NTNU

Norwegian University of
Science and Technology

Dynamic Positioning by Nonlinear Model Predictive Control

Åsmund Våge Fannemel

Master of Science in Engineering Cybernetics

Submission date: June 2008

Supervisor: Morten Breivik, ITK

Co-supervisor: Jann Peter Strand, Rolls Royce Marine AS

Ivar-André Ihle, Rolls Royce Marine AS

Norwegian University of Science and Technology

Department of Engineering Cybernetics

Problem Description

The candidate will consider the problem of dynamic positioning (DP) of marine surface vessels by unscented Kalman filtering (UKF) and nonlinear model predictive control (NMPC). The following elements must be considered:

1. Review the literature on hydrodynamic damping and parameter estimation, and estimate the parameters of a nonlinear damping model using optimization tools and UKF methods.
2. Compare the UKF with the more traditional extended Kalman filter (EKF).
3. Review the literature on NMPC, especially with regard to dynamic positioning applications.
4. Develop a DP controller based on NMPC theory and integrate it with a UKF for estimation of currents and wave-induced motion.
5. Implement, simulate, and evaluate the suggested control scheme in Matlab for a number of relevant 3 DOF DP scenarios. Compare the simulation results with those associated with a traditional DP controller.

Assignment given: 14. January 2008
Supervisor: Morten Breivik, ITK

Preface

> Easy reading is damn hard writing
>
> Nathaniel Hawthorne

I would like to thank my supervisor Morten Brevik and co-advisor Ivar Ihle for their input to this thesis. I would also like to thank everyone up at the gallery for a splendid year of studies and chitchat. My approach to studying has always been to learn as much as possible and I must admit I have done so during the work with this thesis as well. I would like to end my studies here by summing it up with these two quotes:

> "If I have seen a little further, it is by standing on the shoulders of giants." - Isaac Newton

> "Curiosity is only in vanity. Most frequently we wish not to know, but to talk. We would not take a sea voyage for the sole pleasure of seeing without hope of ever telling." - Blaise Pascal

Abstract

This thesis discusses the theoretical aspects of the unscented Kalman filter (UKF) and nonlinear model predictive control (NMPC) and try to evaluate their practical value in a dynamic positioning (DP) system. A nonlinear horizontal vessel model is used as the basis for performing state, disturbance, and parameter estimation, and attempts at controling the vessel using NMPC are made. It is shown that the extended Kalman filter (EKF), which is much used in various navigation applications including DP, is outperformed both theoretically and practically in simulations by the UKF. Much of which is due to the UKF's improved approximation of the estimated system's true stochastic properties. An attempt to estimate the current from the hydrodynamical damping forces have been applied and shown to be working when the vessel is not subjected to other slowly-varying drift forces. It is implemented a dual estimation approach to try to estimate hydrodynamic damping, which is a very real problem for actual vessels and DP systems.

A theoretical evaluation of NMPC is performed and it is concluded that NMPC schemes could fulfill a need in vessel control and DP. Its combination of model based control, optimization approach to achieving performance and predictive properties are indeed useful also for DP. It is found that NMPC could be a step towards a unified control approach combining low and high speed reference tracking, station-keeping and several other control operations which today are handled by separate control approaches. NMPC provides the control designer with an exceptional amount of freedom when quantifying the performance, that it is impossible not to find some use for NMPC.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> So we beat on, boats against the current, borne back ceaselessly into the past.
>
> F. Scott Fitzgerald

1.1 Dynamic Positioning Fundamentals

1.1.1 Background

The International Marine Contractors Association defines a *Dynamic Positioning* (DP) system as "a system which automatically controls a vessel's position and heading exclusively by means of active thrust" [22]. DP systems was developed to keep a vessel at a fixed position with fixed heading, an operation known as station keeping.

The first DP system was installed on the drilling ship Eureka in 1961[] and consisted of an analogue control system receiving position reference from a taut wire. Prior to this, station-keeping operations had been performed with the help of anchor spreads and jack-up barges. The introduction of DP capable vessels made station keeping possible at greater depths and with greatly simplified setup compared with the existing jack-up barge and anchoring approaches. By not depending on fixed mechanical installations or anchors, the vessel has excellent maneuverability and can be positioned at locations where anchoring are not possible/desired, e.g., coral reefs. All three systems have their advantages and disadvantages, but the introduction of DP systems increased the number of possible operations significantly. The disadvantages of DP systems includes the risks of system failures, high fuel

costs, and underwater hazards for divers and remotely operated vehicles due to thrust use. It can be argued that DP systems will overcome some of its disadvantages as the technology is further developed.

DP systems are now increasingly used for accurate low-speed trajectory following, assisted anchoring, and weathervaning as well as station keeping, and to perform these operations, a triplet of actions must be performed; an accurate estimate of the vessel's position and heading must be kept, the forces and moments needed to counteract the environmental disturbances and position deviation must be calculated, and these forces must be applied by the vessel's propulsion units. It is common to divide the functionality of a typical DP system into several sub-systems, including:

- Operator system
- Position and heading reference systems
- Estimators, including signal processing
- Guidance, navigation and control units
- Thrust allocators
- Propulsion units including power generators

where it is said that the DP-system is only as good as it's weakest link. Figure 1.1 shows how these systems are interconnected. Many commercial operators deliver complete DP-systems and the largest by far is Kongsberg Maritime [65], with several other actors including Rolls Royce Marine, Converteam, L-3 Communications, Marine Technologies and Navis Engineering Oy. These commercial systems offer several operational modes such as manual operator control, station keeping, mixed manual and automatic control, low-speed-tracking, and target following. Commercial DP-systems are subjected to several safety and reliability requirements which are enforced by international certification societies such as Det Norske Veritas, Lloyd's Register of Shipping, and The American Bureau of Shipping.

1.1.2 Development of Estimation and Control Systems for DP

The first automatic ship steering mechanism was constructed by Elmer Sperry in 1911 [13]. The first DP system was implemented 50 years later on a drilling ship. Two important reasons for the long development time was the need for a position reference system and that the vessel had to be capable of applying individual forces and moments in the different *degrees*

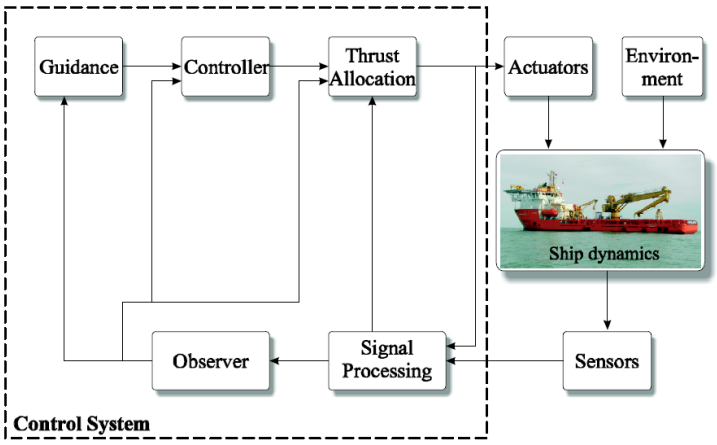

Figure 1.1: Figure depicting control part of DP system. Courtesy of [36].

of freedom (DOF). Vessels that are capable of applying individual forces and moments in all DOFs are known as fully-actuated, Figure 1.2 shows the thruster configuration of a fully-actuated vessel.

The first DP systems made use of linear decoupled PID-controllers in each DOF in cascade with simple low-pass and notch-filtered position measurements. The filters were used to remove the rapid 1. order wave induced motion from the feedback loop, but their averaging effect induced a phase lag into the closed-loop feedback system. The performance was further halted as important coupled effects were ignored.

The second age of DP-control came in the 1970s when model-based LQG-control was implemented. This was suggested in [2] and further developed in cite grimble and several other articles by the same authors. These model based approaches were computationally heavy, but allowed for a more coupled approach as well as an improved separation of low-frequency vessel motion and the high-frequency wave induced motion. They also calculated a slowly-varying bias force that could be used for feed-forward control. The initial implementations used gain-scheduling approaches and later on the extended kalman filter. These aoproaches are still found in use in commercial DP-systems today.

Further research in the 1990s and 2000s lead to the use of nonlinear controllers and estimators which improved the handling of the inherent nonlinear characteristics of the vessel dynamics. Notable contributions include [60] and [63].

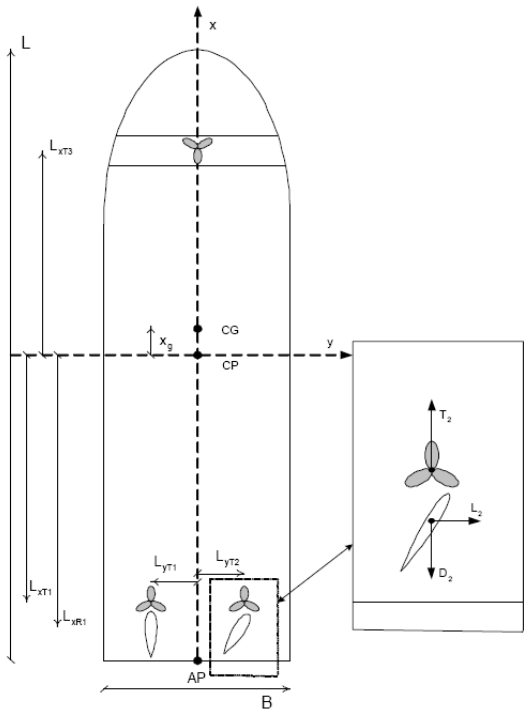

Figure 1.2: Illustration depicting the actuator configuration of Cybership II. Courtesy of [57].

1.1.3 State, Parameter and Disturbance Estimation for DP

The DP system needs measurements of heading and position. The simplest way to obtain these are from the Global Positioning System and gyrocompasses, but any accurate position reference system will do. Measurements of the vessel motion and acceleration can be obtained from inertial navigation systems, but are often not available due to cost issues. An estimate of the vessel's velocity must therefore be obtained using the available measurements and a model of the vessels dynamics. The model should try to accomodate for the external disturbance forces as well. It is obvious that more accurate modeling, and optimal use of the model, will improve the estimates of vessel velocity and disturbances. The modeling problem consists of finding a model that approximates the complex vessel mechanics, and the even more complex systems describing the environmental disturbances, to a satisfactory degree. In an actual implementation, the estimator may also perform sensor integration, i.e., the task of combining measurements from different sensors, performing fault-detection, and error handling. If measurements are lost, the model-based state estimator is also responsible

for keeping an estimate, using only the mathematical model, i.e., dead-reckoning.

It is also desired that the estimator obtaining estimates of the disturbance forces affecting the vessel due to the disturbing sea, wind, and current effects. The wind force and direction is often simply measured and a look-up table is used to approximate the forces due to wind disturbance. The forces due to waves and currents are a bit more difficult to obtain, firstly there are no available measurement systems for obtaining these, nor are they easily separated effects. They are therefore often lumped together and rather separated into a *low-frequency* (LF) component and a *high-frequency* (HF) component. The HF component is mostly due to the oscillating wave induced forces and is often denoted *wave-frequency* (WF) , while the LF components appear as drift forces. Figure 1.3 shows how the motion from the LF component and the WF component are assumed to be superpositioned.

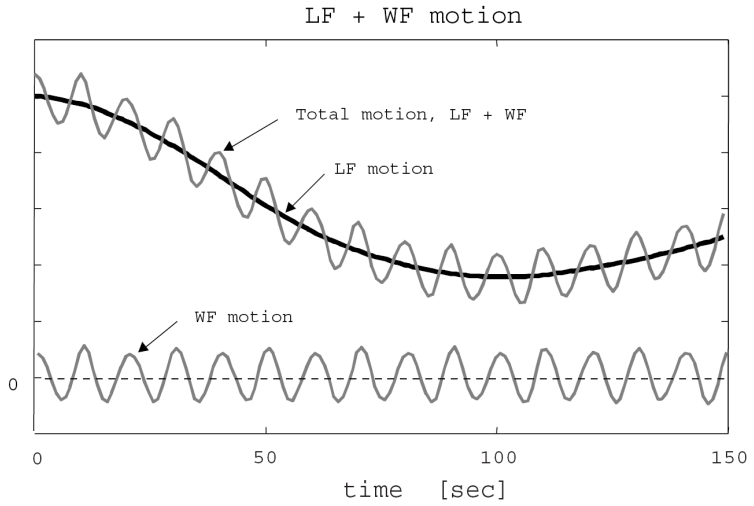

Figure 1.3: Illustration depicting LF and WF motion. Courtesy of [15].

The motion induced by the WF components are typically substantially faster than the vessel's bandwidth, and it is accordingly not able to counteract them. Trying to counteract them would only lead to unnecessary strain on the actuator system and wasted energy since no increased performance can be achieved. The motion estimates are therefore subjected to wave-filtering to remove the WF components. In the earliest DP implementations wave-filtering was performed with the use of low-pass and notch filters, but the induced phase-lag lead to sub-optimal performance. An implementation using optimal filtering methods, i.e., the Kalman filter, was proposed in [2]. The first applications made use of gain-scheduled linear approximations about the yaw axes where the Kalman gain was dependent on the head-

ing. These approaches were difficult to implement since there were so many tuning parameters. Later implementations made use of a extended Kalman filter and thus drastically reducing the number of tuning parameters and included a framework for performing parameter estimation, see for instance Grimble [16]. Recent approaches by Strand and Fossen includes nonlinear passive observers that are simpler to tune [60]. The trend have been to develop observers that were globally and thus decreased the number of tuning parameters. Adaptive estimators estimating vessel parameters and gains for varying sea-states are also proposed [63].

1.1.4 Guidance, Navigation and Control Units

The guidance and navigation units are responsible for creating set-points for the controller. For DP, this task entails setting simple position references, but also path-generation when changing set-points and when the set-point is moving. Such path generation is done using a model reference generator to allow the vessel to keep close to the set-points and not lag at distances that can lead to poor control performance.

The focus here is on the control unit, i.e., the computer program that use measurements, estimates, and setpoints to calculate the necessary thrust-force to perform the necessary maneuvers. The controller consists of a PID controller or a model-based controller, combined with feedforward from estimates of disturbing forces and/or a pathfeedforward. The thrust is most often calculated as forces and moments in the various DOFs and the task of allocating these to the actual thruster system is performed by a separate thrust allocator.

The first DP system consisted of 3 independent, analog linear feedback controllers, one for each DOF. This decoupled approach suffered from neglected coupled effects and phase lag from the filtered feedback error. The LQG approach in the 1970s made way for model-based controllers which inherently considered the coupled effects. Modern approaches include acceleration feedback [36], nonlinear controllers, and coupled PID controllers combined with feedforward terms. A recent commercial system called GreenDP developed by Kongsberg [20], makes use of model predictive control to allow constraint handling and reduced thrust use for operations where high-precision position accuracy is not necessary.

1.1.5 Vessel Models for DP

DP entails low-speed motion, which is a control problem that fits very nicely in the linearization framework. Low velocities means that linear hydrody-

namic effects will dominate higher order effects and the coriolis and centrifugal effects will be negligible. Figure 1.4 shows a common separation of velocity ranges for vessel control. If the heading angle is reasonably fixed, i.e., held within a 10 degree range and, for the duration of the operation, there isn't even a need for considering the vessels heading relative to the earth. Operations such as station keeping are simple set-point regulation and disturbance rejection problems.

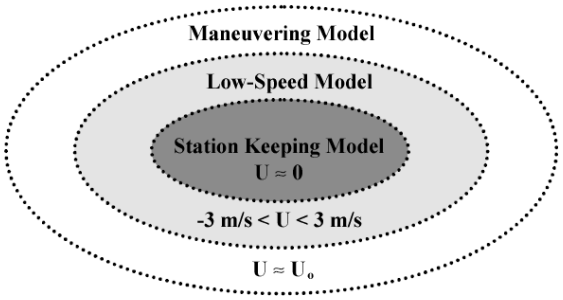

Figure 1.4: The figure shows a common separation of velocity ranges for vessel control. Courtesy of [60].

Yet, as computing power has grown manyfold and improved system identification methods have appeared, there is no good reason for not developing vessel models that are valid for larger velocity ranges by including more complex hydrodynamical damping effects. More accurate modeling can lead to large performance improvements during more complex operations through improved estimation and control calculation.

1.2 Motivation

DP is a mature field and good results are achieved with existing technology, but new marine operations appear and with them the demands on control strategies evolve. PID control, LQR and other feedback controllers seeks to control the position and heading in a bulls-eye fashion and always keep the deviations as small as possible. This aggressive control is not always necessary and reduced thrust force may be valued over position accuracy during some operations. Examples of operations where reducing thrust use can be more important than position accuracy include waiting operations

where anchoring is not an option and to some degree operations as cable-laying and trench digging at great depths, where a position accuracy within 5 percent of depth is required [21].

Kongsberg Maritime has implemented GreenDP, which seeks to reduce the fuel consumption and remove power surges. It divides the control problem into two parts; compensation of the slow drift due to environment forces and then handles other disturbances only when absolutely necessary, instead of, as direct feedback controllers do, acting instantaneously on all disturbances. This leads to a calmer set of control forces and less control peaking. Prediction can also be used for tracking purposes, as the controller can predict that it will need to change the outputs for a turn, or change in set-point. The prediction horizon offers perspective and allows for elements of feed-forward control. This detuning can obviously be achieved using PID-control or LQR as well, but model predictive control offers constraint handling as well, meaning that the vessel can be commanded to use minimize the thrust use, yet still be demanded to stay within a given region. Other constraints can be applied to limit thrust use or thrust change.

Marine control has traditionally been divided into low-speed control, i.e. DP, and high-speed control, i.e., autopilots. This division occurs mainly because of the use of simplified models. By developing a vessel model that is valid for both low- and high-speed maneuvers and include effects due to environmental disturbances, a unified control approach could be made [5]. As this model will be used to predict future behaviour, infer knowlegde about states that are not measured and be used by the controller it is important that the model portrays the dynamics of the system in an adequately accurate manner. It is regrettably impossible to obtain perfect models so a trade-off between identification cost and model accuracy is performed, developing complex models are futile if we are not able to identify the empirical parameters and are clueless about the state of the environmental disturbances.

The environmental disturbances are usually estimated using a triple approach. Wind forces are found by from look-up tables depending on the wind direction, speed and the vessel shape and load. Current and wave effects are typically lumped together and the WF component is separated from the estimates using wave filtering. The slowly varying forces are estimated as a general force bias which also includes other unmodeled effects including unmodeled dynamics. Unmodeled thrust dynamics and poor modeling of hydrodynamical damping will contribute heavily to this force bias [26]. It is known that DP operators wish for a separate estimate of the current velocities as well, and an estimate of those will also lead to improved estimations of the hydrodynamic damping.

It is known that the extended Kalman filter can be troublesome to tune,

and suffer from a lack of robustness. A newly developed alternative named unscented Kalman filtering could be implemented to improve the quality of the estimates.

1.3 Previous Work

The commercial operator Kongsberg has a product called greenDP that includes an extended kalman filter and a nonlinear model predictive controller. Unscented Kalman filter is gaining increased usage as a state estimator for problems where extended Kalman filter has been the selected state estimator. The author knows of two ukf applications in vessel estimation [47], [46]. In addition there are several implementations of ukf used for navigation purposes and sensor integration, see for instance [64] and [12].

A recent PhD Thesis by J. Refsnes [55], focuses on obtaining estimates of current and tracking control of a submersible slender body. His approach is done using nonlinear controllers and nonlinear luenberger observers.

1.4 Contribution

1.5 Thesis Outline

- Chapter 1 is the introduction. Describes the motivation, previous relevant work , the contribution of this thesis and details the applied notation.
- Chapter 2 describes the marine vessel model
- Chapter 3 is a short introduction to Unscented Kalman Filter and details on how model parameters and state variables are estimated.
- Chapter 4 details Model Predictive Control and discusses my implementation of a NMPC controller.
- Chapter 5 containts simulations and results from my application of a UKF state estimator and a NMPC based DP controller
- Chapter 6 discusses the results.
- Chapter 7 contains the conclusion and relevant future work possibilities.
- Appendix A is a short introduction to stability theory.

- Appendix B contains some background material on classical mechanics.
- Appendix C is a primer on statistics needed to develop the UKF.

1.6 Notation

1.6.1 System Description

This paper evolves around *dynamical state-space models* (DSSM). A system is any set of actions and behaviour that we see fit to evaluate simultaneously. The model is a mathematical description of the system's behaviour. It describes how variables relate to each other and how they evolve. Mathematical modeling consists of mapping the state of the system into variables and then describing with equations how the variables relate to each other. The models can be either continous- or discrete-time. The nature as we observe it is continous-time, but discrete-time models are used for implementation on computers. The part of reality that we attempt to model is known as the plant. The term system is used to describe both the plant and our mathematical description of it. The expressions inputs and outputs are used to relative to the plant and measurements and outputs describe the same vector.

It is attempted to keep the notation consistant in this report; with all vectors, matrices, and vector-functions given in bold, and scalars and sets given in regular math font. If a semicolon is used in a function, the parts following the semicolon are parameters. Time dependency is generally not expressed, but sample indexes are applied in subscript for discrete time variables. In Chapter 2; subscript will be used to denote which reference frame vectors are given relative to.

The notation on Kalman filters generally follow the notation of [56], but the sigma-point notation is taken from [64]. The notation on vessel modeling is taken from [45] and [13].

Both Leibniz and Newton notation for differentation is used. The Newton notation is used if simple variables or expressions are differentiated with regards to time and Leibniz notation is used everywhere else.

Variable	Description
$\boldsymbol{\eta}$	Position vector; given relative to NED-frame
$\boldsymbol{\nu}$	Velocity vector; given relative to BODY-frame
x	Position along the x_n-axis
y	Position along the y_n-axis
ψ	Angle about the z_n-axis
u	Speed in the x_b-direction
v	Speed in the y_b-direction
r	Angular speed about the z_b-axis
$\boldsymbol{\tau}$	Force and moment vector
$\dot{\boldsymbol{\eta}}_c$	Current speed; given relative to the NED frame
$\boldsymbol{\nu}_r$	Vessel speed relative to fluid; given relative to the BODY frame
x_G	x coordinate of center of gravity

Table 1.1: The most important variables in the section on vessel model

Variable	Description
k	Sample number
$\boldsymbol{x}$	State
$\boldsymbol{y}$	Measurement
$\hat{\boldsymbol{x}}_k^-$	A priori estimate
$\hat{\boldsymbol{x}}_k^+$	A posteriori estimate
$\boldsymbol{P}_k^+$	(Estimated) A priori error covariance
$\boldsymbol{P}_k^-$	(Estimated) A posteriori error covariance
$\boldsymbol{P}_{\boldsymbol{x}_k\boldsymbol{y}_k}$	Covariance
$\boldsymbol{P}_{\tilde{\boldsymbol{y}}_k\tilde{\boldsymbol{y}}_k}$	(Estimated) Measurement covariance
$\boldsymbol{K}_k$	Kalman gain
W_i^m	Weights for mean
W_i^c	Weights for covariance
$\boldsymbol{\mathcal{X}}$	Sigma points
$\boldsymbol{\mathcal{Y}}$	Transformed sigma points

Table 1.2: The most important variables in the section on Kalman filtering

1.6.2 Abbreviations

AWGN	Additive white gaussian noise
CFD	Computational fluid dynamics
DOF	Degrees of freedom
ECEF	Earth-centered Earth fixed
ECI	Earth-centered inertial
EKF	Extended Kalman filter
GPS	Global Positioning System
LP	Linear programming
LQR	Linear quadratic regulator
MMSE	Minimum mean square error
MPC	Model predictive control
NED	North-east-down
NLP	Nonlinear programming
NMPC	Nonlinear model predictive control
pdf	Probability density function
PDF	Probability distribution function
QP	Quadratic programming
RV	Random Variable
SNAME	Society of Naval Architects and Marine Engineers
SQP	Sequential quadratic programming
SCP	Sequential convex programming
UKF	Unscented Kalman filter
UT	Unscented transform

Chapter 2

Vessel Model

The mathematical description of a system's behaviour is given by a model. Modern developments in control theory has lead to an increased use of model knowledge when designing control systems. It is natural that these model-based controllers benefits from accurate modeling. State estimation also benefits from accurate modeling. The best model is one that is accurate and descriptive enough, yet not too complex to implement or develop. Marine mechanics is a well-known field and a model is derived based on Newtonian and Lagrangian mechanics combined with a semi-empiric evaluation of hydrodynamic forces. The model described here is developed by T.I. Fossen, and more can be found in [13] and [57]. Also, [18] presents a more hydrodynamical study to hydrodynamic damping.

2.1 Kinematics

2.1.1 Reference Frames

A vessel's position and velocity, as well as the forces affecting it, must be defined relative to a reference frame. The dynamics of the evaluated system will determine which frames are used. In terrestrial navigation[1] there are four reference frames of special importance. Two of these are Earth-Centered and two are geographical, i.e., local frames. They are all 3-dimensional (x, y, z) Cartesian frames and follow the right-hand rule.

The two Earth-Centered frames are:

Earth-Centred-Inertial (ECI) frame The ECI frame is defined relative to the position of two distant stars, with the x-axis pointing towards

[1] Navigation on the earth, compared with celestial navigation, i.e., navigation in space.

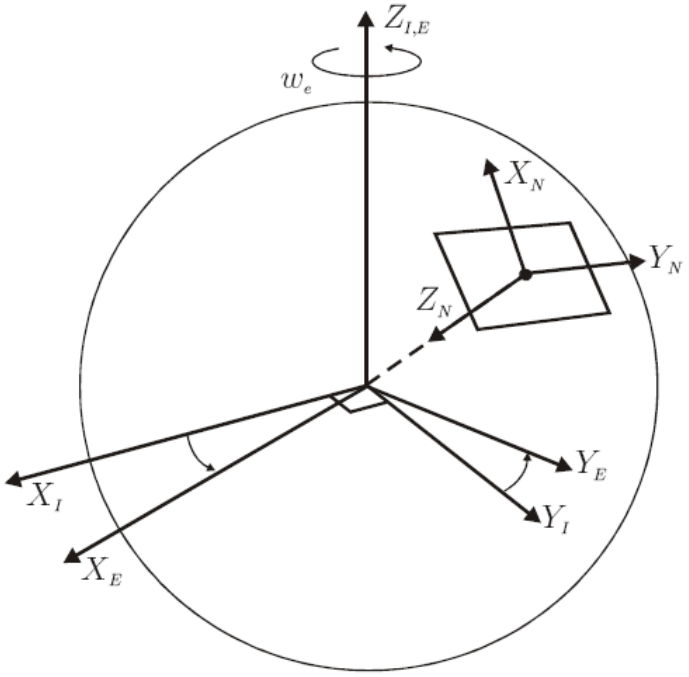

Figure 2.1: Illustration depicting relationship between ECI, ECEF and NED frame. Courtesy of [4].

the Aries point in the Vernal Equinox and the z-axis pointing to the Stella Polaris through the north pole. The y-axis is given from the right-hand rule. The origin is located in the center of the Earth. The ECI-frame is assumed inertial when compared with the remote fixed stars that it is defined relative to. It is further the basis of terrestrial navigation. Points in the ECI frame are given by a vector (x_i,y_i,z_i).

Earth-Centred-Earth-Fixed (ECEF) frame The ECEF frame is defined as a rotation of the ECI frame around the z axis. The x and the y axes rotate as the Earth does with a rate of rotation $\omega_{ei} = 7.2921 \cdot 10^{-5}$ about the z axis. A physical point on the earth's sphere is thus fixed when given in ECEF coordinates and continuously rotating (moving) if given in ECI coordinates. The rotational ECEF frame is obviously not inertial, but the forces attributed to the rotation is often negligable and the frame is thus often viewed as inertial. A point in the ECEF frame is given by (x_e,y_e,z_e).

The two geographical frames are:

North-East-Down (NED) frame The most familiar reference frame is the local NED frame we project all around us. It is defined as a flat-earth approximation of the ECEF frame, i.e., it is a Euclidean frame rotating around the Earth. The origin of the frame is located in a selected point, typically the earth's surface. As the ECEF frame is often assumed inertial, so is the NED frame. The NED frame is the most common reference frame used for local navigation. A position in the NED frame is given by (x_n,y_n,z_n).

Body (B) frame The B frame is fixed to the vessel's hull. The vessel's centre of gravity is usually chosen as the origin of the frame and the coordinate axes are chosen to coincide with the vessel's principle axes of symmetry, see Figure 2.2 This frame is defined to simplify the modeling of the kinetic differential equations relating forces and moments to movement and rotation, velocity and rate measurements are also found relative to this frame. As the B frame is fixed to the vessel, it is continuously rotating and translating relative to the NED frame. Its motion will also typically dominate the acceleration due to the rotation between the ECEF and ECI frame further justifying the assumption that the NED frame is inertial. Position in the B frame is given by (x_b,y_b,z_b).

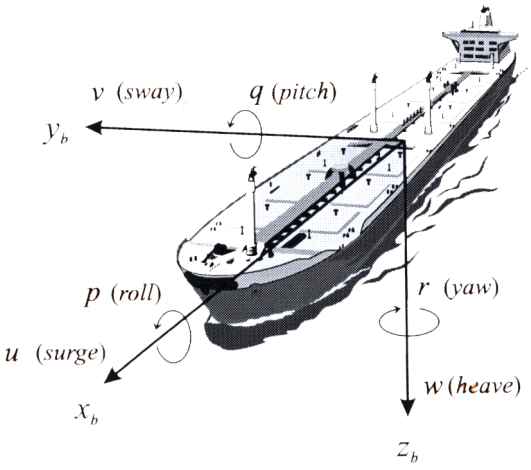

Figure 2.2: Illustration depicting the vessel-fixed body frame axes. Courtesy of [13].

Vectors can be transformed from one frame to another with the use of rotation and homogenous transformation matrices. The general theory on these matrices are not given here, but can be found in several other sources, for example [7], [13].

2.1.2 Vessel States

The vessels pose is composed of position and orientation. The general notation for vessel coordinates comes from [45] and is summarized in tables 2.1 and 2.2 taken from [13].

DOF		Position	Linear velocities	Forces
1	motions in the x-direction,(surge)	x	u	X
2	motions in the y-direction (sway)	y	v	Y
3	motions in the z-direction (heave)	z	w	Z

Table 2.1: Notation for vessel position [45].

2.1.3 3DOF Horizontal Kinematics

This thesis focuses on horizontal control of surface vessels, which means that the vessel model covers only surge, sway and yaw motion, i.e., DOFs 1 and 2 from Table 2.1 and DOF 6 from Table 2.1. This simplification is justified in Section 2.2. The position and orientation angles are gathered in a vector $\boldsymbol{\eta}$, and the linear and angular velocities are gathered in $\boldsymbol{\nu}$, from here on denoted simply as position and velocity. As shown in B.1.2, positions, velocities and accelerations can easily be transformed between frames by using a transformation matrix. In the horizontal 3DOF case, this is a rotation matrix $\boldsymbol{R}(\psi) \in SO(3)$, which is easily found as

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

by examining Figure 2.3.

The position $\boldsymbol{\eta}$ will be given relative to the NED frame while the velocity $\boldsymbol{\nu}$ is given relative to the body frame. More thorough introductions to reference frames and transformations can be found found in [7] and [13].

DOF		Euler Angles	Angular velocities	Moments
4	Rotation about x-axis, (roll,heel)	ϕ	p	K
5	Rotation about y-axis, (pitch,trim)	θ	q	M
6	Rotation about z-axis, (yaw)	ψ	r	N

Table 2.2: Notation for vessel orientation [45].

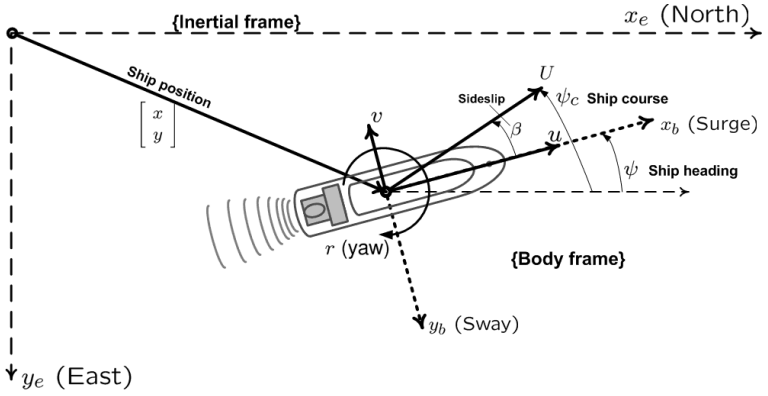

Figure 2.3: Illustration depicting the relationship between the NED frame and the BODY frame, courtesy of [57].

2.2 Kinetics

2.2.1 Maneuvering and Seakeeping

Marine kinetics modeling has traditionally been divided into seakeeping and maneuvering. Seakeeping refers to vessels keeping constant course and speed while subjected to wave excitation, while maneuvering entails more complicated movements in calm waters. These two modeling modes are useful because they explain different hydrodynamic behaviour that fits two separate marine control operations; dynamic positioning and autopilots. The separation allows for different simplifications apropriate for different dominating dynamics. A unified modeling approach that seeks to combine seakeeping and maneuvering is presented in [14] and [48], but will not be considered here. Due to the difficulty of estimating the wave excitation, maneuvering models are often used for DP control as well.

The simulation model developed here will apply a superpositioning of forces that appear due to still-standing vessels in sea, and forces due to vessel motion in calm waters, i.e., a maneuvering model with added forces dependent on wave encounter frequency. This differs from a unified model since all memory effects are neglected.

2.2.2 3DOF Horizontal Kinetic Model

The 6DOF modeling problem includes the motion in surge, sway and heave directions as well as the respective rotations in roll, pitch and yaw. This thesis focuses on DP control, which entails mostly low-speed motion with $U = \sqrt{u^2 + v^2} \leq 4$ knots.

It will be assumed throughout that the controlled vessel is port-starboard symmetric and metacentrically stable. If the vessel is port-starboard symmetric, as most surface vessels are, the dynamics in the surge direction can be decoupled from dynamics in the sway and yaw directions. As most vessels are not aft-fore symmetric, the sway and yaw dynamics are inherently coupled. Metacentric stability depends on the location of the center of gravity and the center of bouancy and the water plane area. Metacentric stability implies that the restoring horizontal forces will limit the deviations from the neutral pitch, roll and heave states. When the vertical states are close to constant, their contribution towards horizontal motion are negligible and as such a 3DOF description of the horizontal dynamics is a suitable approximation.

From Appendix B.1, it is evident that the differential equations describing horizontal motion can be written relative to the B-frame as

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}_{RB}, \tag{2.2}$$

where $\boldsymbol{\tau}_{RB}$ is a generalized vector of forces and moments, $\boldsymbol{M}_{RB}$ is the rigid-body inertia matrix and $\boldsymbol{C}_{RB}$ is the Coriolis matrix. They are found from Newton-Euler considerations of the vessel dynamics. The Coriolis effects can be described in a number of different ways, here a skew-symmetric representation developed in Theorem 3.2 in [13] is chosen. They are given as

$$\boldsymbol{M}_{RB} \triangleq \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_G \\ 0 & mx_G & I_z \end{bmatrix} \tag{2.3}$$

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}) \triangleq \begin{bmatrix} 0 & 0 & -m(x_G r + v) \\ 0 & 0 & mu \\ m(x_G r + v) & -mu & 0 \end{bmatrix}. \tag{2.4}$$

For more details on how these matrices are found, see Chapter 3 in [13].

The components of $\boldsymbol{\tau}_{RB}$ are the net force components in each direction. They consist of the forces produced by the actuators $\boldsymbol{\tau}$, forces due to hydrodynamic effects $\boldsymbol{\tau}_H$ and a vector $\boldsymbol{w}$ containing the unmodeled disturbances including all unmodeled dynamics, i.e.,

$$\boldsymbol{\tau}_{RB} \triangleq \boldsymbol{\tau} + \boldsymbol{\tau}_H + \boldsymbol{w} \tag{2.5}$$

When the fluid is moving, i.e., the vessel's NED velocity given relative to the B frame is different from the vessel's velocity relative to the fluid in the

B frame, where the relative velocity is given as $\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c$ in B coordinates. The forces that appear from hydrodynamic considerations depend on this relative velocity.

The vessel system can be compared with a mass-spring-damper system where the spring forces will move the vessel when the position deviates from the steady state and the damping will counter the movement, i.e., absorb energy. The restoring forces are the gravity and buoyancy forces and they only appear in the vertical direction. The horizontal motion is as such a mass-damper system. It is known that these systems are passive, see for instance [4], and the vessel will, if no propulsion forces are applied, achieve a steady state of velocity $\boldsymbol{\nu}_r = 0$, and drift with the current, i.e., both BIBO stability and asymptotic stability is shown when the dynamics of $\dot{\boldsymbol{\nu}}_r$ is considered. This entails that the vessel control problem consists of counteracting the environmental disturbances and applying force to move the vessel to the desired position.

2.2.3 Hydrodynamic Damping

The hydrodynamic forces affecting a vessel submerged in a fluid are often denoted as hydrodynamic damping or simply as damping. The nature of hydrodynamic damping is inherently complex and analytic approaches are not possible. Approximations applying numerical *computational fluid dynamics* (CFD) are possible, but not tractable when developing models for control systems. The total damping force is commonly viewed as a superposition of several phenomena including radiation damping appearing from wave excitation, and several types of viscous damping due to vessel motion relative to the surrounding fluid.

The basis for modeling hydrodynamical forces is Morison's equation

$$\boldsymbol{\tau}_H = -M_{ij}\frac{\mathrm{d}U_j}{\mathrm{d}t} - \frac{1}{2}\rho A C_{ij}|U_i|U_i, \text{where } i,j \in [1,2,...,6] \tag{2.6}$$

from [40]. As quoted from [18], "for practical marine application, no known model is better suited." Morison's equation is not based on first principles, but is rather a semi-empirical approach that covers the most dominant effects for typical fluid properties and motion ranges. It includes damping due to motion and radiation, but neglects the dependencies on Reynolds number and vortex shedding in the lift direction. This simplification is valid when the Froude number is small $F_r = \frac{u}{\sqrt{gL_{pp}}} < 0.3$. This implies, for a vessel of 87.85 meters, that is valid for up speeds up to $26[\frac{m}{s}]$. For more on this claim, see [18].

A modeling approach based on Morison's equation is found in [13], but with a somewhat differing notation and with additional damping phenomena included. The notation from [13] will be applied from here on, where

$$\boldsymbol{\tau}_H \triangleq \tau_R + \tau_D, \tag{2.7}$$

where

$$\boldsymbol{\tau}_R = -\boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r - \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \boldsymbol{D}_P(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r \tag{2.8}$$

$$\boldsymbol{\tau}_D = -\boldsymbol{D}_S(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \boldsymbol{D}_W(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \boldsymbol{D}_M(\boldsymbol{\nu}_r)(\boldsymbol{\nu}_r), \tag{2.9}$$

$\boldsymbol{\tau}_R$ contains the forces appearing from wave excitation, while $\boldsymbol{\tau}_H$ describes the different viscous damping phenomena.

When a seagoing vessel is exposed to waves, it is forced into a high-frequency harmonic forced motion, induced from pressure differences around the vessel. The resulting forces and moments are dependent on the wave encounter frequency ω_e, i.e., the frequency that the waves hit the vessel with, depending on both the actual wave frequency and the vessel velocity, the acceleration of the vessel and the shape of the hull. As a result, these forces can be interpreted as the result of an added mass which depends on the wave encounter frequency, i.e., $\boldsymbol{M}_A(\omega_e)$. This added mass is generally not a symmetric matrix, i.e., $\boldsymbol{M}_A(\omega_e) \neq \boldsymbol{M}_A(\omega_e)^T$. However, a common simplification in maneuvering models, and for DP-applications, is to assume a constant $\boldsymbol{M}_A = \lim_{\omega_e \to 0} \boldsymbol{M}_A(\omega_e)$ and this constant $\boldsymbol{M}_A$ is often symmetric. Since the kinetic equations are given in the B frame, a coriolis term $\boldsymbol{C}_A$ is added to include the effects of the B frames acceleration. $\boldsymbol{C}_A$ is found using the same calculations as for $\boldsymbol{C}_{RB}$, but the guaranteed symmetric $\overline{\boldsymbol{M}}_A = \frac{1}{2}(\boldsymbol{M}_A + \boldsymbol{M}_A^T)$ is used for the calculation. This leads to the following expressions

$$\boldsymbol{M}_A \triangleq \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix} \tag{2.10}$$

$$\boldsymbol{C}_A(\boldsymbol{\nu}_r) \triangleq \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v_r + \frac{1}{2}(N_{\dot{v}} + Y_{\dot{r}})r \\ 0 & 0 & -X_{\dot{u}}u_r \\ -Y_{\dot{v}}v_r - \frac{1}{2}(N_{\dot{v}} + Y_{\dot{r}})r & X_{\dot{u}}u_r & 0 \end{bmatrix}. \tag{2.11}$$

The interactions between the waves and the vessel are generally well known and the parameters of $\boldsymbol{M}_A$ can be calculated using computer software or estimated from experiments with scale models. A potential damping term

$\boldsymbol{D}_P(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r$ from (2.8) is included to explain the energy which is carried away by the waves and the surrounding fluid.

The latter part of (2.7) is the viscous damping. Several different phenomena are described in (2.9) and all of them appear when the vessel is moving relative to the fluid. These effects are much more difficult to calculate and separate for practical considerations. As a result they are often lumped together with the potential damping in a general velocity dependent damping term $\boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r$.

Skin friction and vortex shedding are described as the most important hydrodynamic damping phenomena in [13]. Skin friction describes the friction forces occuring between the fluid and the vessel hull. This is usually modeled as a linear effect and is a dominating effect for low-frequency motion and low velocities. Vortex shedding is also known as viscous quadratic damping or eddy making and is the cause of cross-flow drag. This force appear because the surrounding fluid is pushed around the vessel when it is moving relative to the fluid, i.e., cross-flow.

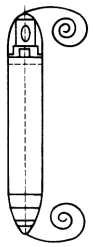

Figure 2.4: Illustration of eddies resulting from vessel motion relative to surrounding fluid. Courtesy of [8].

Since the hydrodynamic forces are impossible to measure, and difficult if not impossible to separate, empirical modeling approaches using Taylor expansions are commonly used. Since the damping is dissipative, or in other words passive, only odd terms should be included. The simple linearization has been used succesfully for DP applications and other operations where the vessel speed is low. Generally, inclusion of more terms leads to smaller errors and thus increasing the range of velocities which the model is valid for. A third-order approach was proposed by Abkowitz in 1963 [1], while a series including quadratic modulus terms were developed by Fedyaevsky and Sobolev [9] and Norrbin [44]. The wish for unifying control applications [5], e.g., applying positioning control at higher velocities, has created a need for more accurate descriptions of the damping forces, and as of late most control applications include nonlinear damping. When choosing a description of the damping effects, parameter estimation should be performed with both models and then choosing the best fit.

The viscous damping terms can then be given by

$$\begin{aligned} \boldsymbol{D}(\boldsymbol{\nu}_r) &= \boldsymbol{D}_L\boldsymbol{\nu}_r + \boldsymbol{D}_{NL}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r \\ &= \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r, \end{aligned} \tag{2.12}$$

where

$$\boldsymbol{D}_L = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix}, \tag{2.13}$$

and the nonlinear damping is given as one of the following:

$$\boldsymbol{D}_{NL}(\boldsymbol{\nu}_r) = \begin{bmatrix} -X_{|u|u}|u_r| & 0 & 0 \\ 0 & -Y_{|v|v}|v_r| - Y_{|r|v}|r| & -Y_{|v|r}|v_r| - Y_{|r|r}|r| \\ 0 & -N_{|v|v}|v_r| - N_{|r|v}|r| & -N_{|v|r}|v_r| - N_{|r|r}|r| \end{bmatrix} \tag{2.14}$$

or

$$\boldsymbol{D}_{NL}(\boldsymbol{\nu}_r) = \begin{bmatrix} -X_{uuu}u_r^2 & 0 & 0 \\ 0 & -Y_{vv}v_r^2 - Y_{vvr}v_r r & -Y_{vrr}v_r r - Y_{rrr}r^2 \\ 0 & -N_{vv}v_r^2 - N_{vvr}v_r r & -N_{vrr}v_r r - N_{rrr}r^2 \end{bmatrix} \tag{2.15}$$

2.2.4 Model Parameters

Finding the model parameters is not a straight-forward task. The parameters of the inertia matrix are easily obtained by measurements, while the added mass parameters can be calculated using strip-theory software such as WAMIT or VERES (ShipX). When these matrices are known, the coriolis terms can be calculated using analytical methods. The real problem consists in calculating the viscous damping parameters. The decoupled parameters can be obtained from simple regression analysis of vessel motion data, while the coupled parameters in sway and yaw are more difficult to obtain. Adaptive methods and online parameter estimation methods are commonly used. It is time-consuming and difficult to perform parameter estimation on full-scale vessels.

The parameters used in the simulation model in this thesis is found in [57]. Measurements, towing tests and adaptive control were performed with the

scale model Cybership II to estimate the parameters. The damping forces are described using a 2nd order modulus damping including an extra third order term in surge. The parameters are summarized in, Table 2.3.

In practice, obtaining accurate parameters is a huge and difficult task, and especially true for the damping parameters. Since they do not consider all effects, they should actually be found anew at every sea-state and vessel operation. As a result, simpler model descriptions are often used for vessel control.

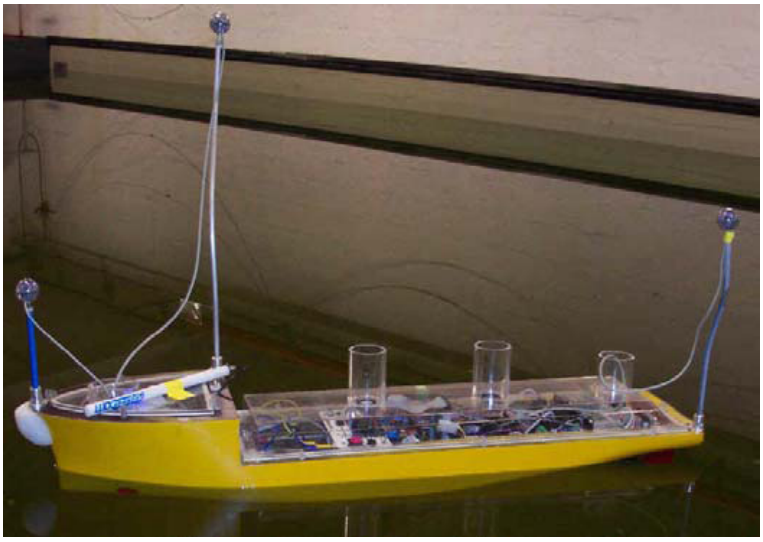

Figure 2.5: Picture of Cybership II. Courtesy of [4].

2.2.5 Scaling of Vessel Parameters and Variables

Scaling is a way to compare variables and parameters of differently sized vessels, e.g., for comparing scaled models with the full-sized vessel. The variables are commonly normalized to a dimensionless form by dividing with the normalization variables relating to the SI-units of the variable. The scaling systems are separated by the selection of normalization variables.

Two common scaling systems exist; the Prime system [45] and the Bis system [44]. The Prime system was originally developed for maneuvering models, and the normalization variable for velocity is simply the total vessel speed U. As a result, the Prime-system can not be used for DP or other low-speed applications where the vessel speed tends to 0.

The Bis scaling is based on vessel mass $\mu\rho\nabla$ and length L_{pp} (perpendicular), where ρ is the mass density of the fluid, and μ and ∇ describe the displacement. It avoids making use of the vessel speed by defining the normalization

Parameter	Value	SI-units
m	23.800	kg
x_G	0.046	m
I_z	1.760	kgm^2
$X_{\dot{u}}$	-2.0	kg
$Y_{\dot{v}}$	-10.0	kg
$Y_{\dot{r}}$	-0.0	kgm
$N_{\dot{v}}$	-0.0	kgm
$N_{\dot{r}}$	-1.0	kgm^2/s
X_u	-0.72253	kg/s
Y_v	-0.88965	kg/s
Y_r	-7.250	kgm/s
N_v	0.03130	kgm/s
N_r	-1.900	kgm^2/s
$X_{\|u\|u}$	-1.32742	kg/m
X_{uuu}	-5.88643	kgs/m^2
$Y_{\|v\|v}$	-36.47287	kg/m
$Y_{\|r\|v}$	-0.805	kg
$Y_{\|v\|r}$	-0.845	kg
$Y_{\|r\|r}$	-3.450	kgm
$N_{\|v\|v}$	3.95645	kg
$N_{\|r\|v}$	0.130	kgm
$N_{\|v\|r}$	0.080	kgm
$N_{\|r\|r}$	-0.750	kgm^2

Table 2.3: Parameters belonging to the scale-model Cybership II.

parameter to time as $\sqrt{L_{pp}/g}$, where g is the gravitation acceleration constant. The scaling parameters for velocity, acceleration, etc. are found as combinations of these. The basic Bis scaling parameters are summarized in 2.4

SI-unit	Scale parameter
kg	$\mu\rho\nabla$
m	L_{pp}
s	$\sqrt{\frac{L_{pp}}{g}}$

Table 2.4: Basic Bis/scaling parameters.

2.3 Disturbance Modeling and Unmodeled Dynamics

A seagoing vessel will be affected by environmental disturbances due to wind, waves and current. It is desirable to know the effects due to these disturbances such that the control system can counteract them in an optimal fashion. It is common to separate the disturbances in high-frequency and low-frequency components, where it is generally not desired to counteract the high-frequency component. Counteracting very quick and often oscillatory motion is often not possible when considering actual thrust capabilities. It would also often induce an unnecessary strain on the propulsion units.

2.3.1 Current

Current is implemented in the simulation model as 2 dimensional nonrotational current modeled by

$$\dot{\boldsymbol{\eta}}_c = \begin{bmatrix} V_c \cos\beta_c \\ V_c \sin\beta_c \\ 0 \end{bmatrix}, \tag{2.16}$$

where V_c and β_c are slowly-varying 1st-order Gauss-Markov processes given by

$$\dot{V}_c + \mu_1(V_c - V_{c0}) = \omega_1 \tag{2.17}$$

$$\dot{\beta}_c + \mu_2(\beta_c - \beta_{c0}) = \omega_2 \tag{2.18}$$

where $\mu_1, \mu_2 > 0$ and V_{c0} and β_{c0} are the mean values.

Both of these variables are given relative to the NED-frame. In the simulation model the vessel motion is affected by current through the hydrodynamic damping.

2.3.2 Waves

This section on wave modeling is based on seakeeping theory and is based on [27]. Waves are generated by wind and are described as developing or fully developed sea. Fully developed seas appear when the wind conditions have been stable over a period and consists of long, stable low-frequency waves that are almost unidirectional. Developing seas, on the other hand, are much more difficult to describe, as the wave composition will be dominated by short crested waves with no single dominant frequency or direction. The mathematical description of the sea state is given by a wave spectrum. which is a stochastic description of the power distribution. It is typically parameterized with the dominant wave height H_s and wave-frequency ω_0. Different wave-spectra are defined for different kinds of sea states and locations.

Here, a modified Pierson-Moskowitz spectrum $S(\omega)$ is combined with a spreading spectrum to allow for several wave-directions. The spreading spectrum is defined with χ_0 as the main wave direction. The wave-directions are found from the range $-\frac{\pi}{2} \leq (\chi - \chi_0) \leq \frac{\pi}{2}$.

Figure 2.6 shows how the power-distribution with regards to frequency and wave-direction for the spectrum

$$S(\omega, \chi) = \frac{2}{\pi} \cos^2(\chi - \chi_0) S(\omega). \tag{2.19}$$

The wave-induced forces are typically assumed to consist of a *high-frequency* (HF) part and a *low-frequency* (LF) part also known as the 1st and 2nd order wave-induced forces. They appear as HF oscillations and a more stable LF drift force.

In seakeeping theory, it is common to model the total wave-induced force as a superpositioning of the Froude-Krilov forces and the diffraction forces. The Froude-Krilov forces are due to pressure differences on the wetted surface of the hull from undisturbed waves. They are calculated by integrating the pressure around the vessel hull. Strip-theory is a simplification of this integation assuming that the vessel hull is divided into strips and summing the force working on each of these strips. The diffraction force is the result of the vessel disturbing the waves, i.e., the vessel stands in the way of the waves.

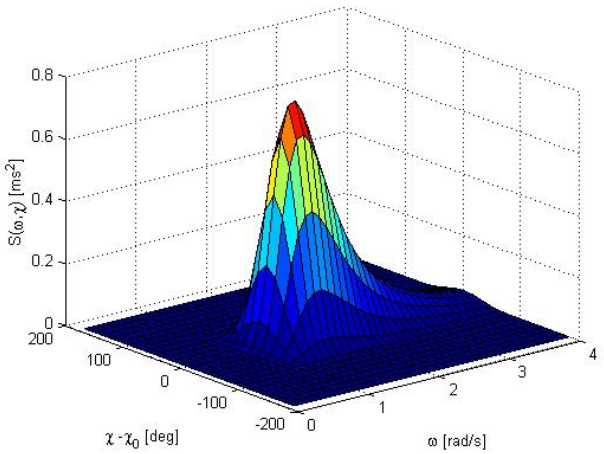

Figure 2.6: The plot shows the power distribution with regards to frequency and wave-direction. Courtesy of [27].

A *force-response amplitude operator* (Force-RAO) relates wave-elevation and wave-acceleration into forces affecting the vessel. The total wave-induced force is then found by summing the effects of several sine waves, all with different amplitude, frequency, direction and phase-shift. N frequencies are picked from the range $\Delta\omega$. They are picked randomly to avoid repeating wave elevation every $\frac{2\pi}{\Delta\omega}$ seconds and the wave-amplitude is found from the wave spectrum. But, a too large or small number of N should be avoided since the total force is equal to the sum of the N waves. Here, a number of 60 waves have been found to be reasonable. If the waves are multi-directional, a random set of M wave directions χ must be found as well. Figure 2.7 shows how the waves are superpositioned.

The total wave elevation in a fixed point is then found from

$$\zeta(t) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sqrt{2S(\omega,\chi)\Delta\omega\Delta\chi}\sin(\omega_i t + \epsilon_i), \tag{2.20}$$

and the wave accelerations are found as

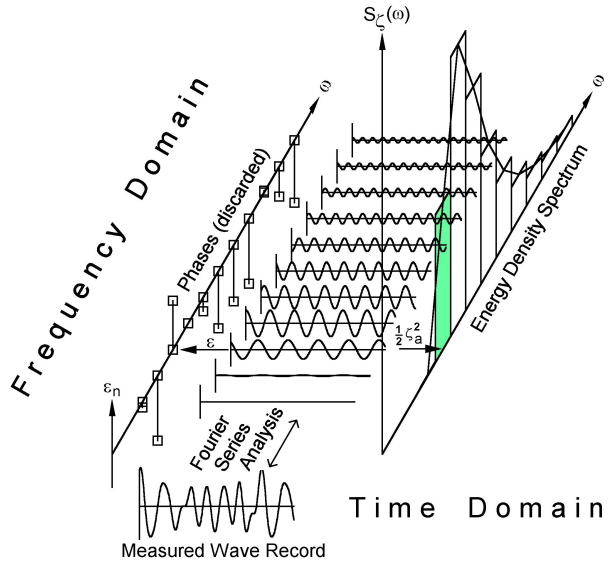

Figure 2.7: The plot shows how several random sine-waves are combined. Courtesy of [27].

$$a_1(t) = \sum_{i=1}^{N}\sum_{j=1}^{M} \omega_i^2 \sqrt{2S(\omega,\chi)\Delta\omega\Delta\chi}\cos(\omega_i t + \epsilon_i) \tag{2.21}$$

$$a_3(t) = \sum_{i=1}^{N}\sum_{j=1}^{M} -\omega_i^2 \sqrt{2S(\omega,\chi)\Delta\omega\Delta\chi}\cos(\omega_i t + \epsilon_i) \tag{2.22}$$

Assuming that the forces are distributed evenly along the vessel's main dimension the Froude-Krilov forces can be found as

$$\tau_{FK} = \begin{bmatrix} \rho V a_1 \cos\chi_0 \\ \rho V a_1 \sin\chi_0 \\ \rho V D a_3 \end{bmatrix}. \tag{2.23}$$

This is a crude simplified version of strip-theory, but is sufficiently accurate for simulation purposes in this thesis. The 2nd order drift forces are modeled as

$$\tau_{WD} = \begin{bmatrix} \frac{1}{8}\rho g(R_x(\omega)\zeta)^2 B\cos\chi_0 \\ \frac{1}{8}\rho g(R_y(\omega)\zeta)^2 L_{pp}\sin\chi_0 \\ \frac{1}{8}\rho g(R_\psi(\omega)\zeta)^2 B L_{pp} D \end{bmatrix} \tag{2.24}$$

,

where $R_x(\omega)$, $R_y(\omega)$, and $R_\psi(\omega)$ are reflection coefficients.

Because these forces are calculated relative to the NED-frame, the wave-direction must also be found from $\chi = \chi_N - \psi$. The equations for wave-elevation (2.20) and wave-accelerations (2.21), (2.22) assume that the vessel's center of gravity stays at a constant point. In addition the wave encounter frequency is considered independent of total vessel speed U. These assumptions are fitting for station-keeping, but is obviously false when performing trajectory-following. The waves are nevertheless modeled in this fashion for both operations.

The total force from the wave-excitation are then given as

$$\boldsymbol{\tau}_W = \boldsymbol{R}^T(\psi)(\boldsymbol{\tau}_{FK} + \boldsymbol{\tau}_{WD}). \tag{2.25}$$

2.3.3 Unmodeled dynamics

An actual vessel is obviously also affected by disturbing forces due to wind. The wind-speed and direction are usually measured and the disturbing forces are then approximated from a look-up table. The approximated forces are then used for feed-forward control. This means that they would be handled in the same way regardless of controller design and are therefore not considered.

Another important aspect is how the thrust $\boldsymbol{\tau}$ appears. They are typically the sum of several thrusters, propellers and other propulsion units. The control of these propulsion units and the distribution of forces between them is done in a thrust-allocation system. In addition the propulsion units will have dynamics such as saturation limits, time-delays and coupled effects. None of these effects are considered in this work.

2.4 Complete model

This leads to the following complete vessel model used for simulation

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)^T\boldsymbol{\nu} \tag{2.26}$$

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{M}_A\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}_A(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau} + \boldsymbol{\tau}_W + \boldsymbol{w}, \tag{2.27}$$

where the relative speed $\boldsymbol{\nu}_r$ and its derivative are defined as

$$
\begin{aligned}
\boldsymbol{\nu}_r &\triangleq \boldsymbol{\nu} - \boldsymbol{R}(\psi)^T \dot{\boldsymbol{\eta}}_c \\
\dot{\boldsymbol{\nu}}_r &= \dot{\boldsymbol{\nu}} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{R}(\psi)^T \dot{\boldsymbol{\eta}}_c\right) \\
&\updownarrow \ddot{\boldsymbol{\eta}}_c = \mathbf{0} \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{R}(\psi)^T\right) \dot{\boldsymbol{\eta}}_c
\end{aligned}
$$

.

Chapter 3

Estimation

Knowing a system's state is essential when deciding upcoming control actions. Since most systems have a dual nature, consisting of hidden states and measured variables, estimation is typically necessary to obtain full state knowlegde. Models are developed to explain how the states develop over time and how they relate to the measurements. There will always be unmodeled dynamics. The main reasons for this are limited knowledge of the physical properties, and by choice due to simplifications. These unmodeled dynamics can be handled mathematically in two ways, deterministic as a perturbed systems or in a stochastic fashion. The first method is much used in nonlinear control and robust stability, but will not be mentioned here, whilst the second has found great usage in state estimation. From a stochastic viewpoint the unmodeled dynamics are denominated as noise. The noise is then described by stochastic processes, often depicted as *additive white gaussian noise* (AWGN) or as augmented states driven by AWGN. Stochastic variables are also known as *random variables* (RV). A stochastic variable is defined by its *probability density function* (pdf)[1] and a stochastic process is defined by its model and the pdfs of the contributing RVs.

[17] describes estimation as the problem of extracting information from data, i.e., separating the signal from the noise. The estimation problem is further divided into three classes; filtering, smoothing, and prediction. The filtering problem consists of estimating the optimal value at current time given all measurements up to and including the current one. Smoothing is estimating something that has happened prior to the current measurement and prediction is estimating a future variable.

[1] NB! Note the difference between probability density function (pdf) and its integrated counterpart the probability distribution function (PDF). For details, see Appendix C.

3.1 Discrete Systems

All physical dynamics are continous[2], but computer calculations are inherently discrete. As such, a discretized model $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{d}_k)$ must be found from $\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d})$. This discretization is usually performed as a simple forward euler integration, where T is the discrete time step.

The forward Euler integration is given as

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\ &\quad \updownarrow \text{ Discrete version, Euler's method} \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + T\boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k), \end{aligned} \tag{3.1}$$

where $\boldsymbol{x}_k$ is the discretized analogue of the continous $\boldsymbol{x}(kT)$. A linearized version can be found as $\boldsymbol{x}_{k+1} = \boldsymbol{F}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k + \boldsymbol{L}_k\boldsymbol{d}_k$, where

$$\boldsymbol{F}_k = (\boldsymbol{I} + T\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{x}}(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}))\mid_{\boldsymbol{x}=\boldsymbol{x}_k, \boldsymbol{u}=\boldsymbol{u}_k, \boldsymbol{d}=\boldsymbol{d}_k}) \tag{3.2}$$

$$\boldsymbol{B}_k = T\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{u}}(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}))\mid_{\boldsymbol{x}=\boldsymbol{x}_k, \boldsymbol{u}=\boldsymbol{u}_k, \boldsymbol{d}=\boldsymbol{d}_k} \tag{3.3}$$

$$\boldsymbol{L}_k = T\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{d}}(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}))\mid_{\boldsymbol{x}=\boldsymbol{x}_k, \boldsymbol{u}=\boldsymbol{u}_k, \boldsymbol{d}=\boldsymbol{d}_k}. \tag{3.4}$$

The numerical integration errors must be kept within an acceptably large bound. If the system has dynamics that are dominantly faster than T, divergence problems, i.e., instability, can occur. Two typical methods to increase numerical accuracy, and thus the numerical stability, is to shorten the sample time or use more complex discretization schemes, e.g., Runge-Kutta methods. The much used fourth order Runge-Kutta is given by

[2] Unless a quantum-physics view of the world is taken, but this would also lead to the potentially catastrophic loss of causality through nonlocality. Luckily, the events we assume causal have a probability of almost 1.

$$
\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \\
&\updownarrow \text{Discrete version, Runge Kutta 4 method} \\
\boldsymbol{k}_1 &= \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) \\
\boldsymbol{k}_2 &= \boldsymbol{f}(\boldsymbol{x}_k + \frac{T\boldsymbol{k}_1}{2}, \boldsymbol{u}_k) \\
\boldsymbol{k}_3 &= \boldsymbol{f}(\boldsymbol{x}_k + \frac{T\boldsymbol{k}_2}{2}, \boldsymbol{u}_k) \\
\boldsymbol{k}_4 &= \boldsymbol{f}(\boldsymbol{x}_k + T\boldsymbol{k}_3, \boldsymbol{u}_k) \\
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \frac{T(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4)}{6}.
\end{aligned} \tag{3.5}
$$

More information on numerical stability and Runge-Kutta methods can be found in [7].

3.2 Probabalistic Inference

Probabalistic inference is defined in [64] as "the problem of estimating the hidden variables (states and parameters) of a system in an optimal and consistent fashion (using probability theory) given noisy or incomplete observations". When states and parameters are estimated simultaneously it is known as dual estimation.

The systems that are considered are 1. order markov processes described by nonlinear discrete-time dynamic state-space models

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k) \tag{3.6}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{v}_k), \tag{3.7}$$

where $\boldsymbol{x}_k$ is the hidden system state, $\boldsymbol{y}_k$ the measured variables, $\boldsymbol{u}_k$ the exogenous inputs, and $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ are stochastic noice processes given by their respective distributions $\boldsymbol{p}(\boldsymbol{v}_k)$ and $\boldsymbol{p}(\boldsymbol{w}_k)$.

The conditional pdf $\boldsymbol{p}(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})$ is then given by (3.6) and $\boldsymbol{p}(\boldsymbol{v}_k)$ while $\boldsymbol{p}(\boldsymbol{y}_k \mid \boldsymbol{x}_k)$ is given by (3.7) and $\boldsymbol{p}(\boldsymbol{w}_k)$. The discrete system can then be described as a bayesian network as shown in Figure 3.1.

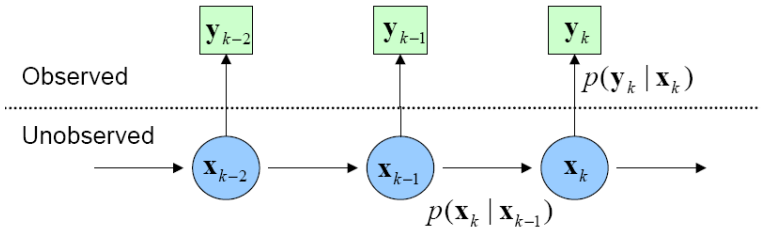

Figure 3.1: The discrete probabilistic system depicting the relationship between states and measurements. Courtesy of [64]

3.3 Recursive Filtering

Recursive filtering is a sequential filtering process where only the current measurement and previous state estimate are used to compute the current estimate. Compared with iterative batch methods, they trade some accuracy for a significant reduction in memory demands and computation cost, allowing on-line estimation.

One choice of filter estimate is the conditional mean,

$$E[\boldsymbol{x}_k \mid \boldsymbol{y}_k] = \int \boldsymbol{x}_k \boldsymbol{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_k) \quad \mathrm{d}\,\boldsymbol{x}_k, \tag{3.8}$$

also known as maximum likelihood.

The conditional mean can be considered as a statistic least squares estimate. Calculating $E[\boldsymbol{x}_k \mid \boldsymbol{y}_k]$ is done in two steps, a time-update step (prediction) calculated prior to obtaining the current measurement and a corrector step (filtering) after the the measurement is available. They are known as the *a priori* estimate $\hat{\boldsymbol{x}}_k^-$ and the *a posteriori* estimate $\hat{\boldsymbol{x}}_k^+$. Mathematically they are defined as

$$\hat{\boldsymbol{x}}_k^- \triangleq E[\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1}] \tag{3.9}$$

$$\hat{\boldsymbol{x}}_k^+ \triangleq E[\boldsymbol{x}_k \mid \boldsymbol{y}_k]. \tag{3.10}$$

We wish to use this in a recursive (sequential) setting such that the estimate is updated using only the previous estimate at every time step. As previously mentioned, the alternative is to propagate all previous measurements and performing a much more computational demanding integral calculation.

If $\boldsymbol{x}$ is a 1. order Markov process, the recursive estimate will be identical with the batch estimate. If $\boldsymbol{x}$ is a Markov process equation (3.8) shows that the recursive estimation task consists of finding the posterior distribution function $\boldsymbol{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_k)$ from $\boldsymbol{p}(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{k-1})$. Given $\boldsymbol{p}(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})$ and $\boldsymbol{p}(\boldsymbol{y}_k \mid \boldsymbol{x}_k)$ and the following bayesian equations

$$\boldsymbol{p}(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) = \int \delta(\boldsymbol{x}_k - \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{v}_{k-1})) \quad \mathrm{d}\,\boldsymbol{v}_k \tag{3.11}$$

$$\boldsymbol{p}(\boldsymbol{y}_k \mid \boldsymbol{x}_k) = \int \delta(\boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{w}_k)) \quad \mathrm{d}\,\boldsymbol{w}_k \tag{3.12}$$

$$\boldsymbol{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1}) = \int \boldsymbol{p}(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1}) \boldsymbol{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1}) \quad \mathrm{d}\,\boldsymbol{x}_{k-1} \tag{3.13}$$

$$\boldsymbol{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_k) = \frac{\boldsymbol{p}(\boldsymbol{y}_k \mid \boldsymbol{x}_k) \boldsymbol{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1})}{\int \boldsymbol{p}(\boldsymbol{y}_k \mid \boldsymbol{x}_k) \boldsymbol{p}(\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1}) \quad \mathrm{d}\,\boldsymbol{x}_{k-1}}. \tag{3.14}$$

the optimal posterior estimate is found from (3.8).

Calculating (3.14) proves difficult in general, but is tractable for simple cases such as linear gaussian systems. If the system is nonlinear or the noise is neither gaussian nor uniformly distributed, approximations are needed. Several approximative methods are known. One method is to approximate the values of the Bayesian integrals (3.11) through (3.14) using finite sums, another to approximate the disturbing noise as gaussian or approximating the general system $\boldsymbol{f}$ and $\boldsymbol{h}$ with linear functions. A third way, which is the focus of this chapter, is to directly approximate the transformed distributions of the non-linear system's states. Option two and three are both contained in the Kalman filter approach and will be presented here, while the first will not be considered here.

3.4 Introduction to the Kalman Filter

The Kalman filter is a recursive linear state estimator. New estimates are made from stochastic calculations and a linear feedback of the innovation error. The innovation error is the deviation between the measurement and an estimated measurement $\tilde{\boldsymbol{y}_k} = \boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-$. The Kalman filter framework can be viewed as a stochastic extension of least squares methods and recursive Bayesian estimation. It consists of a set of simple equations and has been the most applied state-estimator for the last 40 years. In addition to providing a state estimate, the filter also calculates a covariance matrix $\boldsymbol{P}_k^+$ that describes the filter's measure of the quality of the estimate. The covariance also shows how states are statistically coupled. The Kalman filter has a recursive structure as given in Figure 3.2.

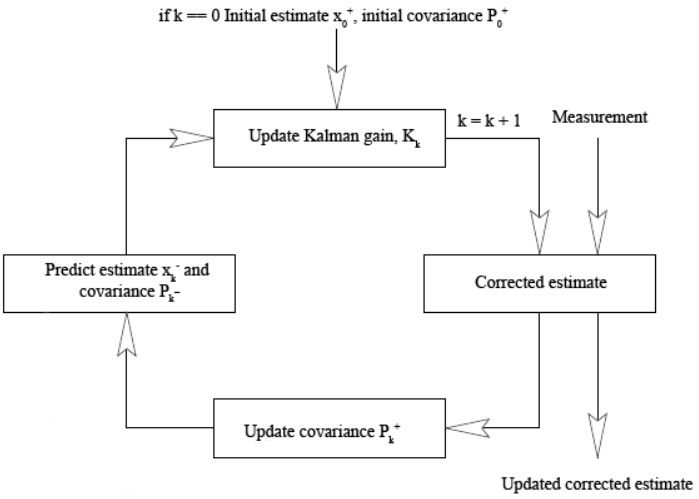

Figure 3.2: The Kalman filter works by recursively updating the estimate and covariance.

3.4.1 Development of the Kalman Filter

The Kalman filter was developed by Rudolf E. Kalman in [31] as an extension to the Wiener filter proposed by Norbert Wiener. Where the Wiener filter was a single input single output filter, the Kalman filter was a multiple input multiple output filter. He also showed that the estimation problem was the dual of the noise-free regulator problem. An important assumption is that the measurement errors and model-mismatch problems can be attributed to random processes.

Kalmans goal was to develop an optimal linear recursive estimator that minimized the mean square error (MMSE), $\boldsymbol{J}_k$, of the estimate. The error vector is defined as $\tilde{\boldsymbol{x}}_k \triangleq \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^+$ and $\boldsymbol{J}_k$ is given by

$$\begin{aligned} \boldsymbol{J}_k &\triangleq E[\tilde{\boldsymbol{x}}_k^T \tilde{\boldsymbol{x}}_k] \\ &= \mathrm{Tr}\, E[\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^T] \\ &= \mathrm{Tr}(\boldsymbol{P}_k^+). \end{aligned} \tag{3.15}$$

This is similar to the projections performed in the deterministic least-squares method. As mentioned previously, Kalman filtering is based on recursive Bayesian estimation with some added assumptions:

1. Consistent minimum variance estimators of the RVs can be recursively maintained from the latest mean and covariance.

2. The a posteriori estimate is chosen as a linear function of the a priori estimation and the new measurement.

3. The system model must be known and observable.

Assumption 1 implies that the pdfs used by the recursive bayesian estimation can be approximated by recursively updating the stochastic first order moment and the stochastic second order central moment of the estimate. Assumption 2 defines the structure of the recursive estimator as $\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-)$. The third assumption states that there must be a observable model of the system and the noise characteristics. An observable model is one where the development of the states can be uniquely calculated from the initial value of states, the applied inputs and the measurements.

As a recursive estimator, the Kalman filter consists of a predictor step

$$\hat{\boldsymbol{x}}_k^- = E[\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1}] = E[\boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{v}_{k-1})] \tag{3.16}$$

and a corrector step (statistic estimator)

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-), \tag{3.17}$$

where

$$\hat{\boldsymbol{y}}_k^- = E[\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{w}_k)]. \tag{3.18}$$

The remaining part of the Kalman filter are equations that calculate the aposteriori error covariance matrix $\boldsymbol{P}_k^+$ and the Kalman gain $\boldsymbol{K}_k$.

The following derivation of the Kalman filter is known as the statistical derivation and is independent of the system structure, i.e. it is not assumed that the state-update $\boldsymbol{f}$ and measurement $\boldsymbol{h}$ equations are linear. Only the assumptions made by Kalman are needed. It is found in the Appendix in [64] and in Chapter 10.5.1 in [56].

$$\begin{aligned}
\boldsymbol{P}_k^+ &= E[\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{x}}_k^T] && (3.19)\\
&= E[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^+)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^+)^T] \\
&\updownarrow \hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-) \\
&= E[\{(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-) - \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-)\}\{(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-) - \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-)\}^T] \\
&= E[\{\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^- - \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-)\}\{(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-)^T + (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k^-)^T \boldsymbol{K}_k^T\}] \\
&= \boldsymbol{P}_k^- - \boldsymbol{P}_{\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{y}}_k} \boldsymbol{K}_k^T - \boldsymbol{K}_k \boldsymbol{P}_{\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{y}}_k} + \boldsymbol{K}_k \boldsymbol{P}_{\tilde{y}_k \tilde{y}_k} \boldsymbol{K}_k^T. && (3.20)
\end{aligned}$$

We wish to find the $\boldsymbol{K}_k$ that minimize the trace of (3.20). It is found from

$$\frac{\partial \boldsymbol{J}_k}{\partial \boldsymbol{K}_k} = 0 \tag{3.21}$$

$$\updownarrow \text{ Solve for } \boldsymbol{K}_k$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{y}}_k} (\boldsymbol{P}_{\tilde{y}_k \tilde{y}_k})^{-1}. \tag{3.22}$$

This $\boldsymbol{K}_k$ is inserted into (3.20) to get

$$\boldsymbol{P}_k^+ = \boldsymbol{P}_k^- - \boldsymbol{K}_k \boldsymbol{P}_{\tilde{y}_k \tilde{y}_k} \boldsymbol{K}_k^T. \tag{3.23}$$

This derivation leads to a set of equations, (3.23) and (3.22), that differs from those typically developed for linear systems, but are mathematically identical. This derivation is more suited to show how gaussian approximative methods can be implemented in the Kalman filter framework. If the system is linear, then (3.16), (3.17) and (3.20) can be evaluated analytically. Otherwise they generally have to be approximated. Two common ways of doing this will be presented later, the *extended Kalman filter* (EKF) and the *unscented Kalman filter* (UKF). It is important to note that the Kalman filter is not the optimal solution to the nonlinear estimation problem, as it is to the linear estimation problem, but is often the best linear estimator. The optimality discussion is of course strictly theoretical as model mismatch, limited numerical accuracy and several other calculational problems will appear when performing estimation on real systems. But, the extensive use of the EKF shows that the Kalman filter extends nicely to nonlinear problems in spite of its assumptions.

3.4.2 Implementation Issues with the Kalman Filter

The Kalman filter is a theoretical algorithm and as such makes some impossible assumptions. Firstly, it assumes that no model-mismatch occurs, including parameter values. Further, optimality is a property of the theoretical filter, same as stability is a property of mathematical equations. Luckily, the filter is generally robust enough to provide converging estimates for most problems. Another important source of error is finite precision arithmetics on computers. Several methods have been implemented to improve robustness for arithmetic problems and model errors. The most important are implementing fictious noise, square root filtering, symmetrizing or other ways of increasing arithmetic presision. More on this can be found in [56]. Observability is a fundamental requirement; if the given model is not observable, another model description must be defined or other measurements

must be obtained. Kalman found simple tests for evaluating observability in [31], while no simple test exists for nonlinear systems.

Divergence, or unstable estimation, is another issue arising in implementations. This appears when the estimator is not consistent, generally because the covariance is under-estimated. Common reasons for this is model-mismatch, poor tuning or poor initialization of the filter. The EKF struggles a lot with model-mismatch when updating the covariance and are often labeled as difficult to tune and troubled with divergence problems.

3.5 Kalman Filter for Linear Systems

The Kalman filter is the optimal solution to

$$\min E[\epsilon_k^T \epsilon_k]$$

for linear systems if $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are zero-mean white Gaussian uncorrelated noise and the optimal linear solution if they are zero-mean white and uncorrelated. The latter part is often misunderstood as if the Kalman filter is only optimal if the noise is Gaussian. The Kalman filter is still the optimal linear filter if the assumptions hold. There are different equivalent expressions for updating the Kalman gain and the error covariance matrix for linear systems, the ones presented here are common and known for their good numerical properties.

3.5.1 Linear Discrete-Time System Kalman Filter

1. Initialize the system
   (a) Model the system

$$\boldsymbol{x}_k = \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1} + \boldsymbol{B}_k\boldsymbol{u}_{k-1} + \boldsymbol{L}_k\boldsymbol{v}_{k-1} \tag{3.24}$$
$$\boldsymbol{y}_k = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{D}_k\boldsymbol{w}_k \tag{3.25}$$
$$E[\boldsymbol{w}_k] = \boldsymbol{0} \tag{3.26}$$
$$E[\boldsymbol{v}_k] = \boldsymbol{0} \tag{3.27}$$
$$E[\boldsymbol{v}_k\boldsymbol{v}_j^T] = \boldsymbol{R}_k\delta_{k-j} \tag{3.28}$$
$$E[\boldsymbol{w}_k\boldsymbol{w}_j^T] = \boldsymbol{Q}_k\delta_{k-j} \tag{3.29}$$
$$E[\boldsymbol{v}_k\boldsymbol{w}_k^T] = \boldsymbol{0} \tag{3.30}$$

(b) Initialize the initial guess of estimation values $\hat{\boldsymbol{x}}_0^+$ and covariance matrix $\boldsymbol{P}_0^+$.

$$\boldsymbol{x}_0^+ = E[\boldsymbol{x}_0] \tag{3.31}$$

$$\boldsymbol{P}_0^+ = E[(\boldsymbol{x}_0 - \boldsymbol{x}_0^+)(\boldsymbol{x}_0 - \boldsymbol{x}_0^+)^T] \tag{3.32}$$

2. For each time step $k = 1, 2, ...,$ calculate the updated estimates and covariance

(a) Predictor step

$$\boldsymbol{x}_k^- = \boldsymbol{F}_{k-1}\boldsymbol{x}_{k-1}^+ + \boldsymbol{B}_k\boldsymbol{u}_{k-1} \tag{3.33}$$

$$\boldsymbol{P}_k^- = \boldsymbol{F}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{F}_{k-1}^T + \boldsymbol{L}_{k-1}\boldsymbol{Q}_{k-1}\boldsymbol{L}_{k-1}^T \tag{3.34}$$

(b) Correction step

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{D}_k\boldsymbol{R}_k\boldsymbol{D}_k^T)^{-1} \tag{3.35}$$

$$\boldsymbol{x}_k^+ = \boldsymbol{x}_k^- + \boldsymbol{K}_k(y_k - \boldsymbol{H}_k\boldsymbol{x}_k^-) \tag{3.36}$$

$$\boldsymbol{P}_k^+ = (I - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^-(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \tag{3.37}$$

3.6 The Extended Kalman Filter

The Extended Kalman filter (EKF) algorithm is the simplest extension of the Kalman filter algorithm for nonlinear systems. It is based directly on the original derivation of the Kalman filter, but approximates the nonlinear transformations of $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ through $\boldsymbol{f}_k$ and $\boldsymbol{h}_k$. The time-update and measurement equations are implemented in their nonlinear form, but the covariance is propagated through a linearized approximation. This simplification is poor when the nonlinearities are severe and divergence is a common problem. Severe nonlinearities basically entails that the first of Kalmans assumptions are not fulfilled. Due to these problems, the EKF is known as troublesome and difficult to tune. It will also be difficult to decide whether the problems are due to poor tuning or simply from the structure of the problem.

Yet, the EKF has been the de facto standard for non-linear estimation since the 60s. A reason for this is that the theory of nonlinear estimation have not been sufficiently developed combined with the fact that nonlinear problems are much more difficult to generalize than linear problems. Linearization is a well known approach to handling nonlinearities and for many problems it provides sufficient accuracy. The linearisation errors in the EKF can be decreased by implementing higher order linearizations, shorter sample-times or iterated kalman filtering, but all of these methods increase the computational expense. EKF is approximately as computationally expensive as the

linear Kalman filter, but calculating the Jacobian can be a difficult mathematical operation.

3.6.1 Nonlinear Discrete-Time System Extended Kalman Filter

1. Initializing
   (a) Model the system

$$\boldsymbol{x}_k = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{v}_{k-1}) \tag{3.38}$$
$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{w}_k) \tag{3.39}$$
$$E[\boldsymbol{w}_k] = \boldsymbol{0} \tag{3.40}$$
$$E[\boldsymbol{v}_k] = \boldsymbol{0} \tag{3.41}$$
$$E[\boldsymbol{v}_k \boldsymbol{v}_j^T] = \boldsymbol{R}_k \delta_{k-j} \tag{3.42}$$
$$E[\boldsymbol{w}_k \boldsymbol{w}_j^T] = \boldsymbol{Q}_k \delta_{k-j} \tag{3.43}$$
$$E[\boldsymbol{w}_k \boldsymbol{v}_k^T] = \boldsymbol{0} \tag{3.44}$$

   (b) Initialize the initial guess of estimation values $\hat{\boldsymbol{x}}_0^+$ and covariance matrix P_0^+.

$$\hat{\boldsymbol{x}}_0^+ = E[\boldsymbol{x}_0] \tag{3.45}$$
$$\boldsymbol{P}_0^+ = E[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0^+)^T] \tag{3.46}$$

2. For each time step $k = 1, 2, ...$, calculate the updated estimates and covariance
   (a) Predictor step

$$\boldsymbol{F}_{k-1} = \frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}))|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}=\boldsymbol{u}_k, \boldsymbol{w}=\boldsymbol{w}_k} \tag{3.47}$$
$$\boldsymbol{L}_{k-1} = \frac{\partial}{\partial \boldsymbol{w}}(\boldsymbol{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{w}_{k-1}))|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}=\boldsymbol{u}_k, \boldsymbol{w}=\boldsymbol{w}_k} \tag{3.48}$$
$$\boldsymbol{P}_k^- = \boldsymbol{F}_{k-1} \boldsymbol{P}_{k-1}^+ \boldsymbol{F}_{k-1}^T + \boldsymbol{L}_{k-1} \boldsymbol{Q}_{k-1} \boldsymbol{L}_{k-1}^T \tag{3.49}$$
$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{f}_{k-1}(\hat{\boldsymbol{x}}_{k-1}^+, \boldsymbol{u}_{k-1}, \boldsymbol{0}) \tag{3.50}$$

   (b) Correction step

$$\boldsymbol{H}_{k-1} = \frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{v}_k))|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k^-, \boldsymbol{v}=\boldsymbol{v}_k} \tag{3.51}$$
$$\boldsymbol{D}_{k-1} = \frac{\partial}{\partial \boldsymbol{v}}(\boldsymbol{h}_k(\boldsymbol{x}_k, \boldsymbol{v}_k))|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k^-, \boldsymbol{v}=\boldsymbol{v}_k} \tag{3.52}$$
$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{D}_k \boldsymbol{R}_k \boldsymbol{D}_k^T)^{-1} \tag{3.53}$$
$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k (y_k - \boldsymbol{h}_k(\hat{\boldsymbol{x}}_k^-, \boldsymbol{0})) \tag{3.54}$$
$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_k^- (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)^T + \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{K}_k^T \tag{3.55}$$

3.7 The Unscented Kalman Filter

The motivation for developing the UKF was improved treatment of nonlinear systems in Kalman filters. The goal was to find an approximate gaussian method that was derivativeless and yet computationally comparable to existing methods. Several differential-free Monte Carlo methods were developed, e.g., Ensemble Kalman Filter, but they were difficult to implement for on-line estimation due to a high computational cost. They approximated $\boldsymbol{x}_k^+$ and $\boldsymbol{P}_k^+$ by selecting many random points, transforming them through the nonlinear function and then findind the mean and covariance of the transformed points. Julier and Uhlmann imagined that by finding a specific collection of points they could achieve similar accuracy with fewer points. This lead to the development of the unscented transform.

UKF was first presented in [28]. It is still in an initial phase of implementation, but is used in an increasingly number of problems that was previously solved by the EKF. It was the subject of R. van der Merwe's PhD degree [64] which compared it with other state estimators as well as applied it for navigational estimation. In addition, one Masters Thesis[12] at Department of Engineering Cybernetics at NTNU submitted in 2007 implemented the filter for use with sensor integration for underwater navigation.

3.7.1 The Scaled Unscented Transform

The unscented transform is a method to approximate the transformations of RVs through nonlinear systems. Especially it calculates the transform of the mean and covariance with up to fourth order accuracy., i.e., as accurate as a fourth order truncated taylor expansion. This is the same precision as truncated second order Gauss filter. It also seeks to estimate the covariance matrix in a consistent, and as such a more conservative, matter such that the covariance is never underestimated. Mathematically, consistency is given as (3.56). This is essential, as the lack of consistency was one of EKF's great deficiencies,

$$\mathrm{Tr}[\boldsymbol{P}_{\tilde{y}_k\tilde{y}_k} - E[(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k)(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_k)^T]] \leq 0. \tag{3.56}$$

An efficient filter implementation will have a minimal value on the right-hand side of (3.56).

The scaled unscented transform is an extension of the unstended transform first described in [29].

A transformation of variables $\boldsymbol{x}$ through function $\boldsymbol{h}$ into $\boldsymbol{y}$ the scaled unscented transform is performed as follows:

1. Choose a set of $n_s = 2n_x + 1$ points that capture the mean and covariance of $\boldsymbol{x} \in \mathbb{R}^{n_x}$
2. Transform them through the function $\boldsymbol{h}$
3. Calculate the weight parameters W_i^m and W_i^c, where $i = 1, 2, ..., n_s$
4. Approximate the statistics of $\boldsymbol{y}$ as the statistics of the transformed points

The selected points are named sigma points and are found using the scaled unscented transform

$$\boldsymbol{\mathcal{X}} = \begin{bmatrix} \bar{x} & \bar{x} + \gamma\sqrt{\boldsymbol{P}_x} & \bar{x} - \gamma\sqrt{\boldsymbol{P}_x} \end{bmatrix} \tag{3.57}$$

$$\gamma = \alpha\sqrt{L + \kappa} \tag{3.58}$$

$$W_0^c = \frac{\lambda}{L + \lambda} \tag{3.59}$$

$$W_0^m = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta) \tag{3.60}$$

$$W_i^m = W_i^c = \frac{1}{2(L + \lambda)}, \quad i \in [1, 2L]. \tag{3.61}$$

The estimates and covariances are then approximated as

$$\boldsymbol{\mathcal{Y}}_{k|k-1} = h(\boldsymbol{\mathcal{X}}^{x-}_{k|k-1}) \tag{3.62}$$

$$\hat{\boldsymbol{y}}_k^- = \sum_{i=0}^{2L} W_i^m \boldsymbol{\mathcal{Y}}_{i,k|k-1} \tag{3.63}$$

$$\boldsymbol{P}_{\tilde{y}\tilde{y}} = \sum_{i=0}^{2L} W_i^c (\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\boldsymbol{y}}_k^-)(\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\boldsymbol{y}}_k^-)^T \tag{3.64}$$

$$\boldsymbol{P}_{\tilde{x}\tilde{y}} = \sum_{i=0}^{2L} W_i^c (\boldsymbol{\mathcal{X}}^x_{i,k|k-1} - \hat{\boldsymbol{x}}_k^-)(\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\boldsymbol{y}}_k^-)^T, \tag{3.65}$$

where i denotes the column number.

3.7.2 Nonlinear Discrete-Time System Additive Noise Unscented Kalman Filter

The additive noise case is a very common special case of the UKF. It can be employed anytime the system can be modeled as

$$\boldsymbol{x}_k = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{w}_{k-1} \tag{3.66}$$
$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{v}_k. \tag{3.67}$$

The equations are similar to the linear Kalman filter since no additional state augmentation occurs as would be the case for general noise implementations. The performance and estimation accuracy will be similar to what one get if implementing an the general algorithm. A reduction in calculation time is expected since fewer sigma-points are created.

1. Initializing
   (a) Model the system
   $$\boldsymbol{x}_k = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{w}_{k-1} \tag{3.68}$$
   $$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{v}_k \tag{3.69}$$
   $$E[\boldsymbol{w}_k] = \boldsymbol{0} \tag{3.70}$$
   $$E[\boldsymbol{v}_k] = \boldsymbol{0} \tag{3.71}$$
   $$E[\boldsymbol{v}_k \boldsymbol{v}_j^T] = \boldsymbol{R}_k \delta_{k-j} \tag{3.72}$$
   $$E[\boldsymbol{w}_k \boldsymbol{w}_j^T] = \boldsymbol{Q}_k \delta_{k-j} \tag{3.73}$$
   $$E[\boldsymbol{v}_k \boldsymbol{w}_j^T] = \boldsymbol{0}_k \tag{3.74}$$
   (b) Initialize the initial guess of estimation values $\hat{\boldsymbol{x}}_0^{a+}$ and covariance matrix P_0^{a+}.
   $$\boldsymbol{x}_0^+ = E[\boldsymbol{x}_0] \tag{3.75}$$
   $$\boldsymbol{P}_0^+ = E[(\boldsymbol{x}_0 - \boldsymbol{x}_0^+)(\boldsymbol{x}_0^- \boldsymbol{x}_0^+)^T] \tag{3.76}$$
2. For each time step $k = 1, 2, ...,$ calculate the updated estimates and covariance
   (a) Predictor step
      i. Calculate sigma points and weight factors
      $$\boldsymbol{\mathcal{X}}_{k-1} = \begin{bmatrix} x_{k-1}^+ & x_{k-1}^+ + \gamma\sqrt{\boldsymbol{P}_{k-1}} & x_{k-1}^+ - \gamma\sqrt{\boldsymbol{P}_{k-1}} \end{bmatrix} \tag{3.77}$$
      $$W_0^c = \frac{\lambda}{L+\lambda} \tag{3.78}$$
      $$W_0^m = \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \tag{3.79}$$
      $$W_i^m = W_i^c = \frac{1}{2(L+\lambda)}, \quad i \in [1, 2L] \tag{3.80}$$

ii. Time-update equations

$$\boldsymbol{\mathcal{X}}^{*-}_{k|k-1} = \boldsymbol{f}(\boldsymbol{\mathcal{X}}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{0}) \tag{3.81}$$

$$\hat{\boldsymbol{x}}^{-}_{k} = \sum_{i=0}^{2L} W^m_i \boldsymbol{\mathcal{X}}^{*-}_{i,k|k-1} \tag{3.82}$$

$$\boldsymbol{P}^{-}_{k} = \sum_{i=0}^{2L} W^c_i (\boldsymbol{\mathcal{X}}^{*-}_{i,k|k-1} - \hat{\boldsymbol{x}}^{-}_{k})(\boldsymbol{\mathcal{X}}^{*-}_{i,k|k-1} - \hat{\boldsymbol{x}}^{-}_{k})^T + \boldsymbol{Q}_{k-1} \tag{3.83}$$

(b) Correction step

i. Recalculate sigma points

$$\boldsymbol{\mathcal{X}}_{k|k-1} = \begin{bmatrix} \hat{\boldsymbol{x}}^{-}_{k} & \hat{\boldsymbol{x}}^{-}_{k} + \gamma\sqrt{\boldsymbol{P}^{-}_{k}} & \hat{\boldsymbol{x}}^{-}_{k} - \gamma\sqrt{\boldsymbol{P}^{-}_{k}} \end{bmatrix} \tag{3.84}$$

ii. Measurement update

$$\boldsymbol{\mathcal{Y}}_{k|k-1} = h(\boldsymbol{\mathcal{X}}^{*-}_{i,k|k-1}, \boldsymbol{0}) \tag{3.85}$$

$$\hat{\boldsymbol{y}}^{-}_{k} = \sum_{i=0}^{2L} W^m_i \boldsymbol{\mathcal{Y}}_{i,k|k-1} \tag{3.86}$$

$$\boldsymbol{P}_{\tilde{\boldsymbol{y}}_k \tilde{\boldsymbol{y}}_k} = \sum_{i=0}^{2L} W^c_i (\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\boldsymbol{y}}^{-}_{k})(\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\boldsymbol{y}}^{-}_{k})^T + R_k \tag{3.87}$$

$$\boldsymbol{P}_{\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{y}}_k} = \sum_{i=0}^{2L} W^c_i (\boldsymbol{\mathcal{X}}_{i,k|k-1} - \hat{\boldsymbol{x}}^{-}_{k})(\boldsymbol{\mathcal{Y}}_{i,k|k-1} - \hat{\boldsymbol{y}}^{-}_{k})^T \tag{3.88}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{\tilde{\boldsymbol{x}}_k \tilde{\boldsymbol{y}}_k} \boldsymbol{P}^{-1}_{\tilde{\boldsymbol{y}}_k \tilde{\boldsymbol{y}}_k} \tag{3.89}$$

$$\boldsymbol{x}^{+}_{k} = \boldsymbol{x}^{-}_{k} - \boldsymbol{K}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}^{-}_{k}) \tag{3.90}$$

$$\boldsymbol{P}^{+}_{k} = \boldsymbol{P}^{-}_{k} - \boldsymbol{K}_k \boldsymbol{P}_{\tilde{\boldsymbol{y}}_k \tilde{\boldsymbol{y}}_k} \boldsymbol{K}^T_k \tag{3.91}$$

3.8 Tuning of UKF

Tuning of a Kalman filter relates to modeling the stochastic processes affecting the evaluated system and approximating the initial state of the system. This is usually performed in a trial and error setting, but adaptive, i.e., self-tuning, methods exist [34], [59].

If the initial state of the system is unknown, it is often simply set to zero with an accompanying high valued initial state covariance matrix, otherwise the best estimate is used as an initial value and the covariance is set as a diagonal

matrix, i.e., only variances are assumed, that expresses the uncertainty of this estimate, e.g., if a state is assumed to be within the range 2 ± 2 an initial value of 2 is used and the variance is chosen as a value larger than 2^2.

The kalman filter should be tuned to be robust enough to compensate for initial errors. By introducing fictitious random processes a robustness towards model error is achieved. The general tuning problem therefore consists of defining the noise covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ such that the filter achieves satisfactory robust behaviour. They are usually diagonal which implies that each state and each measurement is affected by its own random additive variable and larger values relates so larger degree of uncertainty. The weighting of $\boldsymbol{Q}$ related to $\boldsymbol{R}$ is a simple way to express what we are most certain of; the predictor step using the model or the corrector step depending on the measurements. The initial values of the system covariance must also be approximated, but these can be simply set high if they are unknown, similar to the approximated initial state values.

It is quite simple to find the parameters belonging to $\boldsymbol{R}$ if the estimated system can be kept in a fixed state, as they then are found directly as the covariance of the measurement noise, i.e., if there is no state deviation the entire measured covariance is due to measurement noise. The parameters belonging to $\boldsymbol{Q}$ on the other hand are much more difficult to obtain, and this is why the Kalman filter is known as difficult to tune. They must either be tuned using a combination of model knowlegde and trial and error or via adaptive methods that estimate the covariances while performing the actual estimation.

The discrete prediction step depends on the sample time, and as such, the sample time could also be viewed as a tuning coefficient. The accuracy of the linearization in EKF is highly correlated with the numerical accuracy of the discrete integrator, which again is directly depending on the sample time. Shorter sample times will lead to a higher accuracy, but will increase the number of function evaluations. Iterative Kalman filter approaches exist where the the kalman filter sample time is smaller than the actual time between two sets of measurements [33]. Measurement values are then assumed to be constant during the time it takes for them to be updated.

When tuning the UKF, additional tuning parameters are found in the unscented transform. Three variables α, β, and κ can be chosen to influence the selection of sigma points. α is a variable that determines the spread of the sigma points around $\bar{\boldsymbol{x}}$, while κ is used to assure positive-definiteness of the calculated covariances. Usual values for the two variables are $0 \leq \alpha \leq 1$ and $\kappa = 0$. The β variable is used to incorporate knowlegde of the RVs distribution, $\beta = 2$ is optimal for gaussian processes.

3.9 Dual Estimation

Dual estimation is the problem of estimating hidden states and unknown parameters simoultaneously. Two approaches exist within the Kalman filter framework and they are known as joint Kalman filtering and dual Kalman filtering [41]. Both approaches lead to nonlinear equations so EKF or UKF must be used for the estimations.

Joint kalman filtering implies augmenting the state estimation vector with the unknown parameters, and estimating the states and the parameters within the same Kalman filter. This leads to an approach with a strong stochastic coupling between the states and the parameters and have the best theoretical properties [64]. Dual Kalman filters are applied if one wishes to stochastically decouple the parameters and states [19], e.g., if the parameters are completely unknown, and works by estimating the states in one Kalman filter using the previous best estimate of the parameters as known, while the parameters are estimated in its own Kalman filter using the previous best estimate of the state as known. Otherwise the Kalman filters are implemented as usual.

Figure 3.9 illustrates the concepts of joint and dual Kalman filtering.

The parameter dynamics are modeled as random walks,

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \boldsymbol{r}_k \tag{3.92}$$

$$\boldsymbol{d}_k = \boldsymbol{g}\boldsymbol{x}_k, \boldsymbol{p}_k + \boldsymbol{e}_k \tag{3.93}$$

,

where $\boldsymbol{r}_k$ and $\boldsymbol{e}_k$ are RV and $\boldsymbol{d}_k$ is a measurement related to the parameters, quite often it is the same measurement as used for the state estimation. It has been shown that it is advantageous to use UKF instead of EKF for parameter estimation even though the time-update function (3.92) is linear, this is due to the nonlinear measurement function. The UKF also includes stochastic information lost in the linearization process which leads to more robust estimation behaviour.

The measurement noise covariance $\boldsymbol{R}_e$ cancels out of the algorithm when it is diagonal, as it almost always is, and hence can be set arbitrarly, e.g., $\boldsymbol{R}_e = k\boldsymbol{I}$, where k is a constant. A diagonal $\boldsymbol{R}_e$ implies that the parameters $\boldsymbol{p}_k$ are independent. Three approaches on how to tune the covariance $\boldsymbol{R}_r$ describing $\boldsymbol{r}_k$ in the parameter estimation are given in [64].

- Set $\boldsymbol{R}_r$ as a fixed diagonal matrix. This leads to a tracking approach and should be used if the parameter is known to change value. Large

values lead to a high weight on the newest measurements. If the parameter is constant, a similar approach could be used by slowly decreasing the value towards zero as the parameter converges towards the true value.

- Set $\boldsymbol{R}_r = (\lambda_{RLS}^{-}1 - 1)P_{w_k}$, where$0 < \lambda_{RLS} \leq 1$ and P_{w_k} is the covariance a posterori covariance matrix calculated in the Kalman filter. This approach lets the Kalman filter control the noise covariance, meaning that the covariance is assumed to be larger, and thus allows faster changes, when the Kalman filter estimates its own estimate of the parameter to be poor.

- Set

$$\boldsymbol{R}_r = (1+\alpha_{RM})\boldsymbol{R}_{r_{k-1}}+\alpha_{RM}\boldsymbol{K}_k \left[\boldsymbol{d}_k - \boldsymbol{g}(\boldsymbol{x}_k, \hat{\boldsymbol{p}}_k^-\right] \left[\boldsymbol{d}_k - \boldsymbol{g}(\boldsymbol{x}_k, \hat{\boldsymbol{p}}_k^-\right]^T \boldsymbol{K}_k^T \tag{3.94}$$

. This approach is known as the Robbins-Monro stochastic approximation scheme and assumes that the covariance of the Kalman update step should be consistent with the actual update model.

3.10 Comparison of Linearized Transformation and Unscented Transformation

The approximations of the stochastic transformations are the only differing evalution in when comparing the EKF with the additive noise UKF algorithm. Thus, when comparing EKF and the UKF it is natural comparing them. Here, a nonlinear transformation from polar to cartesian coordinates are performed to demonstrate the accuracy of the unscented transform compared with the linearized transform.

The transformation is given as

$$y_1 = x_1 \cos(x_2) \tag{3.95}$$
$$y_2 = x_1 \sin(x_2). \tag{3.96}$$

The comparison is down with x_1 uniformly distributed between 0.99 and 1.01 and x_2 uniformly distributed between ± 0.35 rad.

Figure 3.4 shows how the mean and covariance is approximated with a linearized transformation and the unscented transform. It is easy to see that the linearization is miscalculates both the mean and the covariance while the unscented transform is unbiased and consistent. The unscented transform could probably benefit from tuning though, as the estimated covariance is unnecessarily large.

Another important issue to address, is the linearization approach can stochastically decouple states that are inherently coupled. This will in some cases lead to a Kalman gain that is not controllable. An example based on the estimation of sea current from vessel motion using the model developed in Chapter 2.

Consider the vessel system given by (2.26) and (2.27). The current is also estimated and it is modeled as two separate velocities, one in surge direction and the other in sway direction relative to the NED-frame. No other disturbances are included. An augmented state vector consisting of $\boldsymbol{x}$ and $\boldsymbol{d}$ is created where

$$\boldsymbol{x} = [\boldsymbol{\eta} \quad \boldsymbol{\nu}]^T \tag{3.97}$$

$$\boldsymbol{d} = [\dot{x}_c \quad \dot{y}_c]^T. \tag{3.98}$$

The current states are assumed to be random walks

$$\dot{\boldsymbol{d}} = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}. \tag{3.99}$$

where ω_1 and ω_2 are RVs.

The linearized dynamics of the system is then given by

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{d}} \end{bmatrix} = \boldsymbol{F} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \end{bmatrix} + \boldsymbol{L}\boldsymbol{w} \tag{3.100}$$

$$\boldsymbol{y} = \boldsymbol{H} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{d} \end{bmatrix} + \boldsymbol{D}\boldsymbol{v}, \tag{3.101}$$

where

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_{xx} & \boldsymbol{F}_{xd} \\ \boldsymbol{F}_{dx} & \boldsymbol{F}_{dd} \end{bmatrix} \tag{3.102}$$

is the system Jacobian. In addition, we know that the other system matrices are given by

$$\boldsymbol{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{3.103}$$

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.104}$$

$$\boldsymbol{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.105}$$

$$\boldsymbol{F}_{dx} = \boldsymbol{0} \in \mathbb{R}^{2x6} \tag{3.106}$$

$$\boldsymbol{F}_{dd} = \boldsymbol{0} \in \mathbb{R}^{2x2} \tag{3.107}$$

$$\boldsymbol{F}_{xx} \in \mathbb{R}^{6x6} \tag{3.108}$$

$$\boldsymbol{F}_{xd} \in \mathbb{R}^{6x2}. \tag{3.109}$$

The following calculations do not depend on the actual parameter contents of $\boldsymbol{F}_{xx}$ and $\boldsymbol{F}_{dx}$, but we know from evaluations in matlab that the linearized system is observable and this is sufficient. We will here show that the elements of the Kalman gain that relate to the current estimate will always be zero. Expressed differently, this implies that the stochastic coupling between the current and the position of the vessel is lost.

As shown previously, the calculation of the Kalman gain is done by recursively updating a covariance. Given a previous a posteriori covariance

$$\boldsymbol{P}_k^+ = \begin{bmatrix} \boldsymbol{P}_{xx,k}^+ & \boldsymbol{P}_{xd,k}^+ \\ \boldsymbol{P}_{dx,k}^+ & \boldsymbol{P}_{dd,k}^+ \end{bmatrix}, \tag{3.110}$$

the current Kalman-gain $\boldsymbol{K}_k$ is found from

$$\begin{aligned} \boldsymbol{P}_{k+1}^- &= \boldsymbol{F}_k \boldsymbol{P}_k^+ \boldsymbol{F}_k^T + \boldsymbol{L}\boldsymbol{R}_k\boldsymbol{L}^T \\ &= \begin{bmatrix} (\boldsymbol{F}_{xx}\boldsymbol{P}_{xx}^+ + \boldsymbol{F}_{xd}\boldsymbol{P}_{dx}^+)\boldsymbol{F}_{xx} & (\boldsymbol{F}_{xx}\boldsymbol{P}_{xx}^+ + \boldsymbol{F}_{xd}\boldsymbol{P}_{dx}^+)\boldsymbol{F}_{dx}^T \\ (\boldsymbol{F}_{dx}\boldsymbol{P}_{xx}^+ + \boldsymbol{F}_{dd}\boldsymbol{P}_{dx}^+)\boldsymbol{F}_{xx} & (\boldsymbol{F}_{dx}\boldsymbol{P}_{xx}^+ + \boldsymbol{F}_{dd}\boldsymbol{P}_{dx}^+)\boldsymbol{F}_{dx}^T \end{bmatrix} + \\ &\quad \begin{bmatrix} (\boldsymbol{F}_{xx}\boldsymbol{P}_{xd}^+ + \boldsymbol{F}_{xd}\boldsymbol{P}_{dd}^+)\boldsymbol{F}_{xd}^T & (\boldsymbol{F}_{xx}\boldsymbol{P}_{xd}^+ + \boldsymbol{F}_{xd}\boldsymbol{P}_{dd}^+)\boldsymbol{F}_{dd} \\ (\boldsymbol{F}_{dx}\boldsymbol{P}_{xd}^+ + \boldsymbol{F}_{dd}\boldsymbol{P}_{dd}^+)\boldsymbol{F}_{xd}^T & (\boldsymbol{F}_{dx}\boldsymbol{P}_{xd}^+ + \boldsymbol{F}_{dd}\boldsymbol{P}_{dd}^+)\boldsymbol{F}_{dd} \end{bmatrix} + \\ &\quad \boldsymbol{L}\boldsymbol{R}_k\boldsymbol{L}^T \end{aligned} \tag{3.111}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k+1}^- \boldsymbol{H}_k^T (\boldsymbol{H}\boldsymbol{P}_{k+1}^- \boldsymbol{H}^T + \boldsymbol{D}\boldsymbol{R}_k\boldsymbol{D}^T)^{-1}. \tag{3.112}$$

The elements that are marked with red in (3.111) are zero-matrices for the

evaluated system. Inserted the specific matrices found for the vessel system we find the covariance and the Kalman gain as

$$\boldsymbol{P}_{k+1}^{-} = \begin{bmatrix} (\boldsymbol{F}_{xx}\boldsymbol{P}_{xx}^{+} + \boldsymbol{F}_{xd}\boldsymbol{P}_{dx}^{+})\boldsymbol{F}_{xx} & \boldsymbol{0}_{6x2} \\ \boldsymbol{0}_{2x6} & \boldsymbol{0}_{2x2} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_k \end{bmatrix} \tag{3.113}$$

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^{-}(1:8,1:3)(\boldsymbol{P}_{k+1}^{-}(1:3,1:3) + \boldsymbol{Q}_k)^{-1}, \tag{3.114}$$

where (3.113) shows that the covariance is block diagonal and that the vessel states are decoupled from the current states. As is seen in (3.114) this means that no feedback occurs from the measurements of vessel state into the current estimate. It is important to note that this is not an observability problem as the system is locally observable at every time-step. Yet, (3.111) shows that this could be rectified by including disturbance-dynamics similar to (2.17) and (2.18) such that $\boldsymbol{F}_{dd}$ is non-zero. This is however not necessarily desired, as it would demand extra tuning to find the mean-velocities u_{c0} and v_{c0} or need extra added ficticious noise. The approach of adding fictitious disturbance dynamics have been shown to not work in 5.4, but it could turn out to work for other tuning parameters. The deviation in current estimation leads to extreme deviations in the estimate of the vessel state as well, as seen in 5.5 and the estimation basically fails.

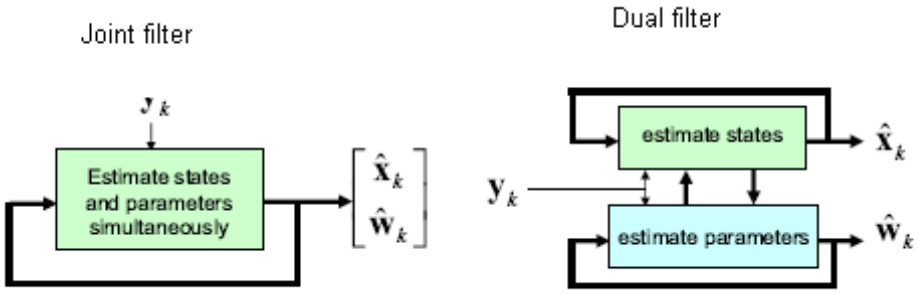

Figure 3.3: Figure illustrating the estimation approaches of joint Kalman filtering and dual Kalman filtering. Courtesy of [64].

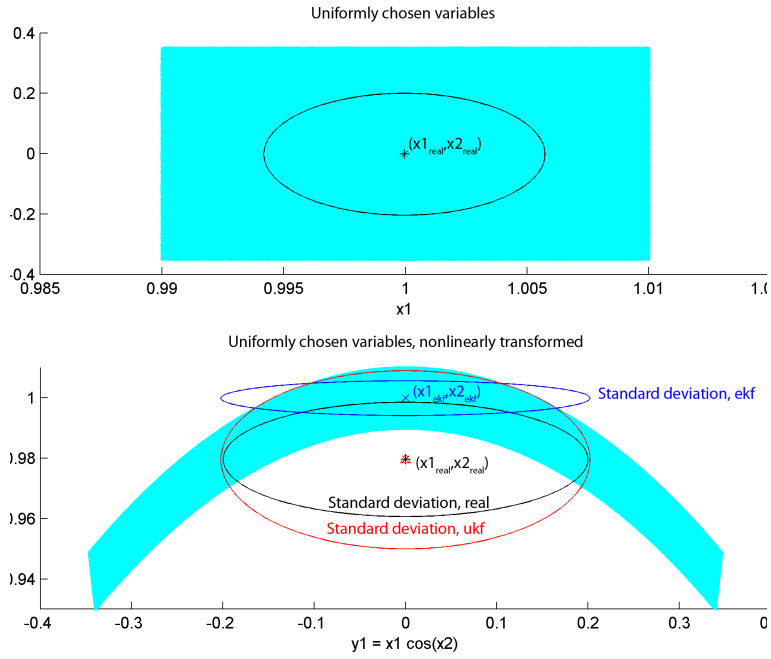

Figure 3.4: Figure depicting the transformation of mean and covariance through linearization and unscented transform

.

Chapter 4

Nonlinear Model Predictive Control

Model predictive control (MPC), also known as *receding horizon control* (RHC), is a class of advanced controllers first thought of by the process industry in the 1960s with the first implementations dating from the 1970s [42]. MPC makes use of a system model and optimization to predict and find an optimal set of inputs relating to maximimize the future performance of the system. By re-solving this optimization problem on-line, a closed-loop behaviour is achieved. Since the optimal parameters are found from solving an optimization problem, constraint handling is inherently supported.

Predictive control is dependent on full state knowlegde, so state estimators must usually be implemented. The alternative is to use a reduced model. Figure 4.1 shows the connection between controller, state estimator and plant.

This chapter will focus on discrete-time systems and discrete-time MPC problems.

4.1 Optimization

Optimization is the problem of finding the minimum or maximum of an objective function. The objective function is a function $f : \mathbb{R}^n \to \mathbb{R}$ that returns a performance index, where n is the number of optimization variables, i.e., the dimension of the optimization variables $\boldsymbol{x}$. The performance index is typically viewed as a cost or a penalty, and smaller values means increased performance. Here we will discuss minimizing, but generality is not lost since similar arguments could be made about maximizing. An un-

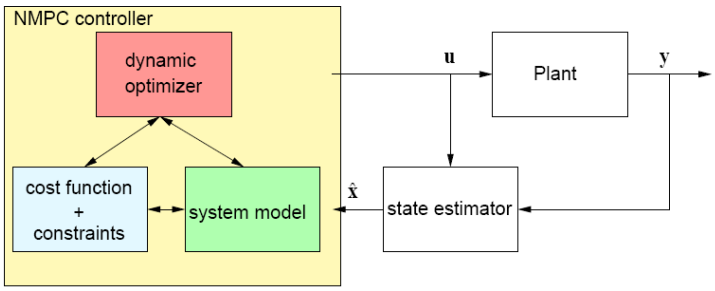

Figure 4.1: The illustration shows the relationship between state estimator, controlller, and, plant. Courtesy of [10].

constrained optimization problem is given as

$$\min_{\boldsymbol{x}\in\mathcal{R}^n} \quad f(\boldsymbol{x}). \tag{4.1}$$

An optimization problem may be subjected to constraints. We separate between two kinds of optimization constraints; hard and soft. Hard constraints are constraints that must be satisfied at all times, while violations of soft constraints entails only added penalties. Constraints are further divided into equality and inequality constraints. Consider a set of hard constraints $\boldsymbol{c}$ given by

$$\boldsymbol{c}_i(\boldsymbol{x}) = \boldsymbol{0}, \quad i \in \mathcal{E} \tag{4.2}$$
$$\boldsymbol{c}_i(\boldsymbol{x}) \leq \boldsymbol{0}, \quad i \in \mathcal{I}, \tag{4.3}$$

where $\mathcal{E}$ and $\mathcal{I}$ are sets defining the equality and inequality indices. The feasible set Ω is then defined as the set where the hard constraints $\boldsymbol{c}(\boldsymbol{x})$ are satisfied, i.e.,

$$\Omega = \{\boldsymbol{x} \mid \boldsymbol{c}_i(\boldsymbol{x}) = \boldsymbol{0}, \quad i \in \mathcal{E}; \quad \boldsymbol{c}_i(\boldsymbol{x}) \leq \boldsymbol{0}, \quad i \in \mathcal{I}\}. \tag{4.4}$$

As previously mentioned, violating soft constraints do not affect the feasibility, but rather induce penalties depending on the magnitude of the violation $\boldsymbol{s}$. The optimization problem given by (4.1) subjected to the hard and soft constraints as

$$\min_{\boldsymbol{x}\in\Omega} \quad f(\boldsymbol{x}) + p(\boldsymbol{s}), \tag{4.5}$$

where $p(\boldsymbol{s})$ is the added penalty function due to the soft constraints. Points $\boldsymbol{x} \in \Omega$ are said to be feasible.

4.1.1 Types of Optimization Problems

A specific optimization problem is characterized by the form of its objective function and its constraints. Several classes exist, the most relevant here are *linear programming* (LP), *quadratic programming* (QP), *convex programming* (CP) and, *nonlinear programming* (NLP):

- LP problems are given by a linear objective function and any constraints are linear.
- QP problems have only linear constraints and a quadratic objective function.
- CP problems have convex objective functions and convex constraints. CP problems are most importantly characterized by the fact that any local minima are global minima. LP and QP problems are examples of CP problems. A strict CP problem have only a single minimum.
- Any problem with nonlinear constraints are characterized as NLP. NLPs are generally not convex.

4.1.2 The Solution to the Optimization Problem

The solution $\boldsymbol{x}^*$ to a optimization problem is also known as a minimum. Two types of minima exist, local minima and global minima. A global minimum is defined by

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \Omega. \tag{4.6}$$

A local minimum is defined as

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{N} \cap \Omega, \tag{4.7}$$

where $\mathcal{N}$ is a neighbourhood of $\boldsymbol{x}^*$ not containing $\boldsymbol{x}^*$, i.e. $\mathcal{N}(\boldsymbol{x}^*) = \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}^* - \boldsymbol{x}\| < r\}$. If the inequalities (4.6) and (4.7) are strict, it is referred to as a strict global minimum and strict local minimum respectively. Figure 4.2 shows an objective function with both local and global minima.

The functions (4.8) and (4.9) are known as the gradient vector and Hessian matrix.

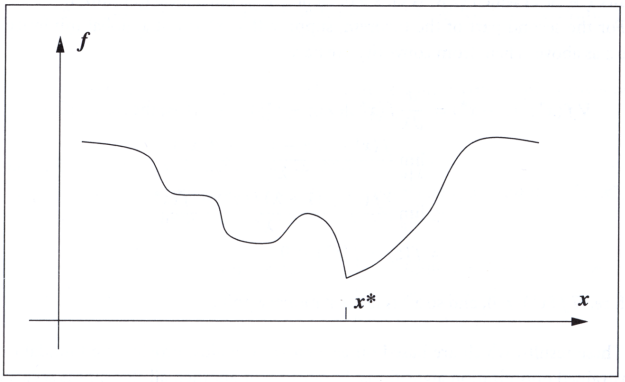

Figure 4.2: The plot shows an objective function $f(x)$ with both local and global minima. Courtesy of [43].

$$g(\boldsymbol{x}) = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{x}} \left(f(\boldsymbol{x}) \right) \tag{4.8}$$

$$H(\boldsymbol{x}) = \frac{\mathrm{d}^2}{\mathrm{d}\boldsymbol{x}^T \boldsymbol{x}} \left(f(\boldsymbol{x}) \right). \tag{4.9}$$

A zero gradient vector tells us that we are at a stationary point, but not whether it is a minimum, a maximum, or a saddle point, see Figure 4.3. The Hessian matrix provides this information; if it is positive semi-definite the point is a minimum, if it is negative semi-definite, a maximum, and if it is neither we are at a saddle point. It is known that gradient and Hessian satisfy

$$g(\boldsymbol{x}^*) = \mathbf{0}$$
$$H(\boldsymbol{x}^*) \geq \mathbf{0}$$

for unconstrained minimization problems.

4.1.3 Constrained Optimization Problems or Unconstrained Optimization Problems

Constrained optimization problems are typically harder to solve than unconstrained problems. For line-search methods this is especially true since they are based on descent-directions, i.e., directions where the gradients

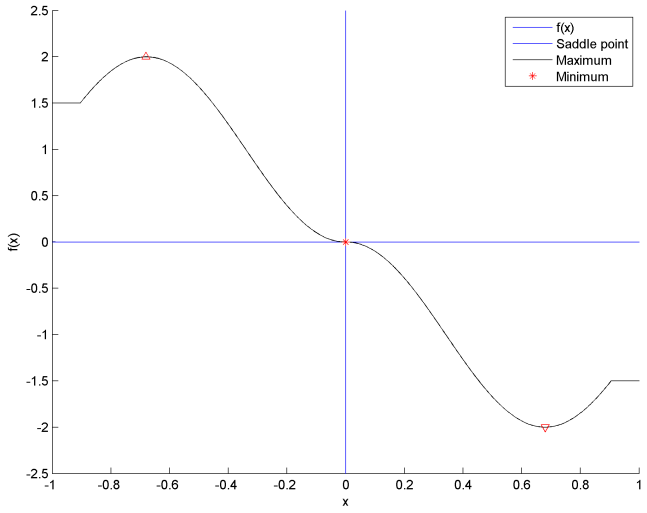

Figure 4.3: The plot shows the minimum, maximum, and, saddle point of a function $f(x)$.

are negative, and these are found directly from the objective function without considering if they are contained in the feasible set. Luckily, we know that equality-constrained optimization problems can be redefined as unconstrained by defining the Lagrangian function

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i \boldsymbol{c}_i(\boldsymbol{x}), \tag{4.10}$$

where $\boldsymbol{\lambda}$ is known as the vector of Lagrange multipliers.

This can be combined with active set methods to construct general unconstrained optimization problems that have the same minima as the original optimization problems. Active set methods are optimization solvers that only consider the set of active inequality constraints, i.e., those that are zero-valued. Given the active set, $\mathcal{A} = \mathcal{E} \cup \{i \in \mathcal{I} \mid \boldsymbol{c}_i(\boldsymbol{x}) = 0\}$, the gradient of the Lagrangian,

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{x}} \left(\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) \right) = g(\boldsymbol{x}) - \sum_i \lambda_i \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{x}} \left(\boldsymbol{c}_i(\boldsymbol{x}) \right), i \in \mathcal{E}, \tag{4.11}$$

shows that the $\boldsymbol{\lambda}_i, i \in \mathcal{A}$ must all be zero for the constraints that are nonzero at the solution. These conditions are summed in the 1st order KKT conditions found on page 328 in [43]. The second order condition demands that the Hessian of the Lagrangian must be positive semidefinite. These

two conditions assure that the optimal solution to the constrained and the unconstrained problems are identical.

4.1.4 Solving the Optimization Problem

Methods that find the minimum are known as solvers and exist in many different flavours. We will here present three methods for solving NLP problems; interval analysis, *sequential QP* (SQP) methods, and the *sequential CP* (SCP) methods.

Interval analysis methods are based on a set of complex mathematical evaluations using interval calculus and can gurantee finding a global minimum, even for non-convex problems, but are typically not performed unless computation-time is a non-issue. Details on interval analysis are found in [25].

SQP methods solve the optimization problem and consists of solving a local QP approximation of the NLP repeatedly until a satisfying solution is found. The approximate QP is based on information from the gradient and Hessian of the Lagrangian function. SQP solvers are obviously not guaranteed to find a global solution, as the approximations can lead to entirely false conditions or feasibility problems. A typical problem is reaching a local minimum and not being able to step out of it and eventually assuming that it is the final minimum. The approximation can also lead to feasibility problems and problems that are impossible to solve, e.g., problems where the solver steps between points eternally without finding a satisfying solution. This last problem is usually solved by limiting the number of allowable steps. If the analytical gradients and Hessians are not supplied, it is possible to approximate them with numerical derivatives, but the solvers are more accurate, faster, and have smaller computational demands when analytical gradients and Hessians are supplied. Both trust-region methods and line-search methods can be applied to solve the local QP problem. See for instance Chapter 3 in [62] or Chapter 12 in [61] for more on SQP solvers.

Line-search and trust-region methods are iterative QP solvers, i.e., they are used to find the minimum of QP problems by stepping from one point to another towards the final solution. Line-search methods work by finding a descent-direction from the objective function and finding the optimal step-length along this direction. Conditions such as the Wolfe conditions or the Goldstein conditions are used to aid the search for the step-length.

Trust-region methods, as line-search methods, approximate the objective function with a quadratic function. This approximation is assumed valid for a certain region, known as the trust-region. Based on the quality of the approximation, the region is expanded or contracted. When the region is found, a solution is found within the region by calcualting both a direc-

tion and a step-length. Since they only search within the trust-region, trust region methods are also known as restricted step methods. Figure 4.4 illustrates the difference in how line-search methods and trust-region methods find the next point during an optimization.

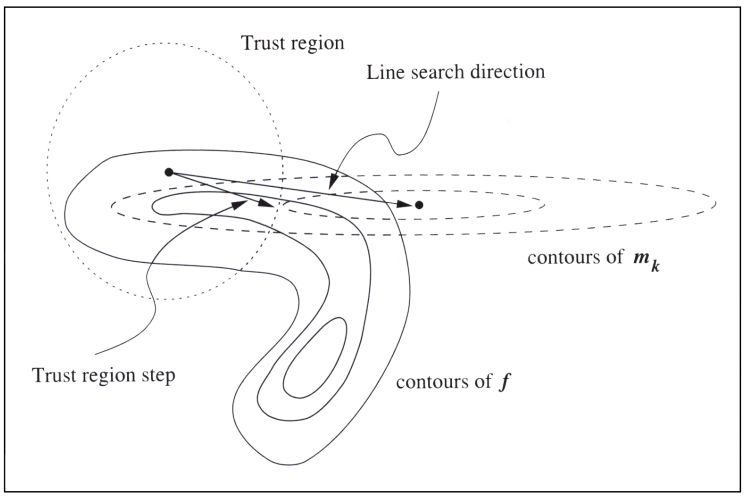

Figure 4.4: The illustration shows the calculated step found from a line-search and a trust-region method. The objective function f is approximated with the quadratic model m_k. Courtesy of [43].

SCP methods apply a similar approach to SQP, by approximating the NLP problem as a general convex problem. As previously mentioned, QPs are convex, meaning that SQP methods are actually included in the SCP framework. SCP methods besides SQP are not that commen, but they have found applications in large scale problems. For more on SCP methods see [11] or [66].

4.2 Optimal Control

Optimal control is about quantifying performance in a objective function and seeking to maximize it, where performance is typically quantified as a cost depending on the state's deviation from reference values and the use of or changes in input. Maximizing the performance therefore means finding the minimum value of the objective function. Since the performance is regarded over a horizon, i.e., a time range, optimal control offers perspective as well. Constraint handling is the final selling point for optimal control methods using online optimization. The controller will use its knowlegde of the system model and its constraints to assure that the system will behave in an allowable fashion. But some problems must be considered, optimiza-

tion takes time. Predictions (simulations) must be made and the actions maximizing the performance must be found. Luckily, advances in computing makes this problem smaller and smaller. Another issue is modeling, prediction is useless if the model is poor. Another point of view is that a predictive controller can get much more out of a good model than for instance PID-controllers, i.e., predictive control makes modeling worth the effort.

4.2.1 Development of Optimal Control

The development of optimal control is highly related to the development of optimal filtering, which is the topic of Chapter 3. It started with Wiener's work in the 1940s and reached maturity in the 1960s with R.E. Kalman's development of *Linear-Quadratic-Gaussian* (LQG) control [30],[58]. LQG is a linear optimal control problem that includes state estimation. The estimation was done using Kalman filtering, there denoted as *Linear-Quadratic-Estimation* (LQE) and the control was done in a deterministic setting with a *Linear-Quadratic-Regulator* (LQR). The LQR control problem consists of finding the optimal control input $\boldsymbol{u}$ to

$$\min_{\boldsymbol{u}} J_k = \lim_{n\to\infty} \int_{t=0}^{n} (\boldsymbol{x}-\boldsymbol{x}_r)^T \boldsymbol{Q}(\boldsymbol{x}-\boldsymbol{x}_r) + \boldsymbol{u}_k^T \boldsymbol{R}\boldsymbol{u}_k \quad \mathrm{d}\,t \tag{4.12}$$

subject to:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}. \tag{4.13}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are positive-semidefinite and positive definite respectively. This unconstrained, no constraints are placed on the actual states nor the inputs, optimization problem has a simple optimal and stabilizing solution[1] given by $\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x}$ where the time-varying $\boldsymbol{K}$ is found from solving the algebraic Riccati equation [23] or simply computing a sub-optimal constant gain matrix using pole-placement techniques. Anyhow, the LQR control problem does not depend on on-line computations. Aerospace engineering, including the Apollo program, took great advantage of this and implemented LQC from the very beginning, while other industries, and particularly the process industry, did not develop many LQC applications [58]. The process industry struggled to fit their nonlinear models into the linear assumptions, leading to poor robustness, and they wanted more flexible performance criterions. But, most importantly they saught a control scheme that would handle inequality constraints directly [50]. These problems lead to the process industry discarding the theoretically developed stabilizing LQR control and instead developed their own heuristic methods.

[1] An infinite horizon controller can never be optimal if it is not stable!

The first online optimization based control implementation came in 1959 at a Texaco refinery in Texas where a computer calculated optimal set-points online [42]. The computational power was naturally limited and the optimization was only performed once every few hours using steady-state models, while the actual set-point tracking was performed by a lower layer of controllers. Other early implementations include ethylene units and oil refinery processes. The industry saw the potential in this new computer assisted control scheme. "The objective function was generally an economic one, but we had the flexibility to select alternative ones if the operating and/or business environment suggested another, e.g., maximize ethylene production, minimize ethylene costs, etc. We were getting the tools to be more sophisticated and we took advantage of them where it made economic sense." [3].

Further implementations were developed by the process industry, and in the late 1960s [51] presented an *model predictive control* (MPC) algorithm that is essentially the same as those currently applied. As the computing capacity and power grew, the first MPC implementations appeared in process plants during the 1970s, and today MPC stands as one of the most widely implemented advanced control technology for process plants [62].

4.3 Model Predictive Control

Model predictive control is a control strategy that involves on-line solving of an optimization problem. The system dynamics during a prediction horizon is simulated using a known model and a set of inputs are found as the solution to a constrained open-loop optimal control problem. The calculated inputs are applied and the optimization problem is solved a new when new state knowlegde appears. As a result, only the first part of the calculated input from each optimization is used, which provides a robust feedback scheme. Since the inputs are found from an on-line optimization problem, MPC supports constraint handling.

Linear MPC is used to refer to MPC implementations described by linear system dynamics, quadratic objective functions, and linear constraints, while the term *nonlinear MPC* (NMPC) usually refers to MPC implementations using nonlinear process models. Linear MPC will form a convex QP problem, but NMPC will always be associated with an NLP problem. This difference is essential when it comes to calculating the optimal inputs and three important problems related to NMPC are presented in [42]:

- "Nonlinear programming, required for the solution of the MPC on-line optimization problem, does not produce exact solutions but rather

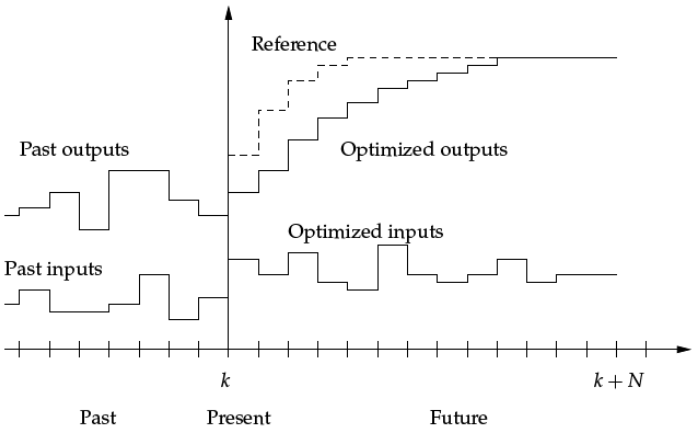

Figure 4.5: The figure displays the MPC principle. Courtesy of [23].

solutions that are optimal within a certain prespecified precision tolerance, or even locally optimal if the optimization problem is nonconvex."

- "Even if the global optimum of the on-line optimization problem is assumed to be exactly reached, MPC behavior may show patterns that would not be intuitively expected. For instance, [53] discuss two simple examples of MPC applied to nonlinear systems, where the state feedback law turns out to be a discontinous function of the state, either because of stability requirements, or due to the structure of MPC. As a result, standard stability results that rely on continuity of the feedback law cannot be employed."
- "A finite prediction horizon may not be a good approximation of an infinite one for nonlinear processes."

It is therefore not recommended to implement NMPC if the nonlinearities are not severe or the process does not operate in several steady states characterized by differing dynamics [23], [49].

The typical MPC optimization problem is stated as

$$\min \sum_{k=1}^{n_p} (\boldsymbol{x} - \boldsymbol{x}_r)_k^T \boldsymbol{Q}_k (\boldsymbol{x} - \boldsymbol{x}_r)_k + \boldsymbol{u}_k^T \boldsymbol{R}_k \boldsymbol{u}_k \tag{4.14}$$

subject to:

$$\boldsymbol{c}_i = 0, \quad i \in \mathcal{E} \tag{4.15}$$

$$\boldsymbol{c}_i \leq 0, \quad i \in \mathcal{I}, \tag{4.16}$$

where the constraints typically represent control demands, e.g., maximum allowed deviation, and given physical relations, e.g., the system model, maximum thrust levels, or valve openings. Unconstrained optimization problems are generally easier to solve than constrained optimization problems so adding constraints will affect the calculation time.

The prediction will be for a finite horizon since infinite-horizon prediction is generally not tractable in finite-time. The predictions far into the future are also of small value since they most certainly will not be as accurate as the first part of the prediction due to model mismatch, numerical issues, etc. It is a rule of thumb to make the prediction horizon long enough to let all the dynamics of the process stabilize after an input-step.

The Objective Function

The objective function can be considered as a general performance index, where decreasing values indicate improved performance. We will here consider only objective functions of the form

$$J_k = \sum_{k=1}^{n_p} \boldsymbol{x}_k^T \boldsymbol{Q}_k \boldsymbol{x}_k + \boldsymbol{u}_k^T \boldsymbol{R}_k \boldsymbol{u}_k, \tag{4.17}$$

where $\boldsymbol{Q}$ is positive-semidefinite and $\boldsymbol{R}$ is positive-definite.

Prediction Shooting Strategies

The prediction shooting strategy is fancy name for how the equality constraints arising from the system model are handled, i.e., assuring that $\boldsymbol{x}_{k+1} - \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) = 0, k \in \mathcal{N} \cap k \leq n_p$. Two different approaches exist; single shooting methods and multiple shooting methods [10].

The single shooting approach is also known as sequential approach, reduced space approach, and feasible path approach and works by solving the prediction as a single initial value problem, i.e., given an initial state value and a set of future inputs, the future states are found by a single shooting, i.e., simulation. It is known as a feasible path approach because its implicit handling of the model constraints assures model feasibility at all times during the optimization. This property can be exploited in implementations if the maximum allowable calculation time it is approaching. Another advantage of this approach is that no added optimization variables, nor constraints, are needed to handle the model constraints. One disadvantage with this approach though, is the optimizer's lack of control over the simulated states, which could be troublesome for unstable modes.

The second shooting method is the multiple shooting method which is also known as full space approach and collocation. This approach implements all states in the prediction horizon, $\boldsymbol{x}_k, k \in \mathcal{N} \cap k \leq n_p$, as optimization variables and model feasibility is assured by adding the model constraints, $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) = 0, k \in \mathcal{N} \cap k \leq n_p$, as explicit optimization constraints. This approach leads to a potentially large increase of optimization variables and model feasibility of the states are only guaranteed when the algorithm has converged. The advantages of this model include; the generally simpler formulations of the objective and constraint functions as well as easier calculations of their respective gradients and Hessians compared with single shooting, and the optimizers' increased control of the future states, i.e., more intuitive handling of unstable modes.

It is of course possible to combine the aforementioned approaches and apply an approach where the total prediction horizon is divided into a set of shooting intervals. Each shooting interval is then calculated as a single shooting and the end-point constraints are handled with multiple shooting, i.e., a form of multiple single shooting. A computationally advante could occur from such formulations as calculating each shooting interval could be done in parallell. Figure 4.6 illustrates how a multiple shooting or combined approach could appear.

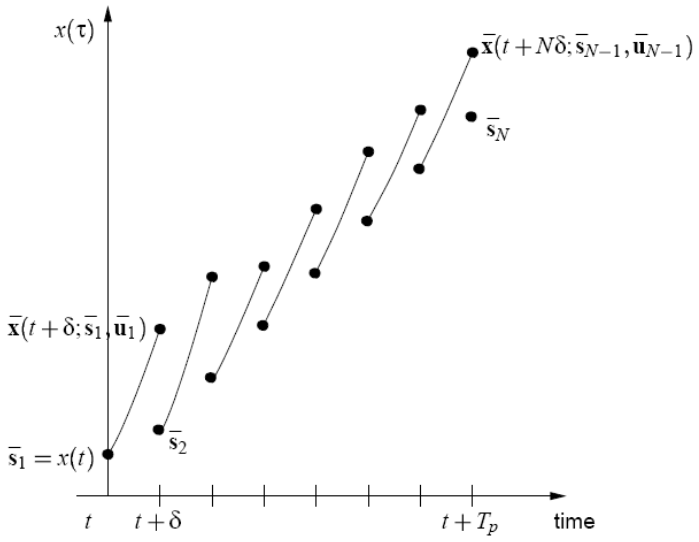

Figure 4.6: The plot illustrates how a multiple shooting strategy depends on the solution of several single shootings. Constraints are added to assure that the endpoints of each shooting are connected. Courtesy of [10].

Discretization

The MPC prediction depends on the selection of discretization method. The approximative discretization is subjected to limited numerical accuracy and these inaccuracies can be influential for the solver. An increase in numerical accuracy is typically attained from increasing the number the function evaluations and simoultaneously decreasing the sample times. This leads to higher computational cost and potentially added states. A comparison of SQP based MPC applications using different numerical integration methods are found in [39] with a comparison of numerical integration methods and different SQP algorithms. A consideration of accuracy vs. consistency is necessary when deciding to use fixed or variable step integrators.

Parameterizing the Inputs

Until now it has never been specified how the inputs $\boldsymbol{u}$ are parameterized in the optimization scheme, where $\boldsymbol{u}$ is given by the vector

$$\boldsymbol{u} \triangleq \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \vdots \\ \boldsymbol{u}_{n_p} \end{bmatrix} \in \mathbb{R}^m, \tag{4.18}$$

where the vectors $\boldsymbol{u}_i, i = 0, 1, ..., n_p$ are the input vectors belonging to step i in the prediction horizon. The dimension of the the vector is given us as the number of system inputs multiplied with the length of the prediction horizon, i.e., $m = n_u n_p$. The most common parameterizing approach is to define every element of $\boldsymbol{u}$ as a free optimization variable. Another approach is to use increment values $\Delta\boldsymbol{u}_i, i \in \mathbb{N} \cap i \leq n_c$ as optimization variables and implement the inputs as

$$\boldsymbol{u} \triangleq \begin{bmatrix} \Delta\boldsymbol{u}_0 \\ \Delta\boldsymbol{u}_0 + \Delta\boldsymbol{u}_1 \\ \Delta\boldsymbol{u}_0 + \Delta\boldsymbol{u}_1 + \boldsymbol{u}_2 \\ \vdots \\ \sum_{i=0}^{n_c} \boldsymbol{u}_i \end{bmatrix} \in \mathbb{R}^m \tag{4.19}$$

.

These two approaches lead to a high number of optimization variables, but blocking methods, i.e., keeping the input constant for a number of samples and keeping the last inputs constant can reduce the number of the variables.

The horizon where the input is allowed change is called the control horizon. A blocked input vector combined with a control horizon which is shorter than the prediction horizon is shown below.

$$\boldsymbol{u} \triangleq \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_{n_c} \\ \boldsymbol{u}_{n_c} \\ \boldsymbol{u}_{n_c} \\ \boldsymbol{u}_{n_c} \end{bmatrix} \in \mathbb{R}^m$$

.

A third possibility is to use parameters $\boldsymbol{p}$ belonging to a state feedback function $\boldsymbol{g}$, including linear feedback, as optimization variables and find the input vector from these as

$$\boldsymbol{u} \triangleq \begin{bmatrix} \boldsymbol{g}(\boldsymbol{z}_0; \boldsymbol{p}) \\ \boldsymbol{g}(\boldsymbol{z}_1; \boldsymbol{p}) \\ \boldsymbol{g}(\boldsymbol{z}_2; \boldsymbol{p}) \\ \vdots \\ \boldsymbol{g}(\boldsymbol{z}_{n_p} 0; \boldsymbol{p}) \end{bmatrix} \in \mathbb{R}^m.$$

Combinations of the various formulations are also possible.

The approaches using direct parameterization and incremental parameterization are most commonly used as they lead to the simplest problem formulations, since they will exhibit a highly structured form, while calculating analytical gradients and Hessians using the feedback parameterization can be quite tricky. On the other hand, the state-feedback approach is usually more robust due to continous disturbance rejection [23].

4.3.1 Solving the NMPC problem

The optimization problem solved by the NMPC controller will be a NLP problem. The solver strategy is often depending on the NMPC formulation Many NMPC formulations are structured and these structures should be made use of when solving the optimization problem. SQP is the preferred solver strategy for NMPC [39], [62], [23], but approaches using interval analysis do also exist [35], [37].

4.3.2 Nominal Stability of NMPC

MPC was initially developed as a heuristic control approach and stability were not considered. The first stability results appeared in [54] in 1993, and they lead to a surge in research evaluating stability and robustness for MPC. It was shown in [6] that closed-loop stability is not guaranteed by sequential finite-horizon open-loop optimal control approaches.

To evaluate the stability of the closed-loop system, we must examine the infinite horizon properties of the system. As the closed-loop system consisting of constrained MPC, including constrained linear MPC, are inherently nonlinear, methods used to show nonlinear stability must be applied. The common approach thereforce considers Lyapunov stability.

Consider a system described by a function $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{f}_k, \boldsymbol{u} + k)$, where $\boldsymbol{f} : \mathbb{R}^n x \mathbb{R}^m \to \mathbb{R}^n$ is continous and $\boldsymbol{f}(\boldsymbol{0}, \boldsymbol{0}) = \boldsymbol{0}$. Consider that the infinite-horizon optimization problem can be separated into a finite-horizon optimization problem and another infinite-horizon problem as

$$\min_{\bar{\boldsymbol{u}}} J_k = \min_{\bar{\boldsymbol{u}}} \sum_{i=0}^{n_p} L(\bar{\boldsymbol{x}}_{k+i}, \bar{\boldsymbol{u}}_{k+i}) + \sum_{i=n_p+1}^{\infty} L(\bar{\boldsymbol{x}}_{k+i}, \bar{\boldsymbol{u}}_{k+i}) \tag{4.20}$$

$$\begin{aligned} &\text{subject to:} \\ \bar{\boldsymbol{x}}_{k+i+1} &= \boldsymbol{f}(\bar{\boldsymbol{x}}_{k+i}, \bar{\boldsymbol{u}}_{k+i}) \\ \bar{\boldsymbol{x}} &\in X \\ \bar{\boldsymbol{u}} &\in U, \end{aligned}$$

where $\boldsymbol{L}$ is a positive definite function, and $X \in \mathbb{R}^n$ and $U \in \mathbb{R}^m$ are convex sets containing the origin. We know from Barbalat's lemma that, if $J_k < \lim_{n\to\infty} n$, then $\lim_{n\to\infty} \bar{\boldsymbol{x}}_n = \boldsymbol{0}$, meaning asymptotic stability. Since we consider a nominal system, it is guaranteed that once $\boldsymbol{x}k$ reaches the origin it will stay at the origin if no input is applied. It is therefore sufficient that it is feasible to reach the origin in finite time to show stability.

Showing stability is then simply done by assuming a sufficiently high n_p, but it is simpler to add a terminal constraint demanding that $\bar{\boldsymbol{x}}_{n_p} = \boldsymbol{0}$ if this is feasible. The difference lies in concept as the first relies on tuning, while the second restricts the NMPC formulation. A more general approach than demanding that the terminal state is at the origin, is to add a terminal constraint demanding that the terminal state is contained in some terminal region Ω, i.e., $\boldsymbol{x}_{n_p} \in \Omega$, for which we know a feedback controller $\boldsymbol{g}(\boldsymbol{x}_k)$ that assures that the closed-loop (using the known feedback controller) system is locally exponentially stable. If either of these terminally constrained

approaches are feasible at a time k, it can be shown that $J_{k+1} \leq J_k$ by simply shifting the previous feasible input vector and applying Bellmans optimality principle. As such there is no need to consider the residual of the infinite-horizon optimization as it is guaranteed to be finite. The optimization problem can then be formed as

$$\begin{aligned} \min_{\bar{\boldsymbol{u}}} J_k &= \min_{\bar{\boldsymbol{u}}} \sum_{i=0}^{n_p} L(\bar{\boldsymbol{x}}_{k+i}, \bar{\boldsymbol{u}}_{k+i}) \qquad (4.21) \\ \text{subject to:} & \\ \bar{\boldsymbol{x}}_{k+1} &= \boldsymbol{f}(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k) \\ \bar{\boldsymbol{x}} &\in X \\ \bar{\boldsymbol{x}}_{k+n_p} &\in \Omega \\ \bar{\boldsymbol{u}} &\in U, \end{aligned}$$

which is called a quasi-infinite-horizon NMPC formulation.

Another possible way to guarantee nominal stability is by adding constraints that demands that the input-output system fulfills some passivity demands [52].

No general separation theorem for nonlinear control exists, which basically means that every combination of estimation and control has to be evaluated when examining the stability properties of nonlinear output feedback approaches. Some recent work on the subject of output-feedback NMPC has been done in [24], and several others by the same authors, but considering only a class of nonlinear observers named high-gain observers. Estimators based on Kalman filtering and receding horizon estimation are not included amongst them and no such similar proofs exist.

4.3.3 Robust Stability

Robust stability means that the system is stable for some perturbations, i.e., model uncertainty. This uncertainty can be due to either unknown parameter values, unknown disturbances, or some unknown model structure. Robust stability is shown by assuring that $J_{k+1} \leq J_k$ even when considering the unknown, but bounded, perturbations. Assuring $J_{k+1} \leq J_k$ is done either through tuning or by adding constraints.

It is unrealistic to assume that a nominal model is known when MPC is applied in practice and the MPC must show some robustness to achieve good control performance[2]. As the MPC is based on optimal control, it has

[2]This is true for all control strategies, and not only NMPC.

some inherent robustness. Unconstrained MPC can even be shown to have gain and phase-margins [38].

Several approaches to achieve robust stability exist. One method is to solve an open-loop min-max problem [42], i.e., formulate the optimization problem as

$$\min_{\boldsymbol{u}} \max_{\boldsymbol{p}} J(\boldsymbol{u}, \boldsymbol{p})_k, \tag{4.22}$$

where $\boldsymbol{p}$ are parameters that express the model uncertainties. This entails finding the worst-case dynamics or a set of models defined by the perturbations and then using the worst-case model in a normal MPC scheme. Conservative results and feasibility issues should be expected as the calculated inputs have to be feasible for all possible seets of models. A less conservative approach is using H_∞ methods, [58], but they are very costly and require finding a global minimum.

Methods using interval analysis can be shown to provide a guaranteed outer bound on the reachable states and a robust MPC scheme guaranteeing convergence [35].

As previosly mentioned, it is also possible to apply continous feedback control by using the parameters of a feedback as optimzation variables in the MPC problem. Continous feedback control is more robust than discrete outputs because it provides continous disturbance rejection.

4.3.4 Feasibility

MPC relies on online solving of an optimization problem and severe problems can occur when no feasible solution can be found. Feasibility is defined in relation to the hard constraints. Feasibility problems are usually solved by relaxing constraints, but optimal constraint relaxation is an NP-hard problem [42]. Penalty functions are introduced in [43] and [10] as a way ensure that hard constraints are only relaxed if necessary due to feasibility problems.

It is highly advantageous to be able to prioritize which constraints should be relaxed first, although all hard constraints should be modeled as hard for a reason. As it is futile to assume that model constraints or other constraints imposed due to physical relations can be relaxed, this basically means that the hard constraints that are implemented due to for instance human safety, environmental considerations, or product quality, must be relaxed. If these are truly hard, it could be a better approach to shut down the controlled system or initiate other extreme measures instead of relaxing the constraints.

Chapter 5

Simulations and Discussion

> Anyone can hold the helm when the sea is calm.
>
> Publilius Syrus

5.1 Simulation Model

The vessel dynamics are modeled as in Section 2 with 2. order modulus damping. Wave disturbances are implemented using only the Froude Krilov forces as the wave-drift is assumed to be contained in the current. Other disturbances or noise are not added.

5.2 Simulations

5.2.1 Parameter Estimation

The linear and nonlinear damping, described by Abkowitz damping model, are estimated using nonlinear least squares method from Tomlab package and Dual UKF. The parameters that are not dependent on both sway speed and yaw rate are found using nonlinear least squares while the four parameters Y_vvr, Y_vrr, N_vvr, and, N_vrr are found using a dual Kalman filtering approach. A PID controller applied reference tracking control for the vessel and only current disturbance is included.

Figures 5.1 and 5.2 show how the least-squares estimated Abkowitz formulations of the damping forces and moments, compare with the 2nd order

modulus formulation provided from [57]. A linear approximation is also seen. The weighted least-squares estimation requires higher accuracy for lower velocities than for higher velocities.

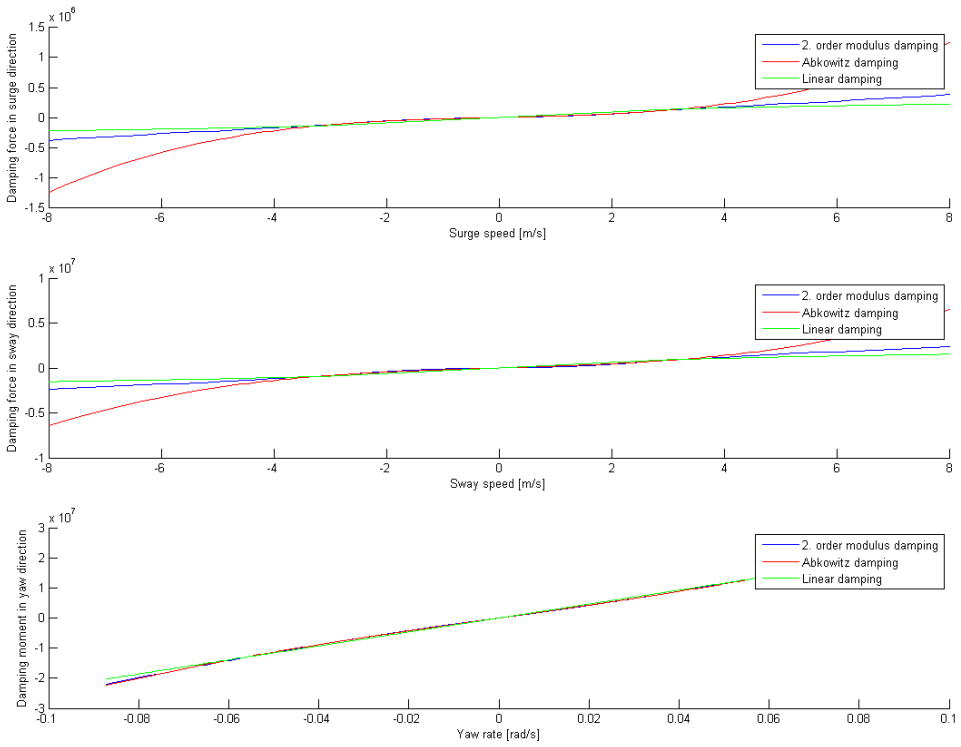

Figure 5.1: Decoupled Abkowitz parameters found using least squares method.

The coupled parameters $Y_v vr$, $Y_v rr$, $N_v vr$, and, $N_v rr$ were found using a dual Kalman approach explained in Chapter 3. The blue dotted line (UKF1) is the vessel state estimates found using the dual UKF estimation, while UKF2 is an UKF estimate using the nominal model.

A fixed diagonal matrix description was used $\boldsymbol{R}_r$ and as 5.5 shows, reasonable estimates were obtained even from this poorly excitated experiment after about 100 samples. Figure 5.4 shows that the current estimate does not converge to the true value, since the vessel estimates are so good this implies that the correct damping parameters are not found, this is most likely due to poor excitation.

5.5

The Figures 5.4 and 5.5 also compares the UKF, given as (UKF2), the

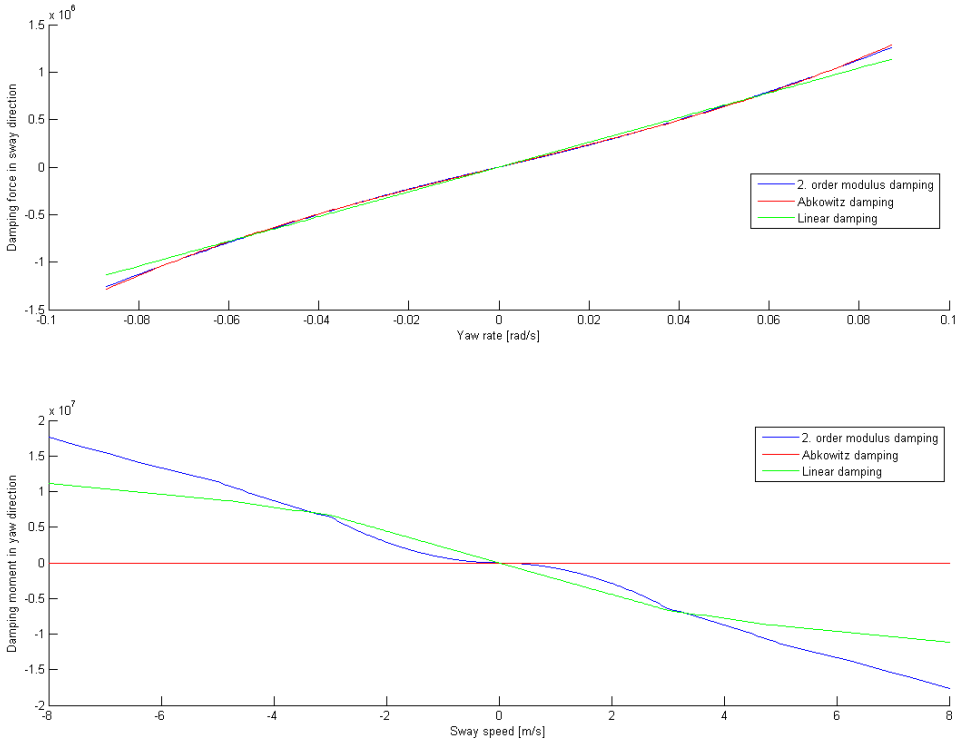

Figure 5.2: Coupled Abkowitz parameters found using least squares method.

green dashed line and a EKF, the red dashed line. Both of these perform estimation using the 2nd order modulus damping model. And it is seen the EKF fails horribly since it does not manage to estimate the current velocity. The reason for this is shown in Chapter 3.

5.2.2 NMPC

Two simulations showcasing the failure of the NMPC controller is provided. The first is a simple pathfollowing operation where constraints are placed on the forces and moments and limiting them to

$$\begin{bmatrix} -10^8 \\ -10^8 \\ -10^9 \end{bmatrix} \leq \boldsymbol{u}_k \leq \begin{bmatrix} 10^8 \\ 10^8 \\ 10^9 \end{bmatrix}. \tag{5.1}$$

No disturbances are applied during the simulation, but the NMPC receives

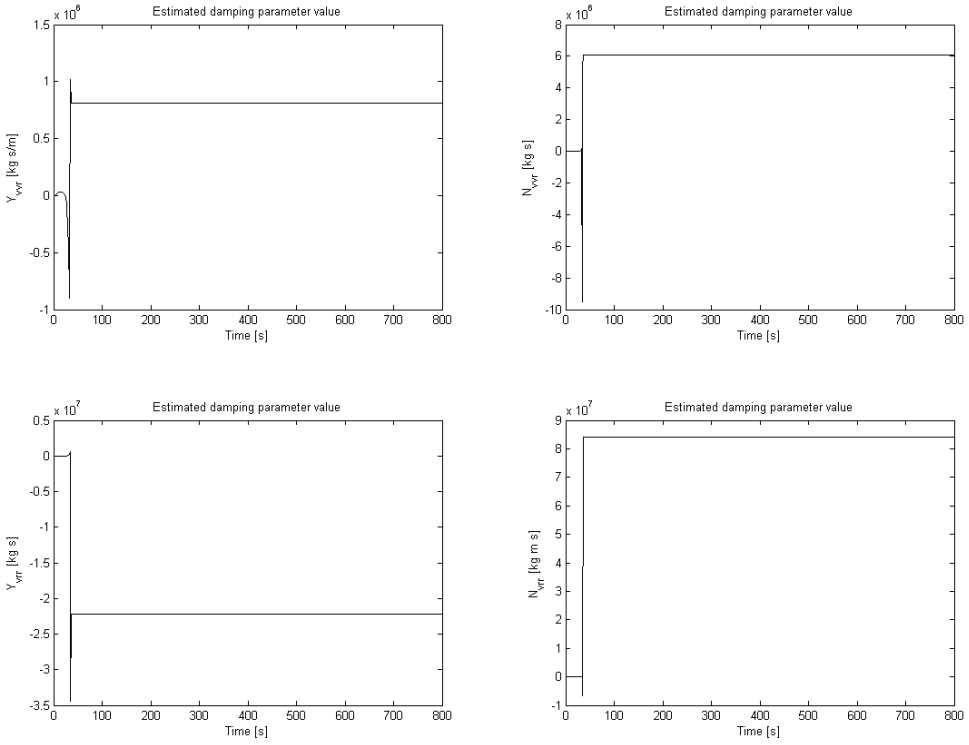

Figure 5.3: Coupled Abkowitz parameters estimated by dual Kalman filter.

its state knowlegde from a UKF estimation. This estimation performed flawlessly, due to lack of disturbances and general unmodeled dynamics, and is therefore not considered to be a source of error. The model implemented in the NMPC is the 2. order modulus model.

Figure 5.6 shows that the NMPC controller fails to follow the path and instead simply follows another straight path with a surge speed which is about 3 times higher than commanded. Figure 5.7 shows that the calculated thrust force is constant during the entire simulation period and it is blatantly obvious that this is not the optimal input. This wrong input occurs either because of errors in my implementation, or from poor solutions due to lack of robustness in the fmincon solver. It is obvious that no feedback effect has been included. These poor solutions can stem from the solver getting "stuck" in a local minimizer and have been able to escape from there.

Figure 5.8 shows a similar failed control operation where the controller was supposed to perform station keeping when subjected to wave motion. The model used in this simulation has linearized dynamics and it is as such not

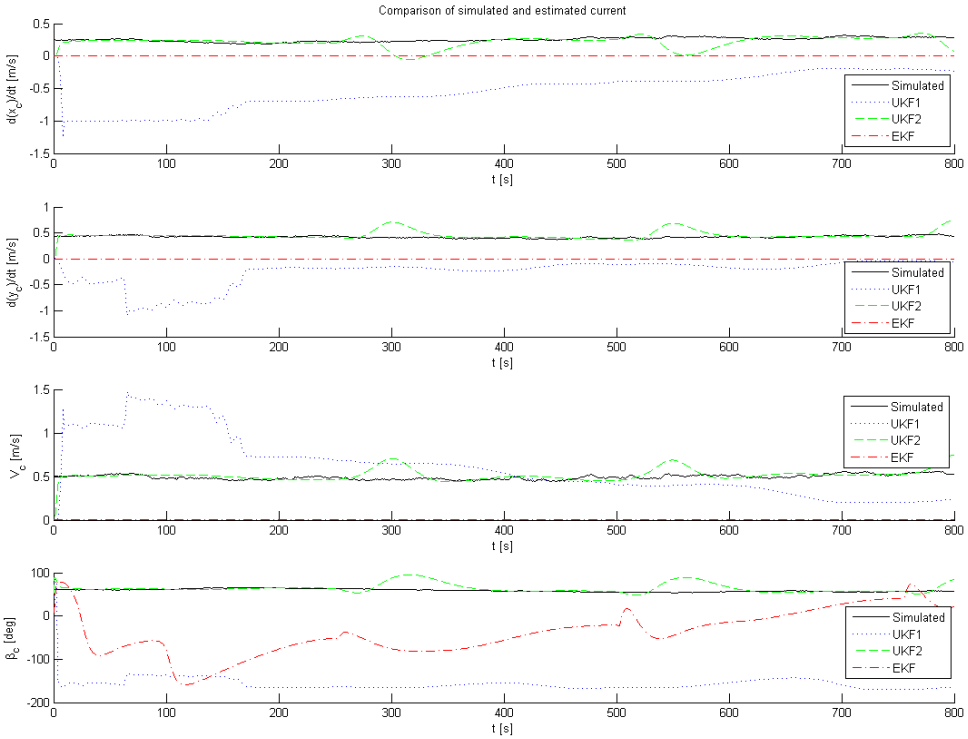

Figure 5.4: Current estimated by dual Kalman filter.

the discontinous damping matrix that causes the poor optimization.

5.2.3 Wavefiltering

Figures 5.9 and 5.10 show how the UKF estimator is capable of estimating correct current values and perform wave filtering when the wavedirection is 180 degrees and the current direction comes from 60 degrees. Only a small deviation in angle is observed. The wave filtering is also performed well as is seen in 5.9. No wave model is implemented here, because the best estimation results when modeling only current disturbance. When the wave model was included the bias force estimation was very poor and made the current estimate deviate. Only the Froude Krilov forces are used as it is assumed that wave drift is absorbed by the current and thus appear as a single disturbance.

Figures 5.11 and 5.12 show that the UKF estimator is unable to correctly separate the current and the wave disturbance when they appear from the

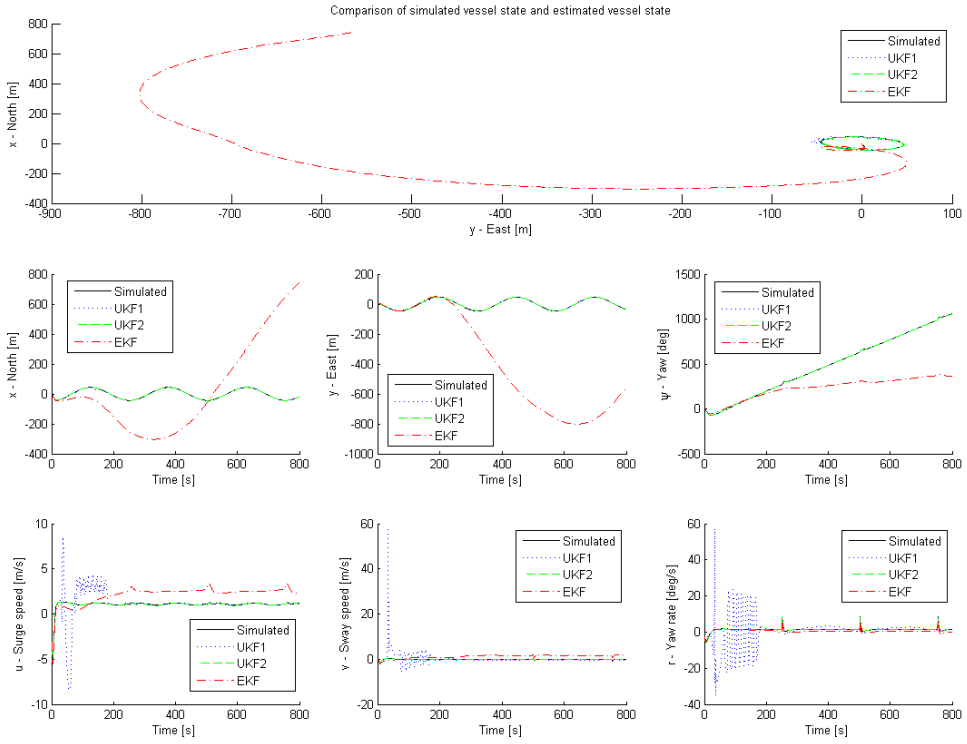

Figure 5.5: Vessel state estimated by dual Kalman filter.

same direction. The wave filtering still works but the deviations in the current estimate leads to deviations in the velocity and state estimations.

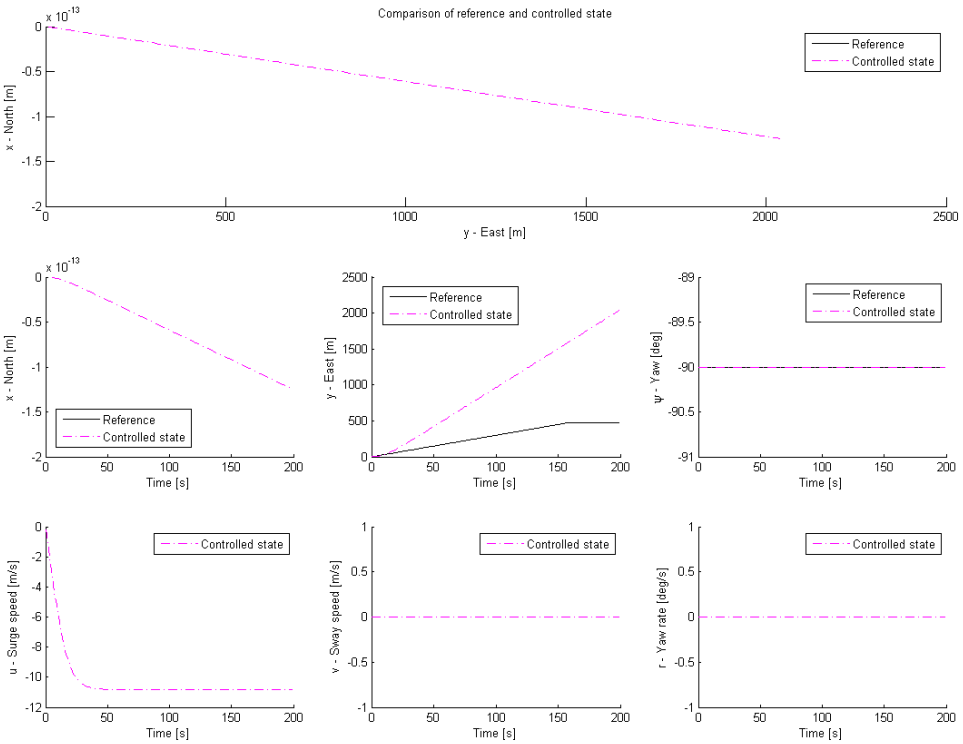

Figure 5.6: Vessel state controlled using NMPC. The vessel is trying to follow a straight line.

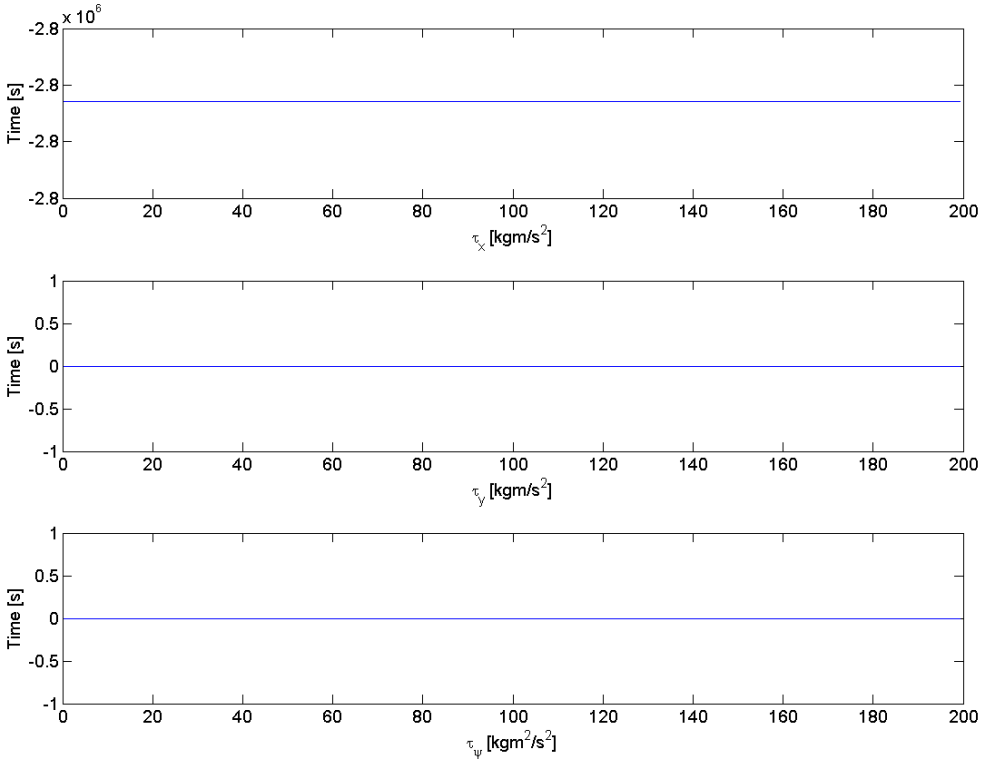

Figure 5.7: Figure shows the thrust values calculated by the NMPC algorithm when trying to follow a straight line.

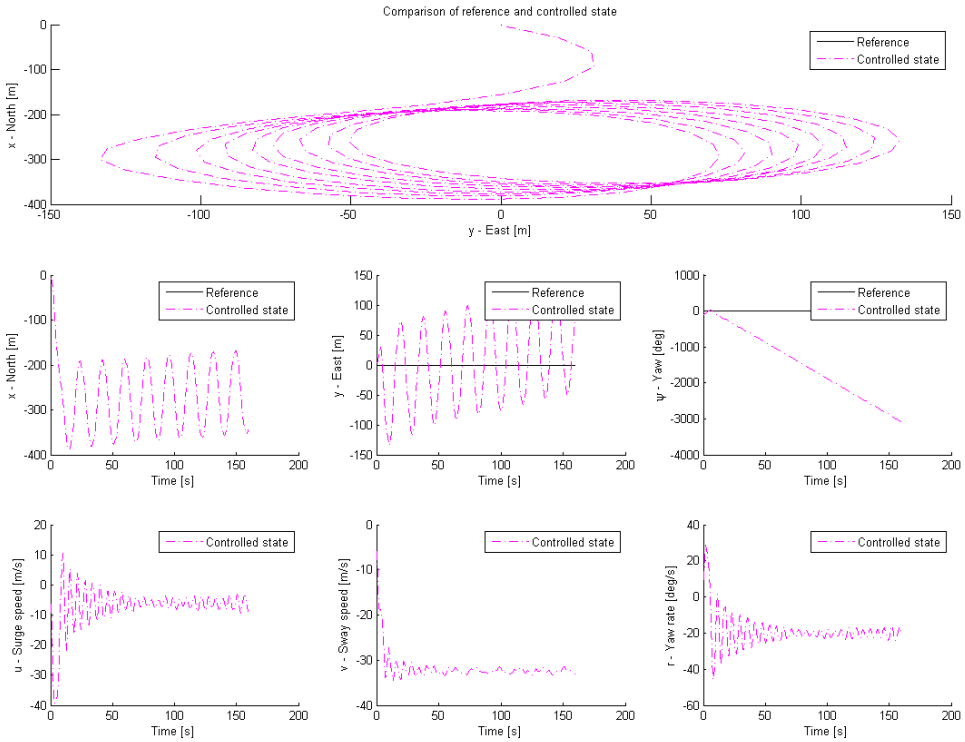

Figure 5.8: Figure showing poor control performance of NMPC during SK mission. The vessel was subject to implemented hard constraints.

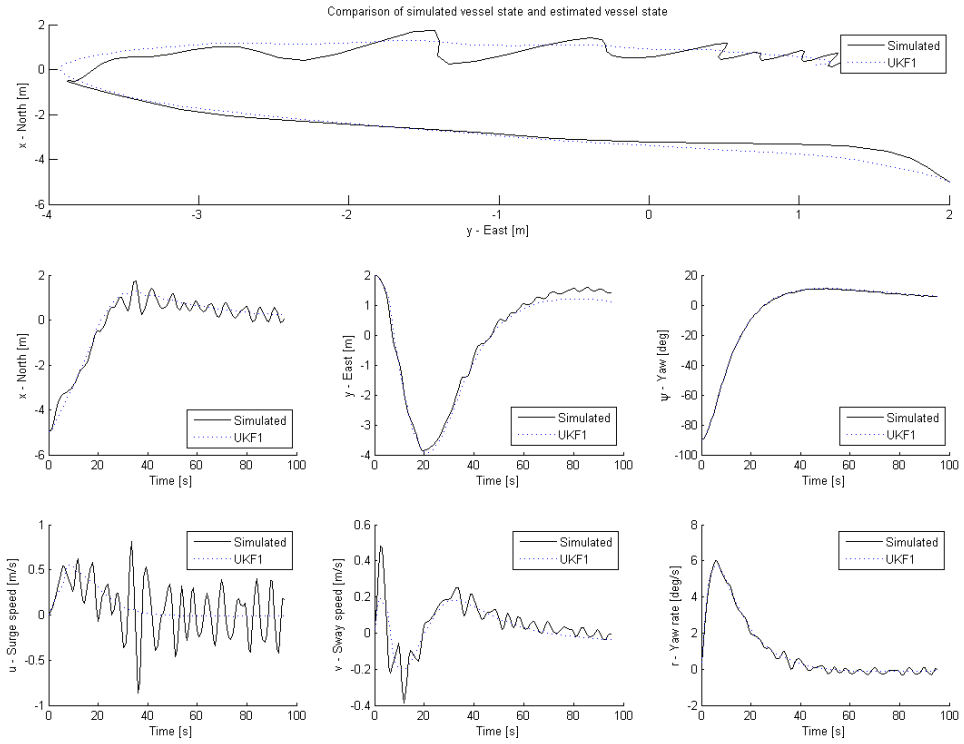

Figure 5.9: Vessel state estimated by UKF. The vessel is subjected to current and wave disturbances coming from separate directions.

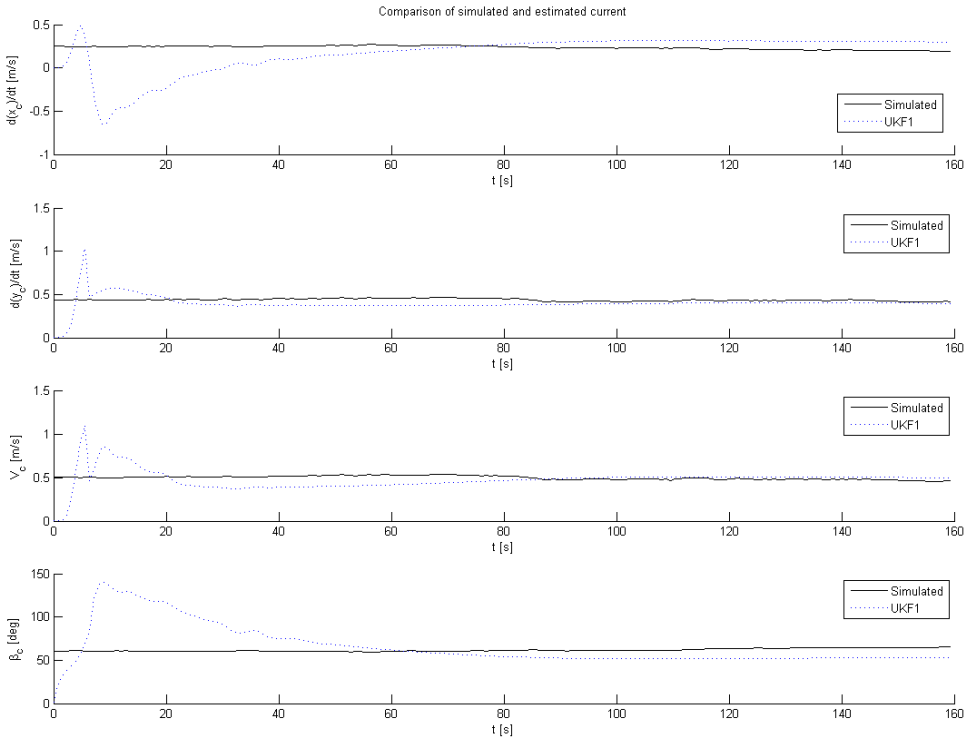

Figure 5.10: Vessel state estimated by UKF. The vessel is subjected to current and wave disturbances coming from separate directions.

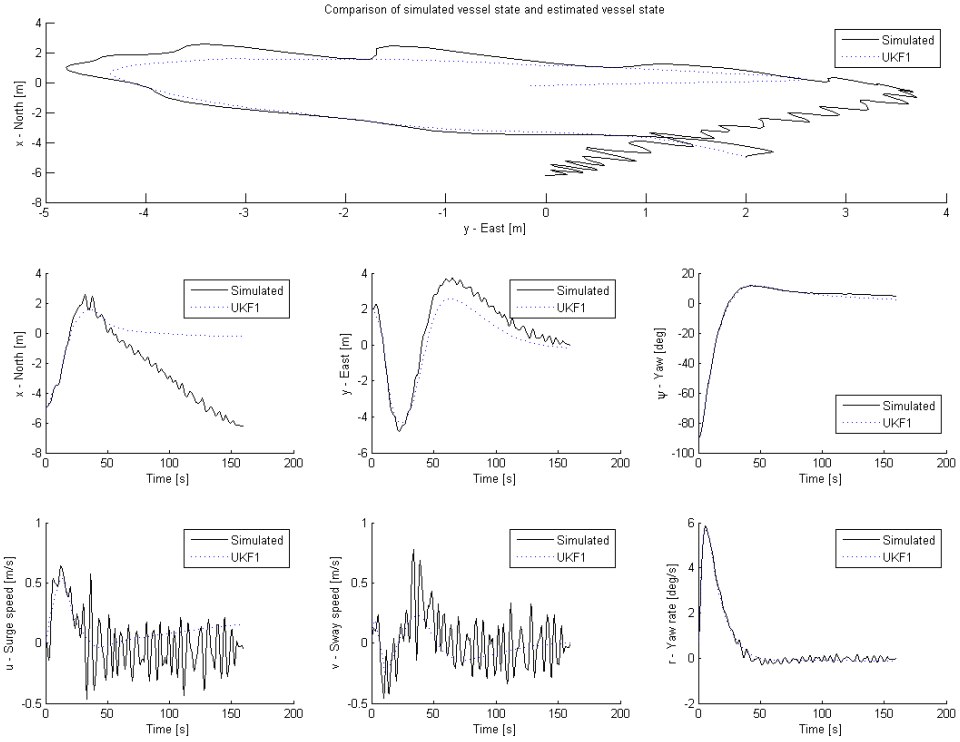

Figure 5.11: Vessel state estimated by UKF. The vessel is subjected to current and wave disturbances coming in the same direction.

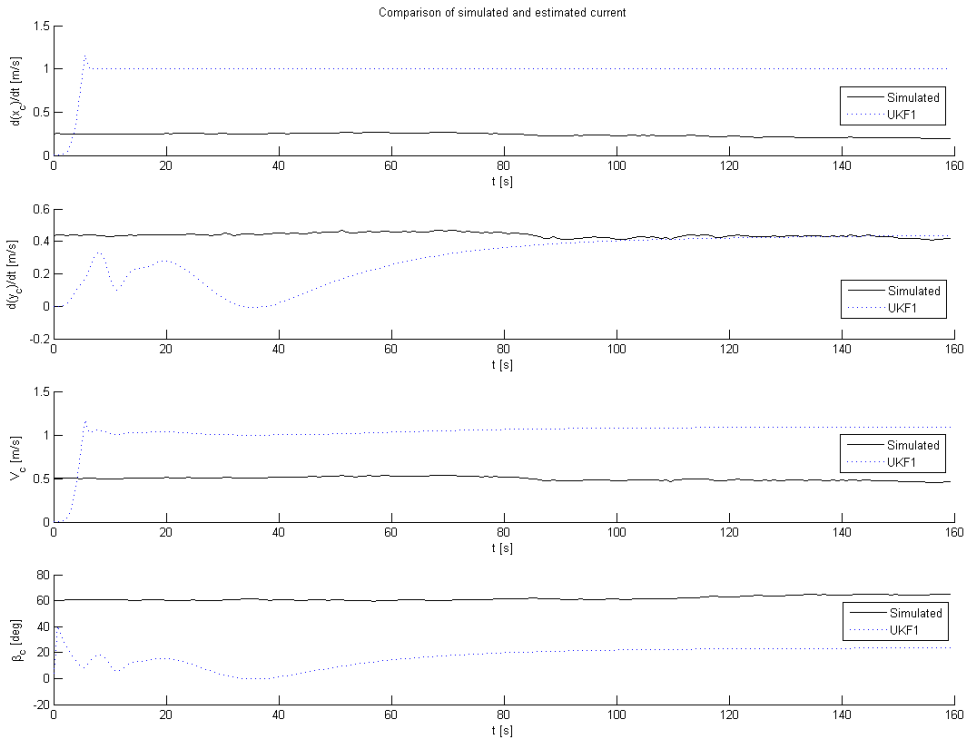

Figure 5.12: Estimate of current found by UKF. Current and waves share direction

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The Unscented Kalman filter and the extended Kalman filter have been compared both theoretically and with simulations. The UKF has consistently outperformed the EKF in all categories. The UKF is much simpler to implement for nonlinear systems as it has no need for the linearized model equations which can be quite hard to calculate for large systems. It is simpler to tune because of its superior handling of the stochastic propagation. Higher accuracy in the calculation of the transformed mean and covariance also reduces the need for increasing robustness by adding fictitious noise, in comparison with the EKF which has to include such noise to overcome the errors developing from the linearized approach to mean and covariance propagation. When little or no fictitious noise have to be added the quality of the estimate also improves. Another problem with the EKF relating to stochastic decoupling of states is identified and simulations show that the UKF has no such problems. Parameter estimation attempts using dual UKF shows there are still more to get from the UKF.

State estimation was performed on a horizontal vessel model to estimate the velocities which are not available through measurements and to estimate the disturbances due to current and wave excitation. The disturbances due to wave excitation are divided into a rapid 1st order oscillating motion and a slower 2nd order drift force, while the effects of the current are appearing due to viscous hydrodynamic damping. It is observed that the slow effects stemming from wave drift and the current are difficult to separate if they act in the same direction. And this will be an even more difficult task for practical considerations where much larger uncertainties, especially regarding hydrodynamic damping coefficients will appear. It has been shown that the estimation of the current velocity through the hydrodynamic damping forces

is possible using the UKF, but impossible using the same system model if an EKF estimator is used. Wave-filtering effects appear in the UKF without the need for modeling the WF motion as a spring damper system.

A control approach using NMPC was reviewed and found to be exciting, but implementations failed as the solutions to the on-line optimization problem were extremely poor. If better solvers were available it is the authors honest belief that the NMPC controller would be a very promising controller for DP operations. NMPC is model based and uses the model to optimize the future performance. This optimization involves a prediction which can be used for feedforward approaches. A DP system developed by Kongsberg Maritime called greenDP uses a NMPC controller and its main advantages have been the possibility to combine reduced thrust use and changes in thrust while still assuring, through the use of state constraints, that the vessel will stay inside a selected region. The prediction strategy lets the controller see the big picture and act calmer instead of using bulls-eye control using rapidly changing thrust and much fuel. A problem that have not been considered here, is how much the NMPC scheme will deteriorate when it is implemented in real life with much poorer model knowlegde than was available in these attempted simulations. If a robust NMPC scheme could be implemented, the NMPC could be imagined being a unified controller as it shows great flexibility in defining the performance function and including state dynamics, i.e., the operations which are now performed by separate types of controllers could all be performed by an NMPC. A final problem with NMPC is the calculation time, this could be difficult problem depending on the length of the prediction horizon and the number of optimization variables.

6.2 Future Work

There are several aspects that should be considered for future work. The most important being implementing a NMPC controller that actually works.

- A functioning NMPC algorithm should be implemented. A better solver must be found. Possible one using analytical gradient information in a better way then fmincon.
- Try to extract information from the predictions to implement feedforward control.
- Combine optimization and thrust allocation. As of today these are two separated tasks, both depending on optimization. Could they be combined?

- Implement a fast solver so the NMPC problem can be solved in real-time
- Compare the accuracy of linear MPC with NMPC for this system. Is nonlinear control necessary? Which nonlinearities are dominating?
- A decoupled approach using one NMPC for surge control and one for controlling yaw and sway should be considered as it could drastically reduce the number of optimization variables
- A continous model of the vessel dynamics should be found and implemented, i.e., find abkowitz damping paramters.

6.3 State Estimation

- Improve the simulations with regards to parameter values etc. Test for model mismatch.
- Consider auto-tuning methods for the filter
- Implement faster square-root versions of the UKF
- Compare 3 DOF model with 6 DOF model

Appendix A

Stability

A.1 Definitions

Stability is loosely defined as boundedness. A system is said to be stable if its states will stay bounded until the end of time. Nominal stability is used to express that a system is stable with no model uncertainty and robust stability implies that the system will be stable for also when there are some uncertanties. The uncertainty can appear both in parameter values and formulation. In addition two kinds of stability definitions exist, relating to forced and unforced system. A system is unforced if no exogenous inputs are applied, i.e., the system can be modeled as $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x})$. This includes the special case of state-feedback systems, since their inputs can be described as $\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{x})$. The two stability definitions are essentially equivalent for linear systems, but separate considerations must be made when nonlinear systems are considerinm.

Some definitions are needed to study stability. Consider a vector $\boldsymbol{x} \in \mathbb{R}^n$. The norm $\|\boldsymbol{x}\|$ is a function that satisfies:

- The norm of $\boldsymbol{x}$ is zero only when all elements of $\boldsymbol{x}$ are zero and is otherwise positive definite, i.e., $\|\boldsymbol{x}\| > 0$ for all $\boldsymbol{x} \neq \boldsymbol{0}$ and $\|\boldsymbol{x}\| = 0$ for all $\boldsymbol{x} = \boldsymbol{0}$.
- The norm of a scaled $\boldsymbol{x}$ is equal to the norm of $\boldsymbol{x}$ multiplied by the absolute value of the scale, i.e., $\|k\boldsymbol{x}\| = |k|\|\boldsymbol{x}\|$.
- The norm satisfies the triangle equality, i.e., $\|\boldsymbol{x}_1 + \boldsymbol{x}_2\| \leq \|\boldsymbol{x}_1\| + \|\boldsymbol{x}_2\|$.

The p-norm of a vector is a general expression of it's magnitude defined by

$$\|\boldsymbol{x}\|_p \triangleq \left(\sum_{k=1}^{\infty} \|\boldsymbol{x}_k\|^p\right)^{\frac{1}{p}}, \tag{A.1}$$

where $1 \leq p \leq \infty$ and $\|\boldsymbol{x}_k\|$ is any norm in $\mathbb{R}^n$.

A.1.1 Stability for Unforced Systems

Consider the general nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{A.2}$$
$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}). \tag{A.3}$$

If $\boldsymbol{f}(\boldsymbol{x}_e) = \boldsymbol{0}$, then $\boldsymbol{x}_e$ is an equilibrium point. But, it is not known whether this equilibrium point is attracting (stable) or repelling (unstable).

From [32], stability is defined as:

- The equilibrium point is stable if:

$$\|\boldsymbol{x}(0)\| < \delta \rightarrow \|\boldsymbol{x}(t)\| < \epsilon, \forall t \geq 0 \tag{A.4}$$

- Asymptotically stable if it is stable and δ can be chosen such that

$$\|\boldsymbol{x}(0)\| < \delta \rightarrow \lim_{t\rightarrow\infty} \boldsymbol{x}(t) = \boldsymbol{0} \tag{A.5}$$

- And unstable if it is not stable

A.1.2 Stability for Forced Systems

Consider the general nonlinear system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{A.6}$$
$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}), \tag{A.7}$$

where $\boldsymbol{u}$ is an exogenous input.

A forced system is said to be input-output stable if a bounded input leads to a bounded output. This is expressed mathematically as

$$\|\boldsymbol{u}\|_p < \infty \Rightarrow \|\boldsymbol{y}\|_q < \infty, \tag{A.8}$$

where $1 \leq p \leq \infty$ and $1 \leq q \leq \infty$. Input-output stability is often not satisfactory and finite-gain stability is typically desired. A system is said to be finite-gain stable if $\sup_u \frac{\|\boldsymbol{y}\|_q}{\|\boldsymbol{u}\|_p} < \infty$.

An excellent in-depth study of nonlinear stability is found in [32].

Appendix B

Mechanics

B.1 Rigid-Body Mechanics

Mechanics is a branch of physics including kinematics and kinetics. Kinematics is the study of how an object is affected by forces and especially how motion is initialized and altered. Kinetics is a purely geometrical consideration of motion. A rigid body is an ideal physical object that does not suffer deformation when affected by forces, i.e., the sum of the object's internal forces is zero. In short, rigid-body mechanics describes how forces contributes to the motion of a rigid-body.

B.1.1 Kinematics

Position and motion have to be given relative to a reference frame. A reference frame is a set of axes allowing observation of position relative to a point. This reference frame can be subject to lateral motion, lateral acceleration and rotation. A frame that is neither accelerated nor rotating is known as an inertial frame. This distinction is essential since Newton's laws are only valid in inertial frames. Only Cartesian[1] frames will be considered here.

The physical space is given in three dimensions, typically (x, y, z), and position is defined as a point in this space. Speed is defined as the change of position, acceleration as the change of speed. If objects are considered, opposed to point-particles[2], orientation[3] and the position of the object's point of origin, are needed to describe the volume of points covered by the object.

[1] Cartesian frames are also known as orthogonal frames.

[2] Point-particles are infinitesimally small and therefore cover no volume

[3] Orientation is a set of angles describing how the object is rotated around the axes of the reference frame.

The combination of position and orientation is also known as pose.

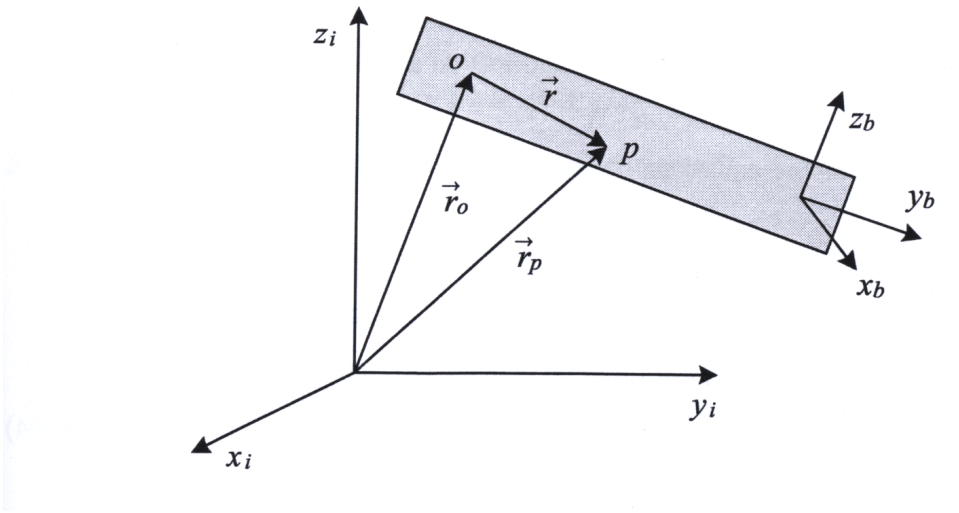

Figure B.1: Illustration depicting an objects pose. Courtesy of [7]

Velocity and acceleration are defined mathematically as

$$\dot{r} \triangleq v \tag{B.1}$$
$$\dot{v} \triangleq a, \tag{B.2}$$

and angular velocity and acceleration as

$$\dot{\theta} \triangleq \omega \tag{B.3}$$
$$\dot{\omega} \triangleq \alpha. \tag{B.4}$$

It is often useful to define several reference frames when modeling a physical system. Subscripts are used to indicate which frame a vector is given in. $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ will then refer to the same vector, but given relative to two separate frames. If frame b is defined as a rotation around the axes of frame a, coordinates can be transformed from a to b with the help of a rotation matrix, $\boldsymbol{p}_b = \boldsymbol{R}_a^b \boldsymbol{p}_b$. Another important property of rotation matrices is that several rotations can be done in succesion by taking the product of the individual rotation matrices. Mathematically this is formulated as

$$\boldsymbol{p}_c = \boldsymbol{R}_c^b \boldsymbol{R}_b^a \boldsymbol{p}_a \tag{B.5}$$

$$\boldsymbol{p}_a = \boldsymbol{R}_a^b \boldsymbol{R}_b^a \boldsymbol{p}_a. \tag{B.6}$$

Transformations due to different points of origin, motion or acceleration are simply done by adding the coordinate, velocity or acceleration given relative to the frame of desired reference. More on transformations can be found in [7] and [13].

B.1.2 Kinetics

The laws of classical mechanics state that changes in an objects momentum (and its rotational counterpart angular momentum), relative to a inertial frame, is equal to the sum of the forces (moments) acting on the object. Mathematically this is expressed as

$$\frac{\mathrm{d}}{\mathrm{d}t}(mv) = \sum_i F_i \tag{B.7}$$

$$\updownarrow \frac{\mathrm{d}m}{\mathrm{d}t} = 0 \tag{B.8}$$

$$ma = \sum_i F_i, \tag{B.9}$$

where the rotational counterpart is given as

$$\frac{\mathrm{d}}{\mathrm{d}t}(I\omega) = \sum_i r_i \times F_i \tag{B.10}$$

$$\updownarrow \frac{\mathrm{d}I}{\mathrm{d}t} = 0 \tag{B.11}$$

$$I\alpha = \sum_i r_i \times F_i. \tag{B.12}$$

The assumptions (B.8) and (B.11) are extensions of the rigid-body assumption.

As mentioned above, Newton's laws are only valid in inertial reference frames. When forces are described in non-inertial frames, fictitious forces[4]

[4] A fictitious force is also known as a pseudo force or as a d'Alembert force and can be applied for cases where the reference frame is accelerating as well as rotational, as rotation is just a form of acceleration.

must be added to explain the observed motion, i.e., one actually describes the forces relative to a inertial frame. These fictitious forces are not real, in the sense that they do not appear from physical interactions, but are rather calculational tools. A special case of fictitious forces known as Coriolis forces appear when the reference frame is rotating around a fixed point relative to an inertial frame. The Coriolis force can be observed from the Coriolis effect; if an object is moving in a straight line and the reference frame is rotating relative to an inertial frame, the object will appear to deflect from its straight path when observed from a point in the inertial frame.

Consider an object with position $\mathbf{0}_b$, velocity $\boldsymbol{v}_B$ and acceleration $\boldsymbol{a}_b$ relative to frame b. Frame b is rotating with a non-constant angular speed $\boldsymbol{\omega}_{ab}$ relative to an inertial frame a, where the rotation vector is given relative to frame a. From (6.405) in [7], we find that the object has an acceleration of

$$\boldsymbol{a}_a = \boldsymbol{a}_b + 2\boldsymbol{\omega}_{ab} \times \boldsymbol{v}_b. \tag{B.13}$$

The term $2\boldsymbol{\omega_{ab}} \times \boldsymbol{v}_b$ is known as the Coriolis term. When applying Newton's laws on the system, the acceleration has to be given relative to the inertial system, giving

$$m\boldsymbol{a}_a = \sum_i \boldsymbol{F}_i \tag{B.14}$$

$$\updownarrow \text{(B.13)}$$

$$m\boldsymbol{a}_b = \sum_i \boldsymbol{F}_i - 2m\boldsymbol{\omega}_{ab} \times \boldsymbol{v}_b. \tag{B.15}$$

This derivation shows that Newton's laws can be extended to rotational reference frames by including a Coriolis term.

Appendix C

Stochastic Background Material

This section is a short survey of stochastic theory needed to follow the section on Kalman filters and probabilistic inference. The equations in this section is mostly derived for scalars, but similar derivations can be made with vectors. If something is valid only for vectors or scalars, it will be clear from context.

C.1 Probability

A deterministic event will always have the same outcome, whilst the outcome of a random action will be randomly distributed. A specific outcome is also known as a sample. The probability of outcome A occuring is defined as:

$$P(A) = \frac{\text{Number of times A occurs}}{\text{Total numbers of occurances}}. \tag{C.1}$$

The conditional probability is defined as the probability of an action A occuring given that action B has already occured $P(A \mid B)$. Bayes's formula says that the conditional probability is equal to the probability of both actions occuring divided by the probability of B occuring.

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)} \tag{C.2}$$

If the outcome of one have no effect on the outcome of the other, the two events are called independent

$$P(A \mid B) = P(A). \tag{C.3}$$

Bayes's theorem then says

$$P(A \wedge B) = P(A)P(B). \tag{C.4}$$

C.2 Random Variables and Probability Density Functions

A Random Variable (RV) is a mapping from events and actions to numbers and functions. The RV is defined by it's Probability Distribution Function (PDF) <>

$$F_X(x) = P(X \leq x), \tag{C.5}$$

or its Probability Density Function (pdf) <>

$$f_X(x) = \frac{\mathrm{d}F_X(x)}{\mathrm{d}x}. \tag{C.6}$$

The PDF is a function that describes the probability that a RV $X \in \mathcal{R}$ will take a value less than or equal to some value x, and the pdf is defined as the derivative of the PDF. Some properties of the PDF are given by the definition:

$$F_X(x) \in [0, 1] \tag{C.7}$$
$$F_X(-\infty) = 0 \tag{C.8}$$
$$F_X(\infty) = 1 \tag{C.9}$$
$$F_X(a) \leq F_X(b) \text{ if } a \leq b \tag{C.10}$$
$$P(a \leq X \leq b) = F_X(b) - F_X(a) \tag{C.11}$$

From the above properties of the PDF, the pdf will by definition have the following properties:

$$F_X(x) = \int_{-\infty}^{x} f_x(z) \quad \mathrm{d}\, z \tag{C.12}$$

$$f_x(x) \geq 0 \tag{C.13}$$

$$\int_{-\infty}^{\infty} f_x(x) \quad \mathrm{d}\, x = 1 \tag{C.14}$$

$$P(a \leq X \leq b) = \int_{a}^{b} f_x(x) \quad \mathrm{d}\, x \tag{C.15}$$

Two important descriptors of an RV is the *mean* and the *variance*. The mean is the average value of infinitely many samples and the variance is a number describing the spread of the samples around the mean. The expectated value of a continous RV, $\bar{x}$, is defined as

$$\bar{x} = E[X] \tag{C.16}$$

$$= \int_{-\infty}^{\infty} x f_X(x) \quad \mathrm{d}\, x, \tag{C.17}$$

and the variance defined as

$$\mathrm{Var}(X) = \sigma^2 \tag{C.18}$$

$$= E[(x - \bar{x})^2]. \tag{C.19}$$

σ is known as the standard deviation. The mean and variance are also known as the first moment and second central moment of the pdf respectively. A nth order moment is defined as $E[x^n]$ and an nth order central moment is defined as $E[(x - \bar{x})^n]$

Sometimes we wish to evaluate the joint probability of two or more variables simultaneously. For the case of two RVs, the joint PDF and joint pdf are given by:

$$F_2(x, y) = P(X \leq x \wedge Y \leq y) \tag{C.20}$$

$$f_2(x, y) = \frac{\partial F_2(x, y)}{\partial x \partial y} \tag{C.21}$$

From the definition of the joint PDF and given the properties of the PDF it is clear that

$$F(x) = F_2(x, \infty) \tag{C.22}$$

$$F(x) = \int_{-\infty}^{\infty} f_2(x, y) \quad \mathrm{d}\, y. \tag{C.23}$$

The covariance is similar to the variance and is an indirect measure of how two RVs relate to each other,

$$\mathrm{Cov}(X, Y) = E[(x - \bar{x})(y - \bar{y})^T]. \tag{C.24}$$

For vector expressions this is known as the covariance matrix.

$$\mathrm{Cov}(X, Y)) = E[(\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{y} - \bar{\boldsymbol{y}})^T] \tag{C.25}$$

Notice the special case where the variance can be found from the covariance matrix.

$$\mathrm{Var}(X, X) = \mathrm{Tr}(Cov(X, X)) \tag{C.26}$$

The two most common distributions are the uniform distribution and the gaussian distribution. The gaussian distribution, also known as the normal distribution, has a pdf given by

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{\frac{-(x-\bar{x})^2}{2\sigma^2}} .. \tag{C.27}$$

As C.1 shows, the pdf is symmetric about the mean. The mean is the most probable value of a gaussian RV.

The pdf and PDF of a gaussian RV X are thus uniquely given when $\bar{x}$ and σ are provided. A common notation for normal RVs is

$$X \sim N(\bar{x}, \sigma^2)$$

The central limit theorem says that the sum of many indepedent and identically-distributed RVs will tend to a normal distribution if the variance of the sum is finite. The normal distribution is therefore often chosen to represent other probability distributions. Another common probability distribution is the uniform distribution where every value in a range is equally probable. If a random variable maps a range from a to b the pdf is given as

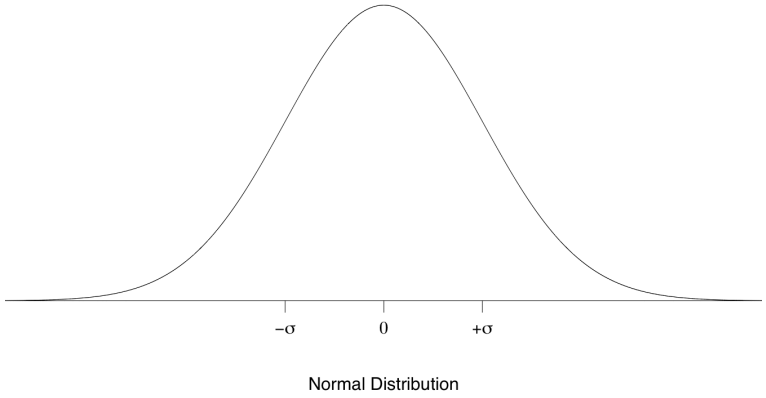

Figure C.1: Graph showing the shape of the normal pdf.

$$f_X(x) = \frac{1}{b-a}, x \in [a,b] \subset \mathcal{R} \tag{C.28}$$

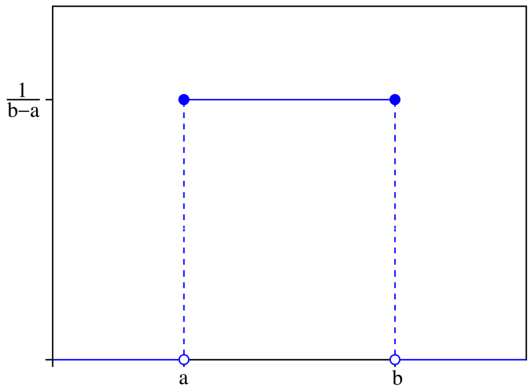

Figure C.2: Graph showing the shape of the uniform pdf.

A random process is a process that is driven by a random variable and as such attributes randomness. An example of a random process is any dynamical system driven by stochastic noise. Some random processes fulfills the Markov property and are known as Markov processes. The Markov property says that the the conditional pdf of future states, given the past and present states are dependent on a fixed number of states. The number of states that needs to be maintained give the degree of the Markov process. A first order Markov process fulfills $p(x_k \mid x_{k-1}, x_{k-2}, \ldots) = p(x_k \mid x_{k-1})$.

Any nonlinear function can be written as a linear Taylor series expansion

$$f(\bar{\boldsymbol{x}} + \tilde{\boldsymbol{x}}) = f(\bar{\boldsymbol{x}}) + \frac{1}{1!}D^1_{\tilde{\boldsymbol{x}}} + \frac{1}{2!}D^2_{\tilde{\boldsymbol{x}}} + \frac{1}{3!}D^3_{\tilde{\boldsymbol{x}}} + \ldots, \tag{C.29}$$

where

$$D^k_{\tilde{\boldsymbol{x}}} f = \left(\sum_{i=1}^{n} \tilde{x}_i \frac{\partial}{\partial x_i} \right)^k |_{\boldsymbol{x}=\bar{\boldsymbol{x}}}.$$

C.3 Random Processes and Transformed Random Variables

If an RV is input to a function, the outcome of the function will be a transformed RV. The stochastic moments of the RV will obviously be transformed as well. Here we will show how the mean and covariance will develop for an RV x transformed through a function $y = g(x)$. Defines $x = \bar{x} + \tilde{x}$ where $\bar{x}$ is the mean of x and $\tilde{x}$ the deviation, i.e. separates the RV into one zero mean RV and a constant.

$$X \sim N(\bar{x}, \sigma^2) \tag{C.30}$$
$$\updownarrow x = \bar{x} + \tilde{x}$$
$$\tilde{X} \sim N(0, \sigma^2) \tag{C.31}$$

The transformed mean and covariance is derived for a linear function $g(x) = ax + b$ by simple use of

$$\begin{aligned} \bar{y} &= E[g(x)] \\ &= E[ax + b] \\ &= E[ax] + E[b] \\ &= aE[x] + b \end{aligned} \tag{C.32}$$

$$\begin{aligned} \operatorname{Cov} Y, Y &= E[(g(x) - E[g(x)])(g(x) - E[g(x)])^T] && \text{(C.33)} \\ &= E[(ax + b - aE[x] - b)(ax + b - aE[x] - b)^T] && \text{(C.34)} \\ &= E[a^2(x - E[x])(x - E[x])^T] && \text{(C.35)} \\ &= a^2 E[(x - E[x])(x - E[x])^T] && \text{(C.36)} \\ &= a^2 Cov(X, X) && \text{(C.37)} \end{aligned}$$

Note that a gaussian RV transformed through a linear function remains a gaussian RV.

If $g(x)$ is a nonlinear function the exact transformations are generally not tractable to derive. It is therefore common to approximate the nonlinear function with a Taylor series expansion.

$$\begin{aligned}
\bar{y} &= E[g(x)] && \text{(C.38)}\\
&= E[g(\bar{x}+\tilde{x})] && \text{(C.39)}\\
&\updownarrow \text{ Taylor series expansion around } \bar{x}\\
&= E[g(\bar{\boldsymbol{x}}) + \frac{1}{1!}D^1_{\tilde{\boldsymbol{x}}} + \frac{1}{2!}D^2_{\tilde{\boldsymbol{x}}} + \frac{1}{3!}D^3_{\tilde{\boldsymbol{x}}} + \ldots] && \text{(C.40)}\\
&\updownarrow\\
&= g(\bar{x}) + E[g(\bar{\boldsymbol{x}}) + \frac{1}{1!}D^1_{\tilde{\boldsymbol{x}}} + \frac{1}{2!}D^2_{\tilde{\boldsymbol{x}}} + \frac{1}{3!}D^3_{\tilde{\boldsymbol{x}}} + \ldots] && \text{(C.41)}
\end{aligned}$$

C.4 Estimators

An estimator is a function depending on sampled data that calculates an estimate of a variable. Here we will deal with point estimators, i.e. estimators that calculate a single value estimate. The optimal estimator is one that is unbiased, consistent, and minimum variance. An unbiased estimator has the same expected value as the true parameter, if it is consistent then the estimation error $|\hat{\theta}-\theta|$ will be strictly decreasing as the number of evaluated samples grow, i.e., it will converge with probability. A minimum-variance unbiased consistent estimator is the unbiased consistent estimator with the smallest variance for all possible values of the parameter. A consistent estimator will never have smaller variance than the real value.

Bibliography

[1] M.A. Abkowitz. Lectures on ship hydrodynamics - steering and maneouverability. Technical report, Hydro- and Aerodynamics Laboratory. Lyngby, Denmark, 1964.

[2] J.G. Balchen, N.A. Jenssen, and S. Sælid. Dynamic positioning using Kalman filtering and optimal control theory. In *IFAC/IFIP Symp. on Aut. in Offshore Oil Field Operation*, pages 183–186, Amsterdam, Holland, 1976.

[3] R.A. Baxley and J. Bradshaw. Personal communication. Texas A&M University, 1998.

[4] M. Breivik. Nonlinear maneuvering control of underactuated ships. Master's thesis, Norwegian University of Science and Technology, 2003.

[5] M. Breivik and T.I. Fossen. A unified concept for controlling a marine surface vessel through the entire speed envelope. In *IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pages 1518–1523, Limassol, Cyprus, 2005.

[6] J.C. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23:756–767, 1978.

[7] O. Egeland and J.T. Gravdahl. *Modeling and Simulation for Automatic Control*. Marine Cybernetics, 2002.

[8] O.M. Faltinsen. *Sea Loads on Ships and Offshore Structures*. Cambridge University Press, 1990.

[9] K.K. Fedyaevsky and G.V. Sobolev. *Control and Stability in Ship Design*. State Union Shipbuilding Industry Publishing House, 1963.

[10] R. Findeisen and F. Allgöwer. An introduction to nonlinear model predictive control. Technical report, University of Stuttgart, 2002.

[11] C. Fleury. Sequential convex programming for structural optimization problems. In *Optimization of large structural systems; Proceedings of*

the NATO/DFG Advanced Study Institute, pages 531–553, Berchtesgaden, Germany, 1993.

[12] H.T. Foss and E.T. Meland. Sensor integration for nonlinear navigation systems for underwater vehicles. Master's thesis, Norwegian University of Science and Technology, 2007.

[13] T.I. Fossen. *Marine Control Systems - Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics, 2002.

[14] T.I. Fossen. A nonlinear unified state-space model for maneuvreing and control in a seaway. *International Journal of Bifurcation and Chaos*, 15:2717–2746, 2005.

[15] T.I. Fossen and J.P. Strand. Passive nonlinear observer design for ships using Lyapunov methods: Experimental results with a supply vessel. *Journal of Automatica*, 35(1), 1999.

[16] P.T-K. Fung and M. Grimble. Dynamic ship positioning using self-tuning Kalman filter. *IEEE Transactions on Automatic Control*, 28(3):339–349, 1983.

[17] A. Gelb et al. *Applied Optimal Estimation*. M.I.T. Press, 1974.

[18] B.K. Golding. Modeling and identification of nonlinear viscous drag for ships. Master's thesis, Norwegian University of Science and Technology, 2006.

[19] D.Y. Hollinger and J.H. Gove. Application of unscented Kalman filter for simultaneous state and parameter estimation in problems of surface-atmosphere exchange. *Journal of Geophysical Research*, 111, 2006.

[20] O.G. Hvamb. A new concept for fuel tight DP control. In *Dynamic Positioning Conference*, Houston, Texas, USA, 2001.

[21] T. Hyakudome, M. Nakamura, H. Kajiwara, and W. Koterayama. Experimental study on dynamic positioning control for semisubmersible platform. *International Journal of Offshore and Polar Engineering*, 2000.

[22] IMCA. Introduction to dynamic positioning. http://www.imca-int.com/divisions/marine/reference/intro.html, accessed june, 2008.

[23] L. Imsland. Introduction to model predictive control. Internal report, Norwegian University of Science and Technology, 2007.

[24] L. Imsland, R. Findeisen, E. Bullinger, F. Allgöwer, and B.A. Foss. A note on stability, robustness and performance of output feedback

nonlinear model predictive control. *Journal of Process Control*, 13:633–644, 2003.

[25] L. Jaulin, M. Kieffer, O. Didrift, and É. Walter. *Applied Interval Analysis*. Springer-Verlag, 2001.

[26] N.A. Jenssen. What is the DP current? In *Dynamic Positioning Conference*, Houston, Texas, USA, 2006.

[27] M.J. Journee. Offshore hydromechanics. http://www.shipmotions.nl/LectureNotes.html, 2001.

[28] S.J. Julier and J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *International Symposium on Aeorospace/Defence Sensing, Simulations and Controls*, Orlando, Florida, USA, 1997.

[29] S.J. Julier and J.K. Uhlmann. The scaled unscented transform. In *Proceedings of the IEEE American Control Conference*, pages 4555–4559, Anchorage AK, USA, 2002.

[30] R.E. Kalman. Contributions to the theory of optimal control. *Bulletin de la Societe Mathematique de Mexicana*, 5:102–119, 1960.

[31] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[32] H.K. Khalil. *Nonlinear Systems*. Prentice Hall, 3rd edition, 2002.

[33] L. Kleeman. Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning. In *International Conference on Robotics and Automation*, pages 2582–2587, Nice, France, 1992.

[34] W. Li, P. Wei, and X. Xiao. An adaptive nonlinear filter of discrete-time system with uncertain covariance using unscented Kalman filter. In *IEEE International Symposium on Communications and Information Technologies*, Beijing, China, 2005.

[35] D. Limon, T. Alamp, J.M. Bravo, E.F. Camacho, D.R. Ramirez, D. Muñoz de la Peña, I. Alvarado, and M.R. Arahal. *Interval Arithmetic in Robust Nonlinear MPC*, pages 317–326. Springer-Verlag, 2007.

[36] K-P.W. Lindegaard. *Acceleration Feedback in Dynamic Positioning*. PhD thesis, Norwegian University of Science and Technology, 2003.

[37] F. Lydoire and P. Poignet. Nonlinear model predictive control via interval analysis. In *IEEE Conference on Decision and Control, and the European Control Conference*, pages 3771–3776, Sevilla, Spain, 2005.

[38] L. Magni and R. Scattolini. *Robustness and Robust Design of MPC for Nonlinear Discrete-Time Systems*, pages 239–254. Springer-Verlag, 2007.

[39] F. Martinsen, L.T. Biegler, and B.A. Foss. A new optimization algorithm with application to nonlinear MPC. *Journal of Process Control*, 14:853–865, 2004.

[40] J.R. Morison, M.P. O'Brien, J.W. Johanson, and S.A. Schaaf. The force exerted by surface waves on piles. *Transactions of the AMIE*, 189:149–154, 1950.

[41] A.T. Nelson. *Nonlinear Estimation and Modeling of Noisy Time-Series by Dual Kalman Filtering Methods*. PhD thesis, Oregon Graduate Institute, 2000.

[42] M. Nicholaou. Model predictive control: A critical synthetis of theory and industrial needs. Technical report, University of Houston, 1998.

[43] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer-Verlag, 1st edition, 1999.

[44] N.H. Norrbin. Theory and observation on the use of a mathematical model for a ship maneoeuvring in deep confined waters. In *8th Symposium on Naval Hydrodynamics*, Pasadena, California, USA, 1970.

[45] The Society of Naval Architects and Marine Engineers. Nomenclature for treating the motion of a submerged body through a fluid. *Technical and Research Bulletin No. 1-5*, 1950.

[46] Y. Peng, J. Han, and Q. Song. Tracking control of underactuated surface ships: Using unscented Kalman filter to estimate the uncertain parameters. In *IEEE International Conference on Mechatronics and Automation*, pages 1884–1889, Singapore, 2007.

[47] Y. Peng, J. Han, and Z. Wu. Nonlinear backstepping design of ship steering controller: Using unscented Kalman filter to estimate the uncertain parameters. In *IEEE International Conference on Automation and Logistics*, pages 126–131, Jinan, China, 2007.

[48] T. Perez, T.I. Fossen, and A.J. Sørensen. A discussion about seakeeping and maneuvering models for surface vessels. Technical report, Norwegian University of Science and Technology, 2004.

[49] S.J. Qin and T.A. Badgwell. An overview of nonlinear model predictive control applications. *Chemical Process Control*, 93:232–256, 1997.

[50] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, 2003.

[51] M.D. Rafal and W.F. Stevens. Discrete dynamic optimization applied to on-line optimal control. *AIChe Journal*, 14:85–91, 1968.

[52] T. Raff, C. Ebenhauer, and F. Allgöwer. *Nonlinear Model Predictive Control: A Passivity-Based Approach*, pages 151–162. Springer, 2007.

[53] J.B. Rawlings, E.S. Meadows, and K.R. Muske. Nonlinear model predictive control: A tutorial and survey. In *Proceedings of ADCHEM '94*, pages 203–214, Kyoto, Japan, 1994.

[54] J.B. Rawlings and K.R. Muske. The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38:1512–1516, 1993.

[55] J.E.G. Refsnes. *Nonlinear Model-Based Control of Slender Body AUVs*. PhD thesis, Norwegian University of Science and Technology, 2008.

[56] D. Simon. *Optimal State Estimation*. Wiley-Interscience, 2006.

[57] R. Skjetne. *The Maneuvering Problem*. PhD thesis, Norwegian University of Science and Technology, 2005.

[58] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control, Analysis and Design*. Wiley, 2nd edition, 2007.

[59] Q. Song, J. Qi, and J. Han. An adaptive ukf algorithm and its application in mobile robot control. In *IEEE International Conference on Robotics and Biomimetics*, Kunming, China, 2006.

[60] J.P. Strand. *Nonlinear Position Control Systems Design for Marine Vessels*. PhD thesis, Norwegian University of Science and Technology, 1999.

[61] W. Sun and Y.X. Yuan. *Optimization Theory and Methods - Nonlinear Programming*. Springer-Verlag, 2006.

[62] H. Torpe. Pipeline liquid control using nonlinear MPC and Olga. Master's thesis, Norwegian University of Science and Technology, 2007.

[63] G. Torsetnes. Nonlinear control and observer design for dynamic positioning using contraction theory. Master's thesis, Norwegian University of Science and Technology, 2004.

[64] R. van der Merwe. *Sigma-Point Kalman Filters in Dynamic State-Space Models*. PhD thesis, Oregon Health and Science University, 2004.

[65] Webpage. Kongsberg maritime. http://www.km.kongsberg.com/, accessed june, 2008.

[66] C. Zillober, K. Schittkowski, and K. Moritzen. Very large scale optimization by sequential convex programming. *Optimization Methods and Software*, 19:103–120, 2004.