlled output, respectively. $f(x, w(t), t)$ is the uncertain function in terms of and w. A with dimension $n \times n$, B_u with dimension $n \times 1$, B_d with dimension $n \times 1$, C_m with dimensions $r \times n$ and c_0 with dimension $1 \times n$ are system matrices, respectively.

In 2.1.8, uncertainty function $f(x, w(t), t)$ is the lumped disturbance, which is a generalized concept, possibly including external disturbances, unmodeled dynamics, parameter variations and complex nonlinear dynamics which may be difficult for the feedback part to handle.

Equation 2.1.8 represents a more general class of systems 2.1.1 as compared with that of system since system 2.1.8 is not confined to integral chain form and may subject to mismatched uncertainties. The matching case is a special case of 2.1.8, by simply taking $B_u = \lambda B_d$, $\lambda \in \mathbb{R}$.

Similar to the standard ESO case, adding an external variable

$$x_{n+1} = f(x, w(t), t) \tag{2.1.9}$$

to linearize system 2.1.8, the extended system equation is obtained

$$\begin{aligned} \dot{\bar{x}} &= \bar{A}\bar{x} + \bar{B}_u u + Lz(t) \\ y_m &= \bar{C}_m \bar{x} \end{aligned} \tag{2.1.10}$$

where variables

$$\bar{x} = \begin{bmatrix} x \\ x_{n+1} \end{bmatrix} \quad , \quad z(t) = \frac{df(x, w(t), t)}{dt}$$

and matrices

$$\bar{A} = \begin{bmatrix} A_{n\times n} & (B_d)_{n\times 1} \\ 0_{1\times n} & 0_{1\times 1} \end{bmatrix}_{(n+1)\times(n+1)}$$

$$\bar{B}_u = \begin{bmatrix} (B_u)_{n\times 1} \\ 0_{1\times 1} \end{bmatrix}_{(n+1)\times 1}$$

$$L = \begin{bmatrix} 0_{n\times 1} \\ 1_{1\times 1} \end{bmatrix}_{(n+1)\times 1}$$

$$\bar{C}_m = \begin{bmatrix} C_m & 0_{r\times 1} \end{bmatrix}_{r\times(n+1)}$$

.
For system 2.1.10, the ESO is designed as follows:

$$\begin{aligned} \dot{\hat{\bar{x}}} &= \bar{A}\hat{\bar{x}} + \bar{B}_u u + G(y_m - \hat{y_m}) \\ \hat{y_m} &= \bar{C}_m \hat{\bar{x}} \end{aligned} \tag{2.1.11}$$

where $\dot{\hat{\bar{x}}} = \begin{bmatrix} \hat{x}^T & , \hat{x}_{n+1} \end{bmatrix}^T$, $\hat{x}$ and $\hat{x}_{n+1}$ are the estimates of the state variable $\hat{x}$, x and x_{n+1} in 2.1.10, respectively. Matrix G with dimensions $(n \times 1) \times r$ is the observer gain to be designed.

In the presence of mismatched uncertainties, the standard ESO law $u = K_x x - \hat{d}$ (where $\hat{d} = \hat{x}_{n+1}$ and K_x is the feedback control gain) cannot effectively compensate the uncertainties in 2.1.8.

The composite control law is disigned accoriding to 2.1 as follows:

$$u = K_x x + K_d \hat{d} \tag{2.1.12}$$

or

$$u = K_x \hat{x} + K_d \hat{d} \tag{2.1.13}$$

where K_x is the feedback control gain and K_d is the disturbance compensation gain, designed as:

$$K_d = -[c_0(A + B_u K_x)^{-1} B_u]^{-1} c_0 (A + B_u K_x)^{-1} B_d. \tag{2.1.14}$$

Note that the disturbance compensation gain K_d in 2.1.14 is a general case and suitable

for both matching and mismatching cases. For the matching case, i.e., $B_u = \lambda B_d$, $\lambda \in \mathbb{R}$, it can be obtained from 2.1.14 that the disturbance compensation gain reduces to $K_d = -1/\lambda$.

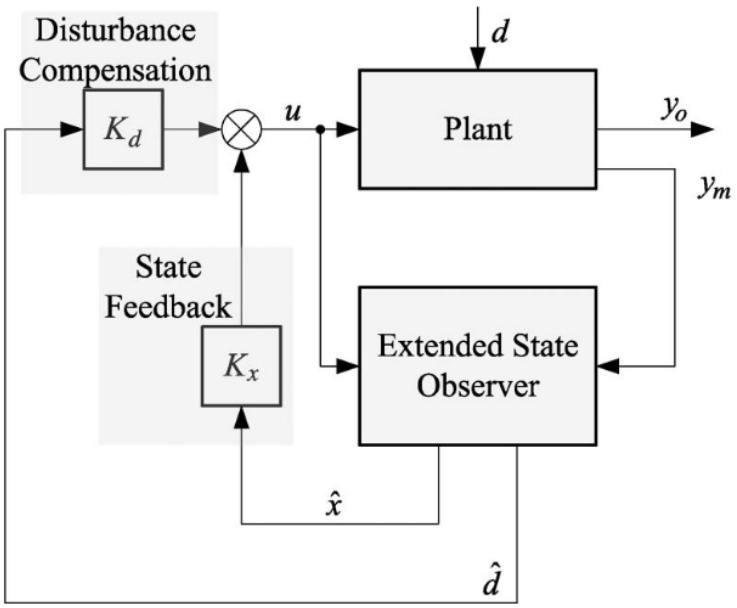

Figure 2.1: GESO and control layout

The state and disturbance estimation errors are defined as:

$$e_x = \hat{x} - x \tag{2.1.15a}$$

$$e_d = \hat{d} - d \tag{2.1.15b}$$

where $\hat{d} = \hat{x}_{n+1}$ represents the estimate of system uncertainties.

Combining 2.1.10, 2.1.11 and 2.1.15a, with 2.1.15b, the estimation error equation is given by:

$$\dot{e} = A_e e - Lz(t) \tag{2.1.16}$$

where

$$e = \begin{bmatrix} e_x \\ e_d \end{bmatrix} \quad , \quad A_e = \bar{A} - G\bar{C}_m. \tag{2.1.17}$$

It should be highlighted that, although there are similarities between the presented GESO and the other state estimation techniques, they have different focuses, so different design philosophies as the motivations are different. The main objective in GESO is to minimize the influence of the disturbance and uncertainty on the output provided that the disturbance has been estimated. Therefore, GESO formula was used to design the adaptive propeller load torque observer described in Chapter 3.

2.2 Kalman Filter

In actual operating conditions the sensors' measurements are contaminated with noise and cannot be used directly by the system's controller. The signals should therefore be filtered and reconstructed by an observer of a different type from the one presented in the previous section. This work is carried out by the "Kalman Filter" observer. Kalman filtering is an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe.

According to [23] the Kalman filter is essentially a set of mathematical equations that implement a predictor-corrector type estimator that is optimal in the sense that it minimizes the estimated error covariance ,when some presumed conditions are met. Since the time of its introduction, the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. This is likely due in large part to advances in digital computing that made the use of the filter practical, but also to the relative simplicity and robust nature of the filter itself. Rarely do the conditions necessary for optimality actually exist, and yet the filter apparently works well for many applications in spite of this situation.

A brief description of the Kalman filtering theory is recorded in [9], while in [23] a more extensive approach to this subject takes place. Detailed description for recursive estimation method are also available on [24]. In this work, the main Kalman filter algorythm is presented ,according to [8].

The Kalman filter is a recursive predictive filter that is based on the use of state space techniques and recursive algorithms. It estimates the state of a dynamic system. This dynamic system can be disturbed by some noise, mostly assumed as white noise. To improve the estimated state the Kalman filter uses measurements that are related to the state but disturbed as well. As shown in Fig. 2.2, the Kalman filter consists of two steps:

a. the prediction,

b. the correction.

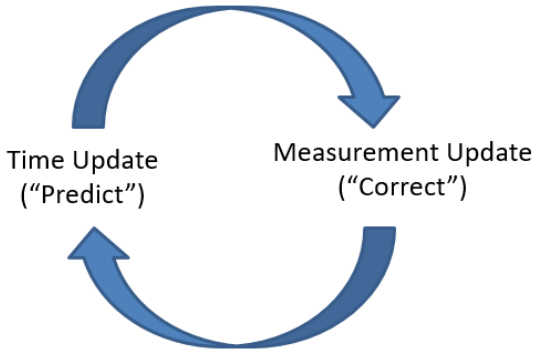

Figure 2.2: Kalman Filter process

In the first step the state is predicted with the *dynamic model*. In the second step it is corrected with the *observation model* , so that the error covariance of the estimator is minimized. In this sense it is an optimal estimator. This procedure is repeated for each time step with the state of the previous time step as initial value. Therefore the Kalman filter is called a recursive filter.

The basic components of the Kalman filter are the state vector, the dynamic model and the observation model, which are described below.

State Vector

As described in 2.1 the state vector $\bar{x}$ contains the variables of interest. The state vector has two values at the same time, that is the *a* priori value, the predicted value before the update, and the *a* posteriori value, the corrected value after the update. In the following the *a* priori value is marked by $\bar{x}^-$ and the *a* posteriori value by $\bar{x}^+$.

Dynamic Model

The dynamic model describes the transformation of the state vector over time. It can usually be represented by a system of differential equations ,as in 2.1.10. For the sake of simplicity, the following dynamic model is considered.

$$\dot{\bar{x}} = \frac{d}{dt}\bar{x} = f(\bar{x}(t), \bar{w}(t)) \tag{2.2.1}$$

In the linear case this can easily be rewritten as:

$$\dot{\bar{x}} = F \cdot \bar{x}(t) + \bar{b}(t) \tag{2.2.2}$$

where F is the dynamic matrix and is constant, $\bar{x}(t)$ is the state vector and $\bar{b}(t)$ is the dynamic noise which is usually assumed as white noise and has the covariance matrix $Q(t)$.

Observation Model

The observation model represents the relationship between the state and the measurements. In the linear case the measurements can be described by a system of linear equations, which depend on the state variables. Usually the observations are made at discrete time steps t_i

$$\bar{l}(t_i) = H \cdot \bar{x}(t_i) + \bar{q}(t_i) \tag{2.2.3}$$

where $\bar{l}(t_i)$ is the vector of the observations at the epoch t_i, H is the observation matrix and $w(t_i)$ is the noise of the measurement process with the covariance matrix $R(t_i)$. Like the dynamic matrix, in a linear system the observation matrix H is a constant matrix as well.

Kalman Filter Algorithm

a. Prediction

Like mentioned before, the prediction is the first step of the Kalman filter. The predicted state, or better the a priori state is calculated by neglecting the dynamic noise and solving the differential equations that describe the dynamic model.

$$\dot{\bar{x}}^- = F \cdot \bar{x}^-(t) \tag{2.2.4}$$

The state vector at time t can be expressed by a Taylor series with respect to an approximate state $\bar{x}^-(t_0)$.

$$\bar{x}^-(t) = \bar{x}^- t_0 + \dot{\bar{x}}^-(t_0)(t - t_0) + \frac{1}{2}\ddot{\bar{x}}^-(t_0)(t - t_0)^2 + ... \tag{2.2.5}$$

By using equation 2.2.4 this can be written as.

$$\bar{x}^-(t) = \bar{x}^- t_0 + F \cdot \bar{x}^-(t_0)(t - t_0) + \frac{1}{2}F^2 \cdot \bar{x}^-(t_0)(t - t_0)^2 + ... \tag{2.2.6}$$

Thus the solution $\bar{x}^-(t)$ of the differential equations, in other words the actual predicted state is a linear combination of the initial state $\bar{x}^-(t_0)$.

$$\bar{x}^-(t) = \Phi_0^t \cdot \bar{x}^-(t_0) \tag{2.2.7}$$

Φ_0^t is called the *state transition matrix*, which transforms any initial state $\bar{x}(t_0)$ to its corresponding state $\bar{x}(t)$ at time t.

From the equations 2.2.4 and 2.2.7:

$$\dot{\bar{x}}^-(t) = F \cdot \bar{x}^-(t) = F \cdot \Phi_0^t \cdot \bar{x}^-(t_0) \tag{2.2.8}$$

and by using 2.2.7 again, it can be seen that:

$$\dot{\bar{x}}^-(t) = \frac{d}{dt}\bar{x}^-(t) = \frac{d}{dt}[\Phi_0^t \cdot \bar{x}^-(t_0)] = [\frac{d}{dt}\Phi_0^t] \cdot \bar{x}^-(t_0) \tag{2.2.9}$$

By comparing 2.2.8 and 2.2.9 it follows that:

$$\frac{d}{dt}\Phi_0^t = F \cdot \Phi_0^t \tag{2.2.10}$$

with the initial matrix $\Phi_0^0 = I$, because $\bar{x}(t_0) = I \cdot \bar{x}(t_0)$.

The covariance matrix $P^-(t_i)$ of the predicted state vector is obtained with the law of error propagation.

$$P^-(t_i) = \Phi_{t_{i-1}}^{t_i} \cdot P(t_{i-1}) \cdot (\Phi_{t_{i-1}}^{t_i})^T + Q \tag{2.2.11}$$

In the more generalized form, where also the covariance matrix of the noise Q is a function of time, the covariance matrix is.

$$P^-(t_i) = \Phi_{t_{i-1}}^{t_i} \cdot P(t_{i-1}) \cdot (\Phi_{t_{i-1}}^{t_i})^T + \int_{t_{i-1}}^{t_i} Q(t)dt \tag{2.2.12}$$

b. Correction

In the correction step the predicted state vector $\bar{x}^-(t_i)$ is improved with observations made at the epoch t_i ,thus the a posteriori state has the form

$$\bar{x}^+(t_i) = \bar{x}^-(t_i) + \Delta\bar{x}(t_i) \tag{2.2.13}$$

with the covariance matrix.

$$P^+(t_i) = P^-(t_i) + \Delta P(t_i) \tag{2.2.14}$$

The Kalman filter is an optimal filter, this means that the state variances in the state covariance matrix P^+ are minimized. As P^- is already known from the prediction step, it follows that ΔP is minimized.

$$\Delta P(t_i) = E[\Delta\bar{x}(t_i)\Delta\bar{x}(t_i)^T] \tag{2.2.15}$$

This combination is complied with

$$\Delta\bar{x}(t_i) = P^-H^T(HP^-H^T + R(t_i))^{-1} \cdot (\bar{l}(t_i) - H\bar{x}^-(t_i)) \tag{2.2.16}$$

$$\Rightarrow \Delta\bar{x}(t_i) = K(t_i) \cdot (\bar{l}(t_i) - \bar{l}^-(t_i)) \tag{2.2.17}$$

where

$$K(t) = P^-H^T(HP^-H^T + R(t_i))^{-1} \tag{2.2.18}$$

K is called the *gain matrix*. The difference $(\bar{l}(t_i) - \bar{l}^-(t_i))$ is called the *measurement residual*. It reflects the discrepancy between the predicted measurement $\bar{l}(t_i) = H\bar{x}(t_i)$ and the actual measurement $\bar{l}(t_i)$.

Finally, the corrected state is obtained by:

$$\bar{x}^{+}(t_i) = \bar{x}^{-}(t_i) + K(t_i) \cdot (\bar{l}(t_i) - \bar{l}^{-}(t_i)) \tag{2.2.19}$$

In this equation the estimated state and the measurements are weighted and combined to calculate the corrected state. That means, if the measurement covariance is much smalles than that of the predicted state, the measurement's weight will be high and the predicted state's will be low. And so the uncertainty can be reduced.

The covariance matrix of the a posteriori state is given with the law of error propagation by:

$$\begin{aligned} P^{+}(t_i) =& P^{-}(t_i) - K(t_i)HP^{-}(t_i) \\ =& (I - K(t_i)H)P^{-}(t_i) \end{aligned} \tag{2.2.20}$$

2.3 Extended Kalman Filter

All of the previous discussion to this point has considered linear filters for linear systems. In practice the dynamic or the observation model are nonlinear. Thus, linear systems do not exist. All systems are ultimately nonlinear. Even the simple relationship of Ohm's Law is only an approximation over a limited range. If the voltage across a resistor exceeds a certain threshold, then the linear approximation breaks down.

However, many systems are close enough to linear that linear estimation approaches give satisfactory results. But "close enough" can only be carried so far. Eventually, a system is run across that does not behave linearly even over a small range of operation, and linear approaches for estimation no longer give good results. In this case, nonlinear estimators should be used.

Nonlinear filtering can be a difficult and complex subject. It is certainly not as mature, cohesive, or well understood as linear filtering. There is still a lot of room for advances and improvement in nonlinear estimation techniques. However, some nonlinear estimation methods have become (or are becoming) widespread. These techniques include nonlinear extensions of the Kalman filter, unscented filtering, and particle filtering. In [25] and [26] some of non linear systems are presented and compared extensively. In this work, the Extended Kalman Filter algorythm is described, since it was used to design the corresponding propeller load observer in Chapter 3.

Thus continuing the reference to the [8], extended Kalman filter was discovered by Stanley F. Schmidt. After Kalman presented his results about Kalman filtering, Schmidt immediately began applying it to the space navigation problem for the upcoming Apollo project for manned exploration of the moon. In this process, he invented the extended Kalman filter.

This Kalman filter linearizes about the current estimated state. Thus the system must be represented by continuously differentiable functions. One disadvantage of this version of the Kalman filter for nonlinear systems is that it needs more time-consuming calculations. The implementation for linear systems can be made more efficient by pre-computing the dynamic matrix F, the state transition matrix Φ and the observation matrix H. But for nonlinear systems, these are functions of the state and consequently change with every time step and cannot be pre-computed.

a. Prediction

In the nonlinear case the dynamic matrix F is a function of the state to be estimated. So the predicted state is calculated by solving the differential equations in the following form.

$$\dot{\hat{x}}(t) = f(\bar{x}^{-}(t)) \tag{2.3.1}$$

By representing this equation by a Taylor series with respect to x at the predicted state $\bar{x}^{-}(t_i)$ and assuming that the higher order terms can be neglected, the dynamic matrix $F(t_i)$ can be calculated with:

$$F(t_i) = \left.\frac{\partial f(\bar{x})}{\partial \bar{x}}\right|_{\bar{x}=\bar{x}^-(t_i)} \tag{2.3.2}$$

And now the other steps of the prediction can be calculated as shown with calculations 2.2.10 and 2.2.12, but it should be noted that, now the used matrices are not constant like in the linear case, but depend on the time step.

$$\frac{d}{dt}\Phi_{t_{i-1}}^{t_i} = F(t_i) \cdot \Phi_{t_{i-1}}^{t_i} \tag{2.3.3}$$

$$P^-(t_i) = \Phi_{t_{i-1}}^{t_i} \cdot P(t_{i-1}) \cdot (\Phi_{t_{i-1}}^{t_i})^T + \int_{t_{i-1}}^{t_i} Q(t)dt \tag{2.3.4}$$

b. Correction

Like the differential equations in the prediction step the corresponding nonlinear observation equations are linearized with the Taylor series about the predicted state $\bar{x}^-(t_i)$ and higher order terms are neglected.

Thus, the approximate observation matrix is

$$H(t_i) = \left.\frac{\partial h(\bar{x})}{\partial \bar{x}}\right|_{\bar{x}=\bar{x}^-(t_i)} \tag{2.3.5}$$

In that case the predicted measurement $\bar{l}^-(t_i)$ for calculating the measurement residual $(\bar{l}(t_i) - \bar{l}^-(t_i))$ is

$$\bar{l}^-(t_i) = h(\bar{x}^-(t_i)) \tag{2.3.6}$$

Further on, the formulas can be used to calculate the corrected state and its covariance matrix like in the linear case but with time depend matrices.

$$\bar{x}^+(t_i) = \bar{x}^-(t_i) + K(\bar{l}(t_i) - \bar{l}^-(t_i)) \tag{2.3.7}$$

and

$$P^+(t_i) = (I - K(t_i)H(t_i))P^-(t_i) \tag{2.3.8}$$

with

$$K(t_i) = P^-H(t_i)^T(H(t_i)P^-H(t_i)^T + R(t_i))^{-1} \tag{2.3.9}$$

Chapter 3

Propeller Load Observer Design

The knowledge of the propeller load disturbance is necessary for the full operation of the model predictive control system. In actual marine conditions propeller load can not be measured or calculated by the corresponding equations presented in the previous sections due to the following reasons.

- The cost of the shaft dynamometer for loading measurements is extremely high.
- In order to calculate the propeller load by equation A.3.14a, velocity V_a should be measured. But GPS navigation sensor's latency cannot serve fast real-time control.
- Propulsion load fluctuations causes unpredictable measurements that are difficult to obtain.
- Environmental uncertainties that affect the propulsion plant are difficult to determine and are not taken into account in the ship modeling equations described in the previous chapter.

It was therefore considered necessary to design an observer in order to assess the state parameters required in the ship's propulsion control plant. In addition to the needs of ship's automatic control, the propeller observer ensures reduced power consumption, higher vessel efficiency and high quality performance monitoring for improving fault detection and thrust allocation in different propeller working conditions.

In this chapter two adaptive propeller estimation schemes are proposed ,based on [7]. The first one arises from the main ship propulsion dynamics equations and the second depends on torque coefficient estimation. Both of them are used to compose the corresponding Extended Kalman Filter observers in the second section ,for noisy measurements. A linear thrust estimation method is also presended ,using the torque coefficient. Finally, the propeller law and the method of calculating its power are mentioned.

3.1 Adaptive Models

In [7] two types of adaptive observers are proposed for propeller load torque estimation. The first arises directly from the shaftline dynamics presented in A.3.4 and is based on linear estimation. The second model is an alternative load torque estimation scheme based on nonlinear parameter estimation. In both methods it is assumed that the only available measurements are the shaft rotational speed and the engine torque.

3.1.1 Propeller Load Torque Observer

The GESO estimation formula, presented extensively in 2.1, was used to design the propeller load torque observer. The system described in A.3.18 can be written in statespace form ,similar to 2.1.11.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{3.1.1}$$

where

$x = \begin{bmatrix} \omega_{shaft} \\ Q_{load} \end{bmatrix}$ is the state vector, $u = \begin{bmatrix} n_m Q_{eng} \end{bmatrix}$ is the input vector

and y is the output.

The rest matrices are defined as follows.

$$A = \begin{bmatrix} 0 & -\frac{1}{J_{shaft}} \\ 0 & 0 \end{bmatrix} \quad , \quad B = \begin{bmatrix} \frac{1}{J_{shaft}} \\ 0 \end{bmatrix} \quad , \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}.$$

The disturbance Q_{load} is unknown and has to be estimated by the observer.

The system is observable from y, since the observability matrix:

$$Ob = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -\frac{1}{J_{shaft}} \end{bmatrix}$$ has full rank.

Using the control plant model described in A.3.18 the proposed propeller load torque observer is:

$$\begin{aligned} \dot{\hat{\omega}}_{shaft} &= \frac{1}{J_{shaft}}(u - \hat{Q}_{load}) + g_a(y - \hat{y}) \\ \dot{\hat{Q}}_{load} &= g_b(y - \hat{y}) \end{aligned} \tag{3.1.2}$$

where $\hat{\omega}_{shaft}$, $\hat{Q}_{load}$ are the observer estimates of the corresponding state variables ω_{shaft} , Q_{load} and g_a , g_b are the observer gains.

On matrix form $G = \begin{bmatrix} g_a & g_b \end{bmatrix}^T$, the observation plant system in 3.1.1 can be written as follows.

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + G\tilde{y} \\ \hat{y} &= C\hat{x} \end{aligned} \tag{3.1.3}$$

The observer error can be also calculated by subtracting 3.1.3 from 3.1.1. Thus

$$\dot{\tilde{x}} = Ax + Bu - (A\hat{x} + Bu + G\tilde{y}) = A\tilde{x} - G\tilde{y} = A\tilde{x} - GC\tilde{x} = F\tilde{x} \tag{3.1.4}$$

where $F = A - GC = \begin{bmatrix} g_a & \frac{1}{J_{shaft}} \\ g_b & 0 \end{bmatrix}$.

Observer Tuning

The main objective in load torque observer tuning is to minimize the influence of the disturbance and uncertainty on the output, provided that the actual disturbance is being estimated. For the appropriate observer gains choice, the pole placement method was used. The values of the observer poles p_i depend on the error dynamics $\hat{x}$ - x of the system and are foud from:

$$p_i = \frac{-g_a \pm \sqrt{g_a^2 + \frac{4g_b}{J_{shaft}}}}{2} \tag{3.1.5}$$

The main parameters of the observer performance are the undamped natural frequence ω_n and damping ratio ζ as they directly affect the selected gains as follows.

$$\omega_n = \sqrt{\frac{-g_b}{J_{shaft}}} \quad , \quad \zeta = \frac{g_a J_{shaft}}{2\sqrt{-g_b J_{shaft}}} \tag{3.1.6}$$

According to [17] the desired values of the poles should be chosen so that the state observer responds at least two to five times faster than the closed-loop system considered ,so that the observation error converge quickly to zero. The damping ratio could be chosen in the range $0.7 < \zeta < 1.1$. The natural frequency depends on the specific thruster of the propeller. Thus $\zeta = 0.7$, $\omega_n = 125 rad/s$ were chosen and the observer gains became:

$$\begin{aligned} g_a &= 2\zeta\omega_n = 175 \\ g_b &= -J_{shaft}\omega_n^2 = -25812.5 \end{aligned} \tag{3.1.7}$$

The evaluation of the performance of the observer was carried out through simulations, the conclusions of which are recorded in Chapter 5. A suitable model was designed for the simulations, based on the characteristics of the observer mentioned above and in the following figure its layout is presented.

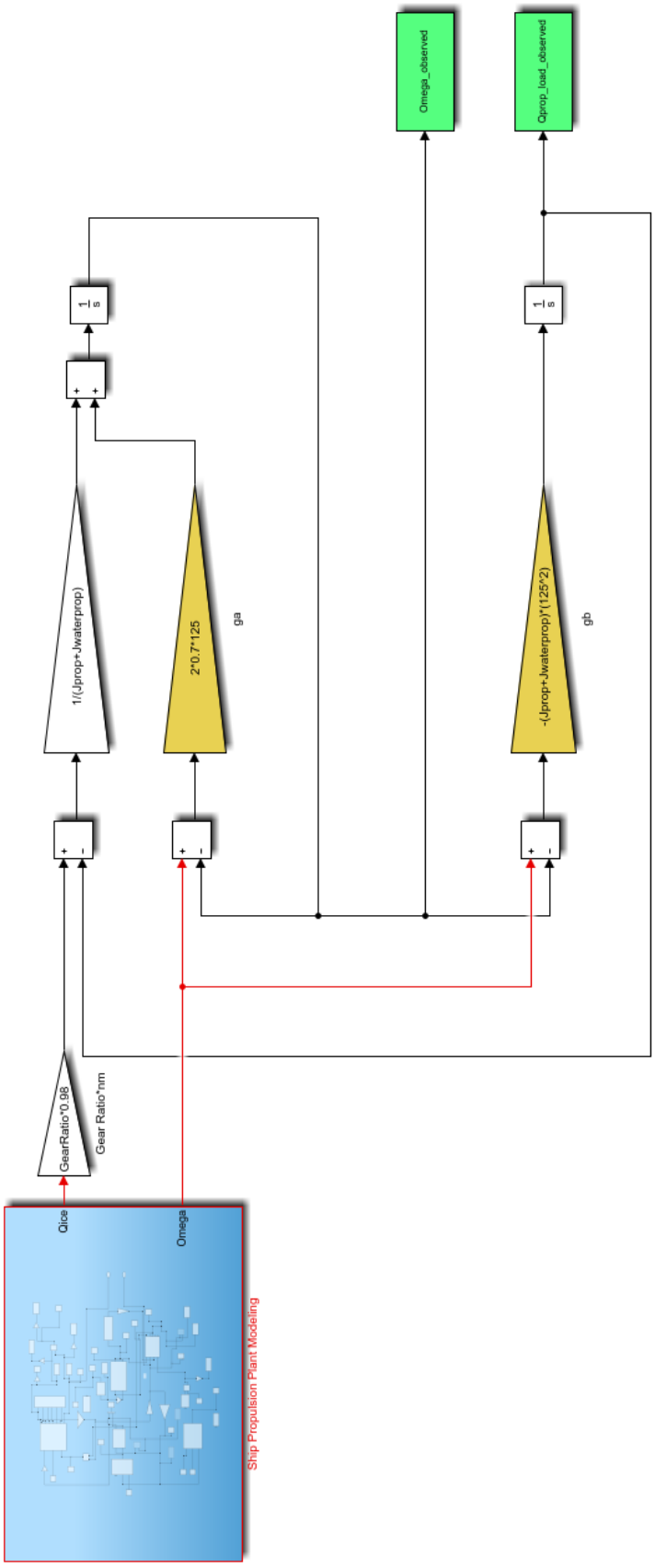

Figure 3.1: Simulink set up of propeller load torque observer

3.1.2 Torque Coefficient Estimation

An alternative load torque estimation scheme based on nonlinear parameter estimation is also presented in [7].

As shown in A.3.14a, the propeller load torque can be calculated from the following formula.

$$Q_{load} = \frac{\pi}{8} C_Q \rho [V_a^2 + (0.7\pi n_{shaft} D)^2] D^3 \tag{3.1.8}$$

In the above equation, however, it becomes necessary to know both the C_Q coefficient and the propeller inflow velocity V_a (which is very difficult to measure in real vessels). Equation 3.1.8 can be rewritten in the form of equation A.3.13a as follows.

$$Q_{load} = K_Q \rho D^5 n_{shaft} |n_{shaft}| = \frac{K_Q \rho D^5}{4\pi^2} \omega_{shaft} |\omega_{shaft}| \tag{3.1.9}$$

where K_Q coefficient is defind as in A.3.15a.

$$K_Q = \frac{\pi}{8} C_Q \left[\left(\frac{V_a}{n_{shaft} D} \right)^2 + (0.7\pi)^2 \right] D^3 \quad and \quad \omega_{shaft} = 2\pi n_{shaft} \tag{3.1.10}$$

In this case, by determining the K_Q coefficient, the propeller load torque can be calculated directly, without the knowledge of any further parameter. Thus, equation A.3.13a is used instead of A.3.14a.

As an alternative to the load torque observer presented in the previous subsection, it is possible to use a nonlinear online parameter estimation scheme to find an estimate $\hat{K}_Q$ of the propeller torque coefficient.

As in propeller load observer, the parameter estimation scheme is based on the shaft dynamics A.3.18, with the addition of the torque equation 3.1.10. The control plant model is rewritten as follows.

$$\dot{\omega}_{shaft} = \frac{1}{J_{shaft}} (n_m Q_{eng} - \frac{K_Q \rho D^5}{4\pi^2} \omega_{shaft} |\omega_{shaft}|) \tag{3.1.11}$$

Defining the input u as in 3.1.1 and the unknown parameter θ as follows,

$$\theta = \frac{K_Q \rho D^5}{4\pi^2} \tag{3.1.12}$$

the system in 3.1.11 ,which is affine in θ, becomes:

$$\dot{\omega}_{shaft} \quad = \quad f(\omega_{shaft}, u, \theta) \quad = \quad F(\omega_{shaft})\theta + g(\omega_{shaft}, u) \tag{3.1.13}$$

where

$$f(\omega_{shaft}, u, \theta) = \frac{1}{J_{shaft}}(u - \theta\omega_{shaft}|\omega_{shaft}|)$$

$$F(\omega_{shaft}) = \frac{\partial f(\omega_{shaft}, u, \theta)}{\partial \theta} = -\frac{\omega_{shaft}|\omega_{shaft}|}{J_{shaft}}$$

$$g(\omega_{shaft}, u) = \frac{1}{J_{shaft}}u$$

The proposed nonlinear parameter estimation scheme is then described by the following equations

$$\begin{aligned} \hat{K}_Q &= \frac{4\pi^2}{\rho D^5}\hat{\theta} \\ \hat{\theta} &= -g_0\frac{|\omega_{shaft}|^3}{3J_{shaft}} + z \\ \dot{z} &= g_0\frac{\omega_{shaft}|\omega_{shaft}|}{J^2_{shaft}}(u - \hat{\theta}\omega_{shaft}|\omega_{shaft}|) \end{aligned} \tag{3.1.14}$$

where g_0 is the estimation scheme gain.

Observer Tuning

Appropriate choice of the gain g_0 in the K_Q estimation scheme 3.1.14 is highly dependent on the application. If the gain is chosen high, the scheme will estimate the instantaneous loading of the propeller, such that Q_{load} from 3.1.9 will be comparable to 3.1.2 from the previous section. If g_0 is chosen low, the high-frequency disturbances in the load torque will not be captured. Instead, an average torque coefficient will be estimated. Taking into account the operating range (and the fluctuations) of the selected tugboat and performing a number of simulations, it was observed that the ideal scheme gain value is the following one.

$$g_0 = 0.005 \tag{3.1.15}$$

The evaluation of the performance of the estimation scheme was carried out through simulations, the conclusions of which are recorded in Chapter 5. A suitable model was designed for the simulations, based on the characteristics of the K_Q estimation method mentioned above and in the following figure its layout is presented.

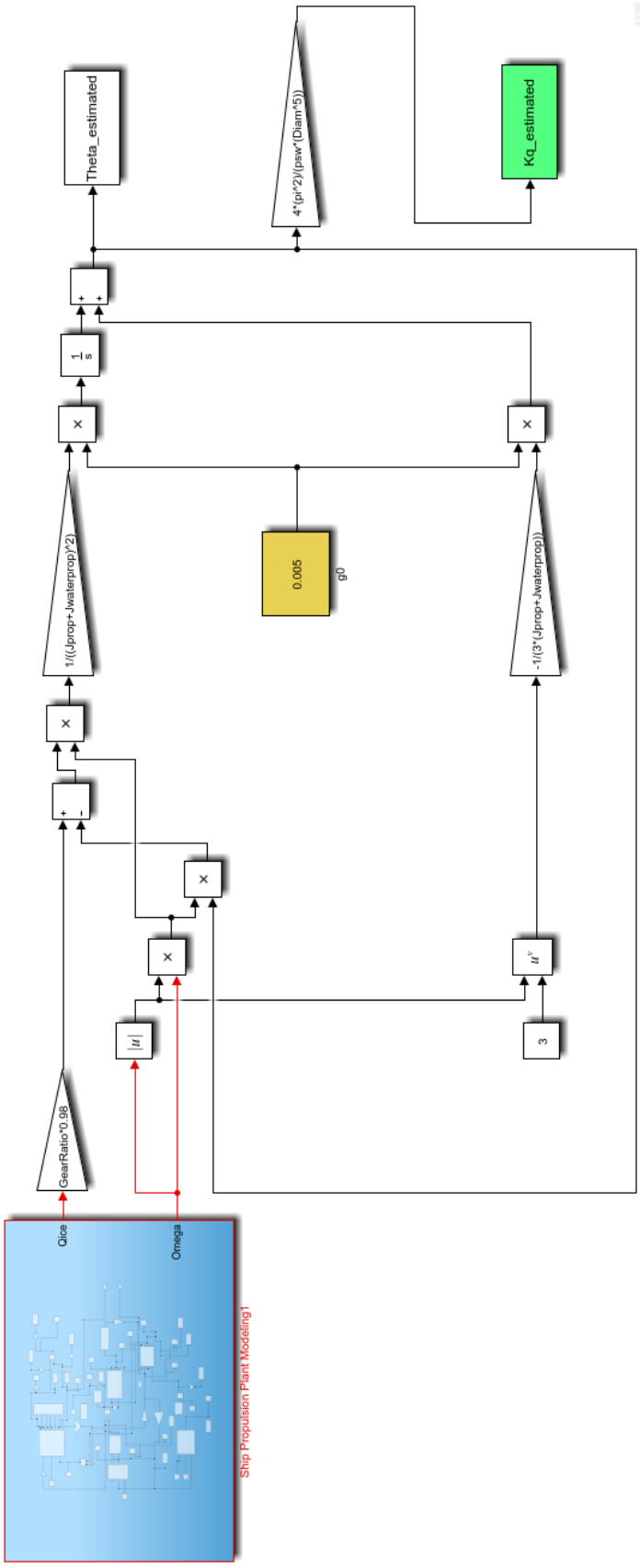

Figure 3.2: Simulink set up of propeller torque coefficient estimation scheme

3.1.3 Propeller Thrust Estimation and Application Restrictions

In marine guidance, navigation, and control (GNC) systems, the low level thruster controllers have traditionally received less attention compared to the guidance system and the high-level plant control. More recently, also the issue of thruster dynamics and control has received more attention because the knowledge of the propeller thrust, together with the thrust induced pressure force on the hull, is fundamental to achieve high vessel control performance. Many resent researches ,as [34], have focused on individual subjects of thrust control. The knowledge of the propeller thrust, either measured or estimated, could also allow the design of controllers for reducing power fluctuations and wear and tear in high sea state. Moreover, the performance monitoring is also useful for improving fault detection and thrust allocation in different propeller working conditions. These considerations motivate the development of schemes to estimate the propeller thrust because, in general, its measurement is not available.

The thrust estimation scheme was designed according to [7] and [6]. It was based on the observer torque estimates ,as found before and on a linear relation between thrust and torque coefficients. Due to the working principles of the propeller, the thrust and torque are closely coupled. According to *Lerbs (1952)* research on the effect of roughness on open propellers a change ΔK_Q of K_Q implies a proportional change ΔK_T of K_T, i. e.:

$$\Delta K_T = c_t \Delta K_Q \tag{3.1.16}$$

where c_t is a constant.

This implies that K_T and K_Q (and their estimates $\hat{K}_T$, $\hat{K}_Q$) can be linearly related by:

$$K_T = a_t K_Q + b_t \quad \Rightarrow \quad \hat{K}_T = a_t \hat{K}_Q + b_t \tag{3.1.17}$$

where a_t , b_t are propeller constants inferred from the open-water characteristics and were chosen as:

$$\begin{aligned} a_t &= 7.52 \\ b_t &= -0.04 \end{aligned} \tag{3.1.18}$$

Propeller thrust estimation $\hat{T}_{load}$ can then be found via A.3.13b.

$$\hat{T}_{load} = \hat{K}_T \rho D^4 \hat{n}_{shaft} |\hat{n}_{shaft}| = \frac{\hat{K}_T \rho D^4}{4\pi^2} \hat{\omega}_{shaft} |\hat{\omega}_{shaft}| \tag{3.1.19}$$

Although the linear relationship proposed above for calculating propeller thrust through torque is stable for a large range of propeller operating conditions, including varying advance ratios, there are cases in which the thrust estimation scheme model does not give correct estimates. Indicative cases in which there may be problems with unreliable estimates are the following.

- In significant changes of the vessel's operating draught, the estimated thrust shows an offset from the actual one. Although the relationship between thrust and torque coefficients continues to be almost linear, it becomes necessary to perform new simulations and redefine parameters a_t and b_t. Therefore, for draught changes, the thrust estimation model parameters cannot be maintained and must be changed accordingly.

- A similar problem with the previous one is also presented in cases of change in the propeller efficiency due to wear and tear or fouling. When a ship performs a marine operation, propellers are often affected by thrust losses due to cross flow, ventilation, in-and-out-of water effects, wave-induced water velocities, interaction between the vessel hull and the propeller and between propellers. Propellers may thus work far from ideal conditions therefore and thrust estimates via the proposed method may not be accurate.

- Thrust estimation scheme described above gives good estimations for a large range of advance ratio. However, when the shaft speed is reversed, the propeller works for a short time outside that range of advance ratio and thrust estimates deviate from the actual values. Thus, thrust-torque coefficients relationship is more accurate for positive shaft rotational speed.

- $K_T - K_Q$ relationship is not hold equally well for ducted propellers, since the shapes of the K_T and K_Q curves are inherently different from corresponding curves of FPP or CPP. As noted in [7], for values of coefficient K_Q greater than 0.04 the relationship between K_T and K_Q is not linear and equation 3.1.17 cannot be applied.

Attached to the following summary table are the basic design parameters of the propeller observer models described in this section.

Scheme - Model	*Propeller Torque Estimation Gains*
Propeller Load Torque Observer Torque Coefficient Estimation	$g_a = 175 \quad g_b = -25812.5$ $g_0 = 0.05$
Scheme - Model	*Propeller Thrust Estimation Parameters*
Thrust - Torque Coefficients Relationship	$a_t = 7.52 \quad b_t = -0.04$

Table 3.1: Propeller Observers' Design Parameters

3.2 Kalman Filter Models

The design of the adaptive observer models presented above was based on the assumption that the only available measurements are the engine speed and torque, with almost zero noise levels (zero noise disturbances were considered). In actual operating conditions these sensors' measurements are contaminated with noise and cannot be used directly by the controller. The signals should therefore be filtered and reconstructed by the observer. For a more realistic observing approximation Kalman Filter was implemented for both state estimation methods described in the previous section. Many Kalman Filter implementations have proposed for ship sensors and navigation systems, as in [5]. Since the torque coefficient estimation system is nonlinear, Extended Kalman Filter was used for both the propeller load torque observer and the torque coefficient estimation scheme.

The description of the basic design parameters of the two observer models, with the Extended Kalman Filter is based on section 2.3 and follows below.

The Extended Kalman Filter estimates the states of a discrete-time nonlinear system using the first-order discrete-time extended Kalman filter algorithm. Considering a plant with states x, input u, output y, similar to 3.1.1 but also with process noise w, and measurement noise v (both with Gaussian distributions), it is assumed that the plant can be represented as a nonlinear system 3.3.

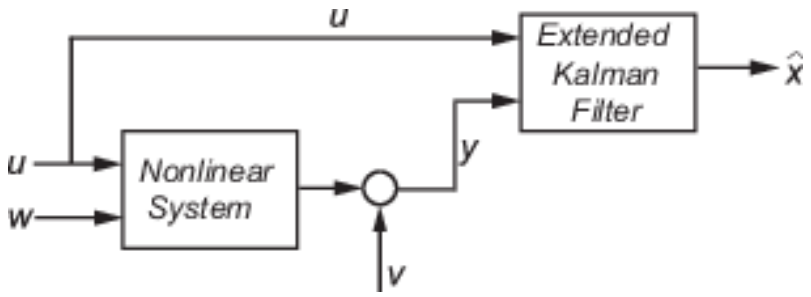

Figure 3.3: Extended Kalman Filter Plant

The filter's algorithm is a two-step process: the first step predicts the state of the system, and the second step uses noisy measurements to refine the optimal estimate of system state, as shown in 3.4. The Extended Kalman Filter linearizes the nonlinear functions around the mean of the current state estimates. At each time step, the linearization is performed locally and the resulting Jacobian matrices are then used in the prediction and update states of the filter algorithm.

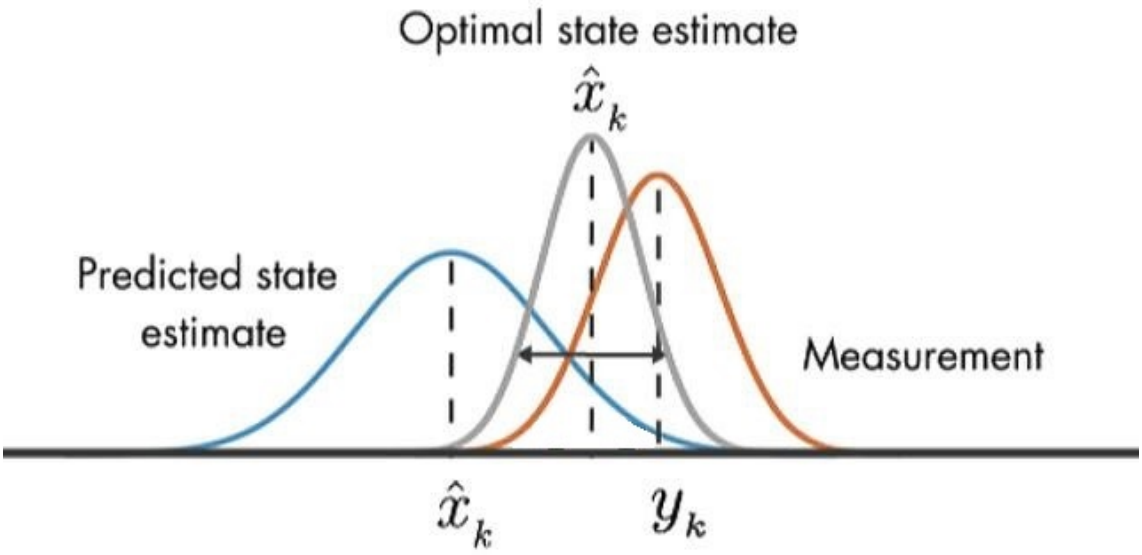

Figure 3.4: State Estimation Probability Density Function

The algorithm computes the state estimates $\hat{x}$ of the system using state transition f and measurement h functions. The systems equations of the observers tested in the previous section were used to compose these functions and the observation system has the following form.

$$\begin{aligned} x[k] &= f(x[k-1], u_s[k-1]) + w[k-1] \\ y[k] &= h(x[k], u_m[k]) + v[k] \end{aligned} \tag{3.2.1}$$

Here f is a nonlinear state transition function that describes the evolution of states x from one time step to the next. The nonlinear measurement function h relates x to the measurements y at time step k. w and v are the zero-mean, uncorrelated process and measurement noises, respectively. These functions can also have additional input arguments that are denoted by u_s and u_m in the equations.

The noise terms in both equations are additive. That is, $x[k]$ is linearly related to the process noise $w[k-1]$, and $y[k]$ is linearly related to the measurement noise $v[k]$. The process and measurement noise terms are assumed to be drawn from Gaussian and white distribution, with zero mean and covariance matrices Q and R, respectively:

$$\begin{aligned} w[k] &\sim (0, Q[k]) \\ v[k] &\sim (0, R[k]) \end{aligned} \tag{3.2.2}$$

In order to design the Kalman observers, the corresponding state transition and measurement functions had to be expressed in the appropriate form. Given the state transition and measurement functions, the performance of the Kalman filter is determined by the choice of the corresponding noise covariances Q and R. By increasing the values of these coefficients, the observed values converge to their actual values, but are infected with more noise. By lowering Q and R, the observed values become smoother, but deviate from their actual values. A large number of simulations were performed to select the appropriate noise covariances, both in the case of the propeller load torque observer and in the case of the torque coefficient estimation. Thus, covariances that ensure satisfactory convergence were selected, along with good noise filtration.

3.2.1 Propeller Load Torque Observer

Knowing that the available measurement of shaft rotational speed $\omega_{shaft}[k]$ is contaminated with noise disturbances, the main design characteristics of this Kalman Filter observation model were based on the scheme equations presented in 3.1.1 and were identified as follows.

State Transition Function

$$x[k] = x[k-1] + \begin{bmatrix} \frac{1}{J_{shaft}}(u[k-1] - Q_{load}[k-1]) \\ 0 \end{bmatrix} dt \tag{3.2.3}$$

where $dt \rightarrow 0.01$

Measurement Function

$$y[k] = \omega_{shaft}[k] \tag{3.2.4}$$

Noise Covariances

The state transition and measurement noise covariances were chosen as follows.

$$\begin{aligned} Q[k] &= 5000 \\ R[k] &= 200 \end{aligned} \tag{3.2.5}$$

3.2.2 Torque Coefficient Estimation

The main design characteristics of this Kalman Filter observation model were based on the scheme equations presented in 3.1.2 were identified as follows.

State Transition Function

$$x[k] = x[k-1] + \begin{bmatrix} \frac{1}{J_{shaft}}(u[k-1] - \theta[k-1]\omega_{shaft}[k-1]|\omega_{shaft}[k-1]|) \\ 0 \end{bmatrix} dt \tag{3.2.6}$$

where $dt \rightarrow 0.01$

Measurement Function

$$y[k] = \omega_{shaft}[k] \tag{3.2.7}$$

Noise Covariances

The state transition and measurement noise covariances were chosen as follows.

$$\begin{aligned} Q[k] &= 0.0007 \\ R[k] &= 5 \end{aligned} \tag{3.2.8}$$

The evaluation of the performance of the Extended Kalman Filter models was carried out through simulations, the conclusions of which are recorded in Chapter 5. The simulation models were designed using the EKF Matlab Simulink block, according to observation characteristics mentioned above.

Attached to the following summary table are the basic design parameters of the propeller observer models described in this section.

EKF Scheme - Model	State Transition Noise Q[k]	Measurement Noise R[k]
Propeller Load Torque Observer	5000	200
Torque Coefficient Estimation	0.0007	5

Table 3.2: Extended Kalman Filter Propeller Observers' Design Parameters

3.3 Propeller Law Constant and Power Estimation

Propeller Law connects shaft rotational speed with propeller power by the following equation.

$$P_{load} = c \cdot n_{shaft}^3 = c \cdot \frac{\omega_{shaft}^3}{8\pi^3} \tag{3.3.1}$$

where c is the propeller law constant. The value of constant c proves to be particularly important for performance monitoring, as it is directly affected by the level of hull fouling and the weather operating conditions.

It is also known that the propeller torque is connected to its power through the following equation.

$$P_{load} = 2\pi \cdot n_{shaft} \cdot Q_{load} = \omega_{shaft} \cdot Q_{load} \tag{3.3.2}$$

Combining equations 3.3.1 and 3.3.2 results:

$$Q_{load} = c \cdot \frac{n_{shaft}^2}{2\pi} = c \cdot \frac{\omega_{shaft}^2}{8\pi^3} \tag{3.3.3}$$

Having estimated the rotational speed and propeller torque through the observers analyzed in the previous sections, it is now possible to determine both the propeller law constant c and the power. The evaluation of these estimates is carried out in Chapter 5 for each model through the corresponding simulations.

Chapter 4

Shaft Rotational Speed Prediction

The observation schemes presented in the previous chapters are based on the assumption that the only available measurements are the shaft speed and the engine torque. To maximize the efficiency of the propulsion control system and ensure better fuel economy, it is advisable to predict shaft rotational speed within a sufficient future horizon. Many methods have been proposed for various system states prediction. In this chapter, a Neural Network (NN) model is described as a well performed deterministic prediction approach to our problem. A Markov Chain prediction model is also being presented, although its performance is not satisfactory in this case.

Dataset

As mentioned above, the focus is on engine rotational speed predictions over 1 to 10 steps prediction horizon. Engine speed was chosen, as it is the reference input to the controller of the propulsion plant. Moreover, engine rotational speed expresses either the ship cruising velocity or the propeller load during maneuvering. In order to train the predition models, it became necessary to provide sufficient relevant data. Specifically, speed data of a tugboat were taken during four consecutive days of operation, in the area of Panama Canal. Using the propulsion plant model, these data where converted to engine rotational speed.

4.1 Markov Chain Prediction Model

Markov Chain has been used as a prediction model in many automatic control applications. In [10] a Markov chain that models the power request dynamics is proposed ,to improve the prediction capabilities of model predictive control, while in [11], [12] Markov chain performance is compared to other stohastic models.

The following is a brief overview of the Markov Chain prediction method.

A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. A simple Markov Chain model is shown in the following figure.

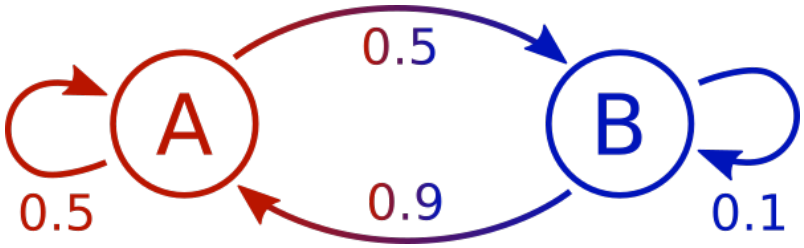

Figure 4.1: Two-State Markov Chain process diagram

In the example illustrated above, there are two states A and B. If the system is in state A at the time instant t, then at the time instant $t+1$, the system could be in state A (with probability 0.5) or in state B (with probability 0.5). Similary, if the system is in state B at the time instant t, then at the time instant $t+1$, the system could be in state B (with probability 0.1) or in state A (with probability 0.9). The Markov Chain is defined by the state transition probability matrix as follows.

	A	B
$P = A$	0.5	0.5
$P = B$	0.9	0.1

Table 4.1: Markov Chain State Transition Probability Matrix

The Markov Chain prediction implementantion was based on state transition probability matrix, where the states capture the possible shaft rotational speeds during ship operation. The state transition probability matrix can be calculated by frequency count of shaft speed transitioning between discrete intervals over a suitable sample time window. Having the probability table, it is possible to predict future speeds over to tested horizon (10steps). Thus, for a previous state, the next predicted state is the one that equalls to the corresponding maximum probability state of the probability matrix. In order to implement Markov Chain prediction model and perform simulations, the following brief algorithm was used.

Algorithm 1: Markov Chain Prediction Model Algorithm

Train Data: selected data in matrix *all_ships*;
for *all elements in Train Data* **do**
    Find unique elements;
    Sort unique elements;
    Put elements in a 2D matrix, in the form of 4.1;
end
Compose State Transition Probability Matrix p, based on the frequency of unique elements;
Test Data: tested data in matrix $MT.(ship)$;
for *all elements in Test Data* **do**
    Find the equal (or closest) element in State Transition Probability Matrix p;
    For this element, find the maximum probability and the corresponding predicted element;
end
Repeat the above prediction method for the next 9 future horizon steps giving the previous state values as input to for-loop;
Plot the predicted values along with the actual values and compare them;

Although the evaluation of the performance of this prediction model is carried out in Chapter 5, a significant drawback of the use of the Markov chain should be noted at this point. As mentioned in the definition of the Markov model, the prediction of the method is based on a single input value of the previous state and not on a sufficient number of multiple previous state values, that "show" the course of the testing data. Given that the next prediction is based solely on the state with the highest probability in the state transition probability matrix, it is possible that in many cases the model deviates from the actual values. This prediction model proves to be particularly effective in cases where the training data is quite similar to the testing data and in cases where the object of the study is to find the state transition probability matrix and not the future forecasts. Otherwise, the model should be updated at each time step depending on the actual state measurements, which, however, are often not available.

4.2 Neural Network Prediction Model

Neural networks have emerged in the past few years as an area of unusual opportunity for research, development and application to a variety of real world problems. Many books as [13], [14], [35], [36] and also scientific papers as [37] have focused on extensive descriptions of neural networks or their applications in new technological problems. In addition to the above literature, information for neural networks is also obtained from [38].

The following is a brief presentation of the Neural Network prediction method.

Neural networks are computing systems inspired by the biological neural networks that constitute animal and human brains. These systems learn to progressively improve their performance for a specified task by considering examples, generally and without task specific programming. This is done by analyzing example data manually labeled without having any priori knowledge about the data or the application. This closely resembles the way humans and animals learn to execute tasks. Although neural networks were firstly inspired by the goal to model a biological neuron, they have since diverged and become a matter of engineering and machine learning. Neural network learning methods provide a robust approach to approximating real-valued, discrete-valued, and vector-valued target functions. For certain types of problems, such as learning to interpret complex real-world sensor data, artificial neural networks are among the most effective learning methods currently known. A neural network derives its computing power through, first, its masively parallel distributed structure and second, its ability to learn and therefore generalize. Generalization refers to the neural network producing reasonable outputs for inputs not encountered during training (learning). These two information-processing capabilities make it possible for neural networks to solve complex (large-scale) problems that are currently intractable.

A neural network consists of many processing elements joined together to form an appropriate network with adjustable weighting functions for each input. These processing elements are called neurons and are usually organized into a sequence of layers with full or random connections between layers. Typically, there are three or more layers: an input layer where data are presented to the network through an input buffer, an output layer with a buffer that holds the output response to a given input, and one or more intermediate or "hidden" layers. According to [35], the simplest Neural Network model (Perceptron with a single neuron) consist of 3 basic elements:

1. A set of synapses or connecting links, each of which is characterized by a weight or strength of its own. Specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}. Thus, neurons are connected with each other with synapses and each neuron receives signals from its dendrites and produces output signals along its one axon. The axon eventually connects to the dendrites of other neurons and so on. In the computational model of a neuron the signals that travel across the axons interact multiplicatively.

2. An order for summing the input signals, weighted by the respective synapses of the neuron.

3. An activation function for limiting the amplitude of the output of a neuron.

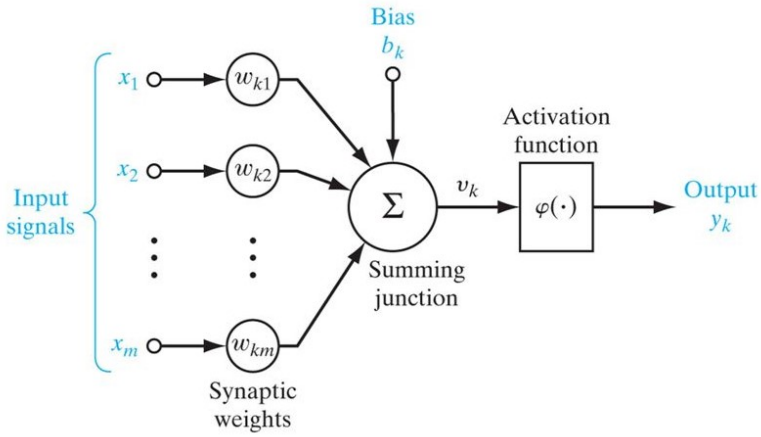

Figure 4.2: Nonlinear Model of a Neuron

In mathematical terms, a neuron k of 4.2 model may be described as follows:

$$u_k = \sum_{j=1}^{n} w_{kj} x_j \tag{4.2.1}$$

and

$$y_k = \phi(u_k + b_k) \tag{4.2.2}$$

where x_1, x_2,...,x_m are the input signals, w_{k1}, w_{k2},...,w_{km} are the synaptic weights of neuron k, u_k is the linear combiner output due to the input signals, b_k is the bias which has the effect of increasing or lowering the net input of the activation function (depending on whether it is positive or negative, respectively), $\phi()$ is the activation function and y_k is the output signal of the neuron. The use of bias has the effect of applying an affine transformation to the output u_k of the linear combiner in the model, as shown by:

$$v_k = u_k + b_k \tag{4.2.3}$$

In particular, depending on whether the bias b_k is positive or negative, the relationship between the induced local field or activation potential v_k, of neuron k and the linear combiner output u_k is modified accordingly.

The bias b_k is an external parameter of artificial neuron k and equations 4.2.1, 4.2.2 could be formulated through 4.2.3 ,as follows.

$$v_k = \sum_{j=1}^{n} w_{kj} x_j \tag{4.2.4}$$

and

$$y_k \approx \phi(v_k) \tag{4.2.5}$$

The activation function, denoted by $\phi(v)$, defines the output of a neuron in terms of the inducted local field v. Some common activation functions are described below.

- **Sigmoid**: The mathematical form the sigmoid non-linearity is $\sigma(x) = 1/(1 + e^{-z})$. It takes a real valued number and squashes it in the range between 0 and 1. In particular, negative numbers become 0 and positive numbers become 1. Although the sigmoid function was frequently used in the past, it is very rarely used nowadays mostly because when it saturates at either 0 or 1 the gradient becomes almost 0 and because it is not zero centered.
- **Tanh**: The tanh non-linearity squashes a real-valued number to the range $[-1, 1]$. Like the sigmoid neuron it's activations saturate but it's output is zero centered. Therefore, it's always preferred to the sigmoid non-linearity.
- **ReLU**: The Rectified non-Linear Unit has become very popular in the last few years. It computes the function $f(x) = max(0, x)$. Thus, the activation is simply thresholded at 0. It was found to greatly accelerate the convergence of stochastic gradient descend due to its non-linear non-saturating form.

The above architecture of a Neural Network consisting of a single neuron which can be used as a linear classifier obviously limits the networks capabilities and has little more to offer in comparison to classic machine learning algorithms. The networks used today (deep neural networks) are networks that consist of numerous layers of neurons that are connected in an acyclic graph. The outputs of the neurons of a layers become inputs for the neurons of the next layer and so on. Typically, a deep neural network will consist of an input layer of neurons and an output layer as well as any number of hidden layers in between that connect the above to layers ,as shown in Fig. 4.3. In general, three fundamentally different classes of network architectures may be identified: *Single − Layer Feedforward Networks*, *Multi − Layer Feedforward Networks* and *Recurrent Networks*.

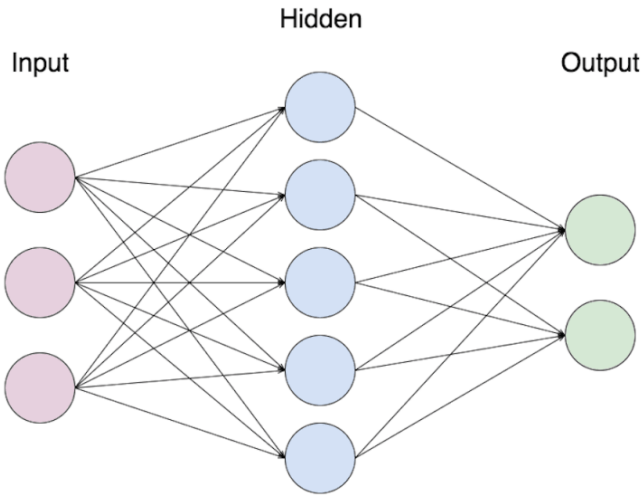

Figure 4.3: Neural Network with a single Hidden Layer

Neural Network Type and Parameters

It is noted that various types of neural networks were examined in order to select and implement the most appropriate one. This choice was based on the following criteria.

- Performance and Accuracy
- Regression
- Training Time

Since 10 previous steps are required to ensure predictions of 10 steps forward, the first type of neural network to be examined was the Time Delay Neural Network (TDNN). It has been observed that this neural network does not provide a good estimate at all, with estimates deviating considerably from the actual ones. The Nonlinear Auto Regresive Neural Network (NARX) was also tested, but it was found to has low performance and regression, with very long training time. In the end, the Feed Forward Neural Network (FFNN) was chosen, which in a short training time ensures great accuracy in predictions and good regression.

The neural network creation and training was developed in Matlab, using the following commands in order.

$$\begin{aligned} net &= feedforwardnet(hiddenlayers) \\ net &= train(net, x, t) \end{aligned} \tag{4.2.6}$$

where:

- Hidden layers as well as the number of neurons in each layer represent the structure of the NN. The number of hidden layers and number of neurons determine the performance of the neural network. In general, as the number of hidden layers and neurons increase, so does the quality of predictions. However, a large increase in the number of these does not lead to the expected improvement of the forecast, overloading the network and at the same time significantly increasing its training time or leading to overfitted results. Taking into account the variation of the collected dataset and performing a number of simulations, it was observed that the ideal NN structure is with 1 hidden layer of 10 neurons, as shown in 4.4. With this structure, satisfactory performance of the neural network is ensured, while minimizing its training time.

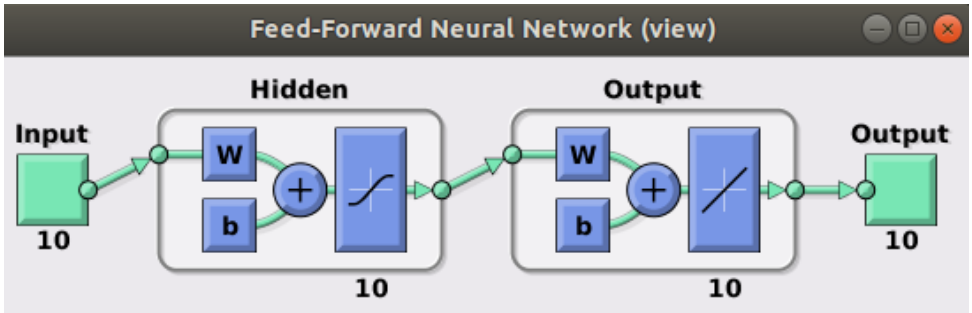

Figure 4.4: Neural Network structure

- x variable represents the given inputs. The previous 10 steps of each dataset element are given as input data to the neural network.
- t variable represents the given targets. The next 10 steps of each dataset element are given as output data to the neural network. For the tugboat collected dataset, it was observed that the neural network gives very good predictions up to 10 steps forward.

The following function was used to connect the neural network to the power plant control model presented in this work.

$$gensim(net) \tag{4.2.7}$$

This function creates a Simulink system containing a block that simulates neural network net. The modeling sample time is 400 ms. This time value is obtained from the number of data taken into account for the simulation and is characterized as particularly satisfactory for the speed and reliability of the system predictions.

The evaluation of the performance of the Feed Forward Neural Network prediction model was carried out through simulations, the conclusions of which are recorded in Chapter 5.

Chapter 5

Observer and Prediction Models Implementation

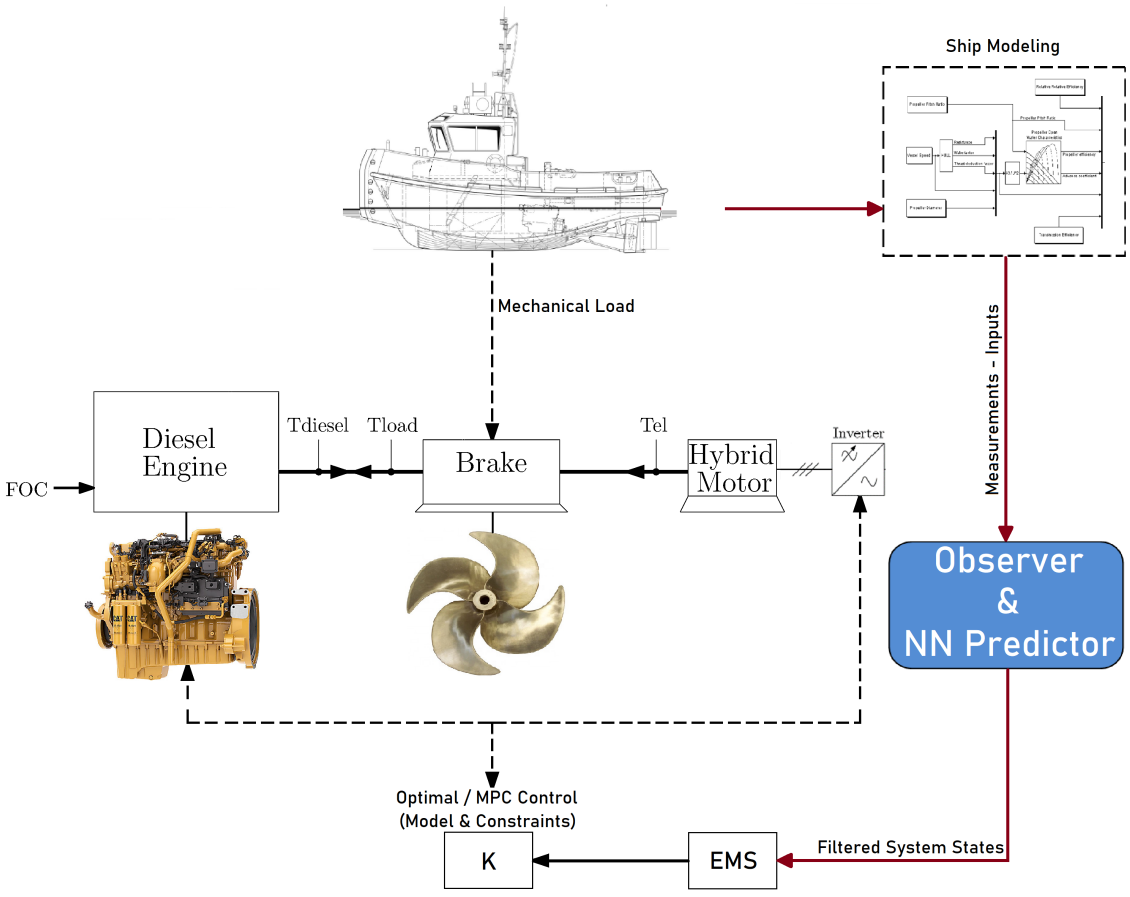

Figure 5.1: Observer and Predictor Implementation

This chapter lists the results-diagrams of the simulations performed, to evaluate the performance of the models presented in the previous chapters. The main conclusions from the execution of the simulations are recorded and the models are compared. This chapter consists of two parts. The first part refers to the implemented models of propeller observer and the second to the models for predicting the rotational speed of the shaft.

The MATLAB/Simulink programming environment was used to perform the simulations and compile the necessary codes. There are a number of bibliographic sources available for quick and effective learning of the program, such as: [39] and [40].

5.1 Propeller Load Observation Models

In this section, the simulations of the two observer models are presented, both in adaptive form (3.1.1 , 3.1.2) and in Kalman Filter format (3.2.1 , 3.2.2). For each of the four models in total, the corresponding propeller thrust estimation is attached, based on 3.1.3. The estimate of the propeller law constant c based on 3.3 is also recorded.

Fast and slow acceleration simulations were performed (settings: Figs. 5.2 and 5.16), which are presented in common diagrams, but also simulation of towing (settings: Figs. 5.12 and 5.26). In every case, it was considered a wave of 1.2 meters significant high and 0.8 rad/s peak frequency, as presented in A.4. Simulations were performed for both Duct Propeller (DuctP) and Controllable Pitch Propeller (CPP).

The diagrams showing the simulation results are summarized in the following table. For each type of propeller, the slow-fast acceleration simulation diagrams are recorded, with sea wave disturbance, while in parentheses () the corresponding towing simulation diagrams are recorded.

Duct	*Propeller Load Observer*		*Torque Coefficient Estimation Scheme*	
	Performance	**Thrust Estimation**	**Performance**	**Thrust Estimation**
Adaptive Form	Fig. 5.3	Fig. 5.4	Fig. 5.5	Fig. 5.6
Kalman Filter	Fig. 5.7	Fig. 5.8	Fig. 5.3 (Fig. 5.13)	Fig. 5.10 (Fig. 5.14)
Propeller Law Constant	Fig. 5.11 (Fig. 5.15)			
CPP	*Propeller Load Observer*		*Torque Coefficient Estimation Scheme*	
	Performance	**Thrust Estimation**	**Performance**	**Thrust Estimation**
Adaptive Form	Fig. 5.17	Fig. 5.18	Fig. 5.19	Fig. 5.20
Kalman Filter	Fig. 5.21	Fig. 5.22	Fig. 5.23 (Fig. 5.27)	Fig. 5.24 (Fig. 5.28)
Propeller Law Constant	Fig. 5.25 (Fig. 5.29)			

Table 5.1: Simulations

5.1.1 Duct Propeller (DuctP)

The settings made to the engine speed controller are shown in the second figure of 5.2, with the final reference value being 1775 rpm.

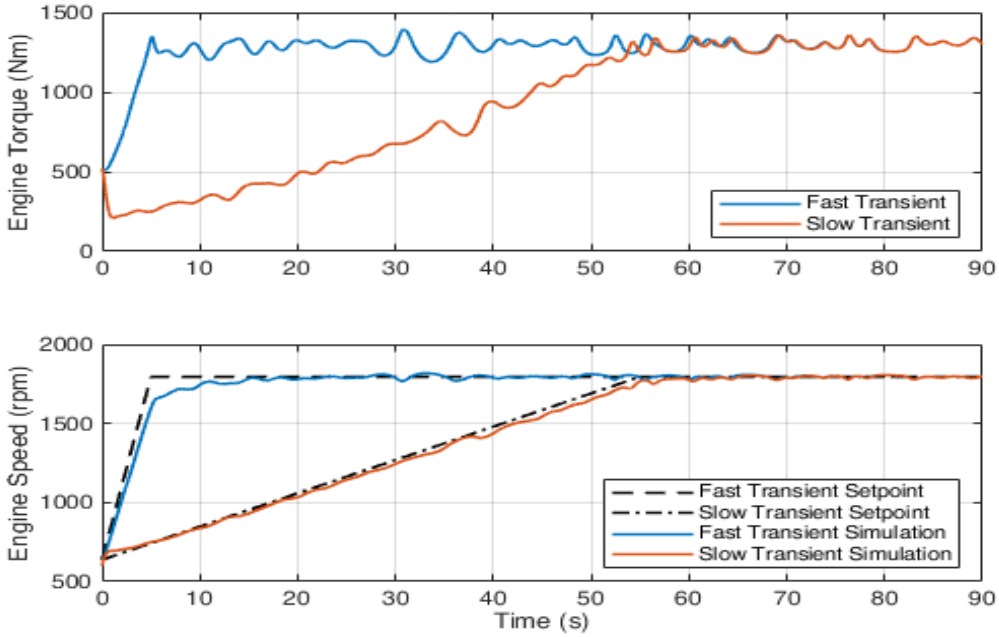

Figure 5.2: Engine setup during slow and fast ship acceleration with wave disturbance (DuctP).

The figure 5.3 shows the performance of the adaptive observer 3.1.1, in relation to the actual values and then in the figure 5.4 the corresponding thrust estimation of the propeller is presented.

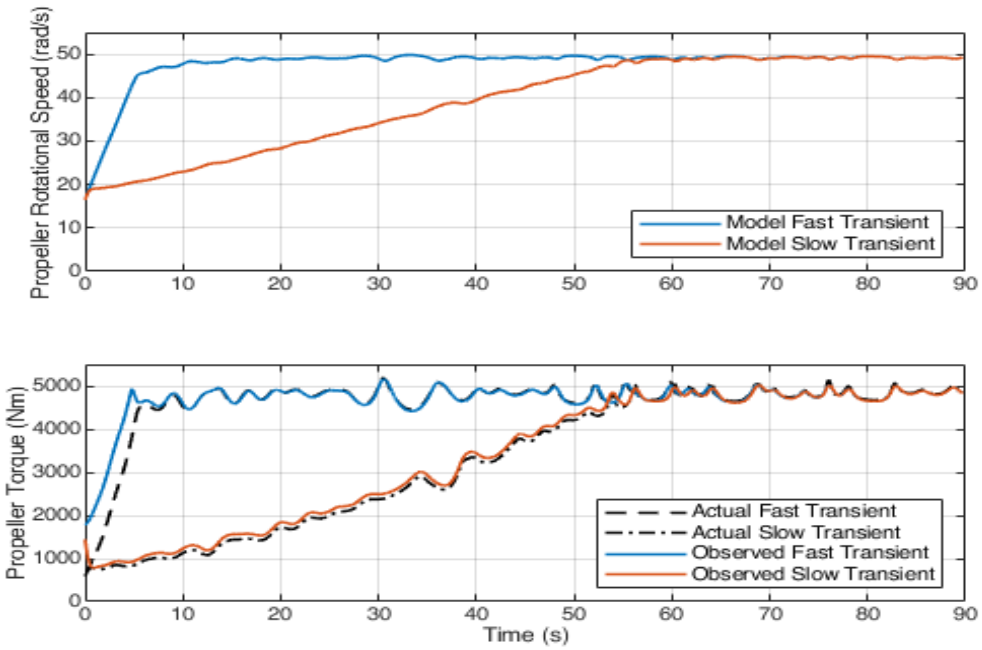

Figure 5.3: Adaptive Observer performance (DuctP).

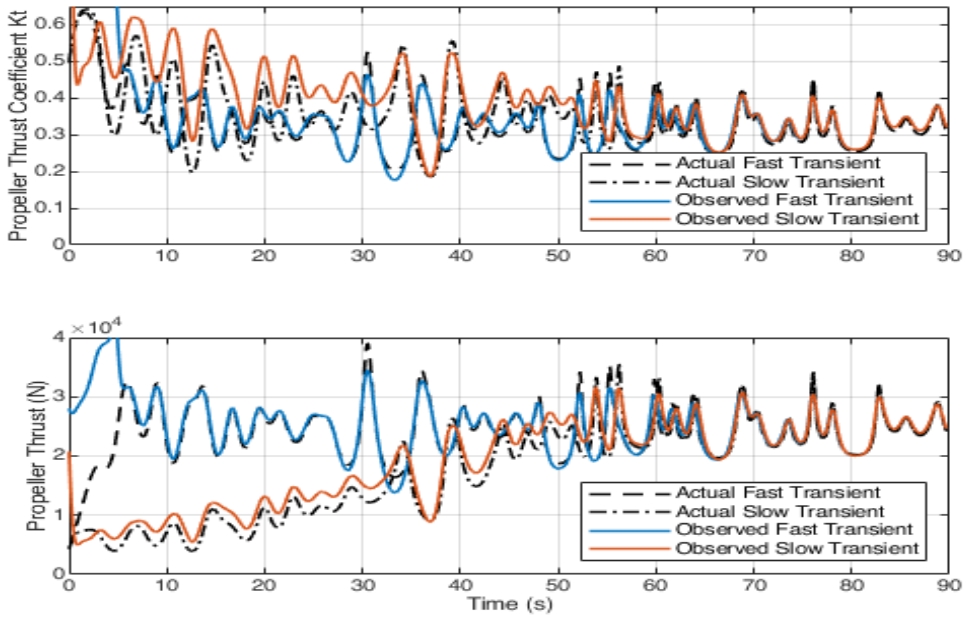

Figure 5.4: Thrust Estimation Scheme performance of 5.3 (DuctP).

The figure 5.5 shows the performance of the adaptive torque coefficient estimation scheme 3.1.2, in relation to the actual values and then in the figure 5.6 the corresponding thrust estimation of the propeller is presented.

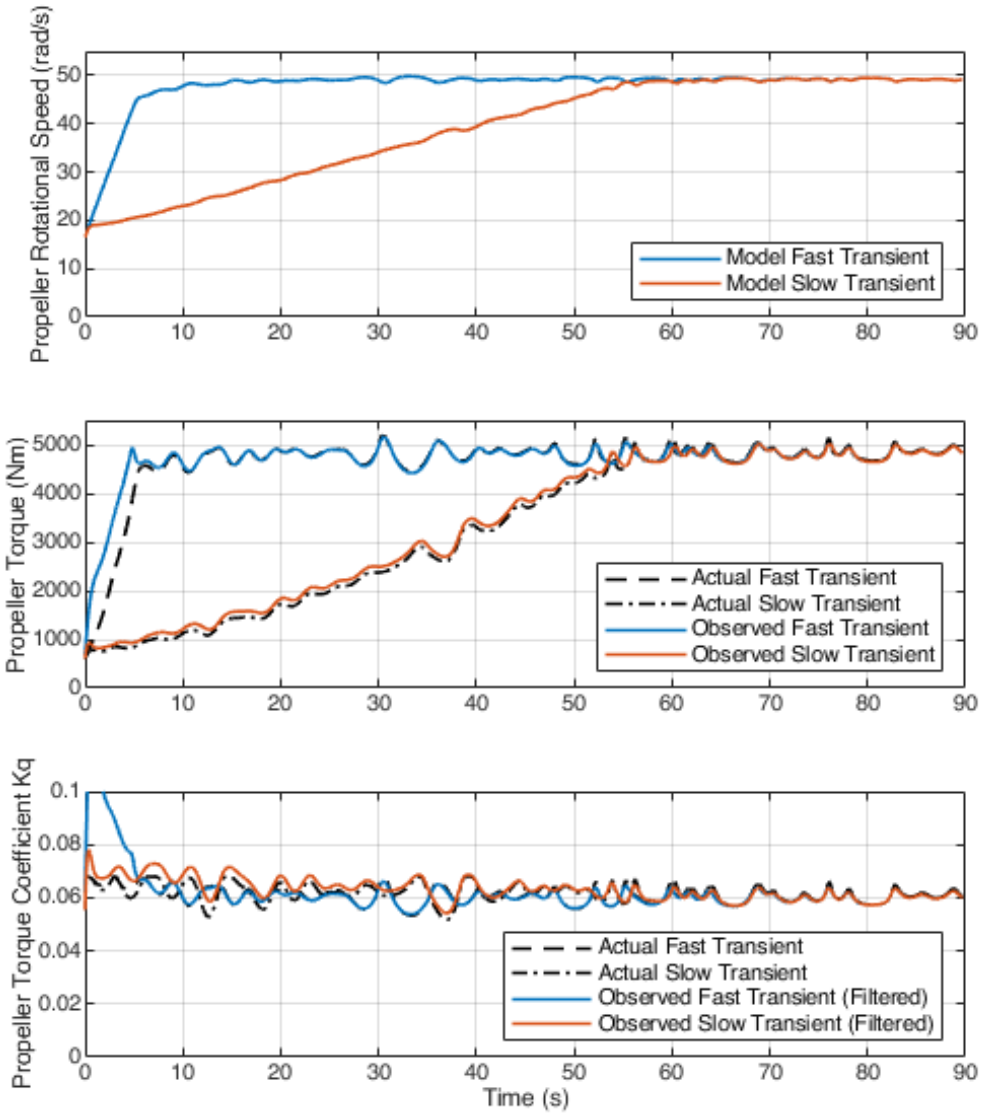

Figure 5.5: Adaptive Torque Koefficient Estimation Scheme performance (DuctP).

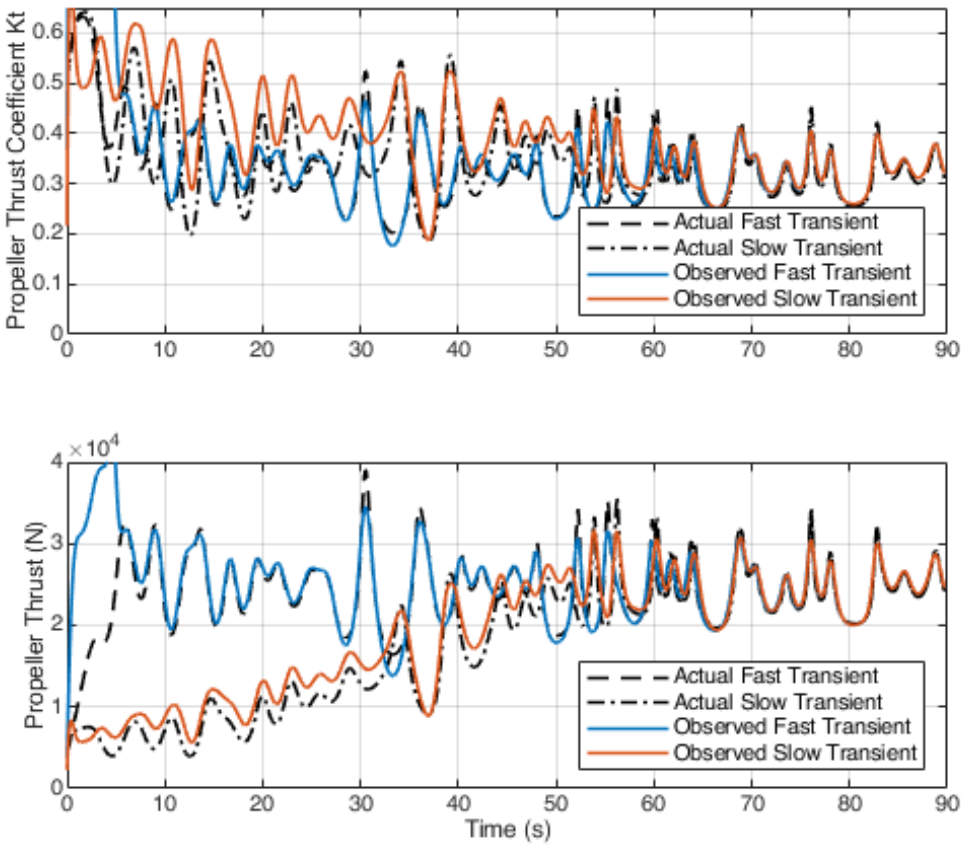

Figure 5.6: Thrust Estimation Scheme performance of 5.5 (DuctP).

The figure 5.7 shows the performance of the Kalman Filter observer 3.2.1, in relation to the actual values and then in the figure 5.8 the corresponding thrust estimation of the propeller is presented.

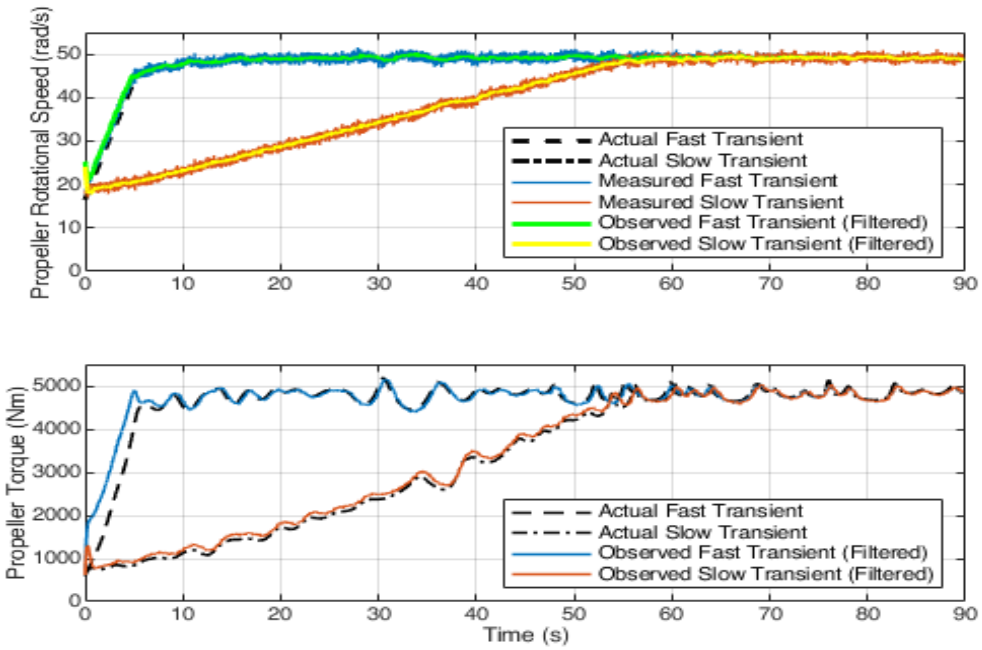

Figure 5.7: Kalman Filter Observer performance (DuctP).

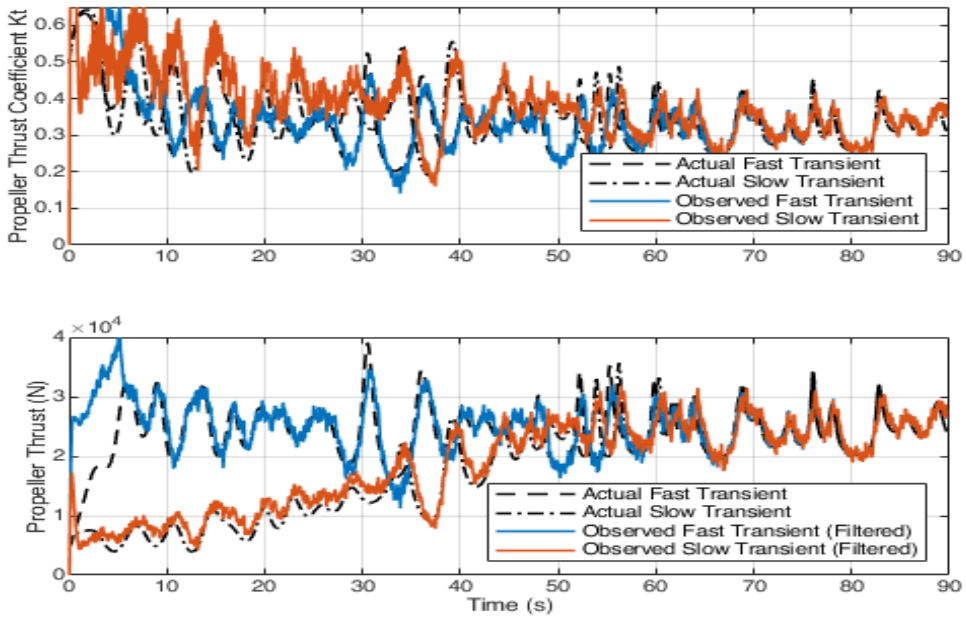

Figure 5.8: Thrust Estimation Scheme performance of 5.7 (DuctP).

The figure 5.9 shows the performance of the Kalman Filter torque coefficient estimation scheme 3.2.2, in relation to the actual values and then in the figure 5.10 the corresponding thrust estimation of the propeller is presented.

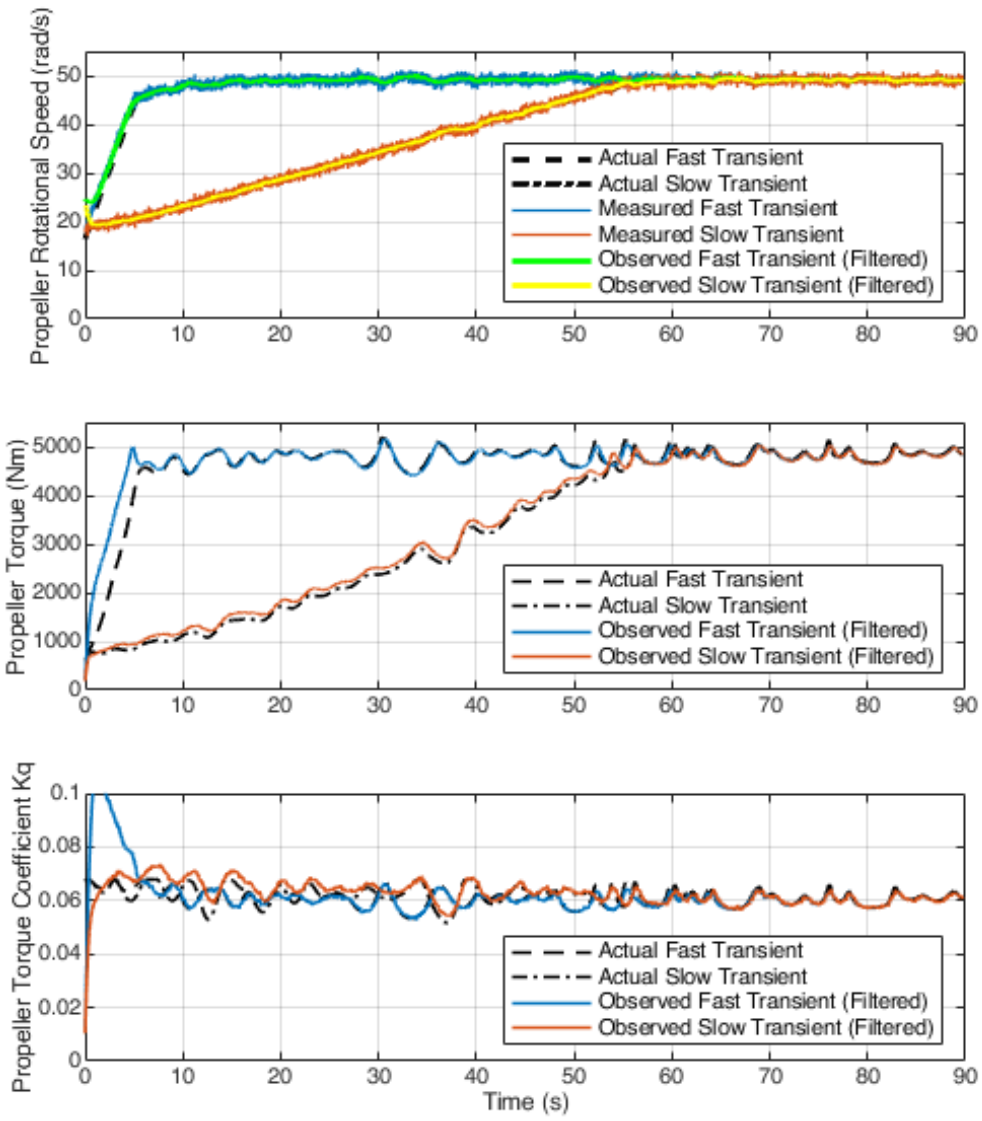

Figure 5.9: Kalman Filter Torque Koefficient Estimation Scheme performance (DuctP).

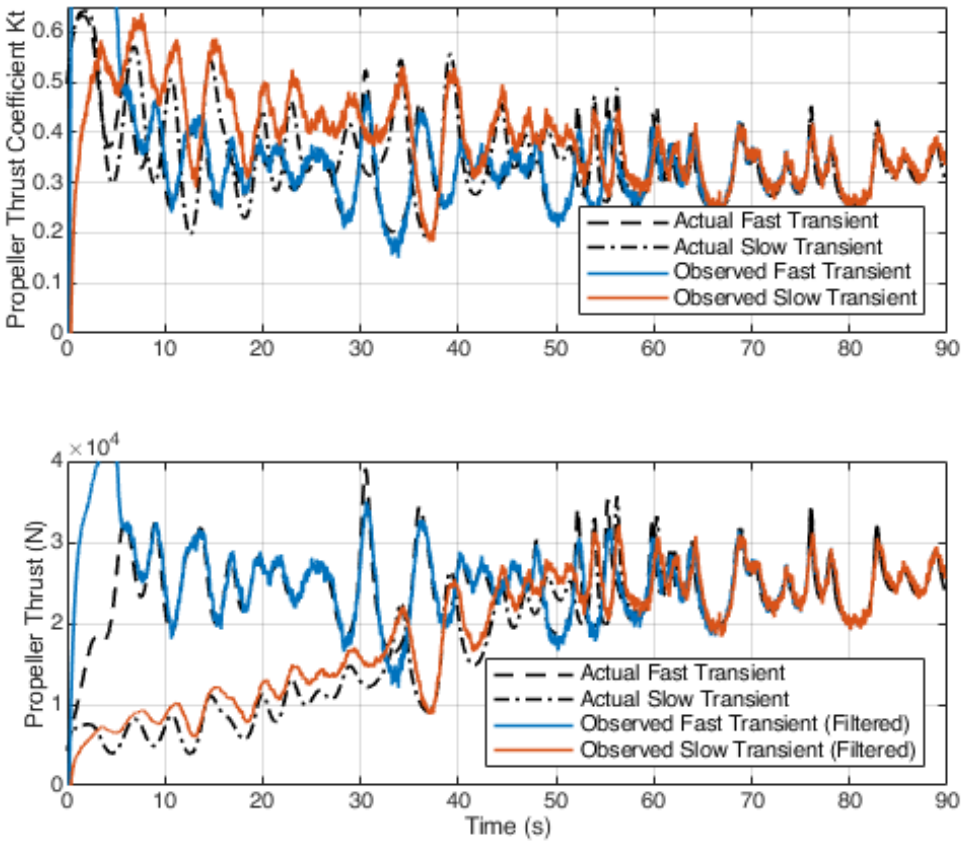

Figure 5.10: Thrust Estimation Scheme performance of 5.9 (DuctP).

The figure 5.11 shows the estimation of the propeller law constant c based on 3.3.

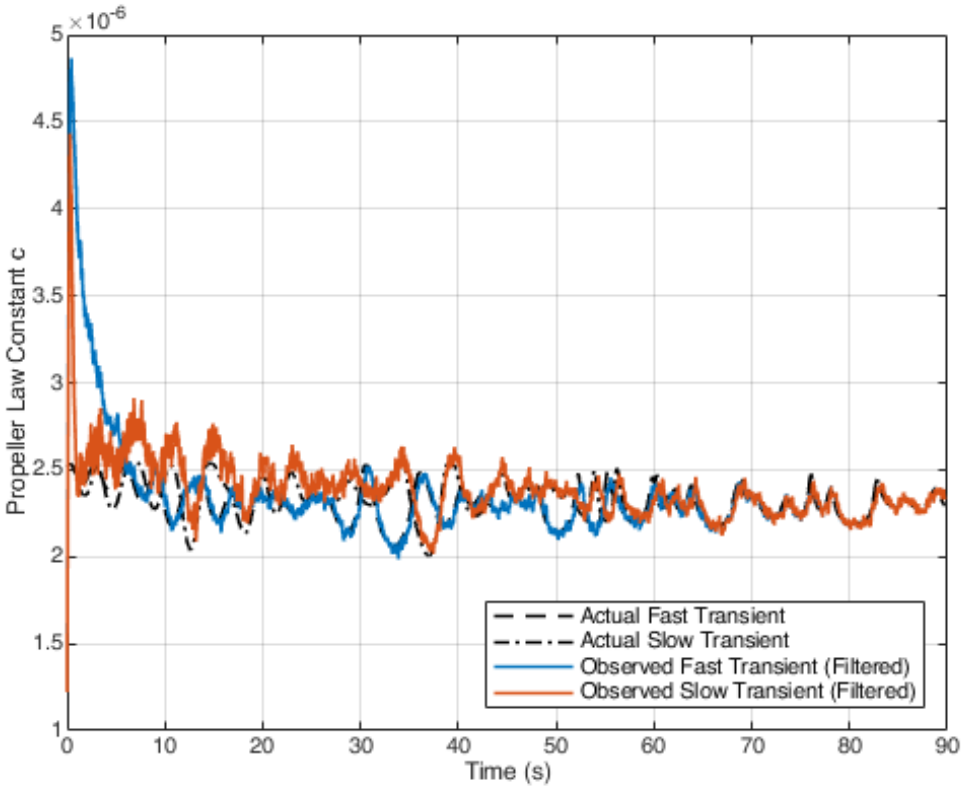

Figure 5.11: Propeller Law Constant Estimation (DuctP).

The settings made to the engine speed controller during towing are shown in the second figure of 5.12, with the final reference value being 1775 rpm.

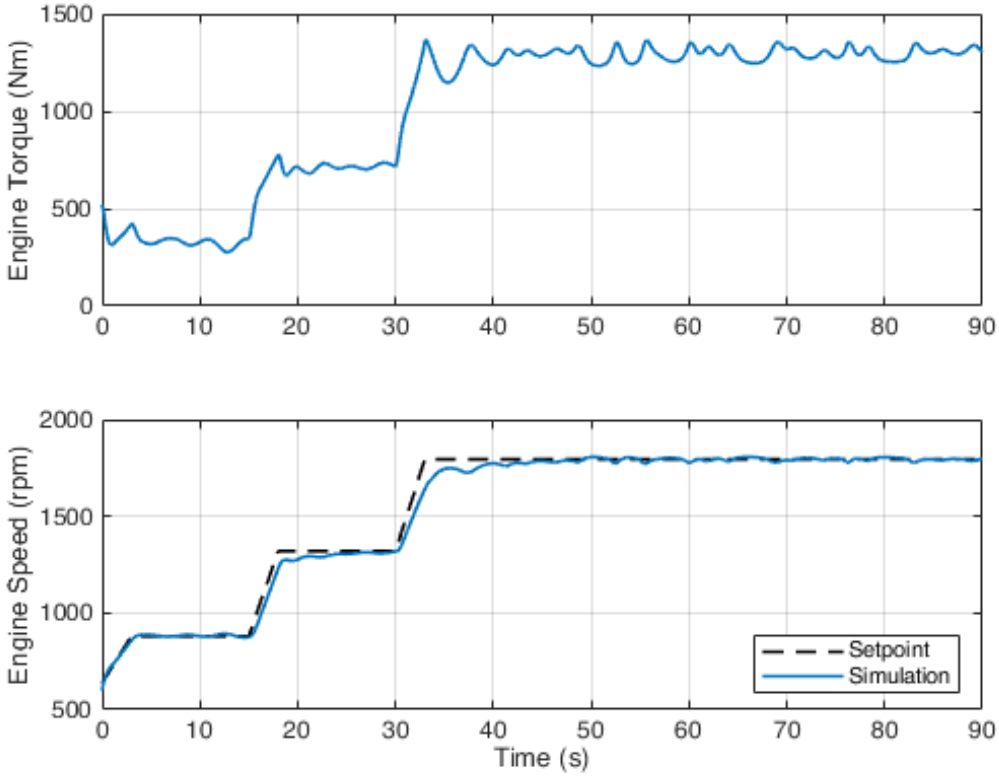

Figure 5.12: Engine setup during ship towing with wave disturbance (DuctP).

The diagrams of the observer performance, the thrust estimation and the estimation of the propeller law constant c follow, respectively.

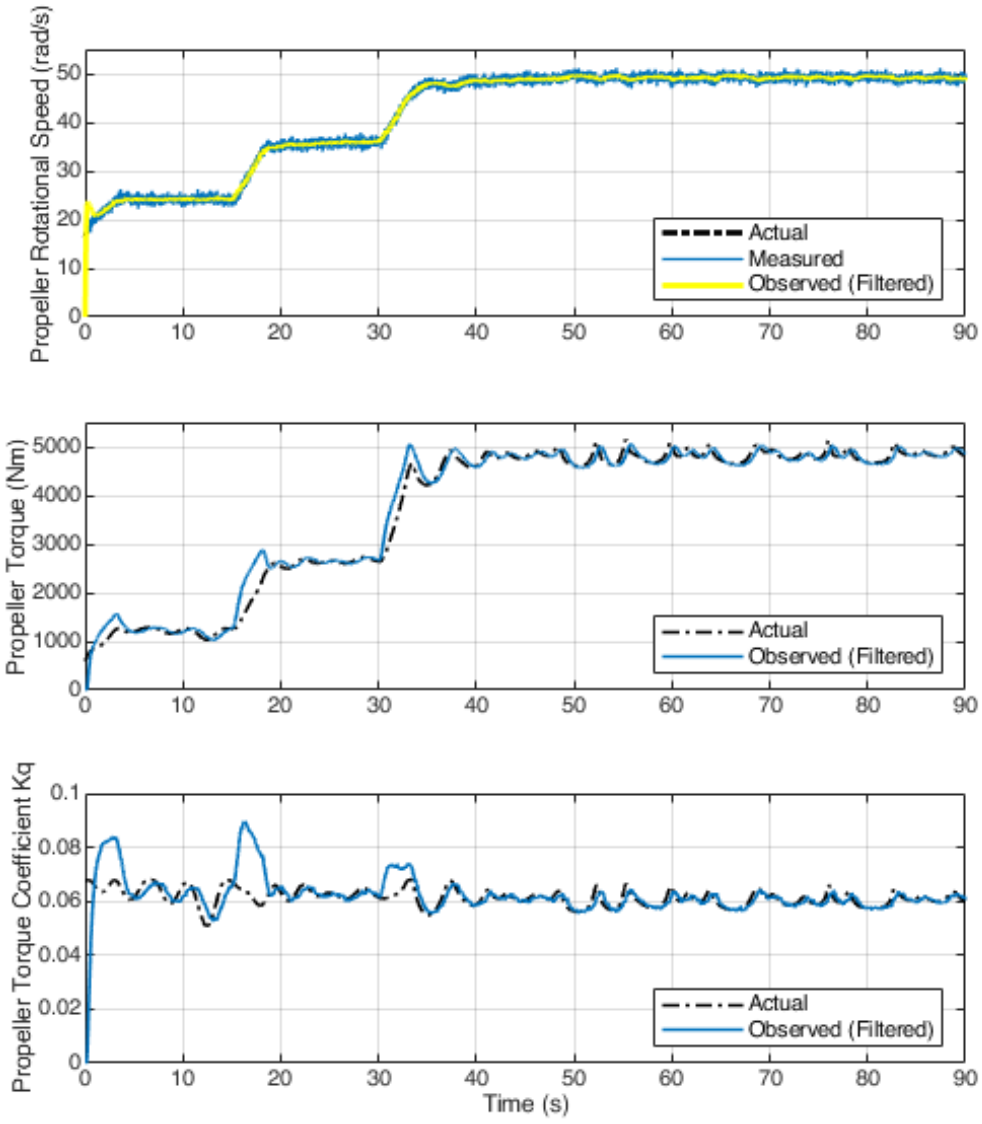

Figure 5.13: Kalman Filter Observer performance (DuctP).

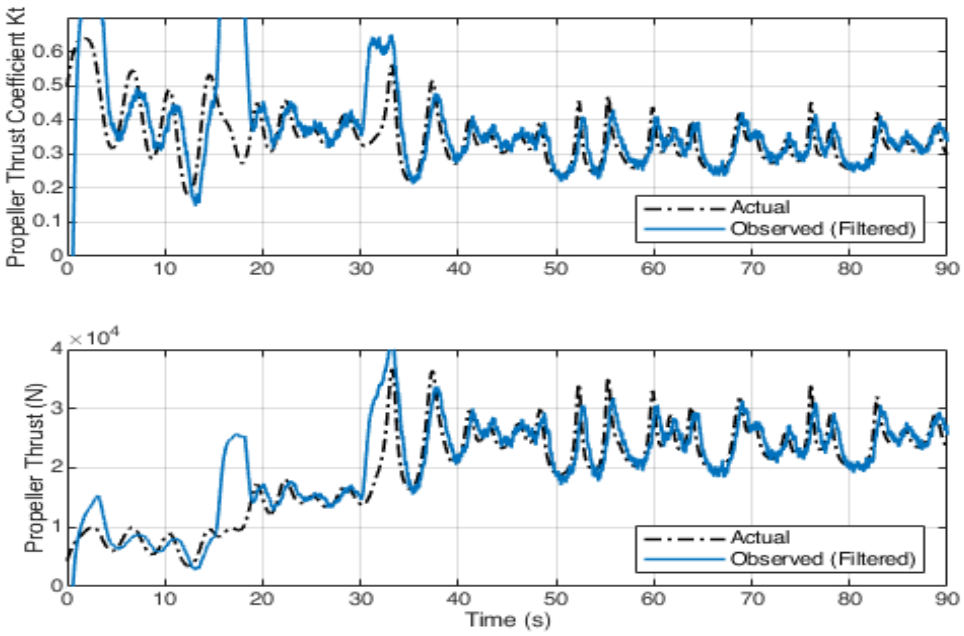

Figure 5.14: Thrust Estimation Scheme performance of 5.13 (DuctP).

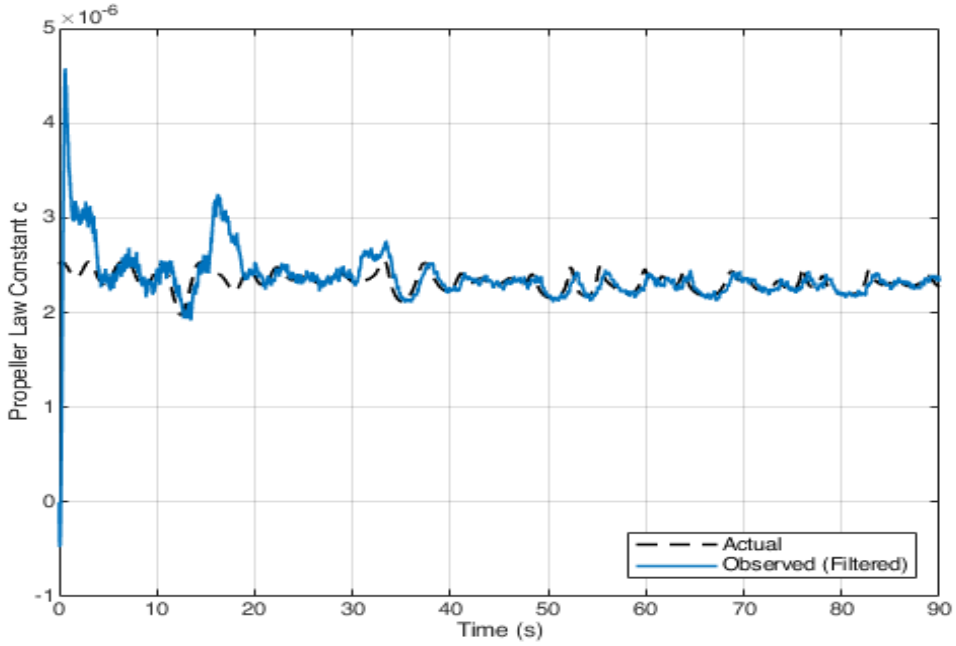

Figure 5.15: Propeller Law Constant Estimation (DuctP).

5.1.2 Controllable Pitch Propeller (CPP)

The propeller pitch is adjusted as shown in the first figure of 5.16. The settings made to the engine speed controller are shown in the third figure of 5.16, with the final reference value being 1775 rpm.

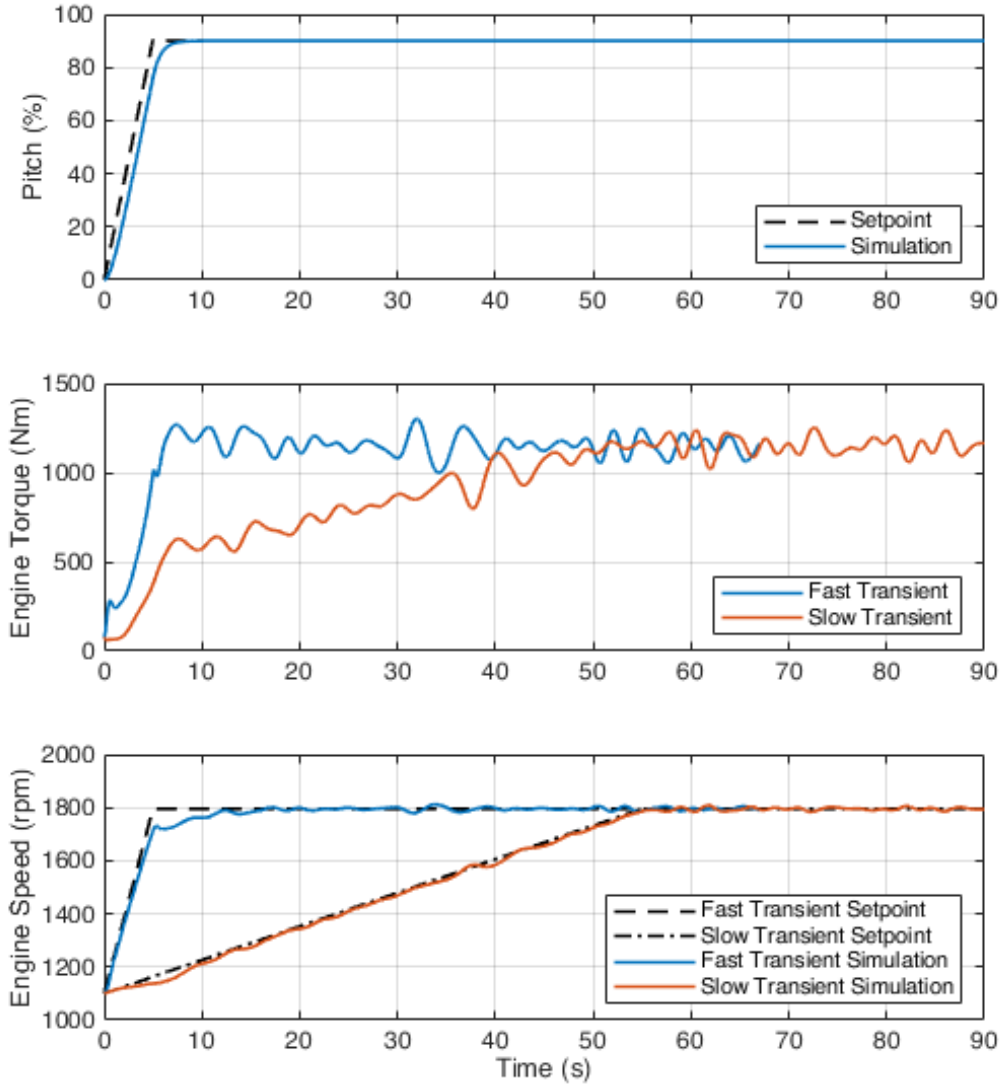

Figure 5.16: Engine and Pitch setup during slow and fast ship acceleration with wave disturbance (CPP).

The figure 5.17 shows the performance of the adaptive observer 3.1.1, in relation to the actual values and then in the figure 5.18 the corresponding thrust estimation of the propeller is presented.

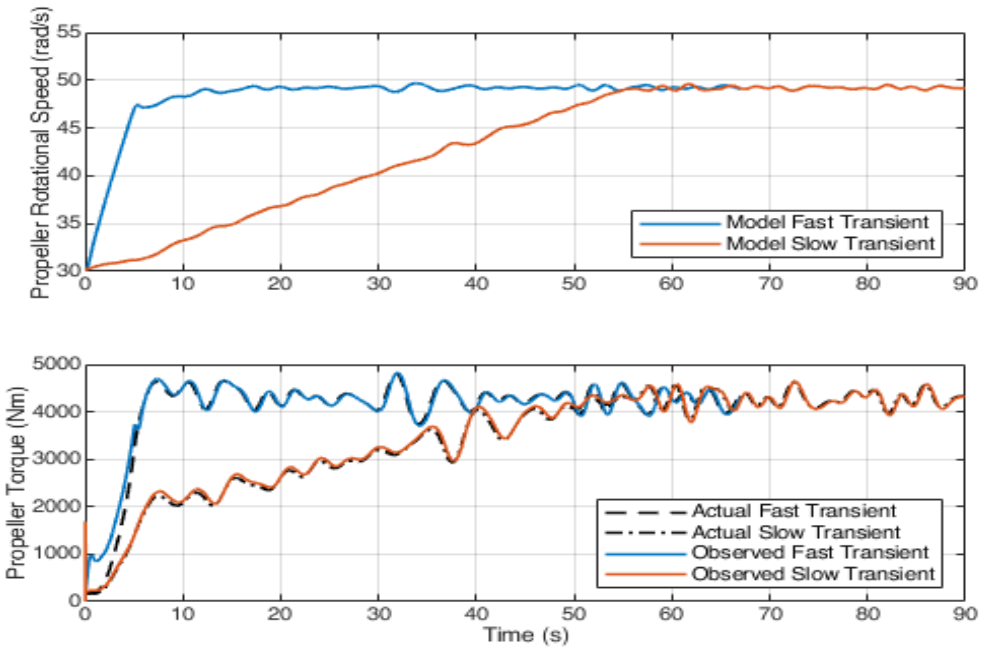

Figure 5.17: Adaptive Observer performance (CPP).

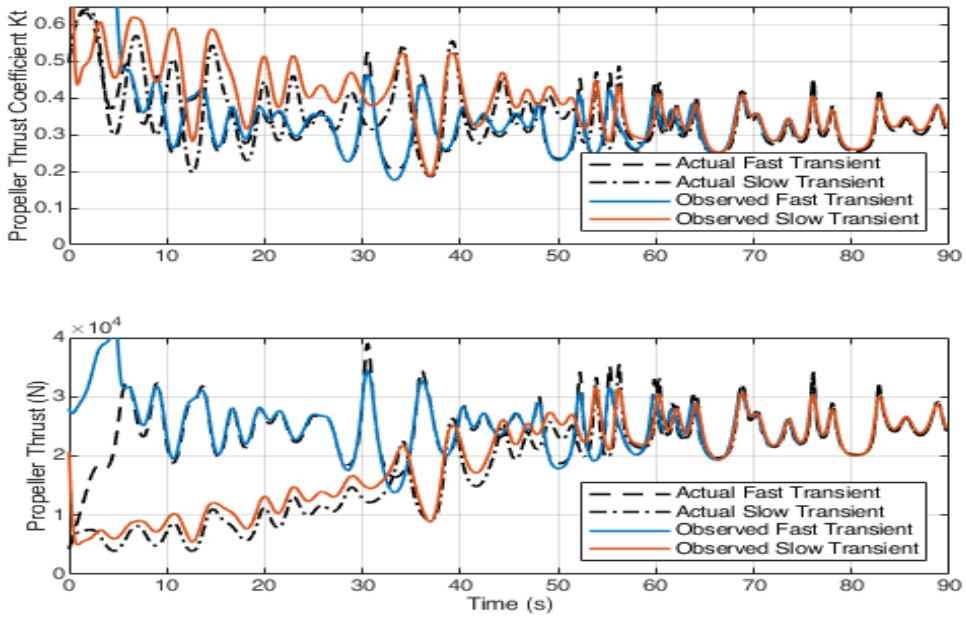

Figure 5.18: Thrust Estimation Scheme performance of 5.17 (CPP).

The figure 5.19 shows the performance of the adaptive torque coefficient estimation scheme 3.1.2, in relation to the actual values and then in the figure 5.20 the corresponding thrust estimation of the propeller is presented.

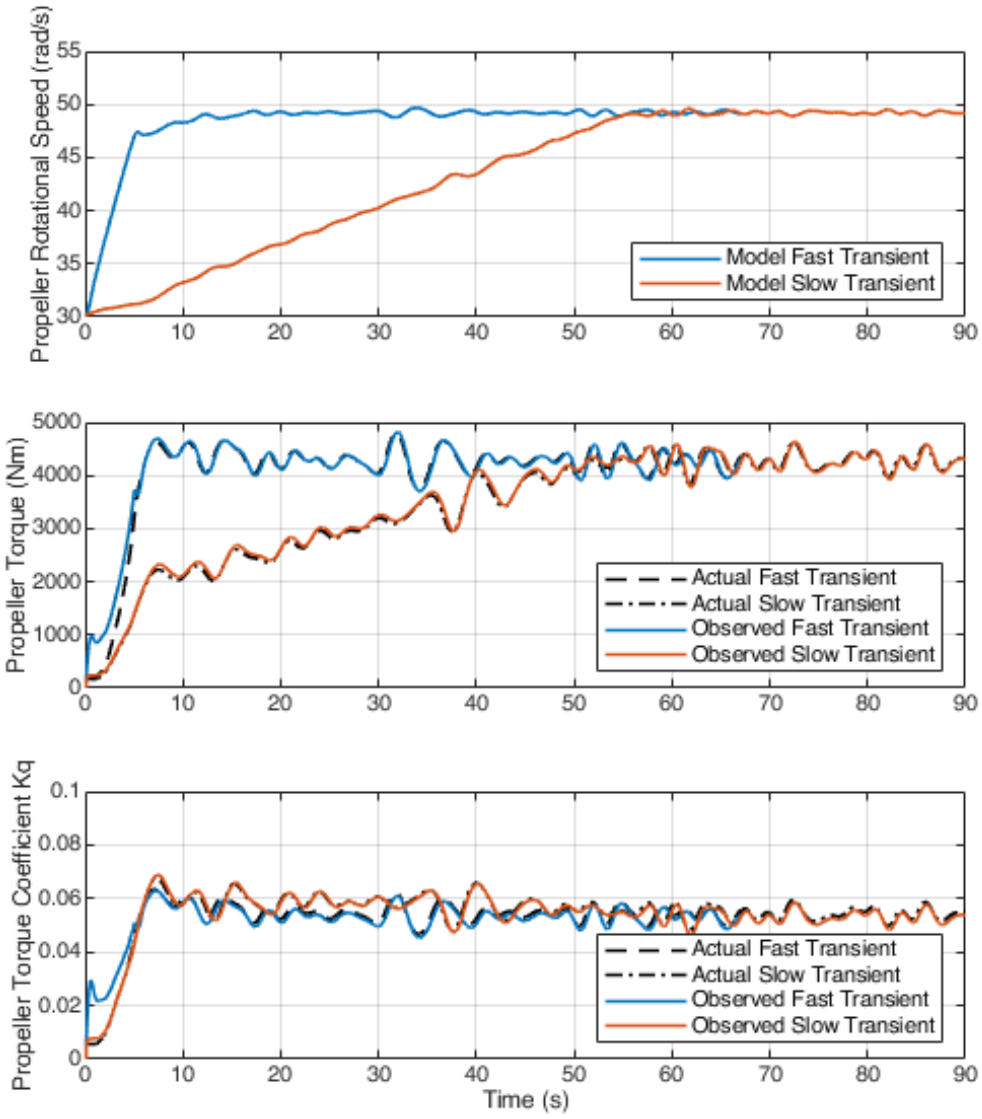

Figure 5.19: Torque Koefficient Adaptive Estimation Scheme performance (CPP).

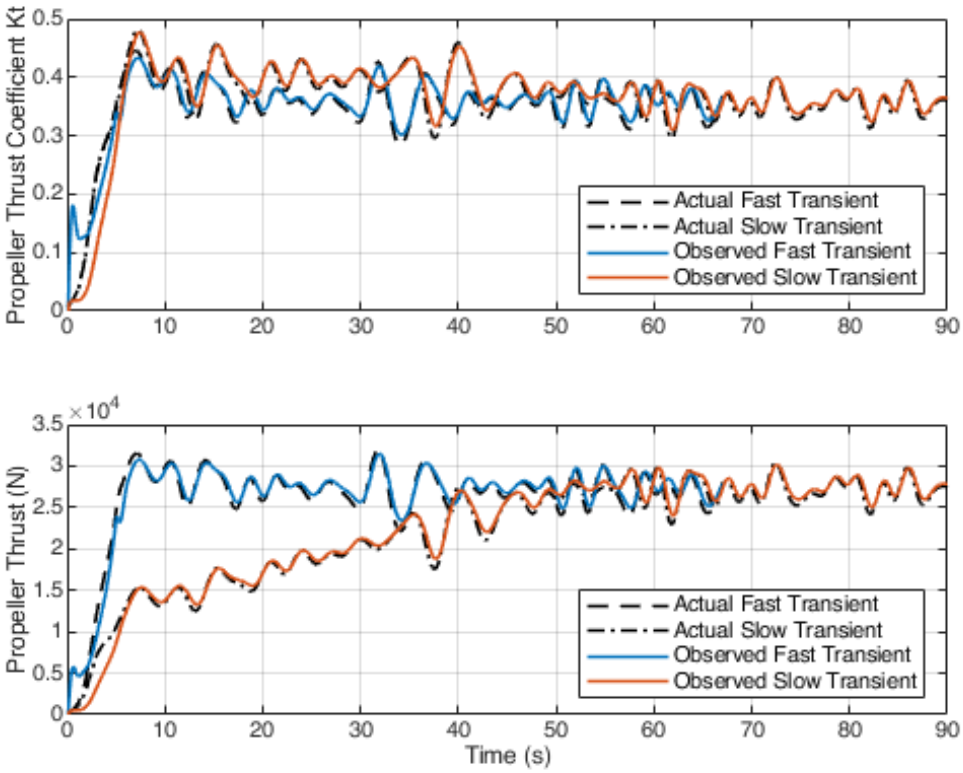

Figure 5.20: Thrust Estimation Scheme performance of 5.19 (CPP).

The figure 5.21 shows the performance of the Kalman Filter observer 3.2.1, in relation to the actual values and then in the figure 5.22 the corresponding thrust estimation of the propeller is presented.

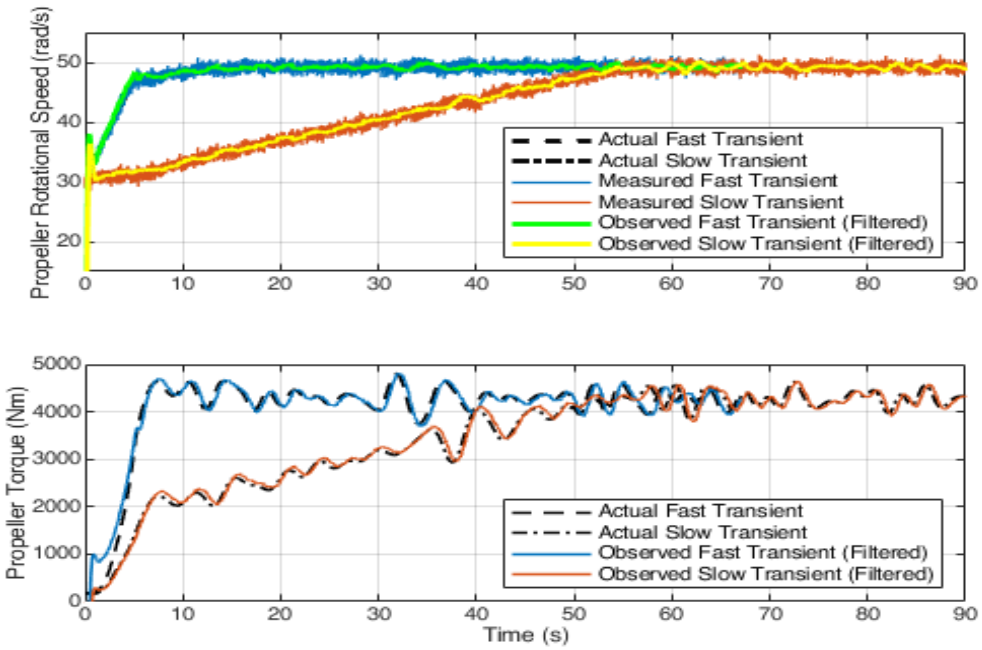

Figure 5.21: Kalman Filter Observer performance (CPP).

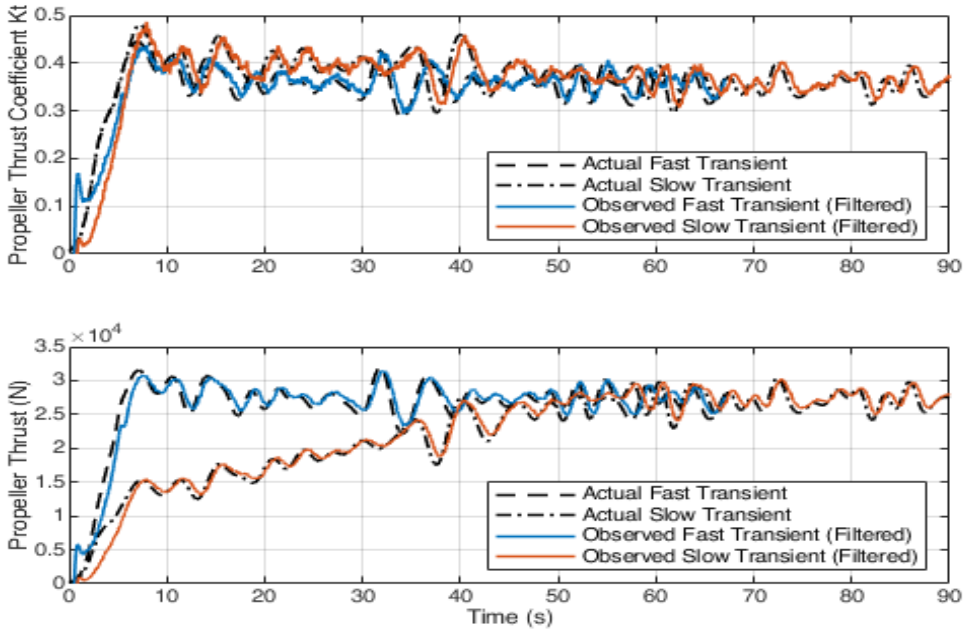

Figure 5.22: Thrust Estimation Scheme performance of 5.21 (CPP).

The figure 5.23 shows the performance of the Kalman Filter torque coefficient estimation scheme 3.2.2, in relation to the actual values and then in the figure 5.24 the corresponding thrust estimation of the propeller is presented.

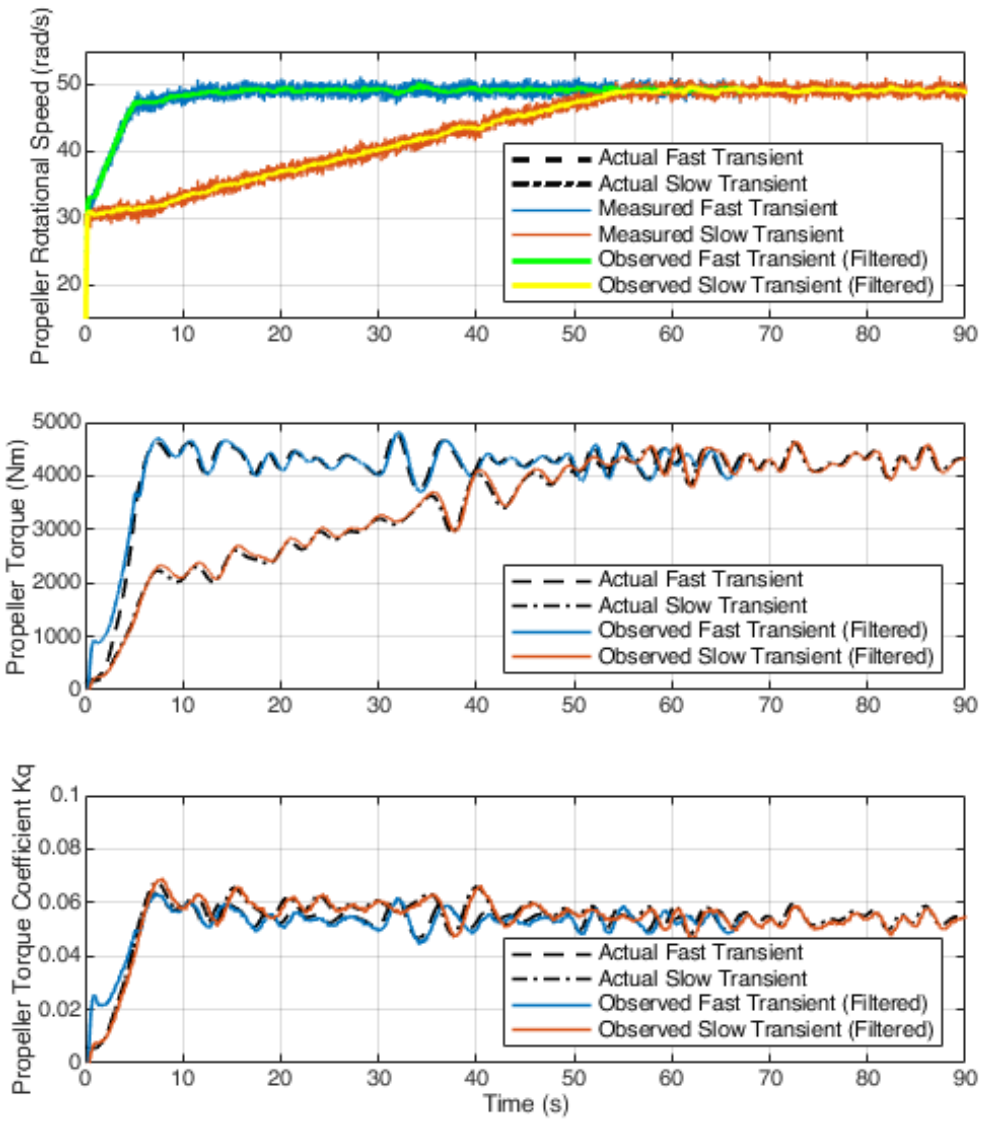

Figure 5.23: Torque Koefficient Kalman Filter Estimation Scheme performance (CPP).

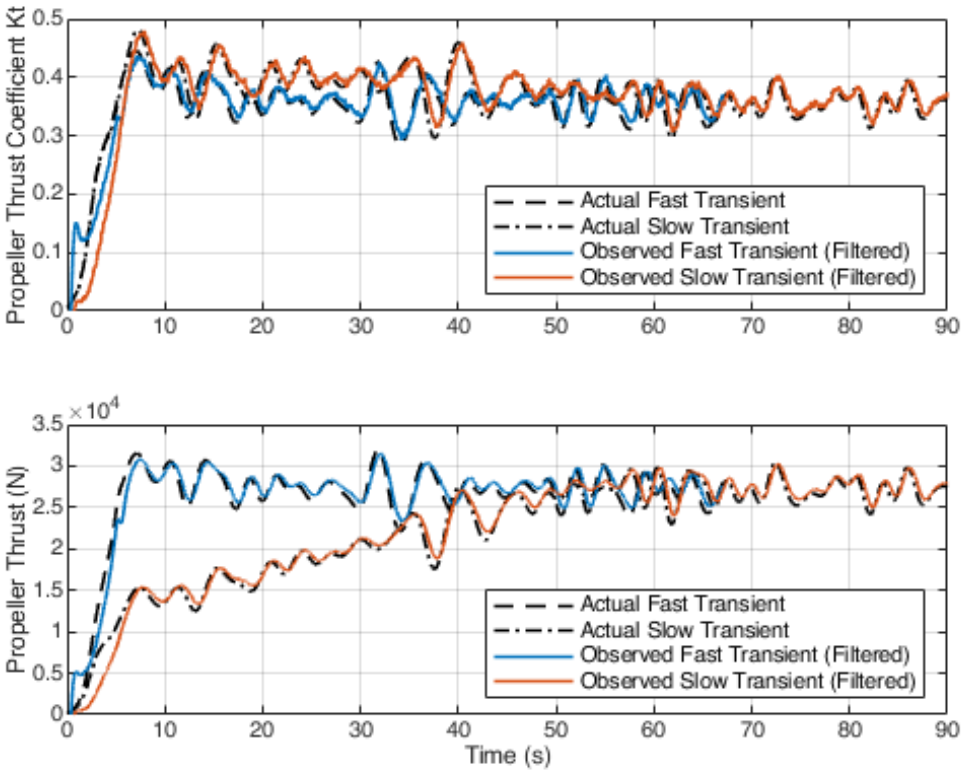

Figure 5.24: Thrust Estimation Scheme performance of 5.23 (CPP).

The figure 5.25 shows the estimation of the propeller law constant c based on 3.3.

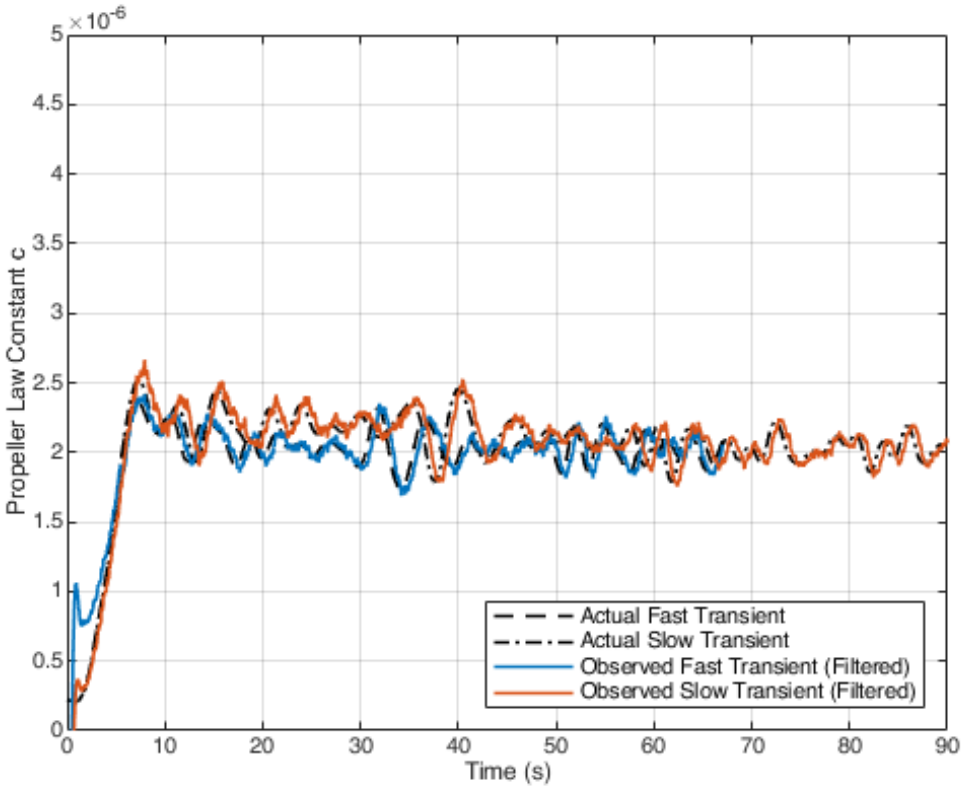

Figure 5.25: Propeller Law Constant Estimation (CPP).

The propeller pitch is adjusted as shown in the first figure of 5.26. The settings made to the engine speed controller during towing are shown in the second figure of 5.26.

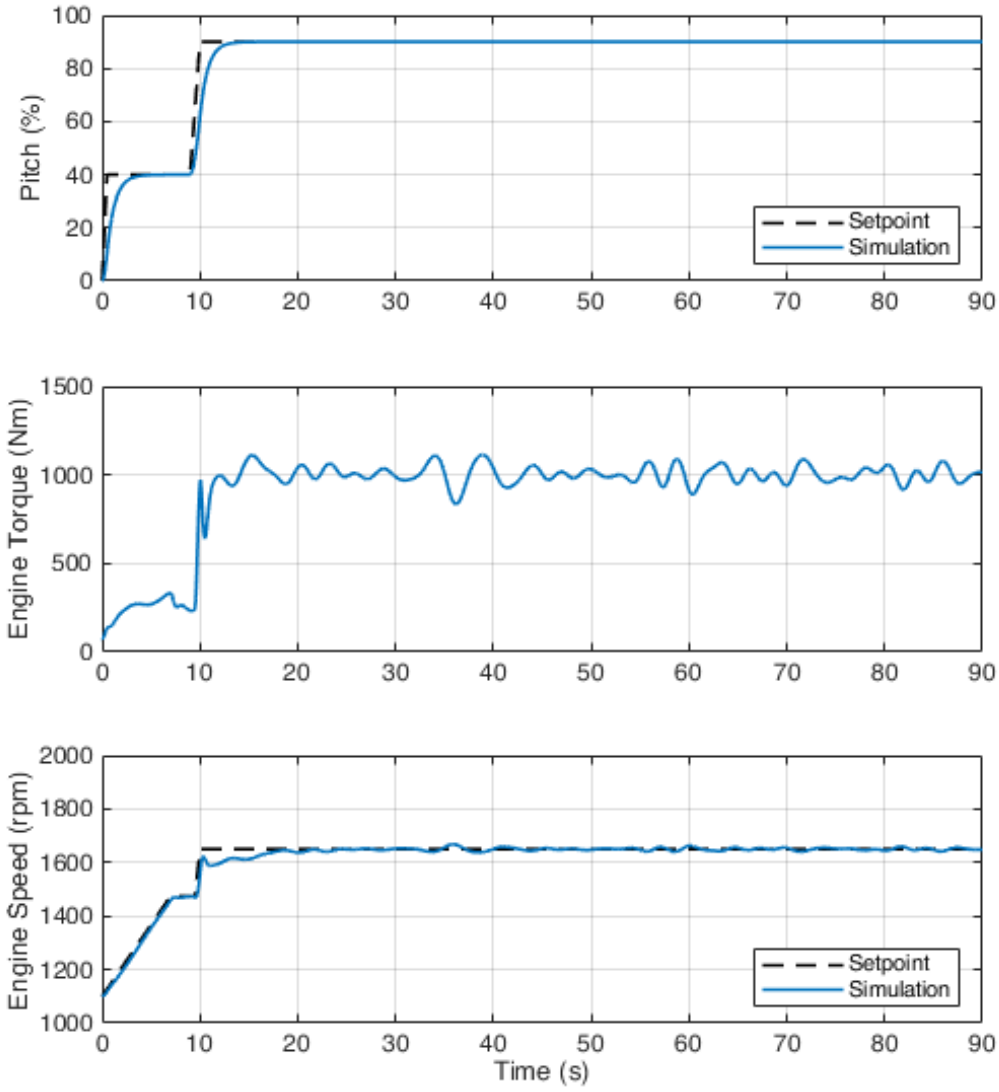

Figure 5.26: Engine and Pitch setup during ship towing with wave disturbance (CPP).

The diagrams of the observer performance, the thrust estimation and the estimation of the propeller law constant c follow, respectively.

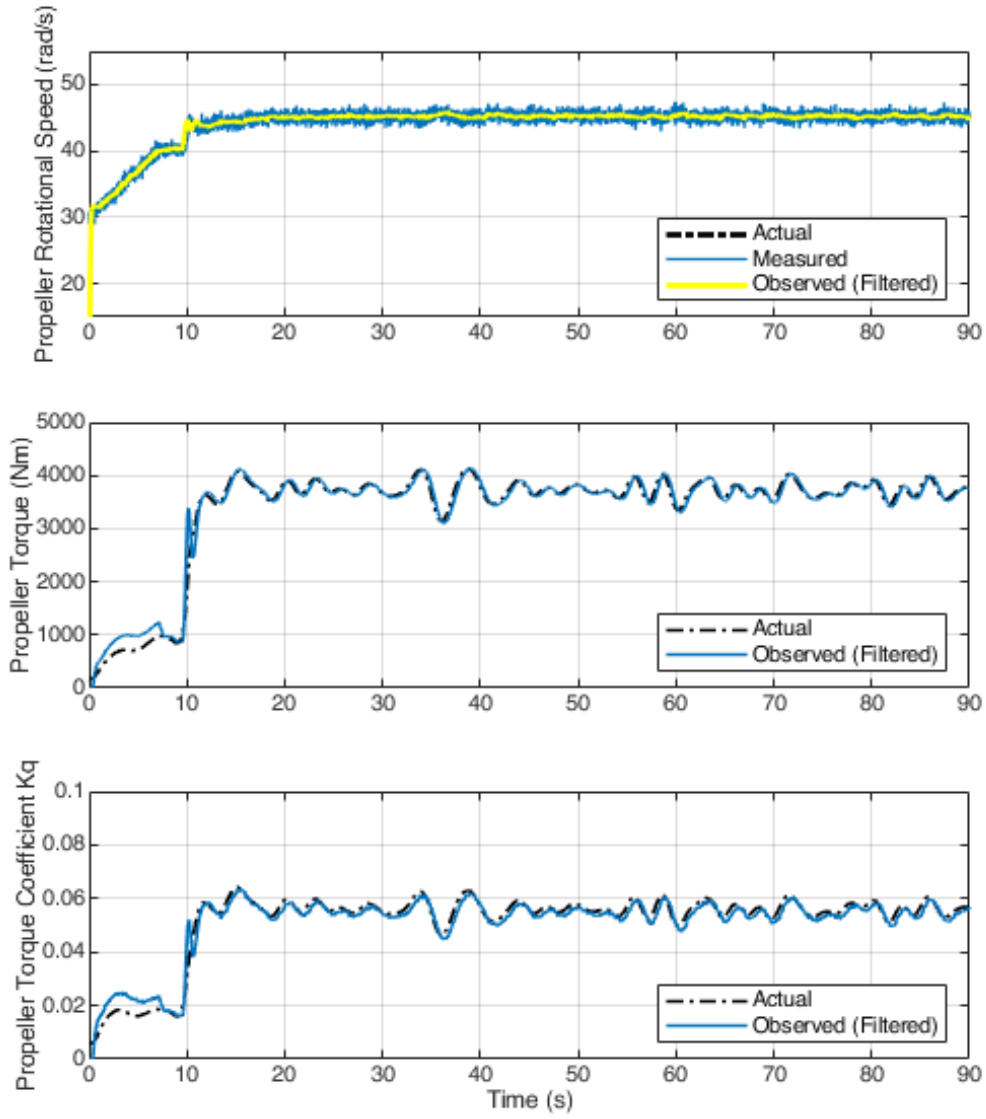

Figure 5.27: Kalman Filter Observer performance (CPP).

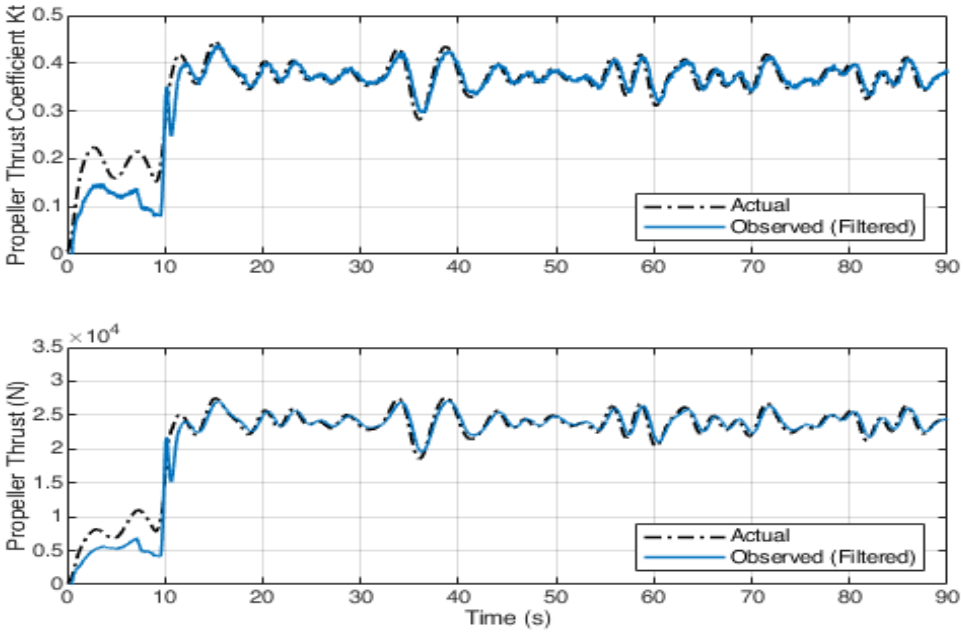

Figure 5.28: Thrust Estimation Scheme performance of 5.27 (CPP).

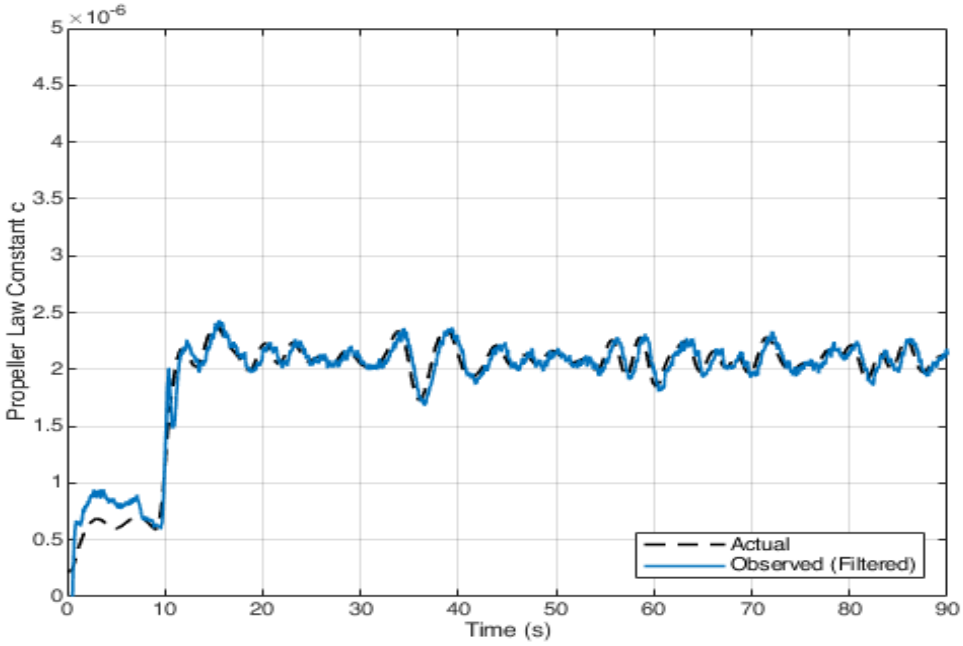

Figure 5.29: Propeller Law Constant Estimation (CPP).

5.1.3 Results Analysis

As shown in Figures 5.3, 5.5 of the duct propeller but also in the corresponding Figures 5.17, 5.19 of the controllable pitch propeller, the performance of both the propeller load observer 3.1.1 and the torque coefficient estimation scheme 3.1.2 is particularly high. Despite the relatively high significant wave height and frequent fluctuations, both models adapt quickly, ensuring satisfactory accuracy at the observed values. It is noted, however, that during abrupt changes in engine speed, during acceleration or towing there is a slight deviation from the actual values. In conclusion, both adaptive observation models can be used with great efficiency in cases where the measurements used do not have high noise levels.

As shown in Figures 5.7, 5.9 of the duct propeller but also in the corresponding Figures 5.21, 5.23 of the controllable pitch propeller, the performance of both Kalman Filter models (3.2.1, 3.2.2) is satisfactory. Measurements are filtered and reconstracted with great accuracy, limiting noise. The observed values follow the course of the corresponding actual values with small error and small time shift. It can be shown, however, that in the case of the torque estimation scheme 3.2.2, smoother results are obtained, but just as accurate as in the case of the propeller load observer. This is probably due to the fact that the system is regulated by lower noise covariances.

In each thrust diagram it can be observed that there is a deviation from the actual values, during acceleration ($0 - 5$ s in fast acceleration and $0 - 55$ s in slow acceleration) or towing. As mentioned in subsection 3.1.3, this is due to the fact that the linear relationship 3.1.17 applies locally to small changes in speed. Thus, until it is stabilized around a certain range, the specific model displays offset.

As expected, the estimated value of propeller law constant c is stabilized, with small changes, near a certain value. Fast or slow acceleration does not so drastically affect the change in this constant as parameters such as weather operating conditions or hull fouling.

Regarding the type of propeller, it is easy to understand that in the case of the controllable pitch propeller the results are more accurate and smooth. This is due to the fact that the $C - series$ of CPP have been designed much more recently than the corresponding of the DuctP. The ability to change the pitch of the propeller while changing its speed, allows the propeller to adapt quicker ,during the acceleration of the ship, passing through more efficient operating points.

5.2 Shaft Rotational Speed Prediction Models

In this section, the simulations of the two prediction models (Markov Chain and Neural Network) are presented in the corresponding Figures 5.30, 5.31. The simulation was performed for a specific period of time (2500 s) during the operation of the tugboat. For the Neural Network model, also the performance and error diagrams are attached.

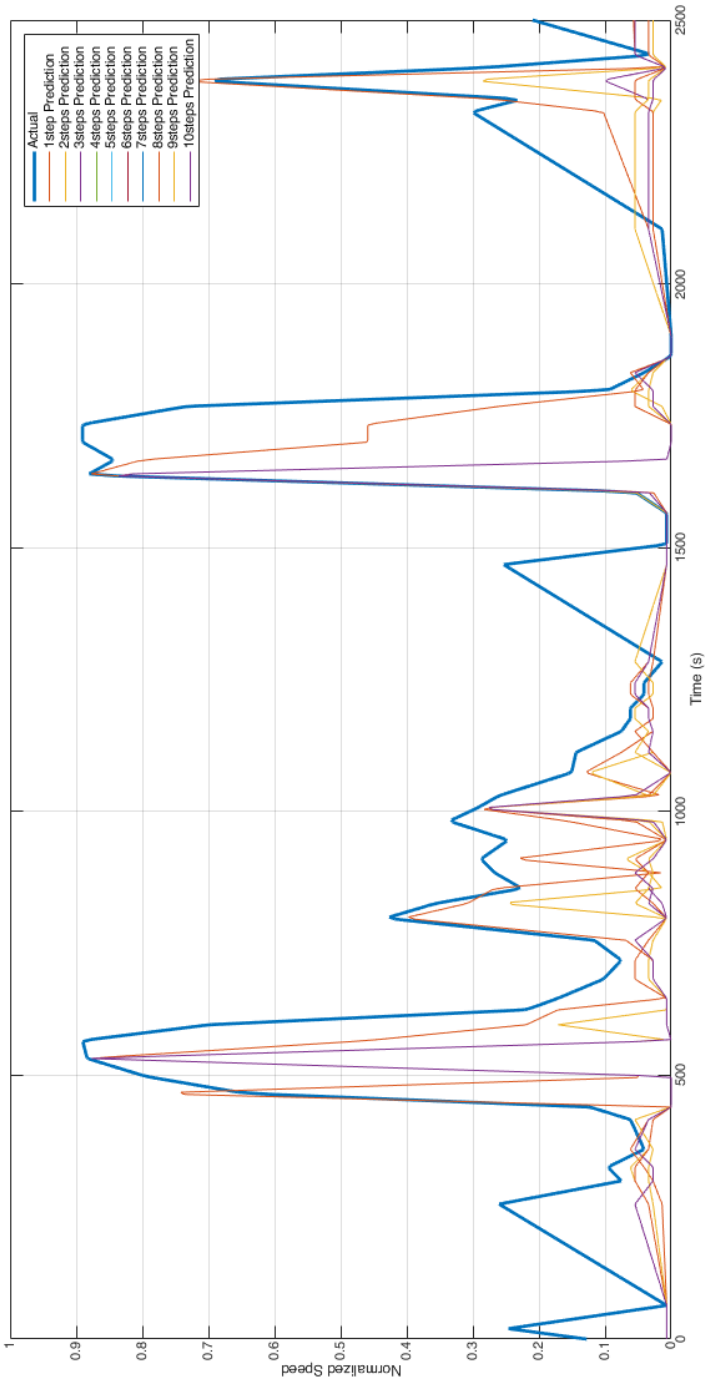

Figure 5.30: Markov Chain prediction model performance

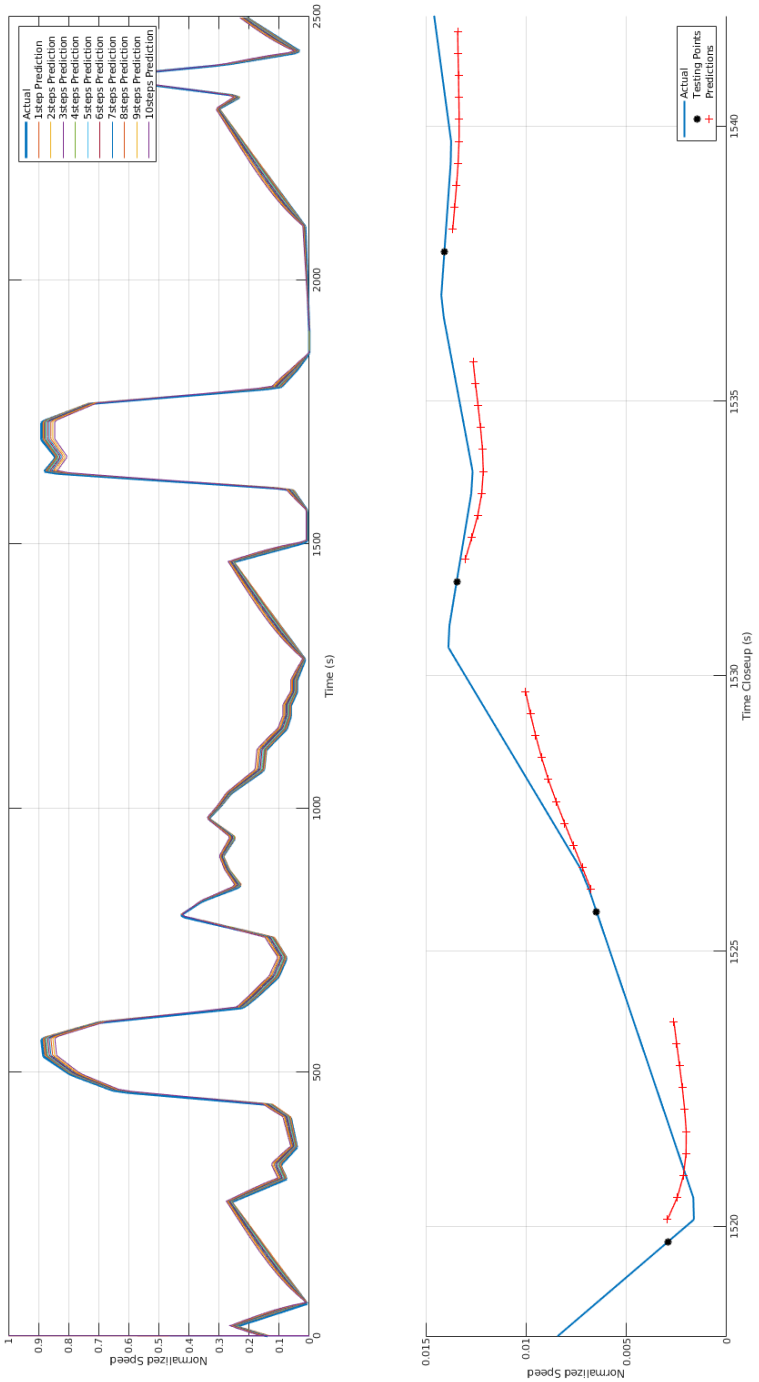

Figure 5.31: Neural Network prediction model performance

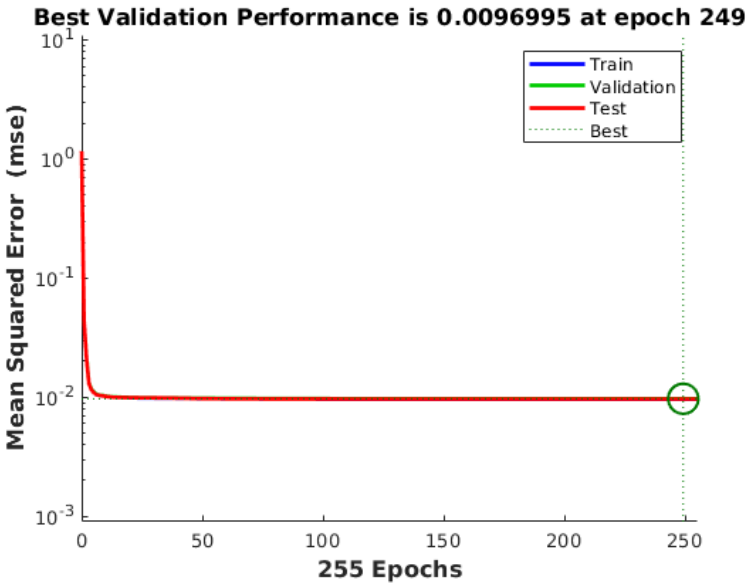

Figure 5.32: NN Performance

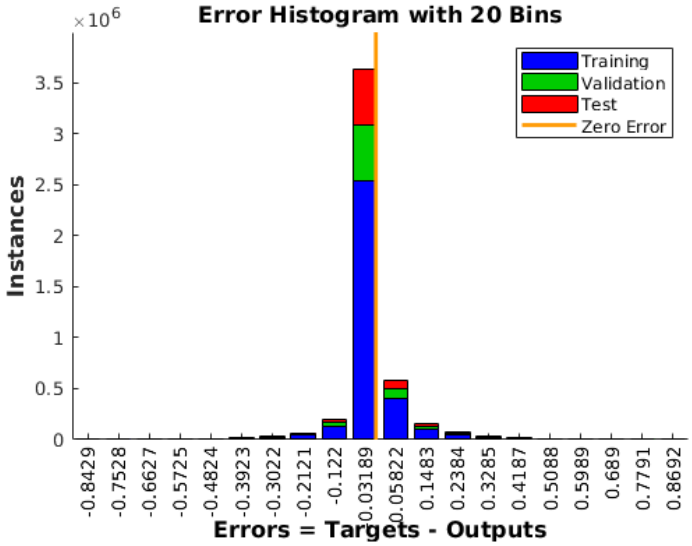

Figure 5.33: Error Histogram

5.2.1 Results Analysis and Models Comparison

Observing the displayed diagrams, it is easy to see that the Neural Network gives much more accurate predictions than the Markov Chain model. More specifically, in the Markov model, although the forecasts tend to follow the course of the actual values, they deviate from them very quickly. From the very first step, the predictions are not reliable, while as the steps progress, the predictions are expected to worsen. It is also noted ,from Figure 5.30, that after the fifth step, the model underfunctions even more. As mentioned in 4.1, this is due to the fact that the estimates of this model depend exclusively on the previous state and not on a sufficient number of previous values. Therefore, the estimated values are limited to a specific set, with the highest probabilities in the state transition probability matrix. In contrast to the Markov model, the Neural Network takes into account states at an horizon of 10 previous steps, giving significantly more accurate results. It can be observed that as the prediction step increases, so does the deviation from the corresponding real values. However, the overall performance and regression of the neural network is particularly high, even though its training time is short. It can also be noted ,from Figure 5.31, that the neural network predictions follow the course of the target values, with both small error and small time shift. The largest deviations are observed in the very high and very low speed values, but also in its abrupt alternations. This is due to the speed variation in the collected dataset, which consist of constant speed segments and intense accelerations and decelerations due to the tugboat operation. In conclusion, for the operation of the MPC, the neural network described above gives satisfactory predictions up to 10 steps forward.

Chapter 6

Conclusions and Future Work

Conclusions

In this work, two observer models were investigated in order to use them in the energy management system of the ship's propulsion plant. At the initial stage, the ship was fully modeled, based on the principles of resistance and propulsion. The type of ship, its main propulsion engine, its propeller and other technical characteristics were selected. After it was tested that this modeling adequately corresponds to the actual operating conditions of a tugboat, observers were designed to assess some states of the system, such as the propeller load torque and the toque coefficient. Observation equations have been used to build both adaptive models and Kalman Filter models. Through a number of simulations in cases of slow, fast acceleration and towing, it was found that the observers respond very efficiently, giving accurate estimates and successfully filtering out the noisy measurements of the sensors. It was also observed from the simulations that the results in the case of the controllable pitch propeller are smoother and more reliable due to its ability to change its step during changes in its speed. A linear model for determining propeller thrust through its estimated torque was examined. It has been found that for a certain range of values of the propeller angle of attack, this method can be applied with very satisfactory performance, but in cases of rapid changes in speed it deviates significantly from the actual values. In order to ensure the best possible control of the propulsion system, two models of predicting the future shaft speed were studied. It was concluded that the neural network gives accurate predictions in a time horizon of 10 seconds, with the mean square error converging quickly to zero. Data from real operating conditions of a tugboat were used to train the model.

Suggestions for Future Work

In this thesis observer models were designed ,using the ship's basic propulsion principles. The evaluation of the performance of these schemes was carried out through simulations. It is suggested to experimentally implement them and to investigate their response using the experimental testbed of LME. The simulation conditions and the noise covariances in the Kalman Filter observers could be alterated, in order to find the optimal ones and to approach effective realistic operating conditions. Regarding the neural network, its use is suggested, along with the observer so as to ensure high efficiency of the closed loop control system.

Appendices

Appendix A

Ship Propulsion Plant Modeling

In order to design the propeller load observer, it was necessary to select and model a ship type. In the previus diploma-thesis [4], a complete modeling of the ship was carried out and the present work was based on its content. Initially, in this chapter the type of ship being modeled is presented. The basic formulas that govern the principles of resistance and propulsion of the ship are described and its basic parts are analyzed, such as the main engine, the gearbox, the shaftline and the propeller. To perform the simulations, a model describing the state of the sea in case of irregular waves is also produced.

A.1 Ship Type

In order to dimension the propulsion system and make simulations, it was deemed necessary to select a type of ship that is similar to the main features of the experimental testbed (see Chapter **??**). Due to the small horsepower of the internal combastion engine of the experimental testbed and the hybrid PTI / PTO device, the tugboat was chosen as the type of ship, in which the most complex charging scenarios are presented. More specifically, the *Stan Tug 1205* tugboat, manufactured by *DAMEN*, was selected. This tugboat, shown in the following figure, has two internal combustion engines, of approximately the same power and speed as that of the experimental testbed. The main particulars of the ship are recorded in the table below.

Length overall	13.08 m
Beam overall	5.28 m
Depth at sides	2.30 m
Draught aft	1.80 m
Displacement	58 t
Max Speed	8.8 knots
Max Bollard Pull	8 t
Main Engine	Volvo D9 MH
MCR	221 kW at 1800 - 2200 RPM

Table A.1: Main Pariculars of DAMEN Stan Tug 1205

Figure A.1: DAMEN Stan Tug 1205

A.2 Ship Dynamics and Resistance

During the longitudinal movement of the ship, Newton's 2nd Law applies. Therefore, through the composition of the forces exerted on the ship, the acceleration is calculated. By integrating accelaration ship velocity can be also calculated.

$$\sum F = ma = m\frac{dV_s}{dt} \tag{A.2.1}$$

As a mass of the ship, its displacement is taken as well as its hydrodynamic mass, which for tugboats amounts to approximately 5% of the displacement. Therefore:

$$m = M_{disp} + M_{hyd} \tag{A.2.2}$$

According to [27]:

$$\frac{dV_s}{dt} = \frac{N_p(1-t)T_p - R + F_{B.P.}}{M_{disp} + M_{hyd}} \tag{A.2.3}$$

where:

- $N_p(1-t)T_p$
  with T_p the thrust produced by each propeller, N_p the number of propellers available on the vessel and t (thrust deduction factor) the rate of reduction of propulsion due to the hull propeller interaction in the stern area.
- R
  the resistance of the tugboat, which is a direct function of its speed. It consists of two parts, the calm water frictional resistance and the additional resistance of the ship due to the waves.
- $F_{B.P.}$
  (Bollard Pull) the external force which acts on the ship during tug operations.

A.3 Ship Propulsion

A.3.1 Diesel Engine

For the simulations, the data of the propulsion engine identified in a previous diploma-thesis [28] were used. In that work, a complete modeling of the main diesel engine was carried out. More specifically, the *Scania 12.7L I6* engine was modeled, with the following technical characteristics.

Maximum Torque	1950 Nm at 1200 RPM
MCR	254 kW at 1300 - 2200 RPM

Table A.2: Scania Engine Particulars

The engine torque Q_{eng} results from the regulation of the fuel supply, through a PI controller.

A.3.2 Shaftline Gearbox and Efficiency

GearBox

A gearbox is inserted between the engine and the propeller of the ship, in order to achieve the optimal speed reduction and to ensure efficient operation of the propeller. The gear ratio of the selected tugboat is:

$$GearRatio = i_{gb} = 3.82$$

Therefore, the speed of the propeller shaft after the gear box is:

$$n_{shaft} = \frac{n_{eng}}{i_{gb}} \tag{A.3.1a}$$

$$\omega_{shaft} = \frac{\omega_{eng}}{i_{gb}} \tag{A.3.1b}$$

The operating torque of the propeller shaft arises as a direct consequence of the principle of conservation of energy and is:

$$Q_{shaft} = i_{gb} Q_{eng} \tag{A.3.2}$$

Also a consequence of the principle of conservation of energy are the following relations between the moments of inertia before and after the gearbox.

$$\begin{aligned} \frac{1}{2} J_{shaft} \omega_{shaft}^2 &= \frac{1}{2} J_{eng} \omega_{eng}^2 \\ \frac{1}{2} J_{shaft} \frac{\omega_{eng}{}^2}{i_{gb}} &= \frac{1}{2} J_{eng} \omega_{eng}^2 \\ \frac{J_{shaft}}{i_{gb}^2} &= J_{eng} \\ J_{shaft} &= i_{gb}^2 J_{eng} \end{aligned} \tag{A.3.3}$$

Shaftline Efficiency

Gearbox Efficiency
The arrangement of the gearbox consists of bearings and other structural elements which come into contact with the shaft, causing friction losses and reducing its efficiency. A good estimate of the efficiency of the gearbox is given through the engine (or propeller) nominal torque, by the following formula.

$$n_{gb} = \frac{Q_{eng}}{Q_{nom}} = \frac{Q_{shaft}}{Q_{load}} \tag{A.3.4}$$

Shaft Efficiency
Power transmission from the main engine to the ship's propeller is via the shaft. The support of the shaft on bearings implies the development of local friction during its rotation, resulting in losses during power transmission and the reduction of the efficiency of the shaftline system. A widely held estimate of the shaft efficiency is as follows:

$$n_{sl} = 1 - a_2 n_{shaft} + \frac{a_1}{1 - a_2}(a_2 n_{shaft} - n_{shaft}^2) \tag{A.3.5a}$$

$$n_{sl} = 1 - a_2\omega_{shaft} + \frac{a_1}{1-a_2}(a_2\omega_{shaft} - \omega_{shaft}^2) \tag{A.3.5b}$$

Setting the parameter values as $a_1 = a_2 = 0.01$ the shaftline efficiency at maximum operating speed becomes $n_{sl} = 0.98$.

Relative Rotative Efficiency
The relative rotative efficiency n_r accounts for the differences in torque absorption characteristics of a propeller when operating at similar conditions in a mixed wake and open water flows. In most cases the value of relative rotative efficiency lies close to unity and is related to the actual torque of the propeller as follows:

$$Q_{load} = \frac{Q_{prop}}{n_r} \tag{A.3.6}$$

For the simulations, it was considered that the relative rotative efficiency is $n_r = 1.02$

A.3.3 Propeller

The propeller diameter of the selected tugboat is:

$$D = 1.05 \quad m \tag{A.3.7}$$

with

$$z = 4 \tag{A.3.8}$$

blades.

The ship's hull efficiency is determined by effective wake fraction w and thrust deduction t. The wake fraction is the actual percentage of the wake, due to the presence of the hull in front of the propellers and the thrust deduction is the rate of reduction of thrust due to the operation of the propeller in the stern area. The values of these coefficients are determined by empirical formulas [32] and are calculated as:

$$w = 0.15 \quad , \quad t = 0.13 \tag{A.3.9}$$

The propulsion velocity of the propeller is determined by the following type:

$$V_a = V_s(1-w) + Vw \tag{A.3.10}$$

where V_s is the velocity of the ship and w is the effective wake fraction, as defined above. V_w is the velocity of the wave that is presented as a disturbance in the propeller region of the ship and is analyzed in A.4.

The angle of attack of the propeller blade at 70% of propeller radius is denoted by β and is the angle formed by the recommended speed of V_a and 0.7 pin_pD seen from the

horizontal axis, which is perpendicular to the axis of rotation of the propeller.

$$\beta = \arctan\left(\frac{V_a}{0.7\pi n_p D}\right) \tag{A.3.11}$$

where V_a is the propulsion velocity of the propeller, n_p is the propeller rotational speed in rps and D is the propeller diameter.

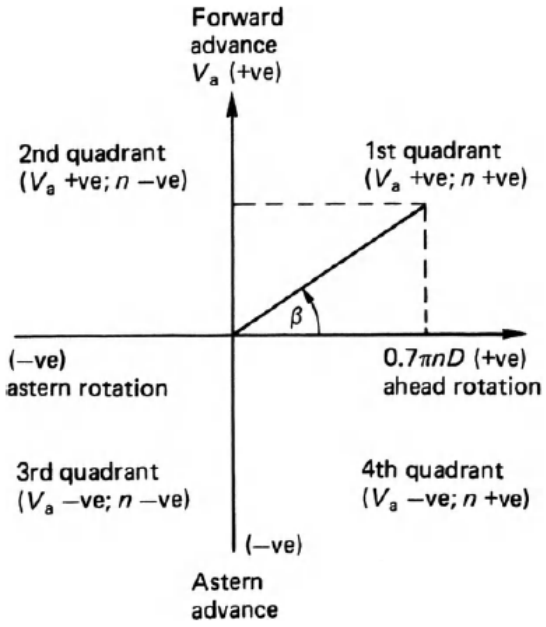

Figure A.2: Angle β

The actual propeller torque Q_p and thrust T_p are influenced by many parameters. As presented in [7] Q_p and T_p can in general be formulated as functions of the shaft speed n_p in revolutions-per-second (rps), time-varying states x_p (e.g. pitch ratio, advance velocity, submergence), and fixed thruster parameters θ_p (e.g. propeller diameter, geometry, position):

$$Q_p = f_Q(n_p, x_p, \theta_p) \tag{A.3.12a}$$

$$T_p = f_T(n_p, x_p, \theta_p) \tag{A.3.12b}$$

More specifically, propeller the torque and thrust are determined by the corresponding coefficients as follows.

$$Q_p = K_Q \rho D^5 n_p^2 \tag{A.3.13a}$$

$$T_p = K_T \rho D^4 n_p^2 \tag{A.3.13b}$$

The last two equations can also be written and functioned by the torque and thrust coefficients C_Q and C_T

$$Q_p = \frac{\pi}{8} C_Q \rho [V_a^2 + (0.7\pi n_p D)^2] D^3 \tag{A.3.14a}$$

$$T_p = \frac{\pi}{8} C_T \rho [V_a^2 + (0.7\pi n_p D)^2] D^2 \tag{A.3.14b}$$

where:

$$K_Q = \frac{\pi}{8} C_Q \left[\left(\frac{V_a}{n_p D} \right)^2 + (0.7\pi)^2 \right] \tag{A.3.15a}$$

$$K_T = \frac{\pi}{8} C_T \left[\left(\frac{V_a}{n_p D} \right)^2 + (0.7\pi)^2 \right] \tag{A.3.15b}$$

It is noted that the density of seawater is taken as follows:

$$\rho = 1025 \quad \frac{kg}{m^3}$$

The torque and thrust coefficients are determined by the following technical characteristics of the propeller.

- z: Number of Blades
- $\frac{P}{D}$: Pitch Ratio
- $\frac{A_E}{A_0}$: Expanded Area Ratio

Therefore, C_Q and C_T coefficients vary depending on the type of propeller. The propeller of the proposed tugboad is ducted with fixed pitch and nozzle, but simulations have also been performed for controllable pitch propeller. In each of the two cases, the torque and thrust coefficients are defined, using propellers of the Wageningen series.

Ducted Fixed Pitch Propeller

For this case

$$Wageningen \quad K_a 4-70, \quad with \quad Nozzle \quad 19A$$

was selected, with the following technical characteristics.

- $z = 4$
- $\dfrac{P}{D} = 1.2$
- $\dfrac{A_E}{A_0} = 0.7$

Fourier series were used to determine the torque and thrust coefficients, with the data given in. In this case the the angle of attack of the propeller blade β takes values in [-90°,270°].

$$\begin{aligned} C_Q &= \frac{1}{10}\sum_{k=0}^{20}\{A(k)\cos(k\beta)+B(k)\sin(k\beta)\} \\ C_T &= \sum_{k=0}^{20}\{A(k)\cos(k\beta)+B(k)\sin(k\beta)\} \\ C_{Tn} &= \sum_{k=0}^{20}\{A(k)\cos(k\beta)+B(k)\sin(k\beta)\} \end{aligned} \tag{A.3.16}$$

It is noted that $A(k)$ and $B(k)$ are different for C_Q, C_T and C_{Tn} and their values are obtained from table B.1 given in [3]. The coefficient C_{Tn} represents the part of the thrust produced by the propeller due to the existence of the nozzle. The factor C_T includes the factor C_{Tn} but both are calculated for completeness reasons. The dimensionless coefficients C_Q, C_T and C_{Tn} are displayed in Fig. A.3

The moment of inertia of the mass and the moment of inertia of the hydrodynamic mass of the propeller are respectively:

$$\begin{aligned} J_M &= 7.25 \quad kgm^2 \\ J_W &= 9.27 \quad kgm^2 \end{aligned}$$

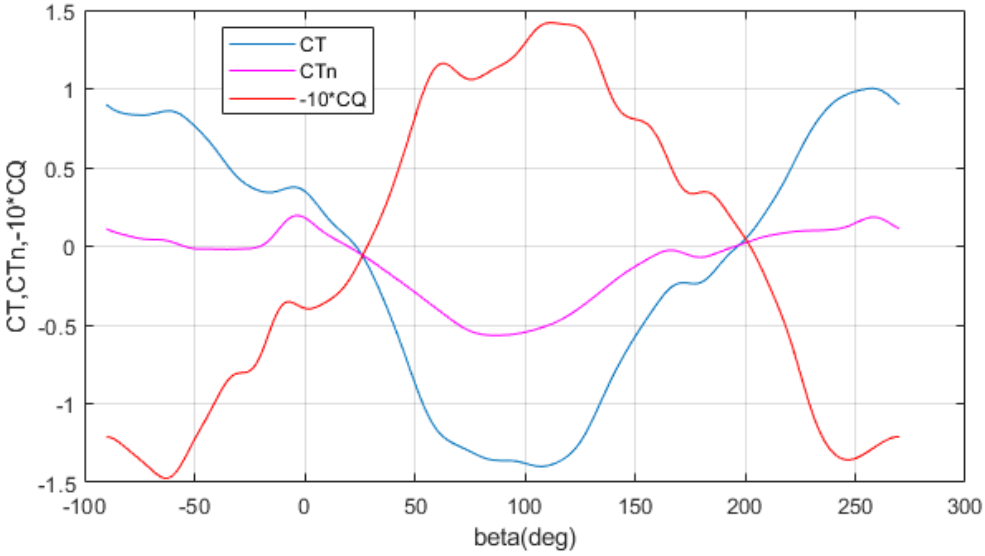

Figure A.3: Torque and Thrust Dimensionless Coefficients

Controllable Pitch Propeller

For this case

$$Wageningen \quad C4-40$$

was selected, with the following technical characteristics.

- $z = 4$
- $\left(\frac{P}{D}\right)_{max} = 1.2$
- $\frac{A_E}{A_0} = 0.4$

MC440 program of MARIN research intitution [1] were used to determine the torque and thrust coefficients. In this case the the angle of attack of the propeller blade β takes values in [-90°,90°] and the pitch ratio $\frac{P}{D}$ is not constant. In ship stopping or retreating, the propeller does not rotate in the opposite direction, but the pitch is changed to negative values, maintaining the propeller rotation at the same time. The dimensionless coefficients C_Q and C_T are displayed in 3D, depending on the angle β and the pitch ratio $\frac{P}{D}$, in the corresponding figures A.4 and A.5. These coefficients can also displayed in 2D, as seen in B.1 and B.2

The moment of inertia of the mass and the moment of inertia of the hydrodynamic mass of the propeller are respectively:

[1] https://www.marin.nl/

$$J_M = 6.11 \quad kgm^2$$
$$J_W = 6.24 \quad kgm^2$$

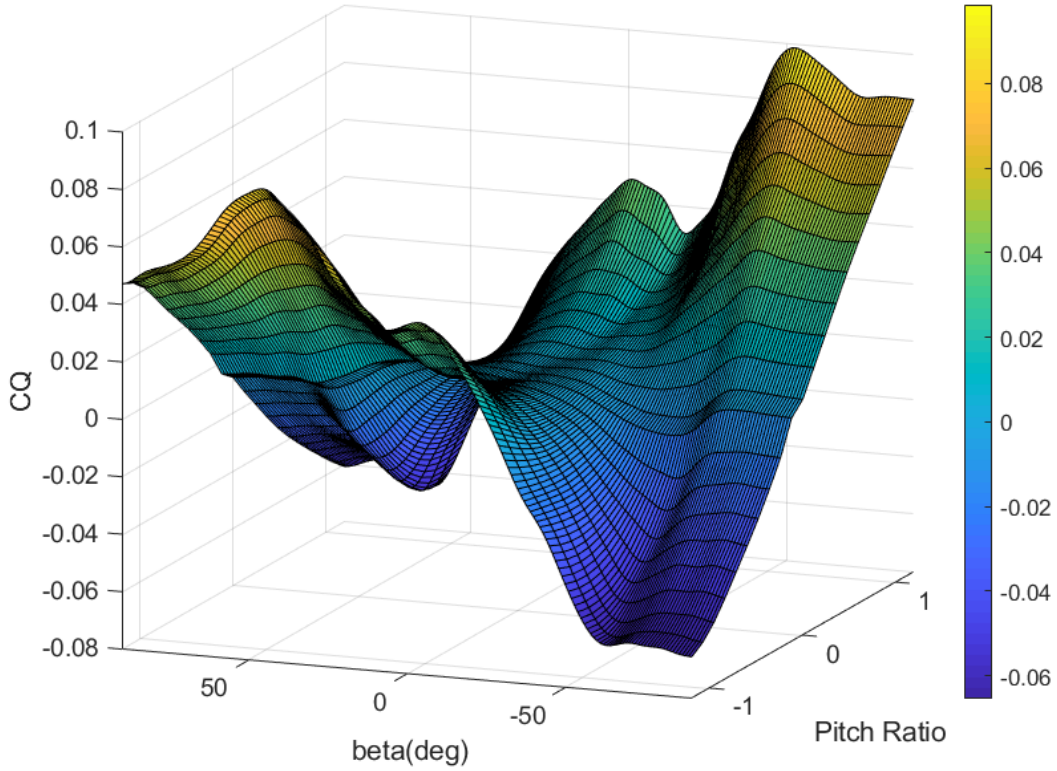

Figure A.4: Torque Coefficient in 3D display

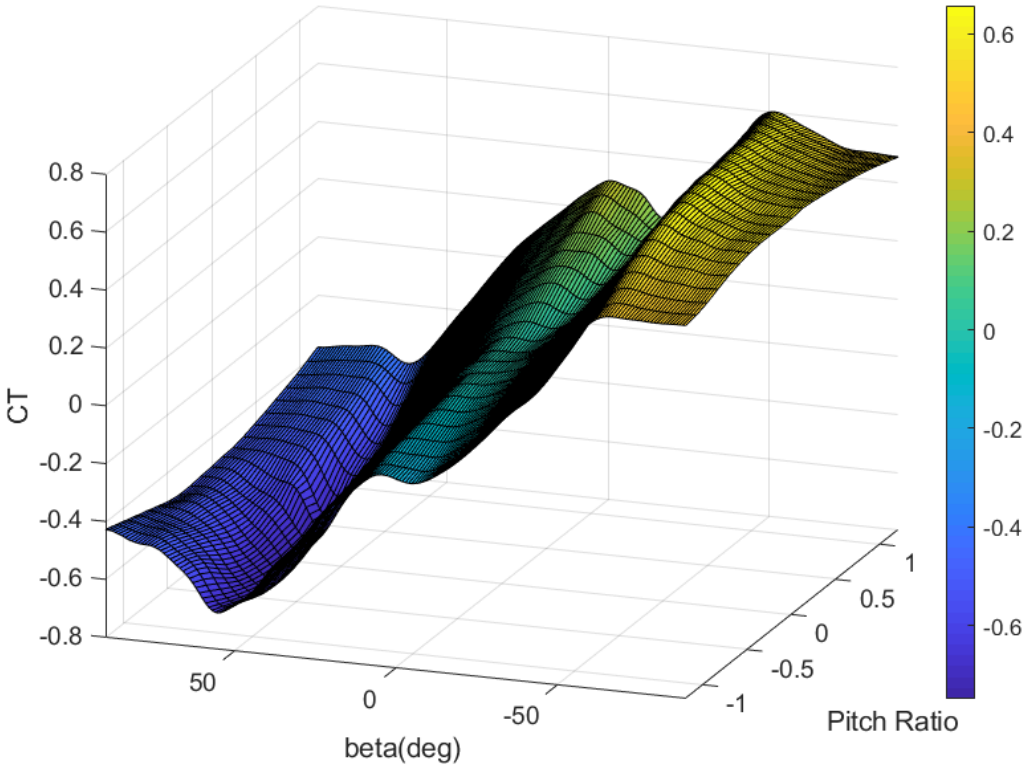

Figure A.5: Thrust Coefficient in 3D display

The selected propellers are capable of producing thrust that overcomes the resistance of the tugboat at maximum speed and the maximum bollard pull at lower operation speeds.

A.3.4 Shaftline Dynamics

During the rotational movement of the shaft, Newton's 2nd Law applies. Therefore, through the composition of the torques exerted on the shaft, the rotational speed is calculated.

$$\sum Q = J\dot{\omega} \tag{A.3.17}$$

Using the relations of the previous section and due to the presence of a gearbox, the above equation can be written as:

$$J_{shaft}\dot{\omega}_{shaft} = n_m Q_{eng} - Q_{load} \quad \Rightarrow \quad \dot{\omega}_{shaft} = \frac{1}{J_{shaft}}(n_m Q_{eng} - Q_{load}) \tag{A.3.18}$$

where:

- $n_m = n_{sl} n_{gb}$
  the mechanical efficiency of the gearbox and the shaftline system, which results as a product of the individual efficiencies n_{sl} and n_{gb}.
- J_{shaft}
  the moment of inertia of the system, which consists of:
  - the moment of inertia of the moving parts of the engine around the crankshaft, ie the pistons, the thrusters, the knobs, with which the crankshafts are connected to each other and finally the flywheel together with the flange at the engine outlet .
  - the moment of inertia of the shafts, the electric motor-generator and the moving parts of the gearbox.
  - the moment of inertia of the propeller, but also of the water trapped between its blades.
- Q_{eng}
  torque generated by the engine in the case of simulations results from the regulation of the fuel supply, as analyzed in A.3.1.
- Q_{load}
  the torque that represents the propeller load is estimated by the observer designed in Chapter 3 and is compared to its actual value Q_p, as calculated by the formulas described in the previous subsection.

A.4 Sea Wave Disturbance

In order to make simulations in case of wave disturbances, it was deemed necessary to develop a model for the production of irregular sea waves. The toolbox included in [1] was used for this purpose. The basic characteristics of the wave disturbance were thus defined ,according to [33].

The direction of the wave is defined by the angle β, which is 0° for the same direction of propagation speed of the wave, with that of the ship and 180° for the opposite direction (frontal waves). For the waves, it was assumed that they are one-directional, that is, all their components have the same direction. Irregular waves can be analyzed in infinite regular waves, having the following basic characteristics.

- H_s the significant wave height.
- ω_p the peak frequency.
- ζ_i the amplitude of each contributing regular wave i.
- ω_i the frequency of each contributing regular wave i.
- ϵ_i the phase of each contributing regular wave i.

It is noted that ζ_i and ω_i depend on H_s and ω_p, while ϵ_i results from a random distribution in $[0, 2\pi]$.

The water surface elevation is given by the following formula.

$$\zeta(t) = \sum_{i=1}^{n} \zeta_i \cos(\omega_i t + \epsilon_i) \tag{A.4.1}$$

n represents the number of regular waves, which are components of irregular waves, as said above. For the simulations, $n = 28$ was obtained.

It is noted that the moving ship "observs" the random wave differently from the stationary observer. For this reason, the meeting frequencies ω_{e_i} must be calculated. Also assuming that the sea water is deep, the wave number is:

$$k_i = \frac{\omega_i^2}{g} \tag{A.4.2}$$

Therefore, the meeting frequencies are calculated as follows.

$$\omega_{e_i} = \omega_i - k_i V_{ship} \cos(\beta) \quad \Rightarrow \quad \omega_{e_i} = \omega_i - \frac{\omega_i^2}{g} V_{ship} \cos(\beta) \tag{A.4.3}$$

Since the propeller is submerged below the sea surface, the effect of each regular wave must be multiplied by the condition $e^{-h_{prop} k_i}$, where $h_{prop} = 1.4m$ is the depth at which the center of the propeller washer is immersed in the sinking state. So the wave disturbance at that depth is:

$$\zeta(t) = \sum_{i=1}^{n} \zeta_i \cos(\omega_{e_i} t + \epsilon_i) e^{-h_{prop} k_i} \tag{A.4.4}$$

By differentiating the above equation, the velocity of the water due to the wave is obtained, in the area of the propeller, regardless of the wave-ship interaction.

$$\dot{\zeta}(t) = \sum_{i=1}^{n} \zeta_i \omega_{e_i} \sin(\omega_{e_i} t + \epsilon_i) e^{-h_{prop} k_i} \tag{A.4.5}$$

In the above equation − ensures that an incidental (frontal) wave, with a propagation speed opposite that of the ship, causes an increase in the speed of progress, while a wave in the same direction as that of the ship causes a decrease.

$$v_w(t) = -\dot{\zeta}(t) \cos(\beta) \tag{A.4.6}$$

The added resistance is calculated using experimental or simulation data for the ship seakeeping response at each frequency that contributes to the irregular wave, as shown in **??**. As such

$$R_w(t) = \sum_{i=1}^{N} R_{wd,i}(V_{ship}, \zeta_i, \frac{g}{\omega_i}) \tag{A.4.7}$$

The characteristics of the regular waves that contribute to the irregular wave formation are selected based on a specific wave spectrum, where the sea state is defined from the significant wave height H_s (mean value of the 1/3 highest free surface elevation observations) and the peak frequency ω_p, which is the wave frequency at which occurs the maximum spectral density.

The peak frequency ,for waves in the North Sea, is calculated as follows:

$$\omega_p = \frac{2\pi}{4.883 + 2.68 H_s^{0.54}} \tag{A.4.8}$$

The simulation of the sea state ,with waves, is displayed in 3D, in Fig. A.4

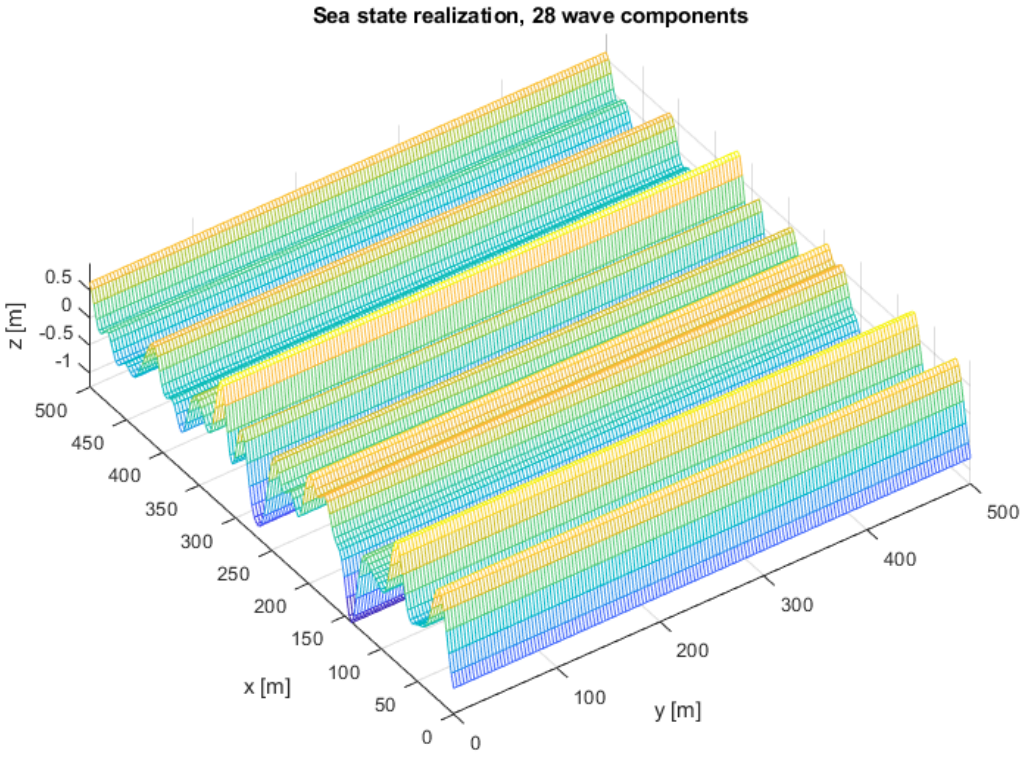

Figure A.6: Water free surface elevation with one-directional irregular waves, $\beta = 180°$, generated with [1] toolbox

Appendix B

Propeller Torque and Thrust Coefficients Calculation Data

The following table shows the hydrostatic data used to calculate the torque and thrust coefficients for the corresponding case of ducted propeller. In the case of the controllable pitch propeller, the illustrations of C_Q and C_T in 2D are also attached, as a supplement to the corresponding 3Ds presented in A.4 and A.5.

k	T-A(k)	T-B(k)	Tn-A(k)	Tn-B(k)	Q-A(k)	Q-B(k)
0	-9.0888E-02	0.0000E+00	-1.0166E-01	0.0000E+00	4.3800E-02	0.0000E+00
1	1.7959E-01	-1.1026E+00	1.8593E-02	-2.7769E-01	3.5299E-01	-1.2949E+00
2	1.4956E-01	6.1459E-02	1.3408E-01	3.5459E-02	-1.0917E-02	5.9030E-02
3	6.5675E-02	1.3715E-01	4.3767E-02	7.2317E-02	4.7062E-02	9.3540E-02
4	5.2107E-03	-1.7280E-02	1.3604E-02	-8.3408E-03	-1.0779E-02	-6.1148E-03
5	-6.8232E-03	9.6579E-02	1.8658E-02	4.4854E-03	-1.0193E-02	1.6121E-01
6	-6.2896E-03	5.8809E-03	2.6598E-03	-3.7642E-03	-8.8824E-04	1.4624E-02
7	1.8178E-02	-2.2587E-02	2.4907E-02	7.5727E-04	-3.7893E-02	-5.3549E-02
8	6.0694E-03	-1.4819E-02	4.7924E-03	-8.8802E-03	-7.0346E-03	-3.1589E-03
9	6.1942E-03	-1.0398E-02	3.6556E-03	4.0541E-04	-8.0130E-03	1.4382E-02
10	2.6482E-03	-2.9324E-03	3.9850E-03	-1.2811E-03	7.2622E-03	9.9836E-03
11	1.2137E-02	4.0913E-03	1.0643E-02	-5.5230E-03	-5.4390E-03	3.8781E-02
12	-3.5705E-03	-4.4436E-03	2.5495E-04	-6.3566E-03	-2.0068E-03	-4.6749E-03
13	3.2985E-03	-1.2190E-03	2.9347E-03	-2.5338E-03	3.9281E-03	1.4944E-02
14	-8.8652E-04	-2.2551E-03	3.6599E-04	-2.0504E-03	-6.5256E-04	-6.3253E-03
15	6.9807E-03	-3.2272E-03	1.3115E-03	-3.8485E-03	1.5414E-02	2.2275E-03
16	-1.7560E-04	1.7533E-03	-1.3511E-03	-6.3908E-04	3.0356E-03	7.1826E-03
17	2.1643E-03	1.4875E-03	1.7101E-03	-1.0819E-03	5.9073E-03	1.0229E-03
18	3.5362E-04	4.5353E-05	3.3765E-04	-9.6321E-04	4.1433E-03	-5.9201E-03
19	2.5772E-03	-8.8702E-04	-3.9681E-04	-2.0969E-03	4.6102E-03	-1.4814E-03
20	-1.8279E-03	-9.4609E-04	-1.1814E-03	-1.9298E-04	-5.7423E-04	-4.3092E-03

Table B.1: $A(k)$ and $B(k)$ for the calculations of C_T, C_{Tn} and C_Q of $Wageningen \quad K_a 4-70$, with $\frac{P}{D} = 1.2$

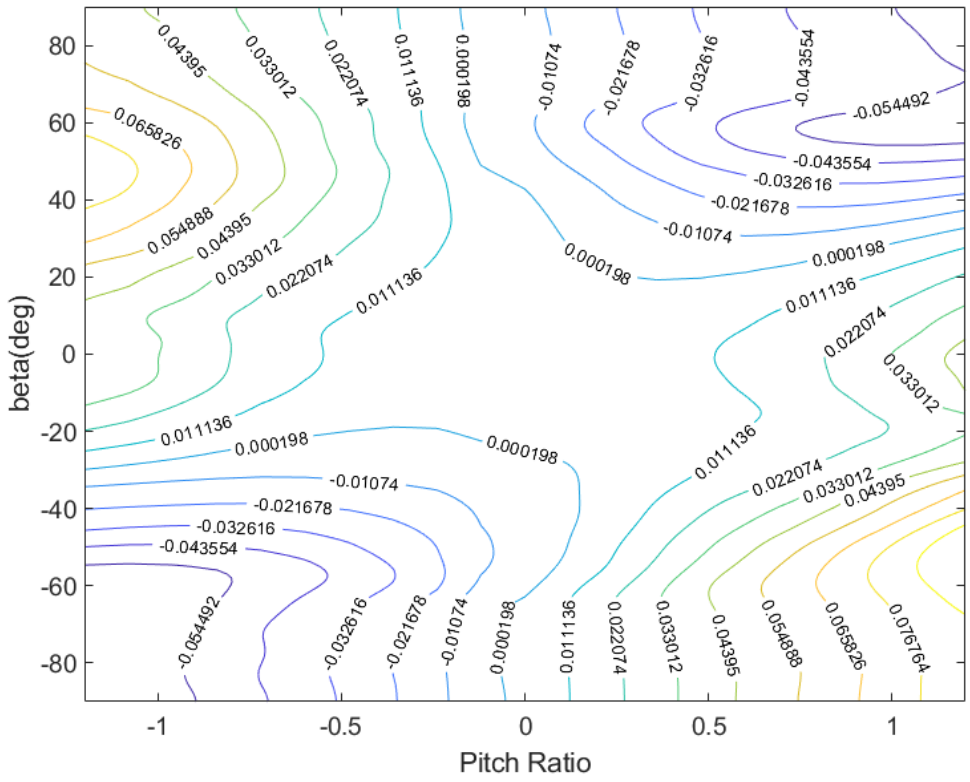

Figure B.1: Torque Coefficient in 2D display

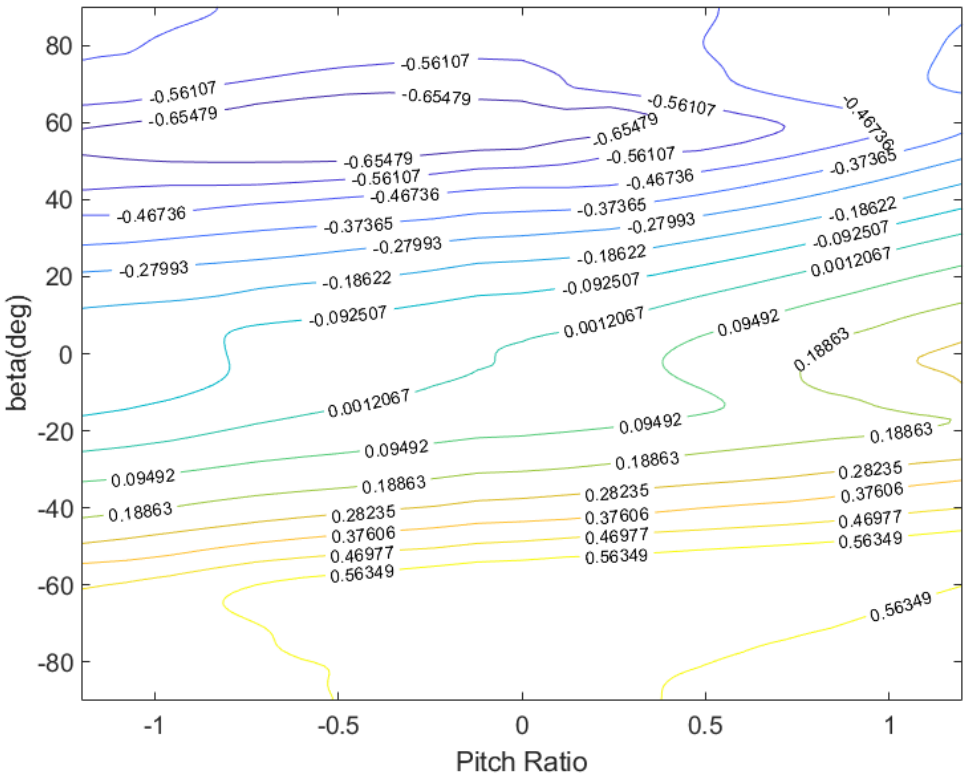

Figure B.2: Thrust Coefficient in 2D display

Bibliography

[1] T. Perez, O. Smogeli, T. Fossen, and A. Sørensen, "An overview of the marine systems simulator (mss): A simulink toolbox for marine control systems," *Modeling, Identification and Control*, vol. 27, 2006.

[2] H. K. Woud and D. Stapersma, *Design of Propulsion and Electric Power Generation Systems*. London: IMarEST, 2002.

[3] J. Carlton, "Marine propellers and propulsion (third edition)," Oxford: Butterworth-Heinemann, 2012.

[4] D. Vasrsamis, *Load Simulation during ship propulasion and application in experimental marine hybrid Diesel-Electric testbed*. Zografou, Athens: NTUA, 2019.

[5] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley and Sons, Ltd, 2011.

[6] L. Pivano, T. Johansen, and O. Smogeli, "A four-quadrant thrust estimation scheme for marine propellers: Theory and experiments," *Control Systems Technology, IEEE Transactions on*, vol. 17, pp. 215 – 226, 2009.

[7] O. Smogeli, *Control of Marine Propellers: from Normal to Extreme Conditions*. PhD thesis, 2006.

[8] R. Kleinbauer, *Kalman Filtering Implementation with Matlab*. Helsinki: Stuttgard University, 2004.

[9] D. Simon, *Optimal State Estimation*. John Wiley and Sons, Ltd, 2006.

[10] S. D. Cairano, D. Bernardini, A. Bemporad, and I. V. Kolmanovsky, "Stochastic mpc with learning for driver-predictive vehicle control and its application to hev energy management," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1018–1031, 2014.

[11] K. Liu, Z. Asher, X. Gong, M. Huang, and I. Kolmanovsky, "Vehicle velocity prediction and energy management strategy part 1: Deterministic and stochastic vehicle velocity prediction using machine learning," 2019.

[12] T. Gaikwad, Z. Asher, K. Liu, M. Huang, and I. Kolmanovsky, "Vehicle velocity prediction and energy management strategy part 2: Integration of machine learning vehicle velocity prediction with optimal energy management to improve fuel economy," 2019.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[14] T. Mitchell, *Machine Learning.* McGraw-Hill International Editions, McGraw-Hill, 1997.

[15] G. Ellis, "Observers in control systems," San Diego: Academic Press, 2002.

[16] G. Papalambrou, *Introduction to Control Systems, Lecture Notes.* Zografou, Athens: NTUA, 2011.

[17] K. Ogata, *Modern Control Engineering.* USA: Prentice Hall PTR, 4th ed., 2001.

[18] D. Luenberger, "An introduction to observers," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 596–602, 1971.

[19] N. L. Ricker, "Model predictive control with state estimation," *Industrial and Engineering Chemistry Research*, vol. 29, no. 3, pp. 374–382, 1990.

[20] W. L. Brogan, *Modern Control Theory (3rd Ed.).* USA: Prentice-Hall, Inc., 1991.

[21] P. Chalupa, P. Januška, and J. Novak, "State observers for model predictive control," vol. 210, 2013.

[22] S. Li, J. Yang, W. Chen, and X. Chen, "Generalized extended state observer based control for systems with mismatched uncertainties," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4792–4802, 2012.

[23] G. Welch and G. Bishop, "An introduction to the kalman filter," tech. rep., USA, 2001.

[24] R. D'Andrea, *Recursive Estimation, Online Lecture Notes.* Department of Mechanical and Process Engineering: ETH Institute for Dynamic Systems and Control-https://idsc.ethz.ch/education/lectures/, 2020.

[25] L. Tamas, G. Lazea, R. Robotin, C. Marcu, S. Herle, and Z. Szekely, "State estimation based on kalman filtering techniques in navigation," in *2008 IEEE International Conference on Automation, Quality and Testing, Robotics*, vol. 2, pp. 147–152, 2008.

[26] F. Daum, "Nonlinear filters: beyond the kalman filter," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57–69, 2005.

[27] D. Stapersma and A. Vrijdag, "Linearisation of a ship propulsion system model," *Ocean Engineering*, vol. 142, pp. 441–457, 2017.

[28] V. Karystinos, *Nonlinear Model Predictive Control of a Hybrid Diesel-Electric Marine Propulsion Plant.* Zografou, Athens: NTUA, 2019.

[29] R. Geertsma, K. Visser, and R. Negenborn, "Adaptive pitch control for ships with diesel mechanical and hybrid propulsion," *Applied Energy*, vol. 228, pp. 2490 – 2509, 2018.

[30] R. Geertsma, R. Negenborn, K. Visser, M. Loonstijn, and J. Hopman, "Pitch control for ships with diesel mechanical and hybrid propulsion: Modelling, validation and performance quantification," *Applied Energy*, vol. 206, pp. 1609 – 1631, 2017.

[31] N. P. Kyrtatos and N. T. Alexandrakis, *Marine Propulsion Plants, Online Lecture Notes.* Zografou, Athens: NTUA-http://www.lme.ntua.gr:8080, 2020.

[32] G. K. Politis, *Ship Resistance and Propulsion, Lecture Notes.* Zografou, Athens: NTUA, 2016.

[33] G. A. Athanasoulis, *Ship Dynamics.* Zografou, Athens: NTUA, 2008.

[34] C. Guibert, E. Foulon, A.-A. Nadia, and L. Loron, "Thrust control of electric marine thrusters," 2005.

[35] S. Haykin, *Neural Networks: A Comprehensive Foundation.* USA: Prentice Hall PTR, 2nd ed., 1998.

[36] H. Demuth and M. Beale, *Neural Network Toolbox For Use with MATLAB.* The MathWorks.

[37] R. E. Uhrig, "Introduction to artificial neural networks," in *Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics*, vol. 1, pp. 33–37 vol.1, 1995.

[38] G. Papalambrou and N. Planakis, *Ship Control Systems, Online Lecture Notes.* Zografou, Athens: NTUA-http://www.lme.ntua.gr:8080, 2020.

[39] J. Dabney and T. Harman, "Mastering simulink," 1998.

[40] M. Ulusoy, *MathWorks.* https://www.mathworks.com.