NTNU
Norwegian University of
Science and Technology

Autotuned Dynamic Positioning for Marine Surface Vessels

Jon Alme

Master of Science in Engineering Cybernetics

Submission date:	June 2008
Supervisor:	Morten Breivik, ITK
Co-supervisor:	Jann Peter Strand, Rolls-Royce Marine dep. Control Aalesund Ivar Ihle, Rolls-Royce Marine dep. Control Aalesund

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Problem Description

The candidate will consider the problem of developing and evaluating automatically tuned dynamic positioning (DP) controllers for marine surface vessels. The following elements must be considered:

1. Review the literature on automatic tuning, especially with regard to dynamic positioning of marine surface vessels.
2. Suggest a set of suitable criteria for evaluating the performance of dynamic positioning controllers.
3. Develop autotuned DP controllers for relevant 3 DOF applications.
4. Evaluate the behavior of your DP controllers on the basis of your suggested performance criteria through numerical simulations in Matlab/Simulink for a set of relevant DP scenarios.

Assignment given: 07. January 2008
Supervisor: Morten Breivik, ITK

Preface

> *Opportunities fly by while we sit regretting the chances we have lost, and the happiness that comes to us we heed not, because of the happiness that is gone.*
>
> *-Jerome K. Jerome, The Idle Thoughts of an Idle Fellow, 1889*

With the fulfilment of this thesis a stage in my life has come to an end and new opportunities arises. The two years I have spent here at the Norwegian University of Science and Technology can be summarized as completing this thesis. It has been challenging and at times frustrating, but an unforgettable good satisfaction it is to finish it and the experience it has given me is indescribable.

First, I wish to express thanks to my supervisor Morten Breivik for his support and advice throughout the work with this thesis. His advice in the progress of the work with this document has been invaluable, also his experience in writing such documents. I also thank my co-advisor Ivar Ihle for all his valuable tips, and for always giving a good answers to all my (not always so) clever questions.

The study presented here comes from an idea that was drawn up in cooperation with Jann Peter Strand, my co-advisor, which I would like to thank for giving me the idea and the support throughout the work with this thesis. During the study many opportunities has arisen and things has not always happened as I thought they would, but the result has gained my interest in the problem and strengthened my belief that it is solvable if enough expert knowledge and experience is used. I can only hope that some of the work also will be useful for others if they embark on a similar task. If anyone does, please keep in mind the words:

> *Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius - and a lot of courage - to move in the opposite direction.*
>
> *-E.F. Schumacker*

With these final words, I finished my M.Sc education and walk out the door looking forward for new opportunities in my life.

Jon Alme
Trondheim, May 31, 2008

Summary

Dynamic positioning of surface vessels involves control of vessels with changing dynamics, shifting conditions, for different operational tasks. A controller with fixed controller parameters cannot have an optimal performance for all these different cases, and autotuning of the controller would be very valuable. However, dynamic positioning is a complex task, and thus automatic tuning of a dynamic positioning controller is not less so. This thesis does not solve all problems that comes with autotuning of dynamic positioning systems, but it gives an overview of the problem and presents a novel performance index for station keeping. Furthermore, a hybrid controller that can function as a first step in solving the autotuning problem is suggested. The hybrid controller has a fixed controller structure and is a combination of a gain-scheduling controller and an adaptive controller. The adaptive controller is used in an idle (training/learning) mode to populate a look-up table with controller parameters, while the gain-scheduling controller work as a fast-changing dynamical controller, using the controller parameters stored in the look-up table.

Each controller parameter set in the look-up table is optimized according to a vessel operational condition, which is defined as a function of environmental conditions (wind, waves, ocean current), vessel draught, and water depth. Optimization of the controller parameters for the different vessel operational conditions is carried out by two different autotuning methods; a genetic algorithm and a rule-based algorithm. Both of these autotuning methods are optimizing the gains in a nonlinear PID-controller. The performance index and the two autotuning methods are implemented in Matlab/Simulink, where simulation tests are performed for a 3 DOF mathematical model of a supply vessel. The test scenario includes two different vessel operational conditions, where the controller has been automatically tuned both for minimal position and heading deviation as well as weighting on the use of forces. A comparison of the two autotuning methods is also performed and finally a discussion of the behaviour and tuning of the suggested performance index is carried out.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The topic of this thesis is to undertake a study of automatic tuning of dynamic positioning (DP) controllers for station keeping of surface vessels. Automatic, or autotuning, is here considered as automatic adjustment of the controller parameters according to shifting vessel operational conditions. The study will involve mathematical vessel models, different controller methods for dynamic positioning, autotuning methods, performance of dynamic positioning systems and operating conditions for a surface vessel.

The autotuning algorithm shall not be a model parameter identification scheme, but optimization of the controller parameters in relation to a specified performance requirement. Furthermore, the autotuning algorithm shall be easy to implement and need little pre-tuning.

Assumptions

- The vessel considered is a fully actuated surface vessel performing station keeping.

- The optimization will work on the force and moment levels, and it will be assumed that the thruster allocation is handled by a separate algorithm already implemented.

- It will be assumed that the vessel position and heading is measured, and that the noise filtration of this signal is perfect, so the measured signals will coincide with the real position and heading. It is also assumed that the velocity is estimated and that the estimated signal coincide with the real velocity.

- The controller is pre-tuned during a sea trial, and this tuning is carried out at design draft with minimum environmental disturbances. Minimum environmental disturbance means calm water with almost no current and wind. We have an idea about the mass of the vessel from the pre-tuned controller as well as the damping term in the ideal environment.

- There is a known upper and lower limit for the controller parameters that guarantees stability.

- Only autotuning of the feedback part of the controller will be considered, which means that it is assumed that potential feedforward terms are properly tuned.

- The environmental condition (at least the sea state condition) is detected (known).

The 4 first assumptions are seen as normal for most DP control systems. An upper and lower stability limit for the controller parameters must also be seen as a realistic assumption, since it is assumed that the controller is pre-tuned and it is normal to have manual gain adjustment possibilities in DP control systems, see Figure 1.4. If there are feedforward terms in the controller, it is must be logical to assume that these are properly tuned during the commissioning, and this assumption is thus also seen as normal for most DP control systems.

When it comes to environmental conditions, wind measurements are normally carried out. According to [3], sea current is impossible to measure, and is normally determined from a continuous integration of the difference between the predicted position of the vessel and the estimated/measured position of the vessel. This is of course not an accurate sea current value and deflection can occur because of model error, improper wind feedforward terms and other forces that influence the vessel position. In [54], it is mentioned that the dominating wave frequency can be measured/estimated on-line by using spectral analysis if it is presented in both the position (x, y) and the heading (ψ) measurements, which is normally present. In [34], it is mentioned that it can be difficult to determine the sea state in calm to moderate sea states by spectral analysis and measurement of only surge, sway and yaw, since the vessel will act as a low-pass filter and it might be difficult to find reasonable data for low sea states. Furthermore, depending on the condition of the developed sea, it may also be difficult to just find one single peak frequency. For moderate seas, measurement of pitch and roll might help, but the natural frequency of the vessel can over dominate the wave frequency. Another factor that can help in detecting the sea state is wind measurement, based on the assumption that waves are wind generated. So if not an accurate environmental condition is known, at least an approximation of some environmental conditions should be known, and the last assumption is thus also considered to be a realistic assumption.

1.1 Motivation

Dynamic positioning of vessels came as a consequence of expansion of oil and gas exploration to deeper water, in the rapidly increasing oil and gas industry of the 1960s and early 1970s. The Norwegian classification society DNV defines a dynamically positioned vessel as a free-floating vessel that maintains its position exclusively by use of thrusters. In [16], the definition is extended from exclusive use of thrusters to also include use of propellers that generate rudder lift forces.

According to [3], the drilling exploration vessel "Eureka" was the first vessel that fulfilled this definition of dynamic positioning with automatically controlled propellers. Eureka was built in 1961 and was fitted with a very basic analogue control system using automatic electro-mechanical devices. The propulsion equipment was two steerable thruster (one fore and one aft) in addition to the main propulsion, see Figure 1.1 for the original sketches of the vessel. Today there are over 1000 DP capable vessels, and the vessel types and their operation tasks are legion, please consult [3] for a detailed description of such types and tasks.

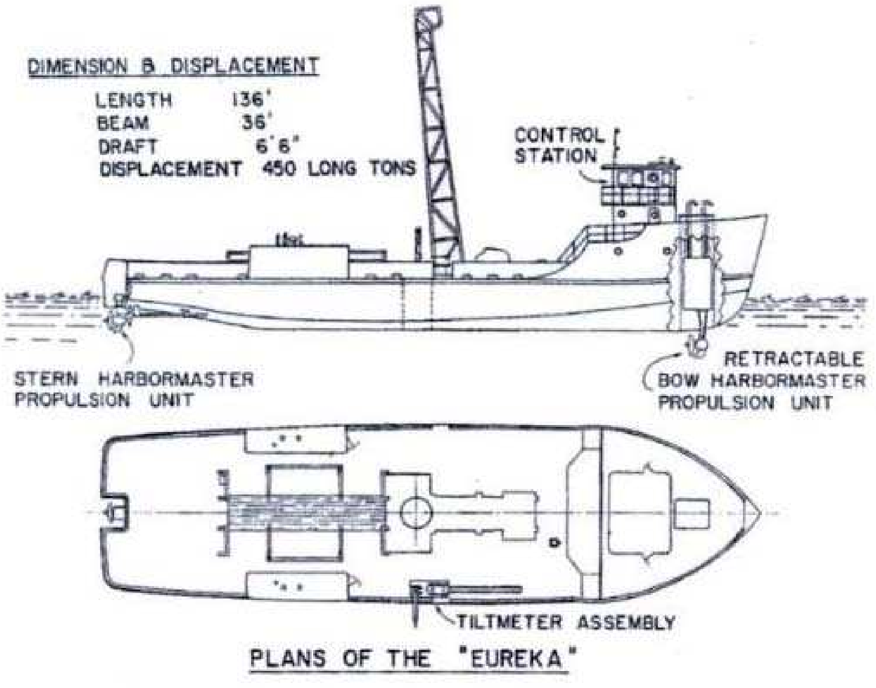


Figure 1.1: The original 1961 drawings by H. L. Shatto and J. R. Dozier of the Drillship Eureka; bow and stern thrusters rotated 360° to maintain position. Courtesy of Howard Shatto and the Shell Oil Company.

Even though the vessel types and their operational tasks have been diverse, they are all still dependent on making profit. Increasing the profit for a vessel performing dynamic positioning can be done in several ways, and examples includes; energy reduction (fuel saving), reduced wear and tear on thrusters, and increasing the vessel performance.

From being a basic analogue control system, DP control systems today are digital, computer-based control systems with sophisticated functions that improve the profit, see Figure 1.2 and Figure 1.3 for a modern DP operator station. Please consult [3] for a detailed description of the different functions implemented in commercial DP control systems. With today's rapidly increasing computer technology and the following price reduction in computer storage and processor power, functions that are more sophisticated and help to improve the profit even further will be implementable in the future.

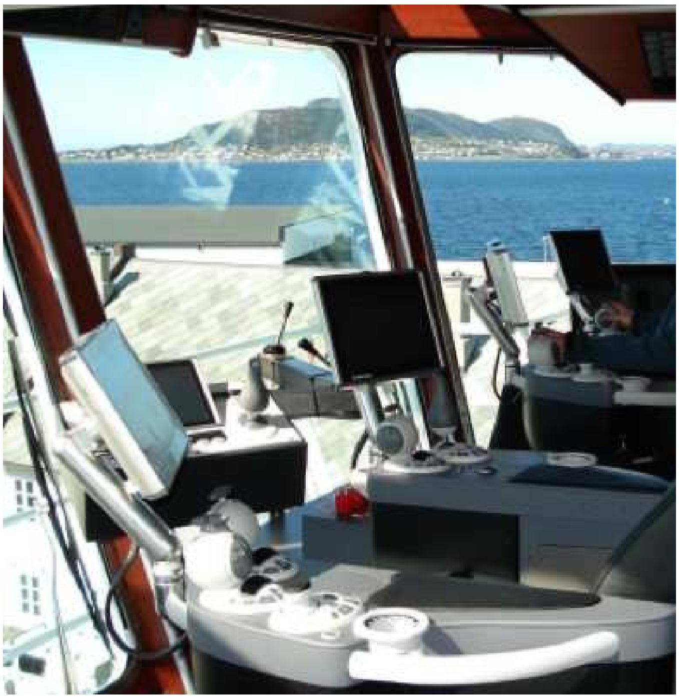


Figure 1.2: Rolls-Royce Marine DP operator station on the supply vessel Volstad Viking. Courtesy of Rolls-Royce Marine AS.

The controller parameters in a dynamic positioning controller is normally tuned with respect to minimizing the deviation in position, heading, velocity, etc., while the minimization in use of thruster forces is carried out by a thruster allocation algorithm. Normally, tuning of the DP controller is carried out at a sea trial during the commissioning of the vessel. At this

Figure 1.3: Rolls-Royce Marine DP operator station. Courtesy of Rolls-Royce Marine AS.

sea trial, a normal vessel operational condition is design draft and minimum environmental disturbances. The controller parameters obtained from this tuning are not necessary optimal for all different operations and shifting conditions that a vessel will operate in. For instance, for some of these operations and conditions, a DP-operator might experience that the DP controller uses an unnecessarily large amount of thruster force to keep the position or heading, while in other situations the vessel might have problem keeping the position or heading. Another motivation in having variable controller parameters for different operations is that different operations do not necessarily have the same requirements in position/heading accuracy, and with other controller parameters, a saving in use of thruster force can be obtained. Saving in use of thruster forces would result in energy saving and less wear and tear on the thrusters. This is of course two conflicting objectives, which must be balanced.

Even though a vessel is exposed to shifting conditions, and different operations require different accuracy levels, the conditions and operations can been seen as static during some time horizon. Theoretically, it should then be possible to find an optimal controller parameter set for this time horizon with respect to the condition and the operation. It is also natural to assume that a condition that is almost equal to the ongoing one will appear sometime in the future. Consequently, by saving the experience from the ongoing condition in a look-up table, the information can be used to find the optimal controller parameters for a similar condition in the future. This can be compared with how a captain will use his experience in operating the vessel for different operations and conditions.

To my knowledge, the most conventional dynamic positioning systems on the market today are only using present values, and not historical data in optimizing the controller. Variable controller gain adjustment to shifting conditions is often solved by a limited manual gain adjustment, or a low, medium or high gain selection, see for instance Figure 1.3.

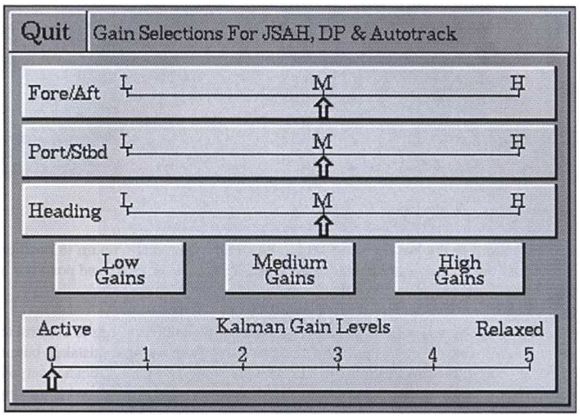


Figure 1.4: Example of manual gain adjustment of DP controllers for the Converteam (formerly known as Alstom) DP system. Courtesy of [3].

1.2 Previous Work

There are not so much work done on autotuning of dynamic positioning controllers, and most of the work is on adaptive estimation of model parameters. However, relevant work for this thesis include:

- In [42], different tuning methods based on model identification were tested without much luck. To separate out wave-induced forces showed to be difficult, and the only method that showed some success was based on off-line optimization. Data used in the optimization was real-time data logged on a sea trial.

- In [58], a neural network predictive controller was proposed, which uses neural networks to estimate the unknown nonlinear terms in a 3 degree of freedom vessel (DOF) model, and combines this estimation with an optimal controller to make a predictive controller.

- A fuzzy logic controller was proposed for manoeuvering of surface vessels in [5], which might in some way be called an adaptive controller since it does not need model knowledge and uses expert rules in deciding controller outputs. In [6], an application of the fuzzy logic controller is developed for a floating structure.

- In [29], different MIMO PID-tuning methods were discussed and tested for simple tuning of dynamic positioning controllers. None of the tuning methods discussed were autotuning, but it was suggested that the proposed tuning method was possible to automate.

- In [1], a model reference adaptive control (MRAC) scheme is developed for an autopilot, where a command generator is used to compensate for nonlinearities because of saturation limits in the actuators.

- In [26], genetic adaptive control was compared against conventional control techniques in on-line control of cargo ship steering. They conclude that the genetic adaptive controller performed very well, but that there were still many uncertainties about the technique, e.g., proofs of stability, convergence and robustness.

- In [30], a genetic algorithm (GA) is used in optimizing the controller parameters in a sliding mode controller for ship steering control of an oil tanker. The controller parameters are optimized with respect to different loading conditions (draught) and different water depths. The optimizations were performed off-line on model basis.

- In [34], a hybrid controller with switching between different controllers with respect to different sea-state levels is proposed. The controllers are pre-tuned and it is the switching that is automated. The different wave states are detected from the estimation of the wave disturbances in an adaptive observer.

- In [52], an MRAC technique is applied to a DP system. The controller is tested out experimentally in surge on a shuttle tanker exposed to environmental forces in an offloading operation, with good results. But the MRAC is only applied in surge and not in sway and yaw, and no considerations about interaction between sway and yaw is discussed. Furthermore, no considerations about nonlinearities are discussed because of actuator saturations.

- In [32], a model experiment on dynamic positioning using a neural network controller is presented. The study is still in an early stage and only simple model experiments in one direction with beam sea is carried out. The results so far show that the controller could change the position without offset even under a disturbance by wave and current.

- In [13], a global output feedback controller for dynamic positioning of surface vessels is presented. The controller has an adaptive approach where neither the ship parameters or the velocity must be known. The drawback is that the global asymptotic position tracking stability proof only holds in an disturbancefree case.

- In [9], a method to design a robust adaptive output feedback controller for surface vessels has been proposed, which does not need measurement of the velocities. This is made possible by introducing a novel adaptive observer that works both as an estimator and a filter, where the adaptive part is model parameter identification in the observer design.

1.3 Contribution

The focus in this thesis is on optimizing controller parameters in a DP-controller with respect to different vessel operational conditions (VOCs) and the performance of DP controllers for station-keeping operations. The contribution in the thesis includes discussion about which VOCs will affect the controller parameters, and a VOC variable is defined on this basis. Furthermore, the thesis includes a discussion about what can be regarded as good performance for dynamic positioning systems, and different performance indices are mentioned. A new performance index for station-keeping operations is proposed, which combines position/heading deviation and the use of force and moment. Also a rule-based tuning algorithm that uses the proposed performance index in autotuning of a PID-controller is suggested. Finally, a hybrid controller structure is proposed, which includes the advantage of using historical data in a gain-scheduling way with respect to different VOCs, and at the same time have the ability to adapt to new knowledge by training/learning. To show the performance of the proposed performance index and the rule-based autotuning algorithm, simulation tests of a supply vessel model performing station keeping in two different VOCs are carried out. The rule-based algorithm is also compared with the results from a genetic algorithm, similar to the one in [26]. The difference between the hybrid controller proposed here and the one in [34] is the focus on finding optimal controller parameters by adaptation with respect to the current VOC, while in [34] different controllers and observers are selected only with respect to subset of the VOCs considered here.

1.4 Outline

The thesis is organized into six chapters with the following contents:

In Chapter 2, notation for modeling of marine vessels, different reference frames and a 3 DOF vessel model for dynamic positioning is presented. Furthermore, linear, nonlinear and intelligent control methods for dynamic positioning are introduced. Autotuning methods are discussed, where the main focus is given to MRAC and GA.

In Chapter 3, a discussion about what is considered as good performances for different VOCs, how this can be defined, and what VOCs will affect the controller tuning are presented. Finally, a novel performance index for station keeping is suggested.

In Chapter 4, a hybrid controller which combines the function of gain scheduling and adaptation is proposed. The goal for the controller is to find the optimal controller parameter set for different VOCs and store them in a database for use in a gain-scheduling scheme. Furthermore, two different adaptation methods for training/learning are mentioned, which both are optimizing the gains in a PID-controller. One of them is a rule-based algorithm, while the other is a genetic algorithm (GA). A simple definition of the VOC variable value is proposed, and a linking between the VOC value and a controller parameter set is defined.

In Chapter 5, simulation results from optimization with a rule-based algorithm and a GA are presented and compared against each other. Furthermore, the suggested performance index for station keeping is used in the simulations and the behaviour of the index is discussed.

In Chapter 6, a summary of the simulation results and the key terms in the thesis are presented, and possible further work recommended.

1.5 Abbreviations

APPC	Adaptive Pole Placement Control
ANN	Artificial Neural Network
CG	Center of Gravity
CLF	Control Lyapunov Function
CP	Center Point
DOF	Degree Of Freedom
DP	Dynamic Positioning
ECEF	Earth-Centered Earth-Fixed
ECI	Earth-Centered Inertial
FAM	Fuzzy Associative Memory
GA	Genetic Algorithm
GNC	Guidance, Navigation and Control
GPS	Global Positioning System
IAE	Integral of Absolute Error
IMC	Internal Model Control
ISE	Integral of Squared Error
ITAE	Integral of Time-weighted Absolute Error
ITSE	Integral of Time-weighted Square Error
LQR	Linear Quadratic Regulator
LQG	Linear Quadratic Gaussian
MIMO	Multiple Input Multiple Output
MRAC	Model Reference Adaptive Control
MRAS	Model Reference Adaptive System
MRC	Model Reference Control
MV	Minimum Variance
MVC	Minimum Variance Control
NED	North-East-Down
NN	Neural Network
PFW	Peak Frequency Wave
SNAME	Society of Naval Architects and Marine Engineers
SISO	Single Input Single Output
STR	Self Tuning Regulators
VOC	Vessel Operational Condition

Chapter 2

Theoretical Background

2.1 Vessel Model for Dynamic Positioning

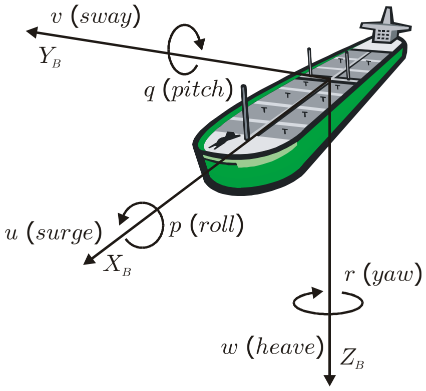


Figure 2.1: Illustration of the SNAME [35] notation. Courtesy of [4].

A model for dynamic positioning of vessels can be represented by a mathematical model of an object operating in a 3-dimensional space, where both the dynamics and the statics of the vessel have to be considered. The statics is concerned with behavior of the vessel when the accelerations are zero, which means that the vessel is at rest in its equilibrium or moving with a constant velocity. The dynamics is commonly divided into two parts:

kinematics is concerned with the geometrical aspects of motion, without reference to the forces that cause the motion. The discipline of kinematics makes it possible to translate motions between different reference frames.

kinetics is concerned with the analysis of the forces that cause the motion.

Since only the dynamical equations of motion is of interest in this thesis, the statics will not be treated further. By describing the vessel as an object in a 3-dimensional space with coordinate axis x y z, the three coordinates will correspond to the position of the vessel rigid body, and the time derivatives of the coordinates will correspond to the translational motions along the axes. To also describe the orientation and the rotational motions of the vessel rigid body, three more coordinates are needed. Totally, this gives the need for 6 independent coordinates to describe the motion of the vessel in 6 degrees of freedom (DOFs). The 6 coordinates for position and orientation, and the translational- and rotational-motion, are illustrated in Figure 2.1.

DOF		forces and moments	linear and angular velocities	positions and Euler angles
1	motion in the x-direction (surge)	X	u	x
2	motion in the y-direction (sway)	Y	v	y
3	motion in the z-direction (heave)	Z	w	z
4	rotation about the x-axis (roll)	K	p	ϕ
5	rotation about the y-axis (pitch)	M	q	θ
6	rotation about the z-axis (yaw)	N	r	ψ

Table 2.1: SNAME notation [35].

The notation and definition of the motion components for marine vessels in a 6 DOF reference frame given by [35] are used. See Table 2.1 for the notation.

By using the notation $\boldsymbol{\eta}$ for position and Euler angles, $\boldsymbol{\nu}$ for linear and angular velocity, and $\boldsymbol{\tau}$ for forces and moments in a vectorial representation, the vectors in a 6 DOF reference frame according to [16] are:

$$\boldsymbol{\eta} \triangleq \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \\ \phi \\ \theta \\ \psi \end{bmatrix}, \quad \boldsymbol{\nu} \triangleq \begin{bmatrix} \mathrm{u} \\ \mathrm{v} \\ \mathrm{w} \\ \mathrm{p} \\ \mathrm{q} \\ \mathrm{r} \end{bmatrix}, \quad \boldsymbol{\tau} \triangleq \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ \mathrm{K} \\ \mathrm{M} \\ \mathrm{N} \end{bmatrix}. \tag{2.1}$$

The rigid-body and hydrodynamic equations of motion are in general described by 6 (complicated) differential equations, one for each degree of freedom. In [15,16], these 6 differential equations are lumped together into a vectorial equation of motion:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\boldsymbol{\nu} \tag{2.2}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \mathbf{w}. \tag{2.3}$$

The vectors and matrices of these equations will be explained in the forth coming sections. In Section 2.1.3, the equation will be reduced to a linear 3 DOF equation of motion for dynamic positioning, which will be used to develop the controllers.

2.1.1 Kinematics and Reference Frames

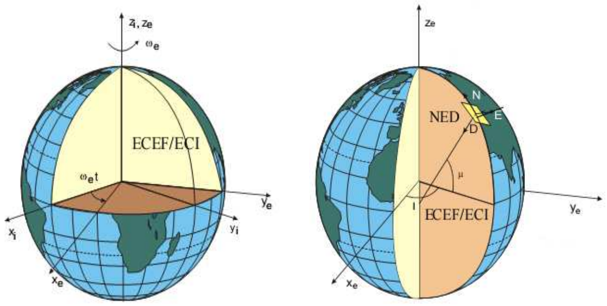


Figure 2.2: Illustration of the ECEF, ECI and NED reference frames. Courtesy of [16].

As already mentioned, the kinematics deals with the geometrical aspect of motion. In the description of the vessels position and orientation on a global space it is convenient to define Earth centered reference frames and local geographical centered reference frames.

When a vessel is in transit between two continents, a reference frame is necessary to describe the motion and the location of the vessel, this reference frame is defined as the Earth-centered Earth-fixed reference frame (ECEF). While to apply Newton's law of motion to the vessel, another reference frame is needed, this is defined as the Earth-centered inertial reference frame (ECI). Both these reference frames has their origin fixed to the center of the Earth, where the ECI frame is fixed in space and the ECEF frame rotate with the velocity of the Earth relative to the ECI frame. See Figure 2.2, for an illustration.

The local geographical reference frames are divided into a North-East-Down reference frame (NED) and a BODY-fixed reference frame. The NED reference frame is the coordinate system we refer to in our everyday life. This coordinate system is defined with the z-axis points down into the earth, and the x- and the y-axis spans a tangent plan on the earth surface, with the x-axis points towards the true north and y-axis towards east. In this thesis, only station keeping will be considered, which means that the vessel is operating in a local area with approximately constant longitude and latitude. This navigation is also refereed to as flat earth navigation, and the position can be decomposed into the NED frame, where it is assumed that the NED frame is also the inertial such that the Newton's law still apply.

The BODY-fixed reference frame is fixed to the vessel, normally with its origin at the center point (CP) of the vessel or to coincide with the center of gravity (CG). In this coordinate-system are the axis in the SNAME notation describe as the x-axis along the longitudinal (stern to bow), the y-axis along the transversal (port to starboard), and the z-axis along the normal axis (top to bottom).

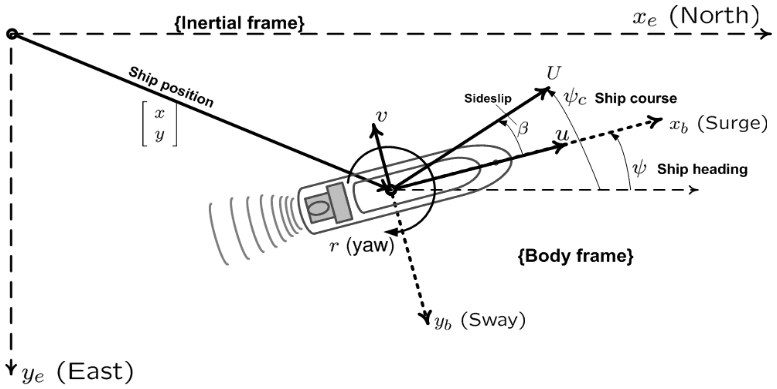


Figure 2.3: NED to BODY reference frame relationship. Courtesy of [46].

While the BODY-fixed coordinate system is used to describe the linear and angular velocities of the vessel, the ECEF or the NED coordinate system is used to describe the position and the orientation of the vessel, since the BODY-fixed reference frame is a moving coordinate frame.

The kinematic relationship between the NED frame and the BODY can in Euler angles be described by a rotation matrix with a rotation ψ about the z-axis, followed by a rotation θ about the current (rotated) y-axis, and finally a rotation ϕ about the current (rotated) x-axis. A relationship between the NED and the BODY-fixed reference frame for a 3 DOF model is illustrated in Figure 2.3. See, for instance [16] for a detailed description about the different reference frames, and the relationship between them.

The kinematic part of the 6 DOF vectorial marine vessel equation of motion is:

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{2.4}$$

where $\mathbf{J}(\boldsymbol{\eta})$ is the kinematic rotation matrix between the BODY-fixed reference frame and the Earth fixed reference frame:

$$\mathbf{J}(\boldsymbol{\eta}) = \begin{bmatrix} \mathbf{J_1}(\boldsymbol{\eta}_2) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{J_2}(\boldsymbol{\eta}_2) \end{bmatrix}, \tag{2.5}$$

where $\boldsymbol{\eta}_2 = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^T$, and $\mathbf{J_1}(\boldsymbol{\eta}_2)$, $\mathbf{J_2}(\boldsymbol{\eta}_2)$ are rotation matrices.

2.1.2 Vessel Dynamics

Like other dynamical system performing some kind of motion, a vessel can be looked at as a mass-damper-spring system, which has an inertia showing some kind of damping behavior and restoring forces that bring the system back to an equilibrium state when the external forces acting on the system subside. By taking a closer look at the kinetic part of the 6 DOF vectorial marine vessel equation of motion proposed in [15, 16]:

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \mathbf{w}, \tag{2.6}$$

where:

$\mathbf{M}$	-system inertia matrix (including added mass)
$\mathbf{C}(\boldsymbol{\nu})$	-Coriolis-centripetal matrix (including added mass)
$\mathbf{D}(\boldsymbol{\nu})$	-damping matrix
$\mathbf{g}(\boldsymbol{\eta})$	-vector of gravitational/buoyancy forces and moments
$\boldsymbol{\tau}$	-vector of control inputs
$\mathbf{w}$	-vector of environmental disturbances (wind, waves and currents),

it is easy to see, that except for the coriolis-centrifugal matrix, equation (2.6) can be looked at as a mass-damper-spring system. The coriolis-centrifugal matrix is included because (2.6) is developed in a BODY-fixed frame, and hence it is not an inertial one. The terms of equation (2.6) represents:

- $\mathbf{M} = \mathbf{M_{RB}} + \mathbf{M_A}, \in \mathbb{R}^{6x6}$. M_{RB} is the vessel's rigid-body system inertia matrix, and M_A is the hydrodynamical added inertia. The hydrodynamical added inertia is added mass due to the inertia of the surrounding fluid, which have to move away and close when the vessel propagate through the fluid. The kinetic energy of this fluid will be bigger than the kinetic energy of the rigid-body.

- $\mathbf{C}(\boldsymbol{\nu}) = \mathbf{C_{RB}}(\boldsymbol{\nu}) + \mathbf{C_A}(\boldsymbol{\nu}), \in \mathbb{R}^{6x6}$. Describes the centrifugal and coriolis forces and moments acting on the vessel. These terms are necessary for describing the rotational motion behavior in the reference frames not fixed in the inertial body frame, and are related to the rigid body system inertia matrix and the added system inertia matrix.

- $\mathbf{D}(\boldsymbol{\nu}) = \mathbf{D} + \mathbf{D_n}(\boldsymbol{\nu}), \in \mathbb{R}^{6x6}$. Describes the hydrodynamic damping forces and moments acting on the vessel, and are mainly caused by; potential damping, skin friction, wave drift damping, and vortex shedding. These damping terms contribute both with linear and quadratic damping, where D is the linear and $D_n(\boldsymbol{\nu})$ is the nonlinear part in $D(\nu)$.

- $\mathbf{g}(\boldsymbol{\eta})$. Describes the gravitational and the buoyancy forces and moments, these are in the hydrodynamic terminology called restoring forces and moments. In a mass-damper-spring system are these forces acting as a spring.

- $\boldsymbol{\tau}, \in \mathbb{R}^6$. Describes the vector of control inputs acting on the vessel for controlling the vessel. The control inputs can come from several different type of propulsion devices, such as main propellers, rudders, tunnel thrusters, azimuth and azipod thrusters, and water jets. The propulsion devices will be referred to as the actuators further in this thesis.

- $\mathbf{w}, \in \mathbb{R}^6$. Describes the environmental disturbances from wind, waves and currents. This can divided into high- and low-frequency forces and moments acting on the vessel, where the low-frequency part can be counteracted by the control inputs. The high-frequency part cannot be counteracted by the control inputs and has to be filtered from the measurements. In this thesis such filtering is assumed implemented, and only the low-frequency part of $\mathbf{w}$ will be considered.

Since this thesis will be considering a 3 DOF model in the development of the controller, the contents of the different matrices in the 6 DOF model will not be illustrated. For the contents of these matrices, and a deeper detailed description of the terms, please consult [16] Chapter 3. In the next Section, the 6 DOF nonlinear model will be reduced to a 3 DOF linear model, and the contents of these matrices will be illustrated.

2.1.3 3 DOF Model for Dynamic Positioning

By taking some assumptions, the 6 DOF model can be reduced to a 3 DOF horizontal model for dynamic positioning (station keeping). The origin of the BODY frame is assumed being taken as the geometric center point (CP) of the vessel structure, other assumptions are:

- **Port-starboard symmetry**. A surface vessel is normally designed with a port-starboard symmetry, and a port-starboard symmetry will place the center of gravity (CG) a distance x_g from the CP along the x-axis, hence $y_g = 0$.

- **Small pitch and roll angles**. By assuming the vessel is longitudinally and laterally meta-centrically stable with small amplitudes $\phi = \theta = \dot{\phi} = \dot{\theta} \approx 0$ the dynamics in roll and pitch can be discarded. This also implies that $\sin\theta \approx \sin\phi \approx 0$ and $\cos\theta \approx \cos\phi \approx 1$, hence the rotation matrix $J(\eta)$ in 2.4 is reduced to a rotation ψ about the z-axis, $R(\psi) = R_{z,\psi}$. The restoring forces in surge will be a function of $\sin\theta$, and in sway and yaw the restoring forces and moments will be a function of $\sin\phi$, and since $\sin\theta \approx \sin\phi \approx 0$ the restoring forces and moments terms $\mathbf{g}(\boldsymbol{\eta})$ will be ≈ 0 in surge, sway and yaw, hence $\mathbf{g}(\boldsymbol{\eta})$ can be neglected in a 3 DOF model for horizontal motions.

- **Surface vessel**. For a surface vessel the mean heave position $z = 0$, and the dynamics in heave can be discarded.

- **Low speed** ($|\boldsymbol{\nu}| < 2-3[m/s]$). By assuming low speed, the quadratic velocity terms in $\mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu}$ and $D(\boldsymbol{\nu})\boldsymbol{\nu}$ will be very small, this gives that $\mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} \approx 0$, and the nonlinear terms in $\mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu}$ will be ≈ 0, hence $\mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \mathbf{D}\boldsymbol{\nu}$.

By employing these assumptions, the 6 DOF equation of motion can be reduced to the low-speed 3 DOF model for dynamic positioning given in [16]:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{2.7}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{D}\boldsymbol{\nu} = \boldsymbol{\tau} + \mathbf{w}, \tag{2.8}$$

where $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$ and the matrices takes the form:

$$\mathbf{R}(\psi) \triangleq \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

$$\mathbf{M}_{RB} \triangleq \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \tag{2.10}$$

$$\mathbf{M}_A \triangleq \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -Y_{\dot{r}} & -N_{\dot{r}} \end{bmatrix} \tag{2.11}$$

$$\mathbf{M} \triangleq \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \tag{2.12}$$

$$\mathbf{D} \triangleq \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix}. \tag{2.13}$$

The system inertia matrix and the damping matrix have some important properties that can be utilized when designing controllers. The properties of the system inertia matrix are:

$$\mathbf{M} = \mathbf{M}^{\mathbf{T}} > 0. \tag{2.14}$$

While the property of positiveness is always true, the symmetry property is on the other hand only true for vessel operating at low speeds, but this thesis is only considering station keeping, hence the properties stated in equation (2.14) are assumed throughout.

The hydrodynamic damping matrix $D(\nu)$ holds the properties that it will be real, nonsymmetric and strictly positive, i.e.,

$$\mathbf{D}(\boldsymbol{\nu}) > 0 \quad \forall \boldsymbol{\nu} \neq 0, \tag{2.15}$$

which is always true. For low speed applications where the damping matrix is reduced to equation (2.13), it can also be assumed that $N_v = Y_r$, hence the damping matrix got the property of symmetry, i.e.,

$$\mathbf{D} = \mathbf{D}^{\mathbf{T}} > 0, \tag{2.16}$$

and since only a low-speed application is considered in this thesis, the properties in (2.16) will be assumed throughout. The property of positiveness means that when all forces acting on the vessel die down, the vessel will finally come to rest. This means that in developing controllers, care should be taken in compensating for such good damping terms.

Unified Model

Normally, low-frequency and wave-frequency models are combined with linear superposition, i.e., $\boldsymbol{\eta} = \boldsymbol{\eta}_{LF} + \boldsymbol{\eta}_{WF}$, where the control system is designed to compensate for the low-frequency motions, and the wave-frequency motions are seen as a disturbance, and is filtered out from entering the feedback loop. From a physical point of view, it is not correct to add the effect of waves directly at the motion level, but to introduce it at the force level, see Figure 2.4. This is the motivation behind some recently published work, for instance [17, 47] where a unified state space model has been proposed:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{2.17}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{D}\boldsymbol{\nu} + \boldsymbol{\mu} = \boldsymbol{\tau} + \mathbf{w}, \tag{2.18}$$

where:

$$\boldsymbol{\mu} = \int_0^t \mathbf{K}(t-\tau)\boldsymbol{\nu}(\tau)d\tau \tag{2.19}$$

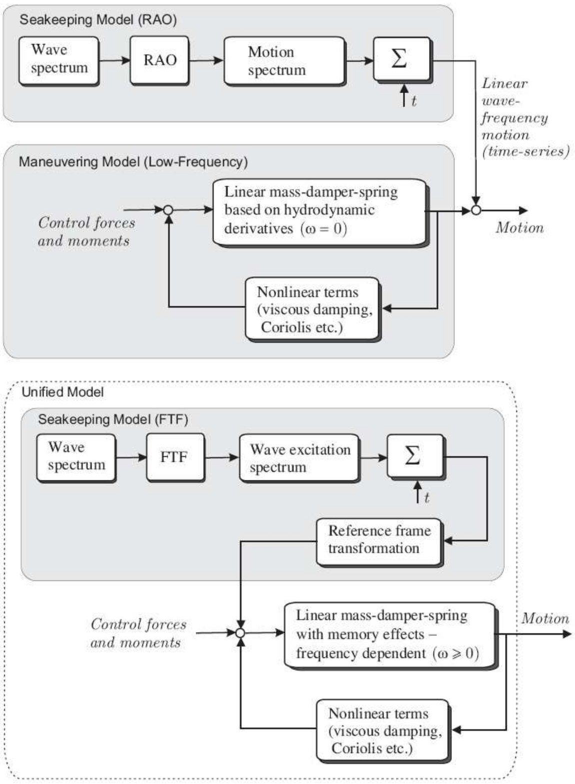

Seakeeping Model (RAO)
Wave spectrum
RAO
Motion spectrum
Σ
t
Linear wave-frequency motion (time-series)
Maneuvering Model (Low-Frequency)
Control forces and moments
Linear mass-damper-spring based on hydrodynamic derivatives (ω = 0)
Motion
Nonlinear terms (viscous damping, Coriolis etc.)
Unified Model
Seakeeping Model (FTF)
Wave spectrum
FTF
Wave excitation spectrum
Σ
t
Reference frame transformation
Control forces and moments
Linear mass-damper-spring with memory effects – frequency dependent (ω ⩾ 0)
Motion
Nonlinear terms (viscous damping, Coriolis etc.)

Figure 2.4: *Upper image:* Superposition of a seakeeping model and a maneuvering model. *Lower image:* A unified model. Courtesy of [17].

is the impulse response from wave loads, also refereed to as the memory effect of the fluid. The term $\boldsymbol{\mu}$ can for instance be described by a linear state space model:

$$\dot{\boldsymbol{\chi}} = \mathbf{A}_r\boldsymbol{\chi} + \mathbf{B}_r\boldsymbol{\nu} \tag{2.20}$$

$$\boldsymbol{\mu} = \mathbf{C}_r\boldsymbol{\chi} + \mathbf{D}_r\boldsymbol{\nu}. \tag{2.21}$$

Details about how to develop the state space model for $\boldsymbol{\mu}$ is found in [25]. The advantage of the model proposed in [17, 47] is that it represents the wave-induced forces in the time-domain, giving a more physical description of the behavior in different sea states. It can be used to simulate vessel and rigs in a seaway for varying sea states and at different speeds (including zero speed). Since it is represented in state-space form it should be easy to use for standard feedback control systems. However, it is more complex than the standard 3 DOF horizontal model (2.8), and more parameters are needed to be determined. For more details about the unified model, please consult [17] and the references therein.

2.2 Control Methods for Dynamic Positioning

Control systems for marine vessels are based on several design techniques, such as PID-control, linear quadratic optimal and stochastic control (LQR, LQG), nonlinear control theory (feedback linearization, sliding mode, integrator backstepping), fuzzy logic and neural networks.

A short introduction to some of these control methods will be given here, and all considered control methods will be based on the 3 DOF model developed in Section 2.1.3. Full state feedback is assumed, or at least that the states are estimated. It will also be assumed that all states are free of measurement and high-frequency wave noise. Theory about estimators (observers) and filters will not be covered, only the assumption that such are implemented. For a more detailed description, stability proofs of the different control methods, and for literature about estimators and filters techniques for marine vessels, please consult [15, 16] and the references therein.

Besides high-frequency wave filtering and state estimation, modern controllers also includes sophisticated features such as:

- Reference feedforward; using a dynamic reference model $\boldsymbol{\eta}_d$, $\boldsymbol{\nu}_d$ and $\dot{\boldsymbol{\nu}}_d$ for position and course-changing maneuvers, where station keeping is obtained by using a constant reference signal $\boldsymbol{\eta}_d$.
- Wind feedforward; for rapid compensation against disturbance from wind.
- Parameter adaptation; because of time-varying model parameters, e.g., changes in mass and CG because of different loading condition (i.e., change in draught, trim and heel), or because of varying environmental conditions or water depth.

2.2.1 Linear Control Theory

Even though the 3 DOF model developed in Section 2.1.3 is nonlinear, linear control theory can be used for low-speed tracking of overactuated vessels. For proof, please consult [28] where the conclusion was that: "Low-speed tracking for overactuated ships can be done with linear tools. When the gains have been determined, simply add the proper rotation $\mathbf{R}^T(\psi)$ in the control law to provide validity for all heading angles".

The two most common control laws with linear control theory are PID and LQR.

PID

According to [49], more than 90% of all control loops are PID. This is mainly because its structure is physical intuitive to understand also for the operator, and it is well proved and reliable. A continuous-time representation of a nonlinear PID-control law for dynamic positioning is

$$\boldsymbol{\tau}_{PID}(t) \triangleq -\mathbf{K}_p\mathbf{R}^T(\psi)\tilde{\boldsymbol{\eta}} - \mathbf{K}_d\tilde{\boldsymbol{\nu}} - \mathbf{K}_i\mathbf{R}^T(\psi)\int_0^t \tilde{\boldsymbol{\eta}}(\boldsymbol{\tau})d\boldsymbol{\tau}, \tag{2.22}$$

where $\boldsymbol{\tau}_{PID}$ is the controller input, $\tilde{\boldsymbol{\eta}} \triangleq \boldsymbol{\eta} - \boldsymbol{\eta}_d$ and $\tilde{\boldsymbol{\nu}} \triangleq \boldsymbol{\nu} - \boldsymbol{\nu}_d$ are the errors in position/heading and velocity, and the controller gains for each DOF are defined as:

K_p	$>$	0	proportional gain constant
K_i	$=$	K_p/T_i	integral gain constant
T_i	$>$	0	integral time constant
K_d	$=$	K_pT_d	derivative gain constant
T_d	$>$	0	derivative time constant.

The control law can easily be extended with feedforward terms as $\boldsymbol{\tau} \triangleq \boldsymbol{\tau}_{PID} + \boldsymbol{\tau}_{FF}$. Typically feedforward terms are wind feedforward $\boldsymbol{\tau}_{FF} \triangleq \boldsymbol{\tau}_{wind}$, or reference feedforward, e.g., consider the tracking control law mentioned in [28]

$$\dot{\boldsymbol{\xi}} = \tilde{\boldsymbol{\eta}} \tag{2.23}$$

$$\boldsymbol{\tau} \triangleq -\mathbf{M}\left(\mathbf{K}_i\mathbf{R}^T(\psi)\boldsymbol{\xi} + \mathbf{K}_p\mathbf{R}^T(\psi)\tilde{\boldsymbol{\eta}} + \mathbf{K}_d\tilde{\boldsymbol{\nu}}\right) + \mathbf{D}\boldsymbol{\nu}_d + \mathbf{M}\dot{\boldsymbol{\nu}}_d, \tag{2.24}$$

where $\boldsymbol{\tau}_{FF} \triangleq \mathbf{D}\dot{\boldsymbol{\nu}}_d + \mathbf{M}\boldsymbol{\nu}_d$.

The control law can also be extended with acceleration feedback, $\boldsymbol{\tau} \triangleq \boldsymbol{\tau}_{PID} - \mathbf{K_m}\dot{\boldsymbol{\nu}}$, where $\mathbf{K_m} > 0$. This yields the expression

$$\dot{\boldsymbol{\nu}} = -(\mathbf{M}+\mathbf{K_m})^{\mathbf{-1}}\mathbf{D}\boldsymbol{\nu} + (\mathbf{M}+\mathbf{K_m})^{\mathbf{-1}}\boldsymbol{\tau}_{\mathbf{PID}} + (\mathbf{M}+\mathbf{K_m})^{\mathbf{-1}}\mathbf{w}, \tag{2.25}$$

and besides increasing the system inertia matrix the acceleration feedback will reduce gain from the disturbance $\mathbf{w}$. If in addition the acceleration feedback is filtered through a filter (e.g., low-pass, notch), it will only have effect at chosen frequencies. If a notch structure is chosen, the acceleration feedback can be used to reduce first-order wave-induced disturbances. For more details about the acceleration feedback in DP systems, please consult [27].

An often practical problem for control laws implemented on physical system is nonlinearities because of saturation limits in actuators. If integral action is applied to a system with saturation limits, an undesired side effect known as integrator windup may occur. This problem mainly occurs when a step is applied to the reference setpoint, which can be reduced by using a reference model, but it can also occur when a step is applied to the manipulated variables because of disturbances. To overcome this problem, several anti-windup techniques has been suggested. A survey of different anti windup techniques will not be given here, it will only be referred to [2] where 4 different strategies are mentioned, and [55] where a modified anti-windup scheme is proposed.

LQR

As the name indicates, a linear quadratic regulator (LQR) is a controller for a linearized system, a second necessary condition is that the system is controllable, see [16] for the controllability definition. The linearized 3 DOF model proposed in Section 2.1.3 can be set up in a state space model with $\mathbf{x} = [\boldsymbol{\eta}^T, \boldsymbol{\nu}^T]^T$ and $\mathbf{u} = \boldsymbol{\tau}$ as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w} \tag{2.26}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x}, \tag{2.27}$$

where:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3x3} & \mathbf{I}_{3x3} \\ \mathbf{0}_{3x3} & -\mathbf{M}^{-1}\mathbf{D} \end{bmatrix} \tag{2.28}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3x3} \\ \mathbf{M}^{-1} \end{bmatrix} \tag{2.29}$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{0}_{3x3}. \end{bmatrix} \tag{2.30}$$

The optimal feedback control law

$$\mathbf{u} = -\mathbf{G}\mathbf{x} \tag{2.31}$$

is found by minimizing the performance index:

$$J = \min_{\mathbf{u}} \frac{1}{2} \int_0^T (\mathbf{x}^\mathbf{T}\tilde{\mathbf{Q}}\mathbf{x} + \mathbf{u}^\mathbf{T}\mathbf{R}\mathbf{u})dt, \tag{2.32}$$

where $\mathbf{R} = \mathbf{R}^\mathbf{T} > \mathbf{0}$ and $\tilde{\mathbf{Q}} = \mathbf{C}^\mathbf{T}\mathbf{Q}\mathbf{C} = \tilde{\mathbf{Q}}^\mathbf{T} \geq \mathbf{0}$ are the weighting matrices. The $\mathbf{Q}$-matrix is defined as $\mathbf{Q} = diag\{\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3\}$, and puts penalty on position/heading and velocity. The $\mathbf{R}$ matrix put weight on the use of actuators. The steady state solution to this problem is:

$$\mathbf{G} = \mathbf{R}^{-\mathbf{1}}\mathbf{B}^\mathbf{T}\mathbf{P}_\infty \tag{2.33}$$

$$\mathbf{0} = \mathbf{P}_\infty\mathbf{A} + \mathbf{A}^\mathbf{T}\mathbf{P}_\infty - \mathbf{P}_\infty\mathbf{B}\mathbf{R}^{-\mathbf{1}}\mathbf{B}^\mathbf{T}\mathbf{P}_\infty + \tilde{\mathbf{Q}}, \tag{2.34}$$

where $\mathbf{P}_\infty = \lim_{t\to 0}\mathbf{P}(t)$.

The LQR can easily be extended to a trajectory tracking controller with integral action. This is usually done by defining $e \triangleq x - x_d$, where x_d is a smooth bounded reference signal, and integral action is obtained by augmenting the system with an integral state $\dot{z} \triangleq e$. By implementing the integral action, the controller can compensate for slowly-varying disturbances. The optimal control law can also be extended with reference feedforward or disturbance feedforward, please consult [16] for more details about this.

2.2.2 Nonlinear Control Theory

As already showed in Section 2.1.3, the dynamic equation of motion (2.6) can be reduced to a linear equation of motion for dynamic positioning (2.8) by taking some assumptions. On the other hand, in the real system there will always be some nonlinearities and especially for a vessel in seaway. With nonlinearities in the real model and the simplicity and design flexibility that comes with nonlinear control methods, nonlinear design methods have become very attractive for dynamic positioning control design.

Feedback Linearization

With feedback linearization, nonlinear system dynamics is cancelled out such that the system is transformed into a linear system. Consider the kinetic equation:

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}, \tag{2.35}$$

where by selecting the control law

$$\boldsymbol{\tau} = \mathbf{M}\boldsymbol{\tau}_{\mathbf{acc\_fb}} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu}, \tag{2.36}$$

the nonlinear term $\mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu}$ is canceled out. The command acceleration $\boldsymbol{\tau}_{acc\_fb}$ can be chosen as one of the linear control laws already mentioned (e.g., PID, LQR). Note that to do the cancelation of the nonlinearities, the $\mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu}$ term must be known exactly, which means that both the values in $\mathbf{D}(\boldsymbol{\nu})$ must be known precisely, and a good measurement (estimation) of $\boldsymbol{\nu}$ is necessary. Note also that this will be a velocity controller and measurement of the acceleration must be available to include the derivative term in the controller. If a position/attitude controller was to be developed, also the kinematic part of the motion equation must be included in the control law. For the LQR control law, the cost function must then be investigated according to the nonlinear equation.

Integrator Backstepping

A drawback with the feedback linearization technique is the necessity in having a precise model, which of course is very difficult in practise. In integrator backstepping on the other hand, the designer is free to exploit "good" nonlinearities and dominate "bad" nonlinearities by adding nonlinear damping, which will give an improved robustness. A recursive design technique using *control Lyapunov functions* (CLFs) is the core of the integrator backstepping technique, see [16] for definition of the CLF. For each step, a sub CLF will be developed, a *virtual control* input defined with a stabilizing function, and a new state variable defined. The new state variable will keep the error to the next sub CLF and finally a control law for the input τ is built up. Examples in using integrator backstepping to develop a nonlinear PID-control for a 3 DOF DP model is mentioned in [16,27].

In [16], an example with adaptive integrator backstepping is mentioned for a mass-damper-spring system. According to [16], a drawback with this law is that it is more sensitive to measurement noise than the integrator augmentation technique. This because the adaptive technique involves integration of the velocity error. In [27], different control laws extended with acceleration feedback are mentioned, most of them related to a PID structure.

Even though the design technique is simple and flexible with integrator backstepping, the control law becomes rather complex and tuning of the control law can be difficult. To illustrate the complexity, an illustrative example is the PID control law

$$\boldsymbol{\tau}_{PID} = -\mathbf{K_i}(r)\mathbf{R}^T(\psi)\boldsymbol{\xi} - \mathbf{K_p}(r)\mathbf{R}^T(\psi_e) - \mathbf{K_d}\boldsymbol{\nu}_e \tag{2.37}$$

$$\dot{\boldsymbol{\xi}} = -\boldsymbol{\Lambda}\boldsymbol{\xi} + \mathbf{R}(\psi_d)\mathbf{e}, \tag{2.38}$$

where:

$$\mathbf{K_i}(r) = (\mathbf{D} + \mathbf{C}_2)\boldsymbol{\Delta}_2^{-1}\boldsymbol{\Delta}_1 - \mathbf{M}\boldsymbol{\Delta}_2^{-1}\boldsymbol{\Delta}_1\boldsymbol{\Lambda} + \mathbf{rM}\boldsymbol{\Delta}_2^{-1}\mathbf{S}^T\boldsymbol{\Delta}_1 \tag{2.39}$$

$$\mathbf{K_p}(r) = \boldsymbol{\Delta}_2 + (\mathbf{D} + \mathbf{C}_2)\mathbf{C}_{12} + \mathbf{M}\boldsymbol{\Delta}_2^{-1}\boldsymbol{\Delta}_1 + \mathbf{rMC}_{12}\mathbf{S}^T \tag{2.40}$$

$$\mathbf{K_d} = \mathbf{C}_2 + \mathbf{MC}_{12}, \tag{2.41}$$

which was mentioned in [27] for a low-speed model. It is easy to see that tuning of the controller gains $\mathbf{C}_{12} \in \mathbb{R}^{3x3}$, $\mathbf{C}_2 \in \mathbb{R}^{3x3}$ and $\boldsymbol{\Lambda} \in \mathbb{R}^{\eta_\xi x \eta_\xi}$ is not a simple task. Notice that $\mathbf{K_p}$ and $\mathbf{K_i}$ has yaw rate dependent gains, where the term $\mathbf{S}^T$ is a skew-symmetric matrix, while the terms $\boldsymbol{\Delta}_1 \in \mathbb{R}^{\eta_\xi x \eta_\xi}$ and $\boldsymbol{\Delta}_2 \in \mathbb{R}^{3x3}$ are positive definite and symmetric matrices with some design restrictions that will affect the controller gains. Please consult [27] for more details about this control law, some thought about the choice of the controller gains and some practical implementation problems.

2.2.3 Intelligent Control Methods

Intelligence can be defined as:

1. Someone‘s *intelligence* is their ability to understand and learn things.
2. *Intelligence* is the ability to think and understand instead of doing things by instinct or automatically.

Essential English Dictionary, Collins, London, 1990

In the second definition, it is not distinguished if it is someone or something that thinks, and in the same dictionary, thinking is defined as:

> *Thinking* is the activity of using your brain to consider a problem or to create an idea.

These two definitions are in [33] gathered into the definition:

> *Intelligence* is the ability to learn and understand, to solve problems and to make decisions.

To extend the definition of intelligence to a definition of *intelligent control*, it must be answered if a computer (machine) can think or be intelligent, and one of the earliest test (definition) of *'machine intelligence'* is the *Turing imitation game*. Turing defined that a computer passes a intelligence test if interrogators cannot distinguish the computer from a human on the basis of the answers to their questions. The Turing imitation game is undertaken in two phases: In the first phase, a person (interrogator) communicates with a man and a woman through a terminal, where the interrogator shall decide who is the man and who is the woman, and the man shall try to convince the interrogator that he is a woman by answer the questions as he is the woman. In the second phase, the man is replaced with a computer program that shall answer the question as it was a man pretending to be woman. The computer program would then make mistakes and provide fuzzy answers in the way a human would, and if it fools the integrator as often as the man did it passed the Turing test. If the computer passes the Turing test it would have the ability to process all the important characteristics of intelligence, and by combing this intelligence with control theory it could be said to be intelligent control. Design of intelligent control is in [40] presented as: "The design of intelligent control system should be based on an attempt to understand and duplicate some or part of phenomena that ultimately produces a kind of behaviour that can be termed 'intelligent', i.e. generalisation, flexibility adaptation etc."

In the last two decades intelligent control methods have become very popular, and in the last decade also intelligent control methods has been proposed for dynamic positioning, see for instance [5, 6, 58]. The reason for this research interest in intelligent control methods is the fact that dynamic positioning is a very complex problem, and even though there has been made good mathematical models of the marine structures and their behavior in water, they are not perfect and often approximation must be taken to be able to solve the mathematical problem. Another drawback by the mathematical model is that it can be difficult and time consuming to find the parameters in the model, and the parameters found for a model might not apply for different vessel operational conditions (VOC).

Fuzzy Logic

A method that is generally accepted as an intelligent control method is fuzzy logic, and in [6] a fuzzy logic controller for a drilling vessel has been proposed.

The idea behind fuzzy logic is that all things admit on degrees. In logic theory this means that most physical state are not just logical on or off, but are somewhere on a sliding scale, e.g.: an error can be very small, small, medium, large, or very large. Fuzzy logic is a logic that describe such fuzziness, and the core in the fuzzy logic theory is the fuzzy sets that calibrates vagueness. A fuzzy set is defined by using linguistic variables (i.e., small, medium, large, or slow, average, fast) as fuzzy variables, and make this as the boundaries for the fuzzy sets. But as already stated, a state is not just small or medium and so on, it can also be defined as very small or slightly small, and a state that is very small will also be consider as small, so then very small will be a subset of small. Hedges like extremely, very, and slightly defines the shapes of a fuzzy subset. Finally the boundaries can overlap each other, that means a state can be both partly small and partly medium, to avoid sharp limits. The function in the fuzzy logic controller is then to use fuzzy associative memory (FAM) rules to correlate a fuzzy input set to a fuzzy output set. A FAM rule is a logical *if-then* type statement, with one or multiple antecedents, and with one or multiple consequences. In Figure 2.5, a schematic of a fuzzy logic controller illustrated in [40] is reviewed. In this illustration the conventional controller blocks is replaced by a composite block including four components:

- The FAM rule base, with 'if-then' rules.
- The fuzzy inference engine, that decides which rules that are relevant to a particular input, and pass the action from the rule to the output.
- Input fuzzification, which converts the input value to a useful value for the fuzzy inference engine.
- Output defuzzification, which converts the value form the fuzzy inference engine to a value useful for the actuators.

Clearly, both to define the fuzzy sets and to set up the FAM rules experience and expert knowledge about the system is necessary. A typical process in developing a fuzzy logic controller will be:

- **Specify the problem and define linguistic variables.** This is probably the most important part, and expert knowledge about the system is necessary to determine the input and output variables of the controller, and the ranges of these variables.
- **Determine fuzzy sets.** Based on the problem specification and the linguistic variables, the fuzzy sets must detected. That means expert knowledge in fuzzy logic is necessary to be able to set up the boundaries and the shapes of the fuzzy sets.

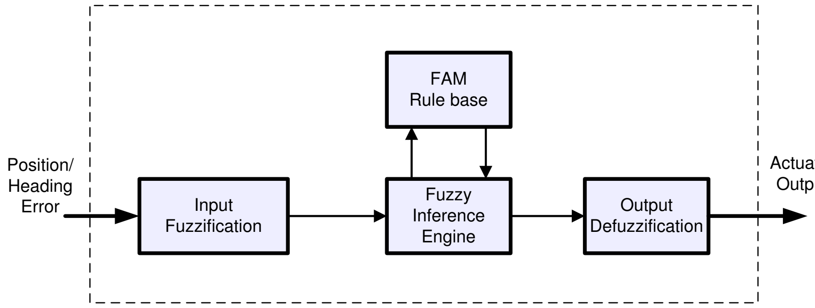


Figure 2.5: Fuzzy Logic controller, where the conventional controller block is replaced by a composite fuzzy logic block. Adapted from [40].

- **Elicit and construct fuzzy rules.** Based on the linguistic variables the problem must be solved by setting up FAM rules. Again expert knowledge about the system is necessary.
- **Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system.** This is the part were the actual fuzzy logic controller is implemented as a computer program.
- **Evaluate and tune the system.** It is not likely to believe that all the fuzzy sets and the rules are set optimal without tuning them, it is more likely to believe that this is the most laborious part of developing the fuzzy logic controller.

The drilling vessel in [6] is equipped with one bow thruster, a single main propeller and a rudder. For solving this problem, 52 fuzzy set are detected and in total 224 FAM rules are set up. By assuming port-starboard symmetry in the hull, the amount of FAM rules to be determined was reduced to 116. In general this means that by using this method, and assuming port-starboard symmetry of the hull, at least 116 FAM rules has to be determined and 52 fuzzy set has to be detected in developing a fuzzy logic controller on the force-moment level for a marine vessel. And finally this has to be evaluated and tuned.

Neural Network

A *neural network*, also called an *artificial neural network* (ANN), is a mathematical model or computational model based on biological neural networks. The fundamental and essential characteristic in a biological neural network is learning, and it has been shown that ANN are capable of learning by using their experience to improve their performance. ANN has been shown to recognise hand written characters and observe patterns that human experts have failed to recognise [33].

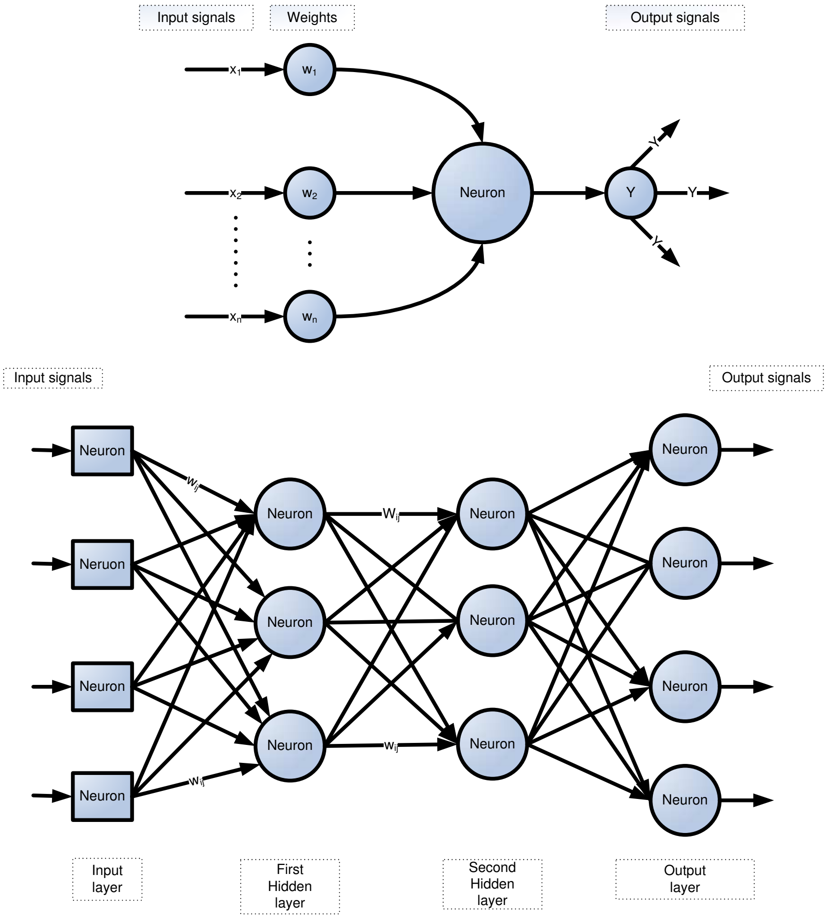


Figure 2.6: A single neuron and a multilayer neural network. Adapted from [33].

The core in ANN are groups of artificial neurons which are interconnected, and where each connection passes signals between the neurons. The connection is a numerical-weighted link, where the values of weights are adjusted during the learning process, based on external or internal information that flows through the network. The value of the weight can be seen as the importance of the link, low value means low importance. The weight can also be thought as the basic means of long term memory in an ANN, and by learning/adjusting the weighting factors, a structure will be built up between the inputs and the outputs of the network. Typically the ANN consist of an input layer and an output layer of neurons, where the input and output layer can be directly connected or consists of one or several hidden layers of neurons. According to [33], any continuous function can be represented with one hidden layer, and with two hidden layers even discontinuous functions can be represented. However, it has to be mention that the computational burden increases with the number of layers, and the computational burden and the long learning time are the largest drawback when it comes to ANN and using them in control systems. To reduce the computational burden and decrease the learning time, a lot of different learning techniques for adjusting the weight has been proposed, but so far none of them have seen to be perfect. Please consult [33], for more details about ANN and different learning techniques. See Figure 2.6 for an illustration of neurons and a multi-layer network.

In [58], a dynamic positioning controller for floating structures that uses an on-line training neural network predictor to predict the output of the MIMO system has been described. The described network was applied with 2-input and 6-output, and is a single layer network that predicts unknown parts in the dynamic equation of motion. This means that a stationary model is used as basic for the controller and the predictor only predict errors beyond this model. The controller is a combination of the neural network predictor and an optimal controller, which is called a neural network controller. The optimal controller is also implemented with constraints on the actuators saturation limits and the actuators rate limits. Please consult [58] for simulation results and conclusion about the described controller.

An extension to the fuzzy logic and neural network is to combined them, which is called an Adaptive Neuro-Fuzzy Inference System (ANFIS), then the shape of the membership functions can be trained with input/output data rather than specifying them manually. Please consult [33], for a more detailed guide about fuzzy logic, artificial neural networks and other intelligent system. In [40] a historical perspective of intelligent ship autopilots is presented, where the conclusion was that: "Until the stability issue of intelligent control is properly addressed and generic solutions formulated, these kinds of advanced control systems cannot be fully developed and will not gain acceptance, especially in certificated and safety-critical applications".

2.3 Autotuning Methods

Autotuning is a process where the controller parameters are (in some way) automatically adjusted. The automatic adjustment can be done by demand of the operator or be a continuous process. An autotuning procedure can be divided into an identification phase and a controller design (parameter calculation) phase. To do the identification, the process must be brought to some kind of excitation, which is normally carried out by putting a step or one or several sine signals on the input of the process. Another method is the frequency response method, which is based on finding the critical frequency and the critical gain of the process by setting the process in some kind of oscillation. The frequency response method will not be covered further because it seems inconvenient to put a vessel into oscillation (instability).

Other used phrases for autotuning controllers are self-tuning controllers or adaptive controllers. According to Cambridge's dictionary, adaptive means "possessing an ability to change to suit different conditions", which is the goal of this thesis. Adaptive control is covered briefly in Section 2.3.1.

Use of intelligent methods for automatic tuning of controllers has attracted much interest in the research area of controller design during the last decade. A method which there has been published several articles about is *genetic algorithms* (GA), e.g., [26,30,45,48,60], which is briefly covered in Section 2.3.2. Other examples on intelligent methods used for autotuning are fuzzy neural networks, reinforcement learning, and immune algorithms. For instance; in [44], a fuzzy neural network is used for tuning a PID-controller; in [41, 56], adaptive PID controllers based on reinforcement learning are mentioned; and in [10], the immune algorithm is used in tuning a PID-controller. However, GA seems to be the most used method so far and will therefore be the only method covered here.

Regardless of which method that is used in the autotuning procedure, a definition of what would be considered as good performance of the system must be made. This is often called performance assessment, which leads to the necessity of a performance index. Performance assessment and performance indices are discussed in Chapter 3.

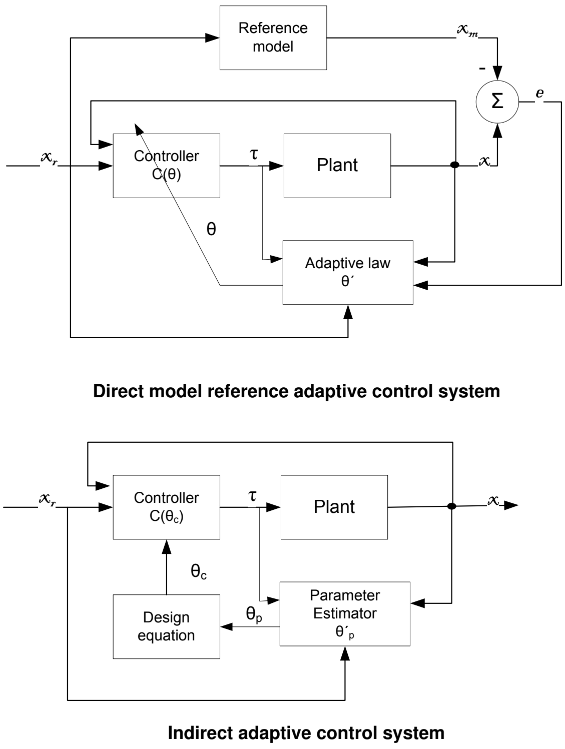
2.3.1 Adaptive Control

According to [21], adaptive control was primary motivated from the need of good regulation system for high performance aircraft, which are operating over a wide range of speeds and altitudes. From the first adaptive control system in 1950, active research in the field has proposed different type of adaptive control scheme. The different types of adaptive controls can generally be divided into; gain scheduling, model reference adaptive control (MRAC) and adaptive pole placement control (APPC). In some literature the MRAC has been called model reference adaptive system (MRAS), and the APPC has been called self-tuning regulators (STR), e.g., in [50].

The gain scheduler is an adaptive controller where controller parameters are precalculated off-line and fixed according to different operation conditions/points. This gives N sets of controller parameters for N different operation conditions/points, which are stored in a look-up table. Then an auxiliary measurement are used to detect the different operation conditions/points and behalf of this measurement, appropriate controller parameters are selected from the look-up table. The advantage with gain-scheduling is that the controller parameters will change rapidly when the operation condition/point changes. The disadvantage is that controller parameters are fixed and will not compensate for changes in plant parameters, or changes that goes beyond the predefined conditions. Another disadvantage is that it can be rather time consuming to define different operations and precalculate controller parameters for them, hence the implementation can be very expensive.

MRAC is derived from model reference control (MRC) where a reference model is designed behalf of the desired I/O properties of the closed-loop plant. In MRC the objective is to find a feedback control law that follows the I/O properties of the reference model, i.e., the output of the plant follows the output of the reference model. In the adaptive case (MRAC) the controller gains of the MRC are updated by some kind of on-line estimation.

APPC is derived from pole placement control (PPC), where the feedback control law is designed to place the poles of the closed-loop plant at a desired locations. Then as for MRAC, the controller gains are updated by some kind of on-line estimation in the adaptive case (APPC). For example, the LQR control law in Section 2.2.1 can be used as a basis of PPC control law, and with adaption of the controller gains it would be an APPC. See Figure 2.7 for a block schematic illustration of a direct MRAC and a APPC.



Direct model reference adaptive control system

Indirect adaptive control system

Figure 2.7: Illustration of MRAC and APPC. Adapted from [53].

In [21], where a lot of different MRAC and APPC schemes are shown, it is also shown that the MRAC can be consider as a special class of the APPC, so the distinction between them are more historical than conceptual. Either the adaptive controller is MRAC or APPC it is formed by combining an on-line parameter estimator, which provides estimate of unknown parameter at each instant, with a control law that is motivated from the known parameter case. Estimation of the controller parameters can be direct or in-direct, and the estimation is referred to as the *adaptive law*. In the indirect case, plant parameters are estimated on-line, and then used to calculate controller parameters. While in the direct case, the controller parameters are estimated directly without intermediate calculation of estimated plant parameters.

In direct estimation of the controller parameters, the plant model must be parameterized in terms of the controller parameters, which gives restriction to certain necessary properties of the plant. A class of plant that have these properties are all SISO LTI plants that are minimum-phase, i.e., their zeroes are located in $Re[s] < 0$. For the direct MRAC schemes shown in [21], it is necessary that the plants are minimum phase, while the direct APPC schemes are restricted to plant that are scalars, or to plants where the desired parameters of the pole placement controller can be expressed in the form of linear or bilinear parametric models. For the indirect schemes, MRAC or APPC, the minimum phase restriction is relaxed, but the mapping between the controller parameters and the plant parameters cannot be guaranteed to exist at all time steps, which can give stabilization problem according to [21].

An Indirect Adaptive Pole Placement Control Scheme

A pole placement controller (PPC) for a PID-controller is mentioned in [16], where the poles are selected in the choice of the natural frequency ω_n and the damping term ζ (which implicitly is the closed loop bandwidth, see [16] for details). Controller parameters K_p, K_d are chosen according to perfect model matching and $K_i = K_p/T_i$ are selected by using the rule of thumb that the integral time constant can be chosen to $T_i \approx \frac{\omega_n}{10}$. This gives the following controller parameters:

$$K_p = m\omega_n^2 \tag{2.42}$$

$$K_d = m2\zeta\omega_n - d \tag{2.43}$$

$$K_i = \frac{\omega_n}{10}K_p = m\frac{\omega_n^3}{10}, \tag{2.44}$$

An indirect APPC would then be a controller where the controller gains are adaptive adjusted by some on-line estimation of m and d. Model parameter estimation is not an issue for this thesis and will not be covered.

Direct Adaptive Control

For a direct MIMO MRAC scheme model requirements are more stringent than for a SISO case. For instance, for a minimum phase SISO system stability can be proven by knowing the sign of the input matrix, while for a MIMO system the high-frequency gain matrix K_p is the stumbling block. In [21], it is assumed that a matrix S_p is known such that $\mathbf{K_p S_p} = (\mathbf{K_p S_p})^T > \mathbf{0}$, which is very strict and can be difficult to fulfil. To lighten this, different decompositions of the $\mathbf{K_p}$ matrix is done in [53], which is valid given that all leading principle minors of $\mathbf{K_p}$ are none zero. A **SDU** factorization (**S** is symmetric positive definite matrix, **D** is diagonal matrix and **U** is unity upper triangular matrix) of the $\mathbf{K_p}$ matrix is shown in [7], where only the sign of the entries of **D** are assumed to be known. The **SDU** factorization is also used in [8, 31], where the MIMO MRAC scheme is extended to systems with relative degree two (in [8]) and for systems with state delay (in [31]). All factorization/decomposition schemes in these references are restricted to system where the sign of all leading principle minors of $\mathbf{K_p}$ are known and nonzero. A solution for system where this is not fulfil is found in [57], where a complex hysteresis switching logic is used. Even tough the factorization is simpler than the switching logic it is not simple and require some parameters to set.

Dynamic positioning of vessel is indeed a multivariable task and with the rotation matrix it can seem difficult to use standard MIMO MRAC schemes. But the rotation matrix is a known value and the error can be rotated to BODY before implying adaptation laws, which reduces the model to a standard linear model. This have been used in passivity proof of adaptive filters for DP (e.g., [18]). Furthermore, by assuming that cross-coupling terms are very small in the system inertia matrix (**M**) and the damping matrix (**D**), they can be neglected. With this assumption the model is decoupled down to a 2nd order SISO model for each DOF. This is of course not a correct assumption, but the cross terms are rather small compared to the diagonal elements, and with CG coinciding with CP of the vessel, the system inertia matrix will in many cases become close to diagonal, since the added mass matrix is typical much smaller than the rigid-body system inertia matrix. The minimum phase requirement is also met for almost all vessels, and a SISO MRAC scheme for each DOF should be implementable.

In [52], a MRAC scheme with a PID structure is implemented in surge on a model of a shuttle tanker. It is by my knowledge not implemented in 3 DOF, but with the previous assumption it should be possible. This MRAC scheme is very equal the scheme used for an autopilot in [1], where a practical drawback with the MRAC scheme is stated. The problem is that it is based on the assumption that perfect model matching can be achieved. Since almost all physical system contains strong nonlinearities in saturation limits and rate limitation in the actuators, this can be very difficult to obtain. A solution is to implement nonlinearities in the reference model, another solution was suggested by [1] and reviewed in [15]. This solution is based on modifying of the control input so the reference model remains linear, and the estimated parameter remains bounded. The idea is to make a command generator which has mechanism that compensates for the actuator limitations, in front of the input to the reference model and the controller. Inputs to actuators should then remain bounded inside their limitations.

2.3.2 Genetic Algorithms

According to [33], genetic algorithms (GAs) are a class of stochastic search algorithms based on biological evolution, which involves the principal of natural selection and genetic modification. GA is an optimization method that operates with a population of points, which all are possible solutions to the optimization problem. Each point is designated as an individual, often called chromosome, and given a fitness value that indicates how well it solves the optimization problem. In the transition from one generation of a population to the next, an evaluation of the chromosomes in the population is carried out to build up a new population for the next generation. The evaluation is carried out by the genetic operators: selection, crossover and mutation.

- **Selection** will select the fittest chromosomes to survive to the next generation. There are several way for how the selection is carried out, but the most common methods are the roulette wheel selection and the tournament selection. In the roulette wheel selection method, each chromosome is given a slice of a roulette wheel. The size of the slice depends on the fitness value of the chromosome, which will favour the chromosome with the best fitness value. The wheel is to be spun the size of the population, and for each spin, a chromosome is selected to survive to the next generation. In the tournament selection method, the fittest individual of z randomly chosen chromosomes, will survive to the next generation. This process has to be repeated the size of the population. By increasing the number of chromosomes (z) in the tournament, chromosome with higher fitness value will be favoured. This leads to faster convergence, but can also lead to convergence to a local optimal solution, in stead of the global optimal solution. The tournament selection is in many application carried out only between two chromosome, i.e. z = 2.

- **Crossover** makes new genetic material to make progress in the optimization. New chromosomes are made by taking two chromosomes and dividing them into two parts at a random crossover point, and then exchanging the parts between the two chromosomes. This means that one of the chromosomes will now be a combination of the first part of chromosome one and the last part of chromosome two, and the other chromosome will be a combination of the opposite. For real number representation it is suggested in [22] to make two new chromosomes as a linear combination of an random chosen percentage factor (crossover parameter) of two chromosome:

$$c_1^{new} = \lambda c_1^{old} + (1-\lambda) c_2^{old} \tag{2.45}$$

$$c_2^{new} = \lambda c_2^{old} + (1-\lambda) c_1^{old} \tag{2.46}$$

where $\lambda \in [0, 1]$ is a crossover parameter. The probability that a crossover will occur is defined by a fixed parameter before the optimization process starts, and a normal value is according to [33], 70%.

- **Mutation** involves that a randomly chosen chromosome changes its value, and then creates a new chromosome. Like for crossover, a predefined probability for a mutation to occur is defined, this probability is normally chosen low (e.g. 0.1%). The role of the mutation, is to provide that the optimization process not got trap in a local solution.

For each generation, the fitness calculation for each chromosome and the evaluation process is done, and new generation will be made until a predefined termination factor is achieved. The termination factor is normally given as a predefined max number of generation, or it can be a limit for the fitness value. A basic genetic algorithm is presented in Figure 2.8.

The features of GA according to [22] is that it is able to find an optimal or suboptimal solution in a complex and large search space. GAs are also applicable to nonlinear optimization problems with constraints that can be defined in discrete or continuous time. By using a population of possible solutions, compared to an individual one, the search will be carried out in parallel and several solution are examined at the same time, which gives a higher probability for convergence to an optimal solution.

This can be summarized in that GA have a high probability in finding an optimal or suboptimal solution to optimization problems, which not obviously have an analytic solution, or to problems that are very difficult to solve analytical. On the other hand, the probability in finding an optimal solution depends on the size of the population and the setting of the probability-parameters for crossover and mutation. Values for this parameters, which are increasing the probability in finding a solution, are often decreasing the convergence rate, which again can make the algorithm very time consuming. Even with a high probability in finding a solution you are never 100% certain in finding an optimal solution. The probability in finding the optimal solution will also depend on the definition of the fitness value. For more details about use of GA in controllers please consult [14,22].

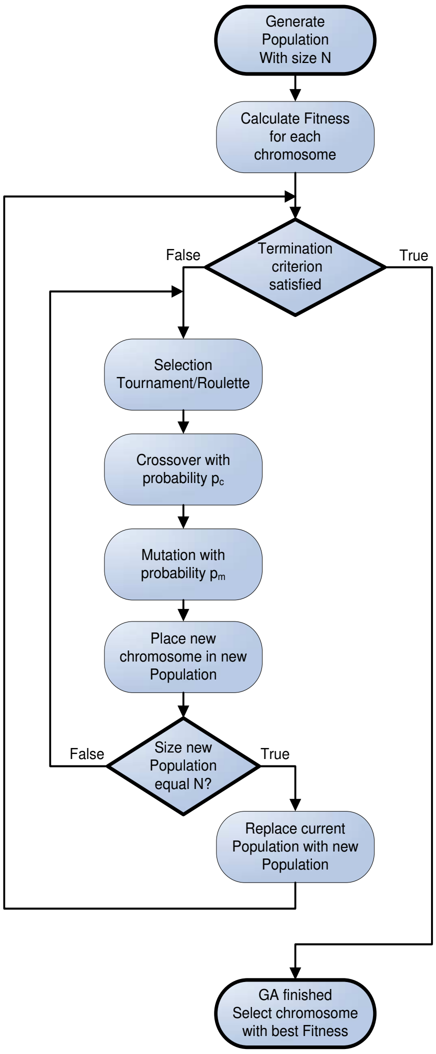

Generate Population With size N
Calculate Fitness for each chromosome
False
Termination criterion satisfied
True
Selection Tournament/Roulette
Crossover with probability p_c
Mutation with probability p_m
Place new chromosome in new Population
False
Size new Population equal N?
True
Replace current Population with new Population
GA finished Select chromosome with best Fitness

Figure 2.8: A basic Genetic Algorithm with selection, crossover, mutation and termination.

Chapter 3

Performance Assessment

3.1 Vessel Operational Conditions

Marine vessels are performing complex operations in shifting conditions. According to [37], a vessel operational condition (VOC) can be defined as $VOC \triangleq (VUM, Env, VC)$, where:

- *VUM* refers to the vessel use mode, which describe the current task of the vessel, e.g.; Replenishment at sea, pipe laying, drilling, etc. A vessel use mode may require different types of guidance navigation and motion control system (GNC), e.g.; course keeping, station keeping, speed regulation, wave motion damping.

- *Env* refers too the state of the environment, i.e., wind, wave and ocean-current condition.

- *VC* refers to the current condition of the vessel. That spans over the loading condition, vessel speed, intact or damaged units, available power, etc.

(In [36], the VOC is defined as $VOC \triangleq (usemode, speed, env, loading)$). In this thesis, only station keeping operations will be considered. Hence, the velocity and the vessel use mode attribute will be constant and can be omitted for the definition. According to [36], the loading attribute is referred to as the mass distribution and to the vessel condition (intact or damaged). From the relation that the buoyancy and the weight of a floating vessel at rest are in balance $mg = \rho g \nabla$, it follows that when the draught of the vessel changes, the mass of the vessel will change proportionally or vice versa. Hence, the model parameter values of the system inertia matrix (2.12) will change. From this relation, the vessel condition (or loading) will be defined as a draught condition, since the draught of the vessel can be related to the mass distribution. By this simplification it also follows that it is assumed that everything is intact. According to [15], changes of water depth can have effect on the performance of the vessel and is another attribute to consider. This leads to the following definition of the VOC variable O:

$$O(\delta, \varepsilon, \varpi) \quad \text{where} \quad \begin{cases} \delta(draught) \\ \varepsilon(environmental - conditions) \\ \varpi(water - depth), \end{cases}$$

which is graphically illustrated in Figure 3.1.

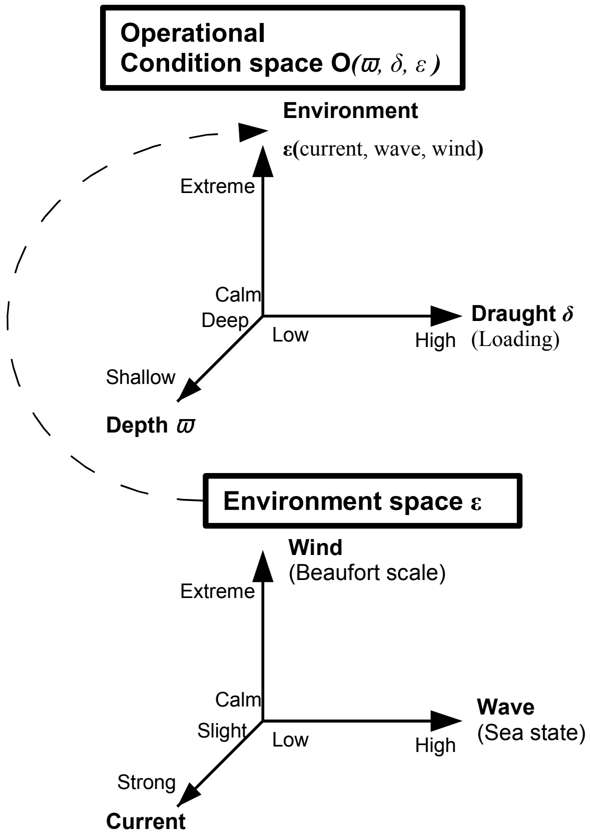


Figure 3.1: Operational Condition space and Environment space.

Environment Space

In practice, the environmental variable $\boldsymbol{\varepsilon}$ will be a function of winds, ocean-currents, and waves, see Figure 3.1. Wind is characterized by a direction ψ_w and a velocity V_w, where the direction basis is fixed to the surface of the earth. The wind experienced onboard the vessel has to be considered relative to the vessel in the rigid body coordinates. The experienced incoming wind direction and speed will be

$$\gamma_r = \psi_w - \psi \tag{3.1}$$

$$V_r = \sqrt{u_r^2 + v_r^2}, \tag{3.2}$$

where

$$u_r = V_w \cos(\psi_w - \psi) - u \tag{3.3}$$

$$v_r = V_w \sin(\psi_w - \psi) - v. \tag{3.4}$$

The wind speed and its direction can be measured by a wind sensor, and the measured signal is normally filtered since only mean wind force and moments can be compensated for by the actuators. In [16], two 3 DOF models as a function of γ_r and V_r are discussed. By using one of these models, wind forces and moments acting on the vessel can be compensated for by a feedforward term τ_{wind} in the controller.

The slowly-varying part of the ocean current, can (in the same way as wind) be considered as a mean speed V_c with an earth-fixed direction β_c. Hence the ocean current, can (similarly to the wind) be decomposed to a speed relative to the rigid body frame. The relative speed of the vessel according to the fluid are:

$$u_r = u - V_c \cos(\beta_c - \psi) \tag{3.5}$$

$$v_r = v - V_c \sin(\beta_c - \psi) \tag{3.6}$$

$$r_r = r, \tag{3.7}$$

or in vector form $\boldsymbol{\nu}_r = [u_r, v_r, r_r]^T$. It is reasonable to assume that the acceleration of the ocean current is approximately zero, at least since only slowly varying forces are consider. In station keeping, the velocity of the vessel is assumed low, so it follows that the ocean current can be considered as a bias to the vessels' velocity. This bias can be compensated for by including an integral term in the controller.

According to [16], wave-induced forces and moments can be separated into *1st-order* and *2nd-order* effects for simulations. The 1st-order effect represents wave-frequency (high-frequency) motion, and the 2nd-order effect represent wave drift forces (low-frequency motion). Wave drift forces, also called higher order loads, encompasses forces and moments whose magnitude are proportional or higher to the square of the waves' amplitude. These forces will have a relatively low frequency compared to the linear, purely oscillatory motion from the wave frequency motion, which oscillate with the wave frequency. An easy model for simulation of wave forces and moments for the 3 DOF DP model are given in [16] as:

$$X_{waves} = \frac{K_{\omega_1} s}{s^2 + 2\lambda_1 \omega_{o1} s + \omega_{o1}^2}\omega_1 + d_1 \tag{3.8}$$

$$Y_{waves} = \frac{K_{\omega_2} s}{s^2 + 2\lambda_2 \omega_{o2} s + \omega_{o2}^2}\omega_2 + d_2 \tag{3.9}$$

$$N_{waves} = \frac{K_{\omega_3} s}{s^2 + 2\lambda_3 \omega_{o3} s + \omega_{o3}^3}\omega_3 + d_3, \tag{3.10}$$

where the first part represent wave-frequency motion and the last represent wave drift forces.

In the wave-frequency motion model, $\omega_i (i = 1, 2, 3)$ are Gaussian white noise processes, and $\omega_{oi} (i = 1, 2, 3)$ is the wave frequency. Wave amplitudes are adjusted by choosing the constants $K_{omega_i} (i = 1, 2, 3)$. The wave spectra are parameterized in terms of λ_i and $\omega_{oi} (i = 1, 2, 3)$, which should be chosen to represent the true physical behavior. A recommended value for the damping coefficient λ_i is 0.1 if the JONSWAP[1] spectrum is used according to [16], where it also is suggested that the wave amplitude constant can be calculated as

$$K_\omega = 2\lambda\omega_o\sigma, \tag{3.11}$$

where σ is a constant describing the wave intensity. In Table 3.1, wave height and frequency for the north sea is given according to the sea state codes of [38].

The wave drift forces $d_i (i = 1, 2, 3)$, can be modeled as slowly-varying bias terms:

$$\dot{d}_1 = \omega_4 \tag{3.12}$$

$$\dot{d}_2 = \omega_5 \tag{3.13}$$

$$\dot{d}_3 = \omega_6, \tag{3.14}$$

where $\omega_i (i = 4, 5, 6)$ are Gaussian white noise processes. A modification to these equations is to implement saturation limits to avoid d_i to exceed maximum physical limits.

[1] Joint North Sea Wave Project (JONSWAP) was carried out in the North Sea in 1968 and 1969, as an extensive measurement program. From these measurements, a spectral density function which is used to describe *non-fully developed seas* is derived. The spectral formulation is representative for wind-generated waves under the assumption of finite water depth and limited fetch.

Sea state code	Sea state description	Wave height $H_s(m)$	PFW ω_p (rad/s)	Percentage probability
0	Calm (glassy)	0	1.29	
1	Calm (ripples)	0 - 0.1	1.29 - 1.11	6.0616
2	Smooth (wavelets)	0.1 - 0.5	1.11 - 0.93	
3	Slight	0.5 - 1.25	0.93 - 0.79	21.5683
4	Moderate	1.25 - 2.5	0.79 - 0.68	40.9915
5	Rough	2.5 -4.0	0.68 - 0.60	21.2383
6	Very Rough	4.0 - 6.0	0.6 - 0.53	7.0101
7	High	6.0 - 9.0	0.53 - 0.46	2.6931
8	Very high	9.0 - 14.0	0.46 - 0.39	0.4346
9	Phenomenal (Extreme)	> 14.0	< 0.39	0.0035

Table 3.1: Sea state code definitions. Column 4 (peak frequency wave (PFW)) and 5 (percentage probability) are valid for the Northern North Sea. Notice that the percentage probability for sea state codes 0, 1 and 2 is summarized. Adapted from [16, 34].

3.2 Performance Indices

In making a definition of what constitutes good performance for a controller, some considerations about what the performance of the controller are a function of must be undertaken, e.g., the controller performance is a function of:

- **Ability to follow a reference value**.
  In regulation against a static setpoint a good performance could be to minimize the integral of the absolute (or squared) error, on the other hand for a setpoint change a good performance could be to minimize the settling time, and at the same time minimize or be inside a limit for the overshoot. Because the integral of the absolute (or squared) error might be low for a short settling time with large overshoot, this might not be the best way of measuring the performance. A performance defined on the bases of the settling time and the amount of overshoot might be a better choice.

- **Response to disturbance**.
  Often called disturbance rejection (means against changes in external loads, which for vessels will mainly be ocean currents, winds, and waves). In this case the definition of a good performance might be seen as the same as with a step change in the setpoint. The preferable is a fast response back to the setpoint, with minimum (or no) overshoot.

- **Efficiency**.
  For a DP control system, efficiency performance means that it shall keep/follow the reference value, and at the same time it shall do this with minimum use of forces, and low rate of change in the forces. The amount of force should be minimized to reduce the energy use, and the rate of change in force should be minimized to reduce wear and tear on the actuators.

- **Ability to be inside actuator limitations**.
  For most regulation systems there will also be saturation limits in the amount of the forces, and a limit in the rate of change in the forces. Breaking these limits can in worst case make the system unstable.

To summarize, the definition of a performance index for a controller can be seen as a function of the controllers' abilities to follow the reference value, its response to load disturbance, its efficiency in the use of forces, and its ability to be inside the force limitations. Another reflection is that a considerable use of forces or a large rate of change in the force might not give a better set-point regulation or trajectory following. This leads to that it can be preferred to have a small deviation in the position, compared to the amount of force use or the rate of change in the force. This can especially be the case for DP operations in higher sea states, where it might be preferable to relax the accuracy in positioning against the use of forces.

3.2.1 Traditional Performance Indices

Several definitions of performance indices can be found and a good survey is [23], which provides very good collection of recent developments in the control performance assessment area. A survey of the different performance indices that are mentioned in [23] will not be given here, since the paper handle performance indices specialized for different kinds of problem (e.g., valve stiction, time delay detection, nonlinearity detection, oscillation detection). For this study, the position/heading error and the use of forces/moment are the topics (for writing/reading simplicity will position/heading further be named pose, and forces/moment named forces).

According to [23], the most widespread controller performance assessment criterion is the variance of the output (or, equivalently, the standard deviation), which is based on minimum-variance control theory (MVC). A typical minimum variance (MV) index is

$$J_{MV} = \frac{\sigma^2_{MV}}{\sigma^2_y}, \tag{3.15}$$

where σ^2_{MV} is the obtainable/desirable MV and σ^2_y is the output variance. To calculate this index, estimation of the impulse response from the noise-to-output transfer function must be done, which is typically done recursively online by using an ARMA or ARMAX model. This means that minimum variance control requires significant process information and is typically based on a parameter identification techniques, which is not the issue for this study.

In [39], which gives an overview of MIMO control performance monitoring, it is concluded that for MIMO system: "covariance based monitoring is more appropriate when strong interactions occur among control variables". In this reference it is also proposed to have a user defined covariance benchmark, instead of a theoretical calculated benchmark. Where the user defined benchmark can be a covariance output measurement taken from an exemplary operation.

Another traditional performance index is to minimize the integral of absolute error (IAE)

$$J_{IAE} = \int_0^\infty |e(t)|dt, \tag{3.16}$$

which is putting equal weight on all errors during the simulation time. For constant setpoint regulation this might be a preferable performance index, but with a setpoint change it have a drawback in giving large values at the start of a setpoint change. This is mainly a problem if a step is given as the change in the setpoint and can be reduced by using a reference model.

This performance index is typically also defined as the integral of squared error (ISE)

$$J_{ISE} = \int_0^\infty e(t)^2 dt, \tag{3.17}$$

which is putting more weight on large errors, which might be convenient, but it will at the same time increase the drawback in giving large punishment for errors at the start of a setpoint change.

For a setpoint change, the integral of time weighted absolute error (ITAE)

$$J_{ITAE} = \int_0^\infty t|e(t)|dt \tag{3.18}$$

performance index is a better choice, which will put more weight at the end of the sample. By putting more weight at the end of the sample it will favour short settling time and relax large errors at the start of a setpoint change.

The time weighted error can also be defined as integral of time weighted squared error (ITSE)

$$J_{ITSE} = \int_0^\infty t^2 e(t)^2 dt, \tag{3.19}$$

which is combing punching large errors and long settling times, and relaxing errors in the start of a setpoint change. It must be noted that a time-weighted performance index must be reset for each setpoint change, and for trajectory tracking with constant setpoint-changes, a time weighted performance index might not be the best choice.

A performance index that is very universal is proposed in [11]. In this index, the performance calculation is divided into 18 subindices that handle different characteristics (i.e., steady state value, settling time, IAE, ISE, ITAE, ITSE, summations of system responses peaks, cumulative area over and under steady state value, etc.) and then collect them into a total index, with weighting between the subindices. The drawback with this performance index is that it involves many parameters to tune and it handles disturbances only indirectly. Another drawback, which is common for all the performance indices mentioned so far, is that none of them focus on minimizing the use of forces. A performance index that would do that is

$$J = \min_{e,u} \int_0^\infty e(t)^2 + \lambda u(t)^2 dt, \tag{3.20}$$

which is a version of the performance index for the LQR, and where λ is weighting between error in the setpoint and use of forces. For a mass-damper system, the weighting factor λ can be linked to the proportional gain as $K_p = \frac{1}{\sqrt{\lambda}}$ in a PD-controller, see [15, 16]. So the value of λ will directly involve the proportional gain in the controller. To also put focus on the rate of change in the forces, the performance index must be extended to

$$J = \min_{e,u,du} \int_0^\infty e(t)^2 + \lambda_1 u(t)^2 + \lambda_2 du(t)^2 dt, \tag{3.21}$$

where du denotes the rate of change in forces, which gives one more parameter to adjust. Another solution might be to set a constraint on the rate of change, but such a constraint would also make the performance index nonlinear.

3.2.2 A *Novel* Performance Index for Station Keeping

Different types of performance indices were mentioned in Section 3.2.1. Few of these indices focus on the use of forces. The focus is setpoint deviation or rejection against disturbances, and they are all very general. Another "drawback" is that they do not give a clear picture of how good the performance actually is for an operator of the system, other than lower (or higher) value is better performance.

Based on these reflections, a novel performance index for station keeping is proposed. The goal and the motivation behind the performance index is that it must focus on the performance balance between setpoint deviation and use of forces. Besides being useful in an autotuning algorithm, the performance index should also give a logical and understandable picture of the performance for a DP-operator. Furthermore, potential parameters in the index should be simple and logical to set, or optimally be set automatically based on available measurements.

Performance Index for Autotuning

For autotuning, it will be preferable that the autotuning function is a continuous function that can be analysed back to the model and the manipulated variables. Furthermore, it will be preferable that performance indices give a function with a diminishing return.

> The law of diminishing returns refers to a situation in which a smaller result is achieved for an increasing amount of effort.
>
> *Cambridge Advanced Learner's Dictionary,*
> *Cambridge Dictionaries Online.*

For a control system, the diminishing return will be a function of different controller parameters settings, e.g., bandwidth for a PID controller, different settings of K_p, K_i and K_d for a PID controller, or different gains in a LQR controller. An illustrative example of the performance index value as a function of the bandwidth for a PID controller with a diminishing return is given in Figure 3.2.

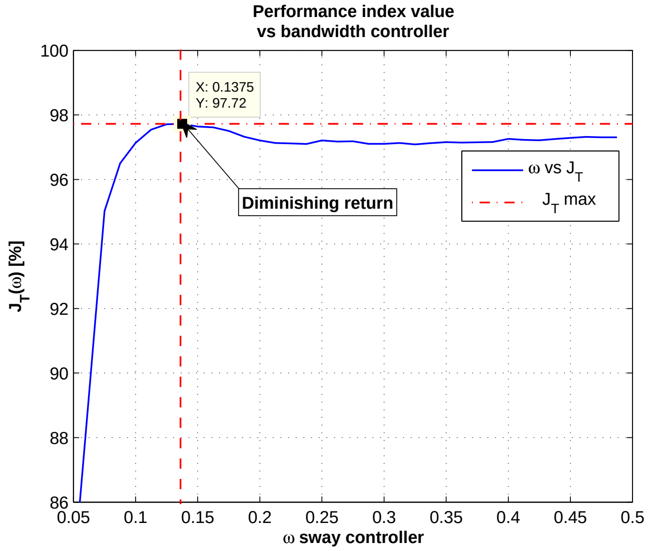


Figure 3.2: Illustration of the performance index value as a function of the bandwidth for a PID-controller with a diminishing return.

Performance Index for a DP Operator

A performance index for a DP operator is seen as a decision support tool, where valuable performance information include:

- How large has the pose deviation been the last x minutes.
- Is it the average pose deviation that is bad or is it peaks and how large are the peaks.
- Is it possible to achieve better performances in pose (implicitly increase controller gains) or are the actuators already performing their maximum.
- Is it possible to reduce the use of forces (implicitly reduce controller gains) and still be inside pose limits.
- If the controller gains were changed (manually or by autotuning), what did actually change in the performances and how much did it change (i.e., was there a reduction in use of forces and how large was it, did the pose deviation decrease and how much did it decrease).
- What DOF has the best and worst performances, according to use of forces and keeping of pose.

Based on this listing, the performance index should illustrate how large percentage share of the last x minutes, the pose has been outside some limits. Furthermore, it should visualize the balance between pose and force performances, as a total balance or a balance for each DOF. A log of the change of the performances as a function of the controller gains could also be informative, see Figure 3.3 for an illustrative example.

With such information, the DP operator would have a better understanding of how well the DP system performs. It would help the DP operator in tuning controller gains alternatively perform an autotuning. Furthermore, it can help him reduce the use of forces, if it possible to slacken the limits in the DOF that has the worst performance.

Based on the previous argumentation, the new performance index is defined as a total index (J_T) as a function of how well the vessel keeps the setpoint in position and heading inside some predefined limits as well as the use of forces. J_T will by this definition be a function of position-error (J_p), heading-error (J_h) and use of forces (J_τ). Each sub-performance index J_p, J_h and J_τ, is defined in the following subsections and finally the total index (J_T) is defined.

Position Deviation Levels

For a vessel performing station-keeping operations in a seaway, some setpoint deviation in position will always occur because of disturbance from, e.g.,

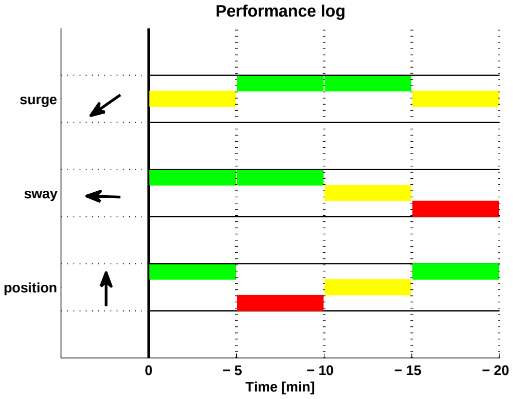


Figure 3.3: A log illustration of performance index values for surge forces, sway forces and position deviation as a function of different controller gains. A time window of 5 minutes was used to detect the performances for each controller tuning. In the figure, red is bad, yellow ok and green good performances. The arrows indicates the evolution of the performance values.

waves, wind and sea current. The size of this position deviation will depend on the vessel operational condition (VOC), see Section 3.1 for details about VOC.

In a seaway, the deviation will typical be larger in surge than in sway, since the natural damping and the added mass are smaller in surge than in sway. Furthermore, wave-generated accelerations and velocities will often be larger in surge than in sway, because of the smaller damping and added mass, which is more difficult to counter-act for the controller. The conclusion is that the size of the deviation will be related to a BODY-fixed reference frame and with typically larger deviation in surge than sway.

From this conclusion, the position performance index is defined as ellipses in a BODY-fixed reference frame. Three levels are defined, where the inner level represents the minimum possible deviation for the VOC, the mid level represents the preferable operation area and the outer level represents the maximum operation area. This can also be seen as green, yellow and red operation sectors, see Figure 3.4 for an illustrative example.

Besides deviations due to disturbances, there will also be an error in the position because of sensor inaccuracy. This error is dependent on the type of position sensors used and environmental factors. Different type of sensors for DP systems is found in [3], where also some accuracy and weighting between sensors are mentioned. The size of the sensor error will not be discussed here, it will only be assumed that such an error exist, that it is a NED fixed deviation and that it must be considered.

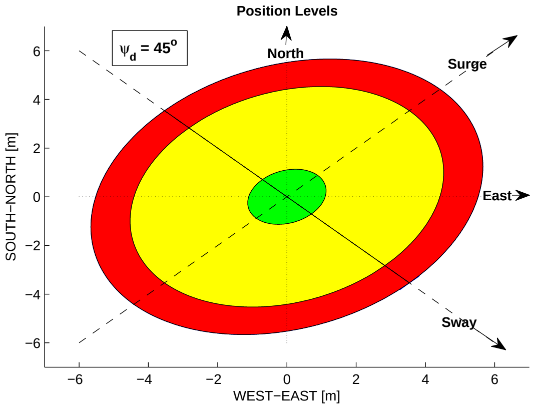


Figure 3.4: Illustration of the position levels, with three operation levels (green, yellow and red).

From these reflections, the definition and the setting of the three levels are summarized into:

Level 1 (inner level) is defined as maximum sensor error plus an offset. The offset is a function of the VOC and is defined as the minimum obtainable deviation for the VOC. Furthermore, it is separated in an offset for surge and an offset for sway, which gives the level 1 settings:

$$\begin{aligned} \text{surge L1} &= \max(\text{north-south sensor error, east-west sensor error}) \\ &+ \text{ offset surge} \\ \text{sway L1} &= \max(\text{north-south sensor error, east-west sensor error}) \\ &+ \text{offset sway,} \end{aligned}$$

where "max(north-south sensor error, east-west sensor error)" is the maximum error of measurement in NED frame position. See Figure 3.5 for an illustration of level 1 settings.

Level 2 (mid level) is defined as the preferable operating area. To simplify the level setting, this is set as a factor γ of level 3 (outer level), and gives the level 2 settings:

$$\begin{aligned} \text{surge L2} &= \text{surge L1 } + \gamma(\text{surge L3 - surge L1}) \\ \text{sway L2} &= \text{sway L1 } + \gamma(\text{sway L3 - sway L1}), \end{aligned}$$

where $\gamma \in < 0, 1 >$.

Level 3 (outer level) is defined as the maximum acceptable operation area, i.e., maximum deviation in surge and sway. In practice, this level is

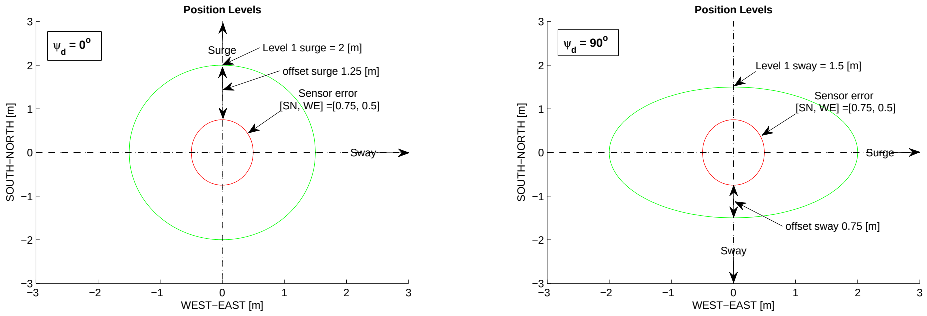


Figure 3.5: Illustration of level 1 settings. *Left image:* Setting of level 1 surge. *Right image:* Setting of level 1 sway. Notice that heading reference is changed to $\psi_d = 90$, while the sensor error is NED fixed and not changed.

set by the DP operator inside some limits (limits because it gives no meaning to set outer level less than inner level, and it is logical to have a maximum distance/factor between inner and outer level). Since this is seen as a maximum operation area from a DP operator view point, a potential sensor error must be subtracted from this setting, and gives the level 3 settings:

$$\begin{aligned}\text{surge L3 } &= \text{max operation area in surge}-\\ &\quad\text{max(north-south sensor error, east-west sensor error)}\\ \text{sway L3 } &= \text{max operation area in sway}-\\ &\quad\text{max(north-south sensor error, east-west sensor error).}\end{aligned}$$

The three levels with an ellipse form and the sensor error ellipse are illustrated in Figure 3.6.

Heading Deviation Levels

The performance index for heading error J_h is quite equal to the performance index for position error. It has the same three-level definitions, i.e.:

$$\begin{aligned}\text{yaw L1} &= \text{maximum sensor error } \psi + \text{ offset yaw}\\ \text{yaw L2} &= \text{yaw L1 } + \gamma(\text{yaw L3 - yaw L1})\\ \text{yaw L3} &= \text{max heading deviation } - \text{maximum sensor error } \psi.\end{aligned}$$

The levels can also here (as for the position) be seen as green, yellow and red operation sectors, see Figure 3.7.

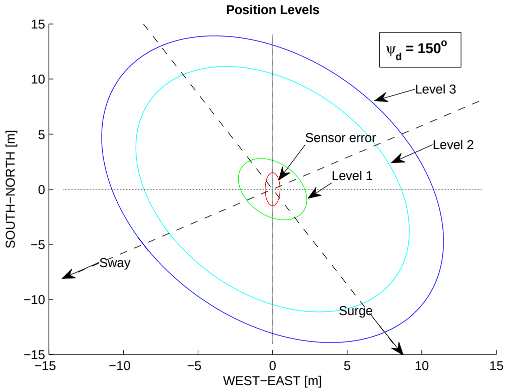


Figure 3.6: Illustration of all three levels with the sensor error. In the illustration, the distance between level 2 and level 3 is set equal to the maximum position sensor error.

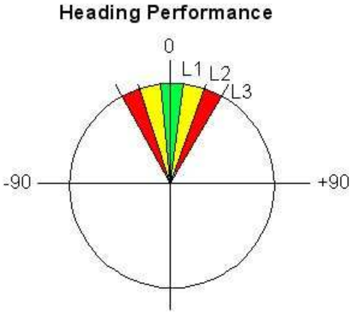


Figure 3.7: Illustration of levels for heading performance index J_h, with green, yellow and red operation sectors.

Calculating Performance Index values (J_p *and* J_h)

According to Table 3.1, the sea state codes with highest probability to occur are 3 to 5 (slight to rough). Furthermore, it is seen as abnormal to perform station-keeping operations in a rougher sea state code than 6 (very rough). A lower sea state code have higher wave frequency and in Table 3.1, the lowest wave frequency for sea state code 6 is $\omega_p = 0.46$ [rad/s]. A wave frequency of $\omega_p = 0.46$ [rad/s] gives a peak period of $T_p = \frac{2\pi}{0.46} \approx 11.85$ [s], which again gives approximately 5 wave peaks per minute. By assuming that the highest wave peak in a sea state condition would occur in a range of 20 wave peaks, it can be assumed that the average disturbance and the peak disturbance can be detected in a time window of 4 minutes. *Please note that it is not claimed that the highest wave peak would occur in a range of twenty peaks,* *it is only assumed.*

If the average disturbance and the peak disturbance are detected in a time window of 4 minutes, it is also logical to look at the performance for a time window of 4 minutes. Based on this assumption, a sub position performance index for each level is calculated as:

1. A counting variable pl_i for each level $i \in \{1, 2, 3\}$ is initialized with $pl_i(0) = T/h$, where T is the time window, e.g., 4 minutes, and h is the sampling time.

2. For each time sample k is:

   (a) The position deviation measured in NED ($P_e^n = [n_e, e_e, \psi_e]^T$) and rotated to BODY ($P_e^b = [x_e, y_e, \psi_e]^T$), i.e.,

   $$P_e^b = \mathbf{R}(\psi)^T P_e^n, \tag{3.22}$$

   since the position deviation levels are defined in BODY.

   (b) A position deviation radius r_{pe} from the setpoint with an angle β is calculated as:

   $$r_{pe} = \sqrt{x_e^2 + y_e^2} \tag{3.23}$$

   $$\beta = \arctan \frac{y_e}{x_e} \tag{3.24}$$

   (Notice that for $x_e = 0$, arctan gives no solution and β is set to $\frac{pi}{2}$. Furthermore, it does not matter what the sign of the β angle is (or, equivalently, the x_e and y_e value), since the deviation levels is defined as an ellipse and has the same radius in all quadrants.)

   (c) A level radius r_{l_i} for the angle β is calculated for the three deviation levels as:

   $$r_{l_i}(\beta) = \frac{ab}{\sqrt{b^2 \cos\beta + a^2 \sin\beta}} \tag{3.25}$$

   for $i \in \{1, 2, 3\}$, where a = surge L_i and b = sway L_i.

   (d) The position deviation radius r_{pe}, is compared with the level radius $r_{l_i}(\beta)$ and:

   $$pl_i(k+1) = \begin{cases} pl_i\mathit{(k)} & \text{if } r_{pe} \leq r_{l_i}(\beta) \\ pl_i\mathit{(k)}{-}1 & \text{if } r_{pe} > r_{l_i}(\beta), \end{cases} \tag{3.26}$$

   for $i \in \{1, 2, 3\}$.

3. The total sub performance index value is calculated as:

   $$J_{pl_i} = 100 \cdot \frac{pl_i \cdot h}{T}, \quad J_{pl_i} \in [0, 100], \tag{3.27}$$

   for $i \in \{1, 2, 3\}$.

A calculation example of the values r_{pe}, β, r_{li} and p_{li} for $i \in \{1,2,3\}$ at time step k is given below.

Example 3.1
Position deviation is $P_e^n = [-10, 10, 0]^T$ *and heading demand is* $\psi_d = 150°$

(a) $P_e^b = \mathbf{R}(150°)^T[-10, 10, 0]^T = [13.66, -3.66, 0]^T$

(b) $r_{pe} = \sqrt{(13.66)^2 + (-3.66)^2} = 14.14$
$\beta = \arctan \frac{-3.66}{13.66} = -0.26 \Rightarrow -15°$

(c) $r_{l1}(-0.26) = 2.8$
$r_{l2}(-0.26) = 11.5$
$r_{l3}(-0.26) = 14.4$

(d) $p_{l1}(k{+}1) = p_{l1}(k)$ - 1, *since* $r_{l1}(\beta) < r_{pe}$
$p_{l2}(k{+}1) = p_{l2}(k)$ - 1, *since* $r_{l2}(\beta) < r_{pe}$
$p_{l3}(k{+}1) = p_{l3}(k)$, *since* $r_{l3}(\beta) > r_{pe}$

An illustration of the values are given in Figure 3.8.

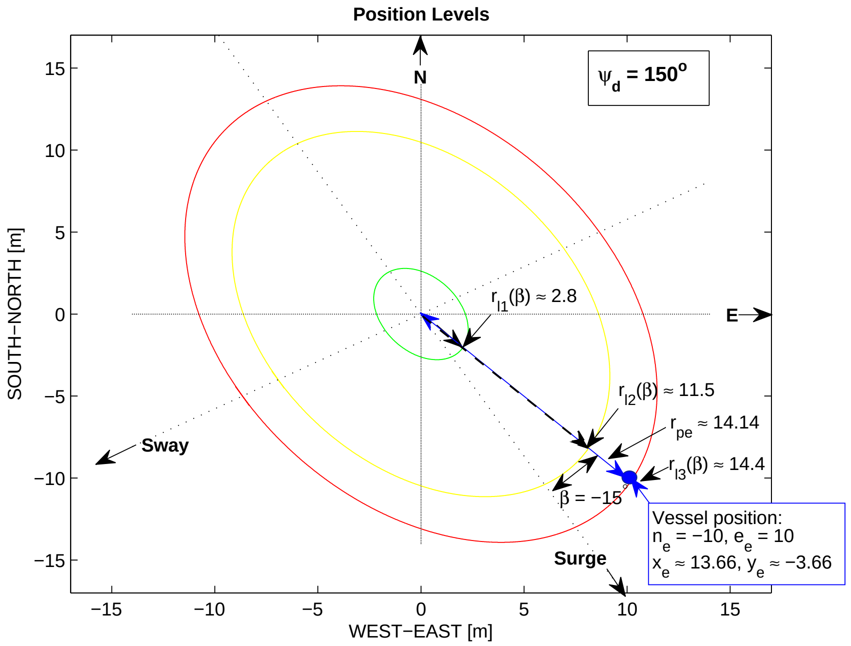


Figure 3.8: Illustration of the values r_{pe}, β and r_{li} for $i \in \{1,2,3\}$ at time step k where the vessel position deviation is $P_e^n = [-10, 10, 0]^T \Rightarrow P_e^b \approx [13.66, -3.66, 0]^T$.

For heading, the NED - BODY considerations can be omitted and the sub-heading performance indices are calculated as:

1. A counting variable hl_i for each level $i \in \{1,2,3\}$ is initialized with $hl_i(0) = T/h$, where T is the time window e.g., 4 minutes, and h is the sampling time.

2. For each time sample, the heading deviation angle ψ_e is compared with the heading deviation levels ψ_{l_i}:

$$hl_i(k+1) = \begin{cases} hl_i(k) & \text{if } |\psi_e| \leq \psi_{l_i}(\beta) \\ hl_i(k) - 1 & \text{if } |\psi_e| > \psi_{l_i}(\beta) \end{cases} \tag{3.28}$$

   for $i \in \{1,2,3\}$.

3. The total sub performance index value is calculated as:

$$J_{hl_i} = 100 \cdot \frac{hl_i \cdot h}{T}, \quad J_{hl_i} \in [0, 100] \tag{3.29}$$

   for $i \in \{1,2,3\}$.

With this definition, each subperformance index is a measurement of how large percentage share of a time window the vessel is outside a deviation level. The total performance index for position and heading is calculated by weighting and summarizing the sub performance index values as:

$$J_p = w_{p_1} J_{pl1} + w_{p_2} J_{pl2} + w_{p_3} J_{pl3}, \text{ where } \sum_{i=1}^{3} w_{p_i} = 1 \text{ and } J_p \in [0, 100], \tag{3.30}$$

$$J_h = w_{h_1} J_{hl1} + w_{h_2} J_{hl2} + w_{h_3} J_{hl3}, \text{ where } \sum_{i=1}^{3} w_{h_i} = 1 \text{ and } J_h \in [0, 100] \tag{3.31}$$

With the weighting factors w_{p_i} and w_{h_i}, a deviation outside the outer level can be punished more than a deviation outside the inner level.

The performance indices J_p and J_h are thus an average of the performance over a time window, where deviations outside some predefined limits is punished and the importance of each deviation level is adjusted with weights. Furthermore, large deviation that last for some considerable time is punished more than short peaks. An example of some resulting performance values is illustrated in Figure 3.9.

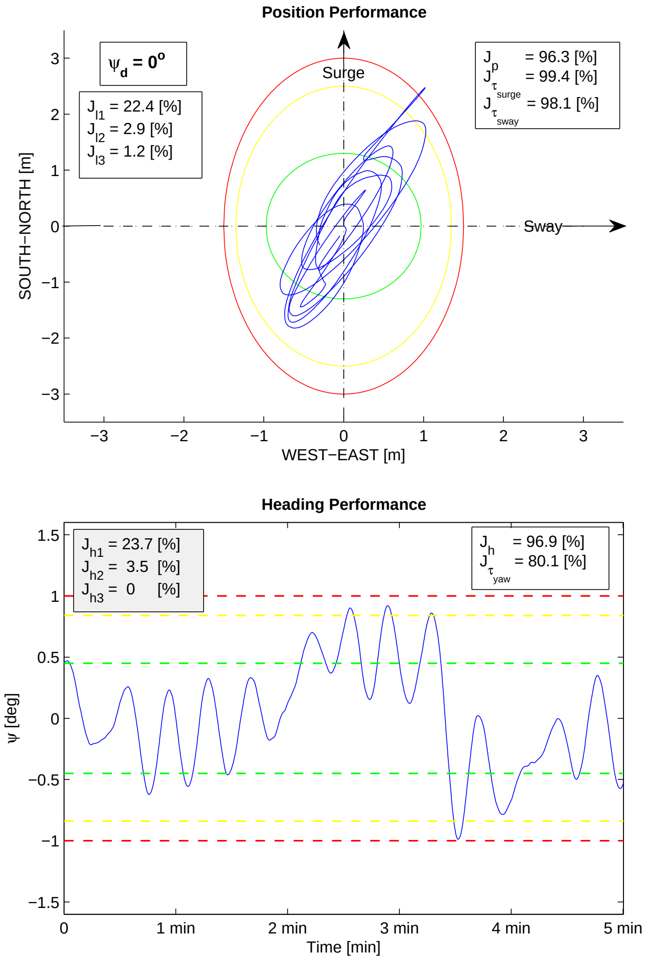


Figure 3.9: Illustration of subperformance index values for a time window of five minutes with the level weights set to $wp = wh = [0.1, 0.2, 0.7]$. *Upper image:* Position performance. *Lower image:* Heading performance.

Performance Index for Dynamical Variation in Force/Moment (J_τ)

The first goal for a controller is of course to keep the setpoint, but a second goal is to do this with a minimum use of forces and dynamical variation in forces. By minimizing the amount of forces and the dynamical variation in forces, energy is saved, wear and tear on actuators is reduced and the comfort for the crew is increased (less noise, vibrations and accelerations in the vessel).

A vessel is often under influence of static force disturbances from sea current, static wave forces, wind, etc. Some of these disturbances are counteracted by feedforward terms (e.g., wind), but other has to be counteracted by a static force from the feedback controller. This static force can be considered as a "necessary" output from the feedback controller and should not be punished in a performance index. On the other hand, dynamical variation around this static force should be minimized and then punished in a performance index. With these reflections, an improvement in the use of forces and dynamical variation in forces are:

- Shorter total movement length for actuators, i.e., the sum of setpoint changes for actuators.
- Lower velocity in actuators and lower frequency in change of actuator positions,
- Smaller maximum peaks in actuator positions.

Larger peaks in the actuator position will implicitly give a longer movement distance for the actuator, which involves larger changes between two samples for the controller output. Larger peaks will also often involve higher frequency in changing of actuator positions, which again involves more frequently changes for the controller output. A higher velocity in actuators will also require larger changes between two samples for the controller output.

The improvement points that were stated are then reduced down to the change in the controller output between two samples and the sum of these changes over a time window. This means that a reduction of the sum

$$\Gamma = \sum_{k=1}^{T/h} |\tau_k - \tau_{k-1}|, \quad T = \text{time window and } h = \text{ sample time,} \tag{3.32}$$

will give an improvement in the force performance. Since a reduction of Γ is a result of less (frequency in the) change of controller output and smaller changes between two samples in the controller output. At the same time, a static output from the controller for counteracting a static disturbance will not give growth in Γ.

For a DP control system, there will always be local actuator controllers. These local controllers will set the limits for maximum obtainable change in actuator force (position) between two samples, and these limits are normally known values. By using this information, a simple performance index for dynamical variation in forces can be defined as

$$J_{\tau_i} = 100 \cdot \exp^{-\frac{\Gamma_i}{\Gamma_{max_i}}} \tag{3.33}$$

$$\Gamma_{max_i} = \frac{T}{h} \Delta\tau_{max_i} \tag{3.34}$$

$$\Gamma_i = \sum_{k=1}^{T/h} |\tau_i(k) - \tau_i(k-1)|, \tag{3.35}$$

where $\Delta\tau_{max_i}$ is the limit set in the local actuator controller and $i \in \{surge, sway, yaw\}$. The performance index can also be extended with a minimum and a maximum offset. A minimum offset is implemented as

$$\Delta\tau_{k_i} = \begin{cases} 0 & \text{if } \Delta\tau_{k_i} \leq \kappa_{min_i} \cdot \Delta\tau_{max_i} \\ \Delta\tau_{k_i} & \text{if } \Delta\tau_{k_i} > \kappa_{min_i} \cdot \Delta\tau_{max_i}, \end{cases} \tag{3.36}$$

where $\Delta\tau_{k_i} = |\tau_i(k) - \tau_i(k-1)|$ and $\kappa_{min_i} \in [0, \kappa_{max_i}\rangle$. While a maximum offset is implemented as

$$\Gamma_{max_i} = \frac{T}{h} \Delta\tau_{max_i}(\kappa_{max_i} - \kappa_{min_i}). \tag{3.37}$$

where $\kappa_{max_i} \in \langle\kappa_{min_i}, 1]$ is used as a factor to reduce $\Delta\tau_{max_i}$. The performance index J_τ is illustrated in Figure 3.10, with minimum and maximum offset.

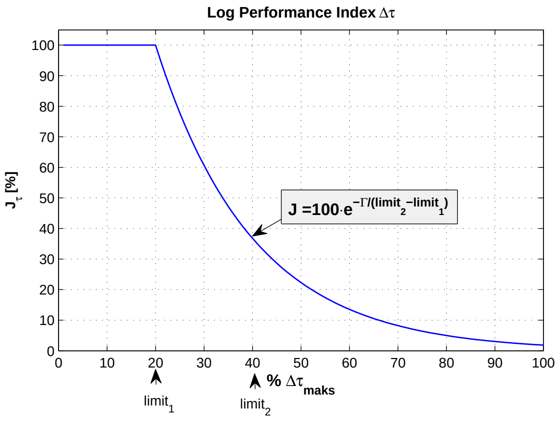


Figure 3.10: Illustration of force performance index J_τ, where minimum offset (limit 1 in figure) is set 20% (i.e., $\kappa_{min} = 0.2$) and maximum offset (limit 2 in figure) is set to 40% (i.e., $\kappa_{max} = 0.4$).

Calculation of J_T

Each sub performance index J_p, J_h and J_τ, have been defined in a range from zero to one hundred percent ($J_j \in [0, 100]$, $j \in \{$p, h, $\tau\}$), where one hundred percent is seen as good performance and zero as bad performance. To separate out the controller output for each DOF, it is natural to define a performance index for each DOF as:

$$J_{T_{surge}} = \lambda_1(J_p - J_{\tau_{surge}}) + J_{\tau_{surge}}, \text{ where } \lambda_1 \in \langle 0, 1] \tag{3.38}$$

$$J_{T_{sway}} = \lambda_2(J_p - J_{\tau_{sway}}) + J_{\tau_{sway}}, \text{ where } \lambda_2 \in \langle 0, 1] \tag{3.39}$$

$$J_{T_{yaw}} = \lambda_3(J_h - J_{\tau_{yaw}}) + J_{\tau_{yaw}}, \text{ where } \lambda_2 \in \langle 0, 1]. \tag{3.40}$$

The optimal in this definition is an automatical specification of the weight as a function of the distance between the inner and the outer level (level 1 and level 3) and the ongoing VOC. Notice that with λ_1 to λ_3 set to 1, the performance index will not consider dynamical use of force in surge, sway and yaw.

With this definition of J_{T_i}, the best performance will be a balancing between $J_{p/h}$ and J_{τ_i}, for $i \in \{$surge, sway, yaw$\}$. Examples of different balances between $J_{p/h}$ and J_{τ_i} is illustrated in Figure 3.11. In the figure, only the balance (difference) between the force performance J_τ and the pose performance $J_{p/h}$ is considered, not the performance value itself. The figure is seen as a lever, where the scale goes from -100% (middle x-axis lower) to + 100% (middle x-axis upper) for J_τ, where the field is marked with green if the balance is positive for the performances, yellow if the balance is between 0% and -20%, and red if it is less than -20%, i.e., if $J_\tau > J_{p/h}$, the left field is green and the lever is pointing at a positive value $J_\tau - J_{p/h}$, while the right field is red or yellow since $J_{p/h} < J_\tau$. Please notice that the figure is a simplification of a similar figure with circle shape.

A total performance index for all DOFs will then be

$$J_T = (J_{T_{surge}} + J_{T_{sway}} + J_{T_{yaw}})/3, \tag{3.41}$$

or with weighting possibilities between the DOFs

$$J_T = \upsilon_1 J_{T_{surge}} + \upsilon_2 J_{T_{sway}} + \upsilon_3 J_{T_{yaw}}, \text{ where } \sum_{i=1}^{3} \upsilon_i = 1, \tag{3.42}$$

and where $J_{T_{surge}}$, $J_{T_{sway}}$, $J_{T_{yaw}}$ and $J_T \in [0, 100]$.

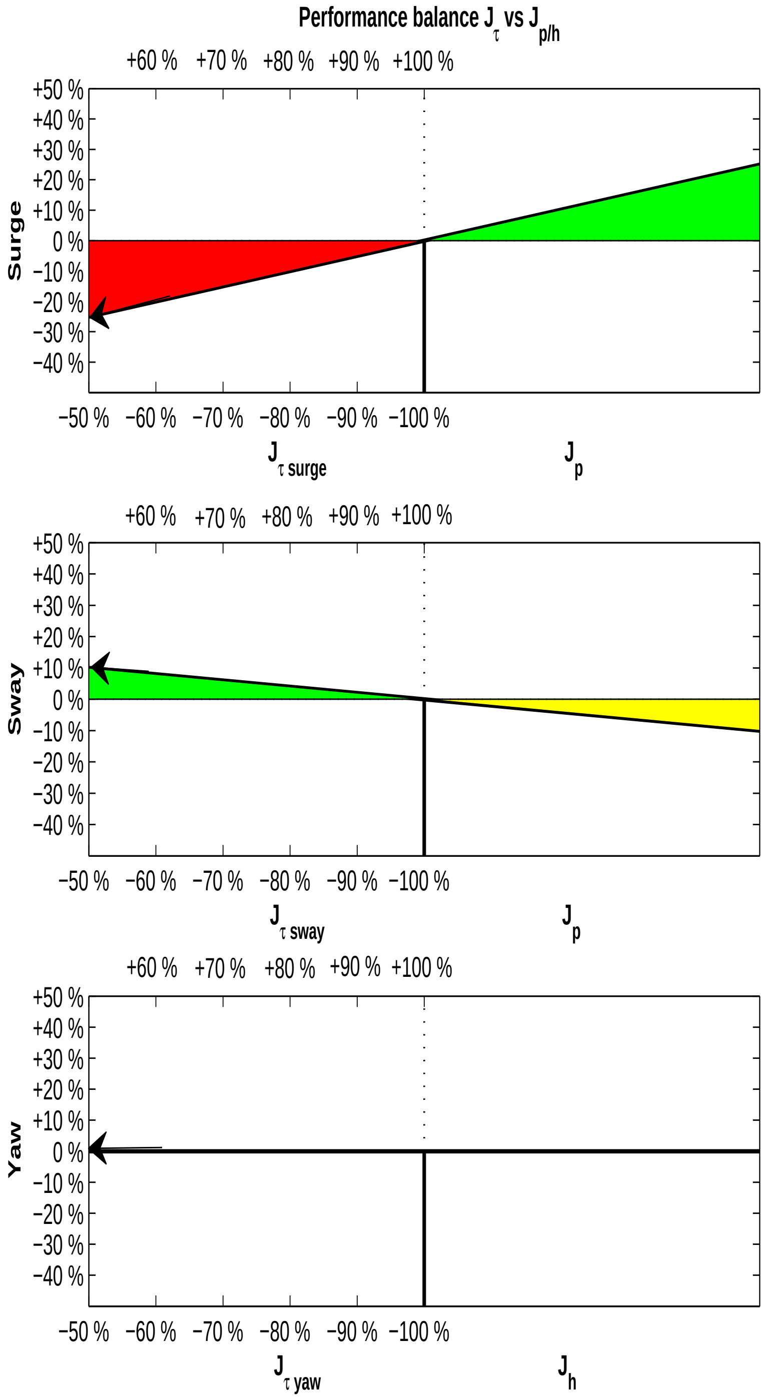


Figure 3.11: Example of performance balance between $J_{p/h}$ and J_{τ_i}, $i \in \{\text{surge, sway, yaw}\}$. *Upper image:* $J_{\tau_{surge}}$ has 25 % lower performance than J_p. *Mid image:* $J_{\tau_{sway}}$ has 10 % better performance than J_p. *Lower image:* $J_{\tau_{yaw}}$ has balanced performance with J_h.

3.2.3 Performance Weighting

In (3.38), (3.39) and (3.40) are the weighting factor λ_i ($i \in \{1, 2, 3\}$) a weighting between pose accuracy and use of forces, and should be set to achieve the performance that is considered as the best. Different vessel operations leads to different definitions about what is considered as good performance, which means that the choice of the weighting factor λ can depend on several factors. As an example, different sea state conditions can have different effects for the use of forces. Similarly, different operations can have different demands on the accuracy of the position. A close loading operation will typical have higher requirement in position accuracy than a waiting operation. Another example is that available power, saturation limits, etc., can change, which can lead to change in what is considered as good performance. The dependency in the choice of λ is illustrated in Figure 3.12.

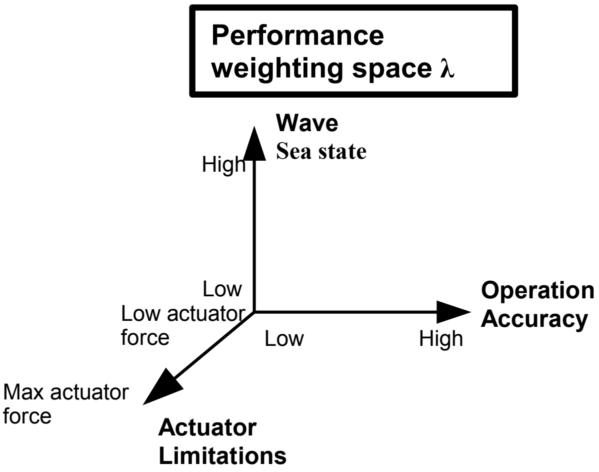


Figure 3.12: Performance space, dependency of weighting factor λ.

Chapter 4

Control Design

4.1 High-Level Controller Function

The idea behind the proposed controller function came as a combination of several factors; First, it was desired to combined the advantage of using already learned knowledge with the advantage of learning with a totally open mind. This can been seen as a combination of the gain scheduling controller, which only uses already learned knowledge, and an adaptive controller, which does not use any historical knowledge (except from design parameters). Another factor was the self-tuned memory(data)-based internal model control (IMC) PID-control [43, 51, 59], where they stored experienced parameters in a memory, and used the stored parameters as a local area (neighbor area) when selecting controller parameters. The memory did also help in decreasing the tuning time in a local area. Area and parameters were in these articles linked to the setpoint, which is convenient for process control. In the controller proposed here, they are in-stead linked against the VOC, which will have more influence on the controller performance than the setpoint in DP operations. The idea is that training can be performed when the vessel is idle, while the gain-scheduling function will supply the advantage of quickly selecting the optimal controller parameters for the actual VOC during other operations.

The controller function is illustrated in Figure 4.1, and the function is as follows:

Initialization Before the controller can be activated, it has to be initialized with some default controller parameters from the commissioning. These controller parameters will be used as an initial value in the training, and are also used in a default mode called normal. The default controller parameters will also be used as a competitor in the training.

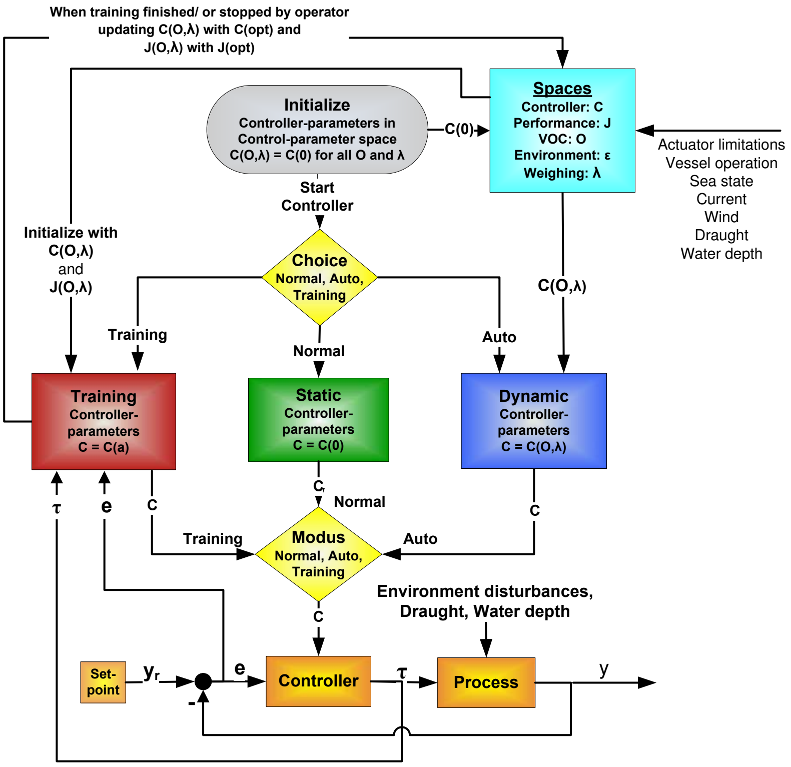


Figure 4.1: Block diagram illustration of the function for the proposed hybrid controller, with normal mode (green block), auto mode (blue block) and training mode (red blocks). Note that the normal mode controller and the controller space is initialized with default parameters (grey block).

Mode selection The controller has three different operation modes (marked with yellow in Figure 4.1); normal, auto and training, which can be manually activated by the operator. The function of each mode is:

- *In normal mode* (marked with green in Figure 4.1), the controller parameters are static at the value they were set to during the commissioning, this can also be seen as a default or safe mode.
- *In auto mode* (marked with dark blue in Figure 4.1), the controller parameters are changed dynamically according to the VOC.
- *In training mode* (marked with red in Figure 4.1), the controller parameters are trained according to the current VOC. A detailed illustration of the training mode is given in Figure 4.2. The goal for the training is to find the optimal controller parameters for the

VOC based on a performance index, see Section 3.2 for a definition of performance indices. The training can either be stopped by the operator, by a time limit, or if some predefined performance goal is achieved. A predefined performance goal can be linked up to historical data (e.g., "historical best"). Notice that with some autotuning methods, the process must be brought to some kind of excitation, which leads to necessity in special inputs (set-points) when in training mode. This means that the input must be fitted to the chosen autotuning method as well as the process.

Switching between the different controller modes are done manually and a filtering effect between the different controller parameters must be implemented. This to avoid peaks in the controller output when a switching is performed.

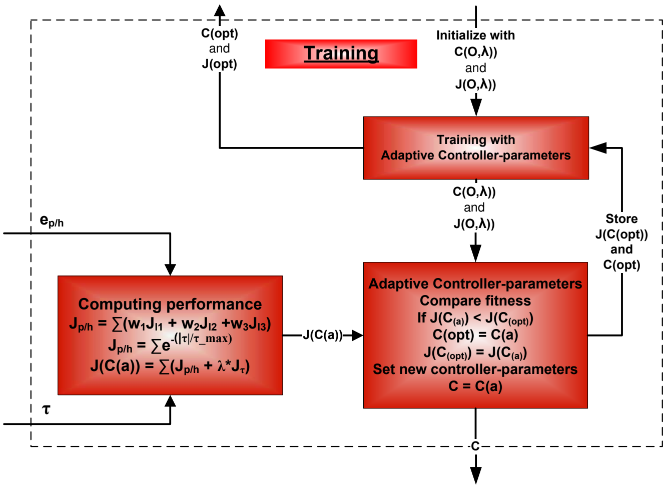

C(opt)
and
J(opt)
Training
Initialize with
C(O,λ))
and
J(O,λ))
Training with
Adaptive Controller-parameters
C(O,λ))
and
J(O,λ))
Store
J(C(opt))
and
C(opt)
$e_{p/h}$
Computing performance
$J_{p/h} = \sum(w_1 J_{l1} + w_2 J_{l2} + w_3 J_{l3})$
$J_{p/h} = \sum e^{-(|\tau|/\tau\_max)}$
$J(C(a)) = \sum(J_{p/h} + \lambda * J_\tau)$
J(C(a))
Adaptive Controller-parameters
Compare fitness
If $J(C_{(a)}) < J(C_{(opt)})$
C(opt) = C(a)
$J(C_{(opt)}) = J(C_{(a)})$
Set new controller-parameters
C = C(a)
τ
C

Figure 4.2: Detailed illustration of the Training block. The training mode is initialized with default parameters or earlier-found parameters and score for the VOC and the weight λ. The block returns the controller parameters that gives the best score for the performance index as well as the corresponding performance index value.

Spaces The different spaces (marked with light blue in Figure 4.1) are used to detect the environment condition (ε), the VOC condition (O) and the weighting factor (λ). A detailed illustration of these spaces is given in Figure 4.3. The block is also used as a data-base with controller parameters for different VOC and weighting factors. To have a comparison basis, earlier scores for different VOC and weights are stored as well. For a definition of each of the spaces, see Section 3.1, 3.2.3, and 4.4.

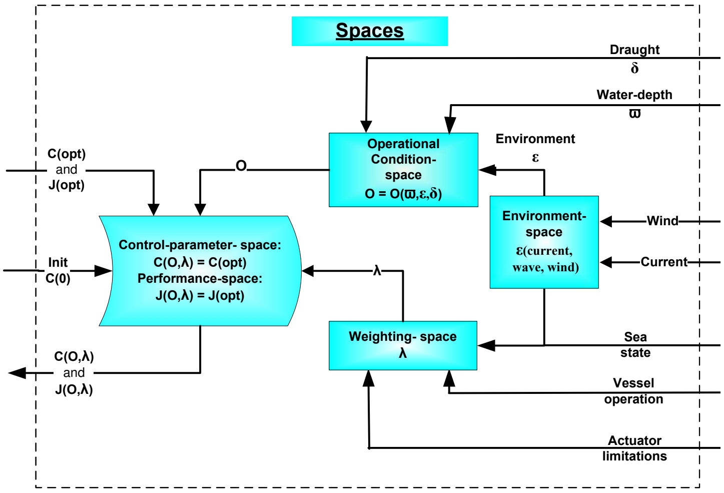


Figure 4.3: Detailed illustration of the Spaces block. The controller space is initialized with default parameter from the commissioning. The VOC and weighting factor λ is detected, and controller parameters and performance values are stored according to VOC and λ weights. The block sends out the controller parameters and performance index value from the database for the VOC and λ weight.

Notice that it is assumed that the controller has a fixed structure and the training is only for adapting the controller parameters.

4.2 Controller Structure

The considered controller is a nonlinear PID-controller based on (2.24), where the reference feedforward term is omitted, since only constant setpoint regulation is considered. Furthermore, the control law is extended with the back-calculation integrator anti-windup scheme mentioned in [55], which gives the control law

$$\boldsymbol{\tau}_{contr} = -\mathbf{M}\boldsymbol{\tau}_{PID} + \mathbf{D}\tilde{\boldsymbol{\nu}} \tag{4.1}$$

$$\boldsymbol{\tau}_{PID} = \mathbf{K}_i\left(\mathbf{R}^T(\psi)\int_0^t \tilde{\boldsymbol{\eta}}(\tau)d\tau + \int_0^t \mathbf{I}(\tau)d\tau\right) + \mathbf{K}_p\mathbf{R}^T(\psi)\tilde{\boldsymbol{\eta}} + \mathbf{K}_d\tilde{\boldsymbol{\nu}}, \tag{4.2}$$

where $\tilde{\boldsymbol{\eta}} \triangleq \boldsymbol{\eta} - \boldsymbol{\eta}_d$, $\tilde{\boldsymbol{\nu}} \triangleq \boldsymbol{\nu} - \boldsymbol{\nu}_d$ and $\mathbf{I}$ is the integrator anti-windup function

$$\mathbf{I}_{(i,i)} = \begin{cases} \frac{1}{T_{t_{(i,i)}}}\left(\tau_{act_i} - \tau_{contr_i}\right) & \text{if } \tau_{act_i} \neq \tau_{contr_i} \\ 0 & \text{otherwise,} \end{cases} \tag{4.3}$$

for $i \in \{1, 2, 3\}$. T_t is called tracking time constant and must be tuned to fit the process.

4.3 Training with Autotuning

In this thesis, the training is done by two different autotuning/optimization methods; a rule-based algorithm and a genetic algorithm (GA). Two different methods were chosen to compare, and especially to compare an intelligent method against a rule-based method. GA was chosen because it seems to be the most used method in intelligent optimal tuning, and it is easy to implement with the performance index that was proposed in Section 3.2.2.

Adaptive methods such as APPC and MRAC were not considered. APPC requires model parameter identification, which is not an issue for this study. MRAC is difficult to implement with the proposed performance index in Section 3.2.2.

4.3.1 Autotuning with GA

Autotuning with GA is based on the basic GA mentioned in Section 2.3.2, see Figure 2.8. The function of the GA is illustrated in Figure 4.4 and it is using tournament selection between two chromosomes, real value crossover, and mutation.

Each chromosome is representing a set of controller gains in the PID-controller (4.2). To simplify the selections of chromosome values, the control law is compared with the pole-placement algorithm in Section 2.3.1, which gives the controller gains

$$K_{p_{dof}} = \omega_{n_{dof}}^2 \tag{4.4}$$

$$K_{d_{dof}} = 2\zeta\omega_{n_{dof}} \tag{4.5}$$

$$K_{i_{dof}} = \frac{\omega_{n_{dof}}^3}{10}, \tag{4.6}$$

where the GA is optimizing the natural frequency ($\omega_{n_{dof}}$) for $dof \in \{1,2,3\}$. With the controller bandwidth of the system $y = h(s)u$ with negative unity feedback defined as the frequency ω_b at which the loop transfer function $l(s) = h(s) \cdot 1$ is $|l(j\omega)|_{\omega=\omega_b} = \frac{\sqrt{2}}{2}$, the natural frequency ω_n can be related to the system bandwidth ω_b as

$$\omega_b = \omega_n\sqrt{1 - 2\zeta^2 + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}. \tag{4.7}$$

From this relation, the controller bandwidth ω_b is a $\bar{\zeta}$ factor of the natural frequency ω_n, i.e.,

$$\omega_b = \bar{\zeta}\omega_n, \tag{4.8}$$

where $\bar{\zeta} = \sqrt{1 - 2\zeta^2 + \sqrt{4\zeta^4 - 4\zeta^2 + 2}}$. Note that with $\zeta = \frac{\sqrt{2}}{2}$ is $\bar{\zeta} = 1 \Rightarrow \omega_b = \omega_n$ and with $\zeta = 1$ is $\bar{\zeta} = 0.64 \Rightarrow \omega_b = 0.64\omega_n$, see [16] for more details. For writing/reading simplicity are (4.4) to (4.6) called controller bandwidth tuning later.

An alternative is to tune the $K_{d_{factor}}$ $(= \zeta)$, and the $K_{i_{factor}}$ in addition to the bandwidth (ω_n), which would give the controller gains

$$\begin{aligned} K_{p_{dof}} &= \omega_{n_{dof}}^2 && (4.9) \\ K_{d_{dof}} &= 2K_{d_{factor}}\omega_{n_{dof}} && (4.10) \\ K_{i_{dof}} &= \frac{\omega_{n_{dof}}^3}{K_{i_{factor}}}. && (4.11) \end{aligned}$$

for $dof \in \{1, 2, 3\}$.

The last alternative has possibilities for finding controller parameters that gives better performance than the first alternative. On the other hand, the last alternative is more difficult to ensure stability for all solutions, since the chromosome would be $[\omega_{n_{dof}}, K_{i_{factor}}, K_{d_{factor}}]$, where stability interactions occur between the $K_{i_{factor}}$ and the $K_{d_{factor}}$. This means that when bounds for the chromosome values are set, these interactions must also be considered with the upper/lower bound on the bandwidth $\omega_{n_{dof}}$, please consult [49] for a discussion about stability regions for the $K_{i_{factor}}$ and the $K_{d_{factor}}$. For the first alternative, the chromosome is $[\omega_{n_{dof}}]$ or alternatively $[\omega_{n_{surge}}, \omega_{n_{sway}}, \omega_{n_{yaw}}]$ if all three DOFs are tuned simultaneously, which only needs an upper and a lower bound for the chromosome values. Note that for the first alternative it is assumed that proper ζ (and potential $K_{i_{factor}}$) is found during the commissioning.

The mutation is carried out by selecting randomly between $\omega_{n_{surge}}$, $\omega_{n_{sway}}$ and $\omega_{n_{yaw}}$ or between ω_n, the $K_{d_{factor}}$ and the $K_{i_{factor}}$, and exchanging one of them with a randomly chosen new value, inside the bounds for the value. If the chromosome is a single value (i.e., $\omega_{n_{dof}}$) the mutation is exchanging the chromosome with a randomly chosen new one.

Calculation of the fitness value is carried out by using the new performance index proposed in Section 3.2.2. To secure that the GA does not give a solution with worse performance than the default controller set, it is initialized with one of the chromosomes equal to the default controller set. The fitness value for this controller set will then be a competitor for the rest of the chromosomes. The default controller set can be from the commissioning or from an earlier training for the VOC and the λ weight.

The algorithm proposed here will always keep the best chromosome to the next generation. Termination of the algorithm is carried out when the maximum number of generations is reached, if the performance index value is 100% or it is stopped by the operator.

Tuning parameters for this algorithm are population size, number of generations, crossover probability, mutation probability, and max-min bounds for the different chromosomes. The most important values among these are the max-min bounds that must be set with care, since wrong values can lead to instability. For the other tuning parameters, bad values will only lead to slow convergence or failure in finding the optimum.

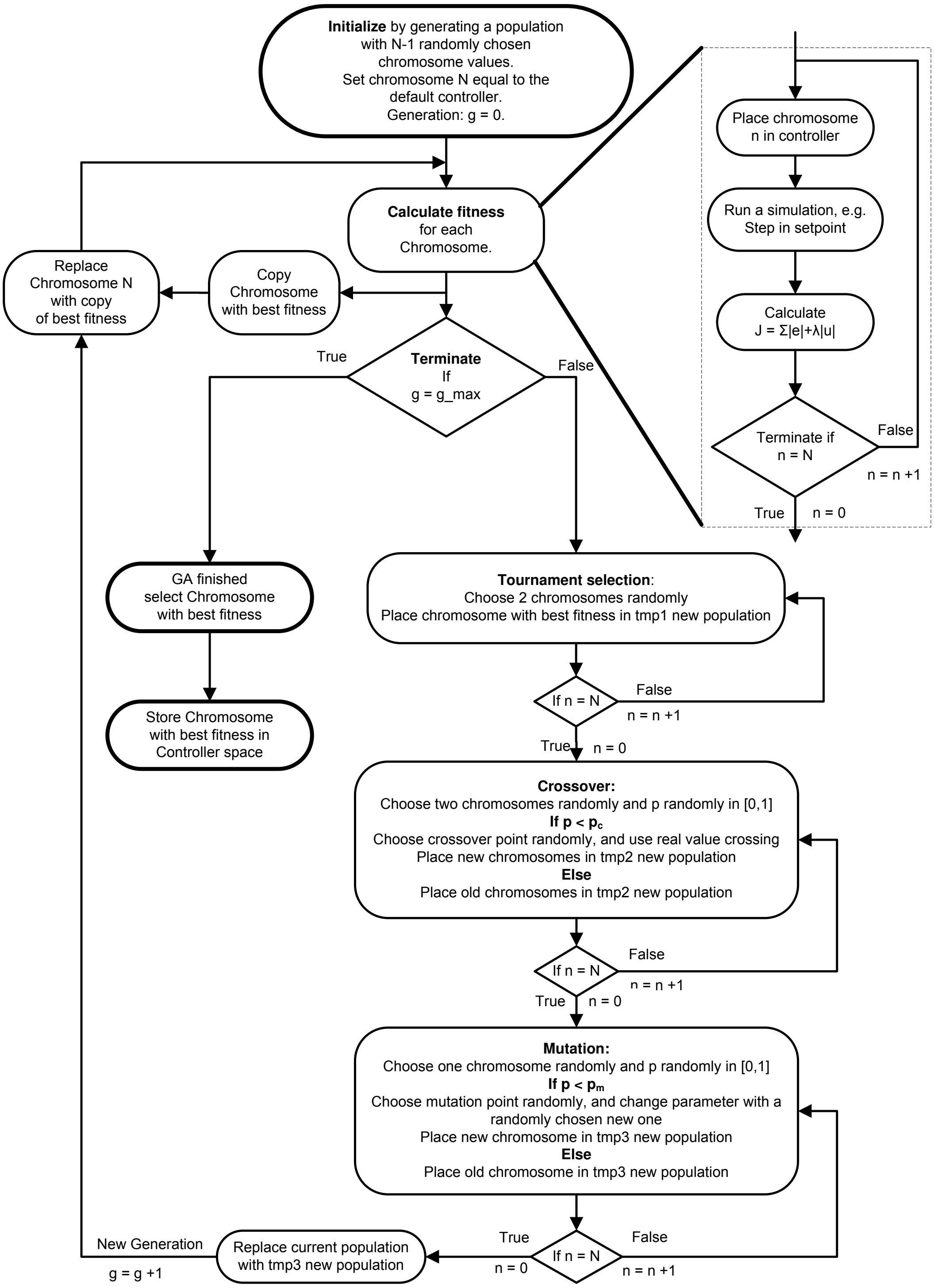

Initialize by generating a population with N-1 randomly chosen chromosome values. Set chromosome N equal to the default controller. Generation: g = 0.
Calculate fitness for each Chromosome.
Copy Chromosome with best fitness
Replace Chromosome N with copy of best fitness
Terminate If g = g_max
True
False
Place chromosome n in controller
Run a simulation, e.g. Step in setpoint
Calculate J = Σ|e|+λ|u|
Terminate if n = N
False
n = n +1
True
n = 0
GA finished select Chromosome with best fitness
Store Chromosome with best fitness in Controller space
Tournament selection: Choose 2 chromosomes randomly Place chromosome with best fitness in tmp1 new population
If n = N
False
n = n +1
True
n = 0
Crossover: Choose two chromosomes randomly and p randomly in [0,1] If p < pc Choose crossover point randomly, and use real value crossing Place new chromosomes in tmp2 new population Else Place old chromosomes in tmp2 new population
If n = N
False
n = n +1
True
n = 0
Mutation: Choose one chromosome randomly and p randomly in [0,1] If p < pm Choose mutation point randomly, and change parameter with a randomly chosen new one Place new chromosome in tmp3 new population Else Place old chromosome in tmp3 new population
If n = N
True
n = 0
False
n = n +1
Replace current population with tmp3 new population
New Generation
g = g +1

Figure 4.4: Function diagram for the GA, where the fitness calculation is done by running a simulation of the model with the actual chromosome implemented in the controller. Note that the best chromosome is retained to the next generation.

4.3.2 Autotuning with a Rule-Based Algorithm

A rule-based tuning algorithm is proposed for the performance index in Section 3.2.2 and the PID-controller (4.3). The motivation behind the proposed algorithm is that if the performance index generates a convex function, as a function of the controller parameters a plain search algorithm would do the job as well as the GA. As a basis for the algorithm, is how a person would think when he is tuning a controller. A block schematic function diagram of the algorithm is illustrated in Figure 4.5, and a short explanation follows.

The search starts in a direction that is logical depending on the performance weight λ and the VOC. For each point (natural frequency or equivalent, controller bandwidth) the algorithm visit it has four basis choice for where to go next (called action), which are:

Action 1: Continue direction and step size.

Action 2: Turn direction and halve step size.

Action 3: Continue direction and halve step size.

Action 4: Continue direction and jump over a step (i.e., take a double step).

The algorithm starts with "large" steps and with action 1. This it continue with until the performance decreases over 2 steps or if the performance decreases more than an offset, then it performs an action 2 (i.e., makes the first turn). After the first turn is made, action 2 is selected each time the performance decreases. Since the algorithm is halving the step size for each turn it makes, the second step in the same direction after a turn is made will already be visited. If the next point is already visited the algorithm have 2 choices, action 3 or action 4, where action 3 is selected if the actual point has higher score than the point one step ahead, and action 4 is selected if opposite. Summarized, the basic action rules are

$$\textbf{Action} = \begin{cases} 1 & \text{if } J_{\omega(c)} > J_{\omega(c-1)} \text{ and } \omega(c+1) \text{ not visited} \\ 2 & \text{if } J_{\omega(c)} < J_{\omega(c-1)} \textit{ (NB! special rules for first turn)} \\ 3 & \text{if } \omega(c+1) \text{ visited and } J_{\omega(c)} > J_{\omega(c+1)} \\ 4 & \text{if } \omega(c+1) \text{ visited and } J_{\omega(c)} < J_{\omega(c+1)}. \end{cases}$$

In addition to these basic rules, some special rules are added. Before an action is performed, it checks against upper and lower bound, and revalue the action to perform (i.e., change to action 2 or 3 if a bound is violated). Furthermore, to not loose any solutions the algorithm checks that both side of the optimum (found) is visited with a minimum step size. Finally, if a action 4 was performed, both the performance for the last point visited and the point it jumped over are considered when selecting a new action.

This procedure is performed until the performance is 100 % or the step size is lower than a predefined minimum step size.

Parameters that have to be set in the algorithm are few and logical to select:

- Upper and lower bandwidth for the controller, which is assumed known from the commissioning. If they are not known they can be selected conservatively.
- A start step size.
- A termination step size.

Limitations of this algorithm include:

- It can only tune one parameter at time, i.e., the natural frequency ω_n for the PID-controller gains (4.4), (4.5) and (4.6).
- To ensure stability, an upper and lower bound for the tuning parameters must be known.
- It requires the performance index function to be convex, i.e., give a global vertex or diminishing return.

Tuning of only one parameter at time is not seen as a considerable drawback, since it is common (logical) to tune each DOF separately. Furthermore, by using bandwidth tuning, the K_p, K_i and K_d gains are all tuned simultaneously.

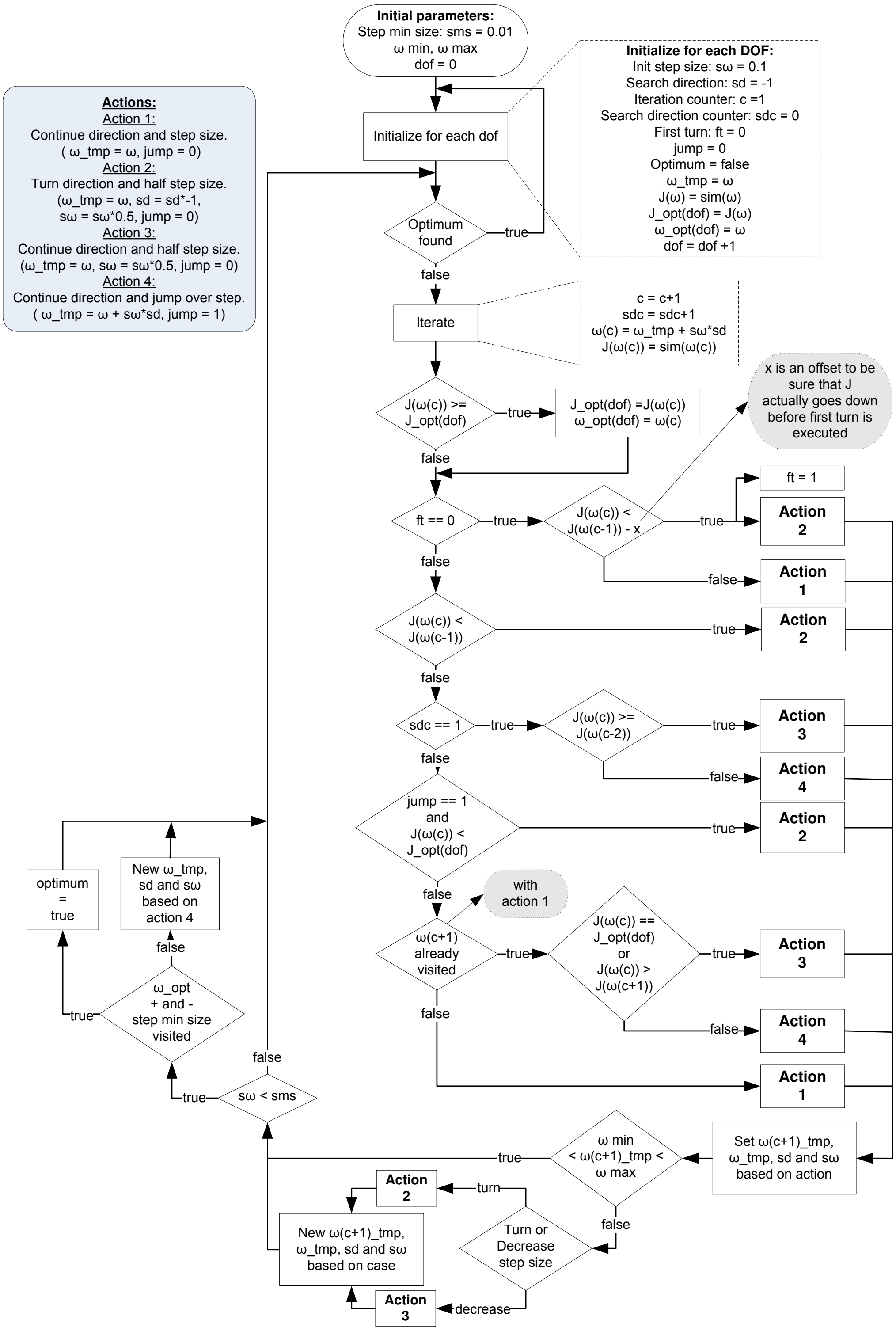


Figure 4.5: Function diagram of the rule-based tuning algorithm.

4.4 Selection of Controller Parameters

In auto-mode, the controller parameters are selected on the basis of an actual vessel operation condition (VOC) and the performance weight λ. Each VOC is given a unique value and have a "unique" set of controller parameters C, linked to the VOC-value O for the λ value. The definition of values for each VOC is defined in 4.4.1, and the linking between the spaces is defined in Section 4.4.2. The same linking is used to employ the controller parameters obtained from the training to populate the controller-space, for the corresponding VOC value. How to detect a VOC is not an issue for this thesis, but it is assumed possible.

4.4.1 The Vessel Operational Condition Space

In Section 3.1, the VOC variable O was given the definition

$$O(\delta, \varepsilon, \varpi) \quad \text{where} \quad \begin{cases} \delta(draught) \\ \varepsilon(environmental - conditions) \\ \varpi(water - depth), \end{cases}$$

which is graphically illustrated in Figure 4.6.

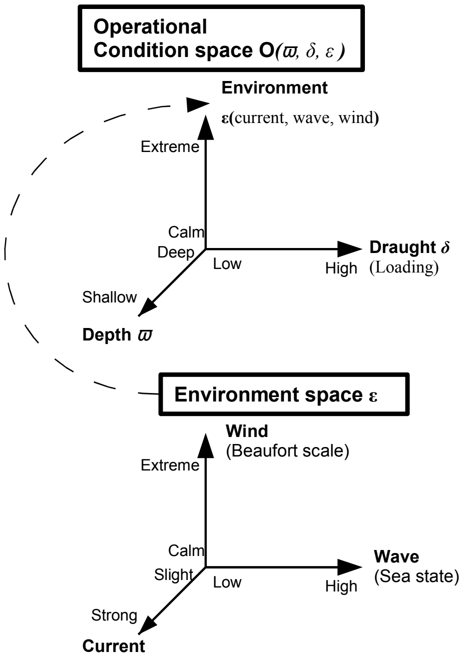


Figure 4.6: Operational Condition space and Environment space.

In the implementation, each of the variables δ, ε and ϖ are (for simplicity) defined as integer values from zero to nine, and their values are selected according to the following rules:

δ is defined such that zero equals minimum draught, and nine equals maximum draught of the vessel. A formula for calculating the value is:

$$\delta = \left(\frac{9}{\delta_{max} - \delta_{min}}\right) \times (\delta_{actual} - \delta_{min}) \qquad (4.12)$$

For example, if the minimum draught is 4 [m] and the maximum draught is 6 [m], then the draught 4 [m] will give $\delta = 0$ and the draught 6 [m] will give $\delta = 9$. Notice that the formula does not give integer values and the result must be rounded of to the nearest integer.

ε is defined such that zero equal calm and nine equal phenomenal (extreme) environmental conditions. The environmental conditions between zero and nine is defined in Table 4.1. In the definition is wave influence favoured over wind, since wind normally generate waves and a sea state condition last longer than a wind condition. Please consult [16] for the relationship between sea state codes and wind conditions (Beaufort numbers). Wind conditions in the area calm to moderate breeze and sea state conditions in the area calm to slight, are seen as a calm environmental condition, since they have low disturbance effect on the vessel. Furthermore, DP operations in rougher environments than a very rough sea state and a moderate gale is seen as abnormal, and is defined as an extreme environmental condition. Stronger wind than moderate gale will also generate high waves (sea state code 7). Current is defined as a high or low condition. See Figure 4.6 for an illustration of the environment space.

ϖ is defined such that zero equal deep water and nine equal shallow water. Shallow water is defined with the highest value, since shallow water has the highest effect on the performance of the vessel.

Since each variable has a range of 10, and there are 3 variables, a total of 10^3 different conditions can occur. The value of the VOC variable O is defined as $O(\varpi, \varepsilon, \delta) = [\varpi, \varepsilon, \delta]^T$, which is a vector with three states, see also Table 4.2. The value of O is used in the mapping between the VOC space and the controller space, see Section 4.4.2. Please notice that with the variables defined as integers a hysteresis function must be used to avoid chattering. An example of a smooth transition function between controllers is found in [34] and the function is revisited in the following.

Scaled-Independent Hysteresis Switching Logic

According to [34], the hysteresis switching logic slow down the switching based on the observed growth of the estimation error $\mathbf{e}_p$. For the case here the estimation error vector $\mathbf{e}_p$ will be the error between the real VOC and VOC models (i.e., how well the measured/detected VOC fit the model/parameters of defined VOCs). The concept of the switching procedure is described in Figure 4.7, and the variables have the following definitions:

- μ_p is a monitoring signal defined in terms of the error norm as

$$\dot{\mu_p} = -\alpha\mu_p + \gamma(\|e_p\|), \quad p \in \{1, \ldots, 999\}, \tag{4.13}$$

  where α denotes a constant non-negative forgetting factor, γ is the class $\mathcal{K}$ function, $\mu_p(0) > 0$, and $\|\cdot\|$ denotes any norm.

- h is a positive *hysteresis constant*.
- $\arg\min \mu_p$ returns the index of the minimum values of the vector μ_p.
- ρ is the VOC switching signal (i.e., $\rho \triangleq O$).
- σ is controller switching signal (i.e., $C(\sigma, \lambda) \triangleq C(O, \lambda)$).

Please consult [34] and the references therein for details about this hysteresis function and definition of other hysteresis functions.

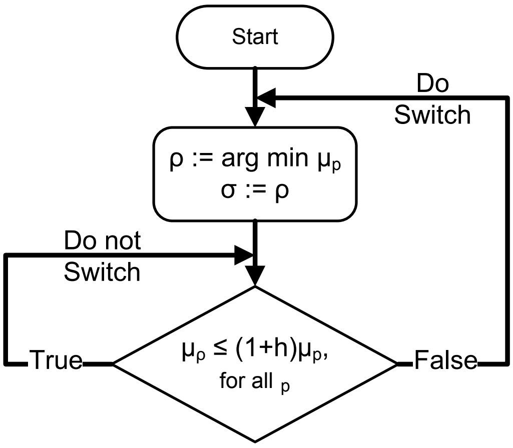


Figure 4.7: The concept of the scaled-independent hysteresis switching logic. Adapted from [34].

ε code	Environmental condition	Wind *Beaufort*	Wave *Sea state*	Current [m/s]
0	Calm	$<$ 5-6	and $<$ 4	and $<$ 1.5
1	Fresh → Strong Breeze	$=$ 5-6	and $<$ 4	and $<$ 1.5
2	Moderate Gale	$=$ 7	and $<$ 5	and $<$ 1.5
3	Moderate Sea	$<$ 7	and $=$ 4	and $<$ 1.5
4	Rough Sea	$<$ 8-9	and $=$ 5	and $<$ 1.5
5	Very Rough Sea	$<$ 8-9	and $=$ 6	and $<$ 1.5
6	Strong Current	$<$ 5-6	and $<$ 4	and $>$ 1.5
7	Rough Combination	$<$ 7	and $<$ 6	and $>$ 1.5
8	Very Rough Combination	$=$ 7	or $=$ 6	and $>$ 1.5
9	Extreme	$>$ 7	or $>$ 6	...

Table 4.1: Environmental code definitions. Notice that wave influence is favoured. See Table 3.1 for definition of sea state codes and [16] for definition of Beaufort numbers.

4.4.2 The Controller Space

The value of the VOC variable O was defined as $O(\varpi, \varepsilon, \delta) = [\varpi, \varepsilon, \delta]^T$, which also can be seen as integer values from zero to nine-hundred and ninety nine. In the mapping between the VOC space and the controller space these values are used as table indices for the controller space C. In addition to VOC, the controller parameters will also be a function of the performance weight λ, which was defined in Section 3.2.3 as a function of *actuator limitations*, *operation accuracy* and *sea state condition*, see Figure 4.8.

An example of the linking between the VOC space O, the weighting factor λ and the controller space C is given below.

Example 4.1
VOC is:

- *Average water-depth*
- *Rough environment*
- *Design draught,*

which gives $\mathrm{O}(4,4,4) = [4,4,4]$*, see Table 4.2. The VOC value* $\mathrm{O}(4,4,4)$ *correspond to controller parameter set number* $\mathrm{C}(4,4,4,\lambda) \Longrightarrow K_p(4,4,4,\lambda), K_i(4,4,4,\lambda)$ *and* $K_d(4,4,4,\lambda)$*, see Table 4.3. An illustration of the mapping is seen in Figure 4.8.*

By using this definition, an easy mapping between the controller and the VOC space is made. The transition between two parameter sets should be

value O	depth ϖ	environment ε	draught δ
$[0,0,0]$	deep	calm	min
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$[4,4,4]$	average	moderate	design
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$[9,9,9]$	shallow	phenomenal	max

Table 4.2: Definition of values for the VOC variable O.

C(O,λ*),*	K_p	K_i	K_d
$[0,0,0,\lambda]$	$K_p(000\lambda)$	$K_i(000\lambda)$	$K_d(000\lambda)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$[4,4,4,\lambda]$	$K_p(444\lambda)$	$K_i(444\lambda)$	$K_d(444\lambda)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$[9,9,9,\lambda]$	$K_p(999\lambda)$	$K_i(999\lambda)$	$K_d(999\lambda)$

Table 4.3: Definition of controller parameters as a function of C and λ.

done smoothly to avoid switching in the controller. Example of ways of doing this is; have a weighting-like function from one controller parameter set to the other, or reduce the error signal into the controller when switching (i.e., move the reference to the measured value and low pass filter it back).

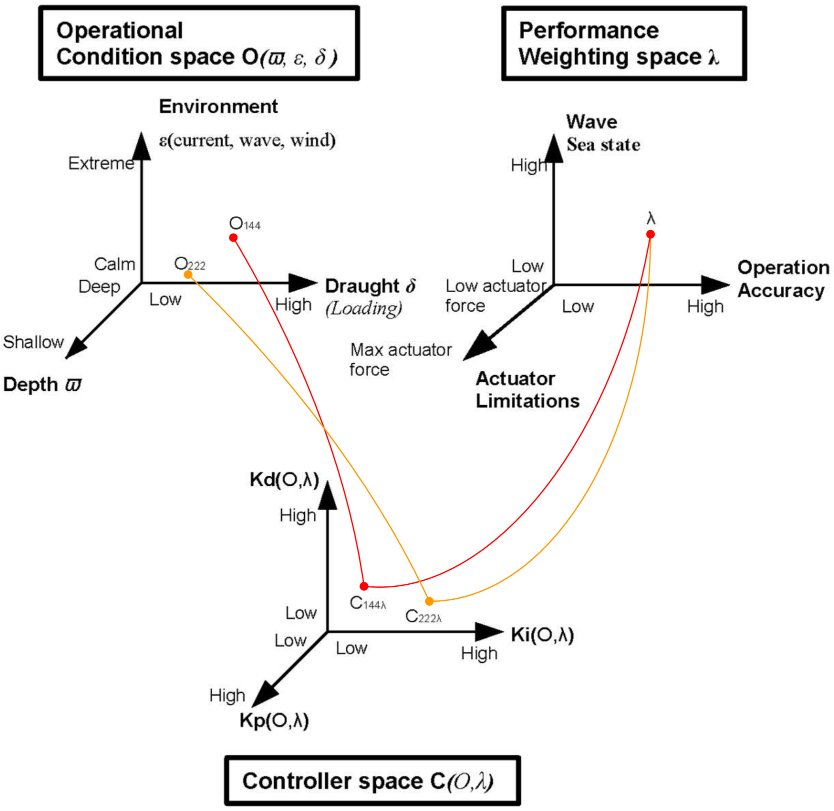


Figure 4.8: Illustration of the vessel operational condition space (VOC), the weighting space (λ) and the controller space (C), with the linking between them.

Chapter 5

Simulation Results

The simulation results verify the behaviour of the suggested performance index for station keeping (Section 3.2.2). Furthermore, the training function for the hybrid controller (Section 4.1) is tested with the mentioned autotuning methods in Section 4.3. The simulation carried out is a scenario for a supply vessel (see Figure 5.1) where the environmental condition goes from one sea state code to another, i.e., 2 different vessel operation conditions (VOCs). For each VOC, the training is carried out with 2 different λ weighting factors in the performance index, i.e., either focus on pose (i.e., position/heading) keeping or focus on weighting forces (i.e., forces/moment) use. Finally, a discussion of the simulation results is carried out, where some results with other parameter settings of the performance index for the same scenario is shown.

Figure 5.1: The supply vessel Northern Clipper. Courtesy of [16].

5.1 Simulation Environment

The mathematical model used for simulation in this thesis is a 3 DOF horizontal model of the supply vessel Northern Clipper (see Figure 5.1). Northern Clipper is a multipurpose supply vessel owned by Trico Shipping AS. Data for the supply vessel are given in Table 5.1, and the vessel is fully actuated with two main propellers, two bow thrusters and two stern thrusters.

Mass:	$4.591 \cdot 10^6$	kg
Length:	76.2	m
Breadth:	18.8	m
Draught:	6.268	m

Table 5.1: Data for Northern Clipper.

A 3 DOF horizontal model of the supply vessel is found in [16, 18], and the model has the following mass matrix $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$

$$\mathbf{M} = \begin{bmatrix} 5.3122 \cdot 10^6 & 0 & 0 \\ 0 & 8.2831 \cdot 10^6 & 0 \\ 0 & 0 & 3.7454 \cdot 10^9 \end{bmatrix}, \tag{5.1}$$

and linear damping matrix

$$\mathbf{D} = \begin{bmatrix} 5.0242 \cdot 10^4 & 0 & 0 \\ 0 & 2.7229 \cdot 10^5 & -4.3933 \cdot 10^6 \\ 0 & -4.3933 \cdot 10^6 & 4.1894 \cdot 10^8 \end{bmatrix}. \tag{5.2}$$

The passive nonlinear observer in [18] was used as filter/observer, which was tested in a full-scale experiment with the Northern Clipper with good results in [18]. The 3 DOF model and the passive nonlinear observer is also implemented in the Marine GNC toolbox, which can be downloaded free of charge from the Marine Systems Simulator (MSS) home page at `http://www.marinecontrol.org/`.

The vessel model was implemented with disturbances from waves and a static sea current. Sea current was implemented according to (3.5), (3.6) and (3.7), as

$$u = u_r + V_c \cos(\beta_c - \psi) \tag{5.3}$$

$$v = v_r + V_c \sin(\beta_c - \psi) \tag{5.4}$$

$$r = r_r, \tag{5.5}$$

where V_c and β_c were implemented as static values, since the simulation time was short. Wave disturbances were generated by a simplified dynamic wave force model [20], which is an internal model developed at Rolls-Royce Marine AS based on [12, 24]. The generated wave disturbances were added as force disturbances $\boldsymbol{\tau}_w$, i.e.,

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{5.6}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{D}\boldsymbol{\nu} = \boldsymbol{\tau}_{contr} + \boldsymbol{\tau}_w. \tag{5.7}$$

A real vessel has actuator saturation limits that affects the controller tuning. In order to also incorporate such effects, the model was implemented with saturation limits in maximum forces and maximum limits in rate of change in forces. Saturation limits were selected by assuming a maximum speed for the vessel model in each DOF (i.e.; 7.5 [m/s] in surge, 2.5 [m/s] in sway and 60 [deg/min] in yaw). Maximum limits in rate of change in forces were selected by assuming that the actuators need 7 [s] from zero to maximum force/moment. See Table 5.1 for the actuators limits used in the simulations.

$\tau_{max}(surge)$:	$7.5 \cdot 5.0242 \cdot 10^4$	N
$\tau_{max}(sway)$:	$2.5 \cdot 2.7229 \cdot 10^5$	N
$\tau_{max}(yaw)$:	$\frac{\pi}{180} \cdot 4.1894 \cdot 10^8$	Nm
$\Delta\tau_{max}(surge)$:	$\tau_{max}(surge)/7$	N/s
$\Delta\tau_{max}(sway)$:	$\tau_{max}(sway)/7$	N/s
$\Delta\tau_{max}(yaw)$:	$\tau_{max}(yaw)/7$	Nm/s

Table 5.2: Magnitude and rate saturation limits in forces and moment.

5.2 Test Scenario

In the test scenario it was desirable that both tuning with performance weights $\lambda = 1$ (minimum deviation) and $\lambda = 0.5$ (balanced deviation and use of forces) should be tested. Furthermore, it was desirable to see the controller performance after a change in the environment, both before and after a new autotuning. The performance index behaviour for static disturbances that requires a static controller output was also desirable to see. Notice that for writhing/reading simplicity it is referred to λ (weight in figures), when it is talked about λ_1 in (3.38), λ_2 in (3.39) and λ_3 in (3.40), since they were all set to the same value.

In Table 3.1, the sea states codes with highest percentage probability to occur are 3, 4 and 5 (slight, moderate and rough), and thus these sea states codes are used in the simulations. Two worst case scenarios among these were chosen, namely sea state 4 and sea state 5. In these sea state codes were the highest wave height and the lowest frequency selected, which is further called sea state code upper 4 and upper 5. Furthermore, to get disturbances both in surge and sway a wave direction of $-45°$ (quartering sea) was used, which also will give some disturbances in yaw. A static sea current with $V_c = 0.5$ [m/s] and $\beta_c = -135°$ (the same direction as the waves) was implemented to generate a static force. The operation requirements were set to

- Surge deviation 3 [m], i.e., offset level 3 surge is 3 [m]
- Sway deviation 3 [m], i.e., offset level 3 sway is 2 [m]
- Heading deviation 1.5°, i.e., offset level 3 heading is 1.5°,

where the sensor error is assumed negligible (i.e., ≈ 0).

The complete test scenario, which is illustrated in Figure 5.2, can briefly be summarized into 4 different cases

1. An autotuning is performed for minimizing deviation in pose (i.e., $\lambda = 1$), in the sea state condition upper 4.
2. An autotuning for balancing deviation and use of forces in side some deviation limits (i.e., $\lambda = 0.5$) is performed in the sea state condition upper 4.
3. An environmental change occur where the sea state condition goes from upper 4 to upper 5.
4. Autotuning for the new sea state condition is performed with $\lambda = 0.5$.

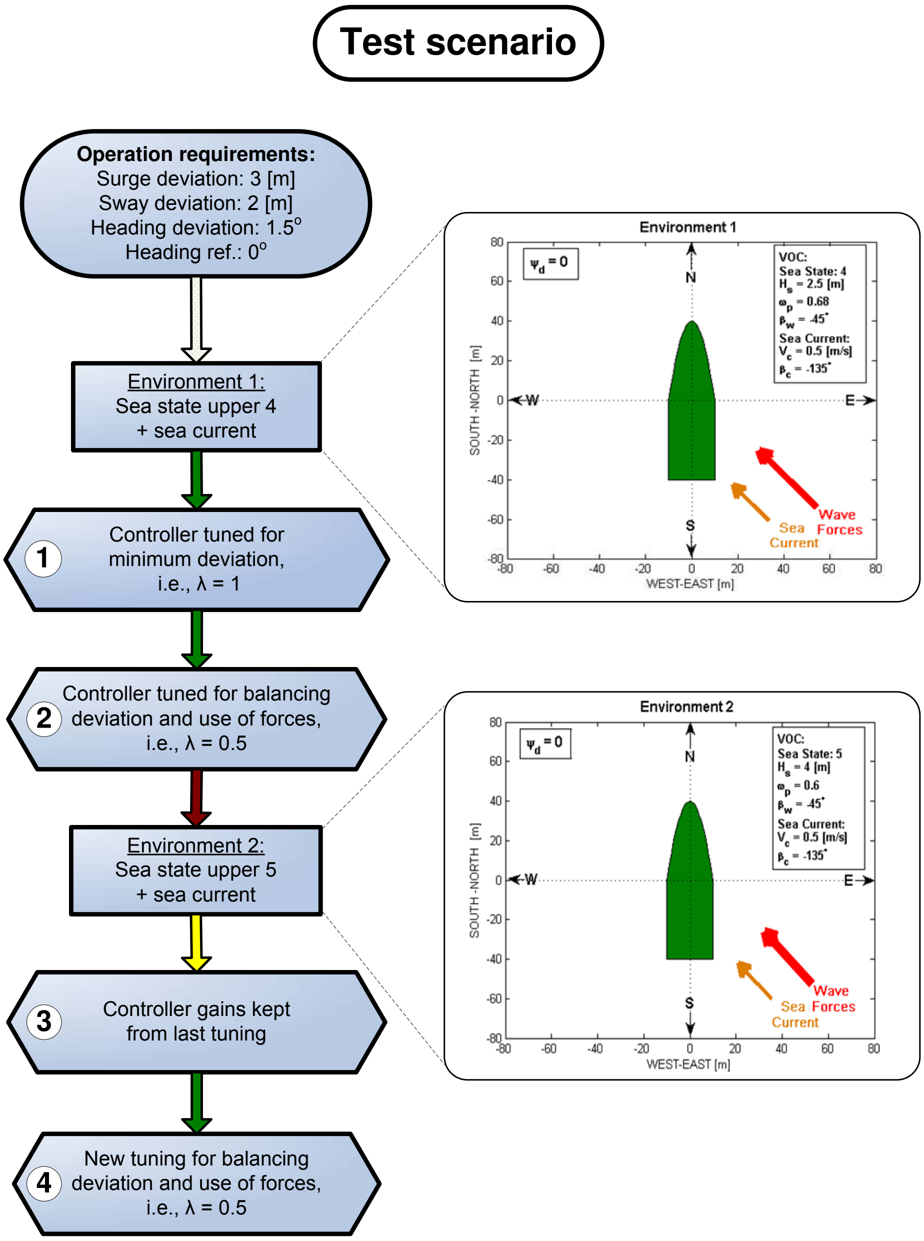


Figure 5.2: Illustration of the simulation scenario, time increases with the arrows. A green arrow indicates that a tuning is performed, while a red arrow indicates an environment change, and a yellow arrow indicates no change in environment or tuning. The numbers (1, 2, 3 and 4) are used as references for later.

5.3 Simulation Results

In the simulation results presented here, only bandwidth (ω_n) tuning of the PID-controller (4.2) is considered, where the other controller parameters were set to

$$K_{d_{factor}} = 1 \Rightarrow \zeta = 1 \tag{5.8}$$

$$K_{i_{factor}} = 10. \tag{5.9}$$

Please remember that ω_n is not the exact bandwidth, but the natural frequency and must be multiplied with a $\bar{\zeta}$ factor (i.e., 0.64 with $\zeta = 1$) to get the exact bandwidth (ω_b), see (4.8). Default controller bandwidths were set to

$$\omega_{n_i} = 0.1, \text{ for } i \in \{surge, sway, yaw\}, \tag{5.10}$$

which are the initial bandwidths for the autotuning in case 1.

The upper and lower bandwidth bounds for the autotuning functions were set to

$$\omega_{n_{max_i}} = 0.5 \qquad i \in \{surge, sway, yaw\} \tag{5.11}$$

$$\omega_{n_{min_i}} = 0.0375 \qquad i \in \{surge\} \tag{5.12}$$

$$\omega_{n_{min_i}} = 0.0625 \qquad i \in \{sway, yaw\}, \tag{5.13}$$

where the upper bounds were chosen 0.1 lower than the maximum wave frequency and lower bounds were set by experience.

In order to avoid initial step disturbance effects, which will never occur in a real situation, calculation of the performance indices $J_{T_{surge}}$ (3.38), $J_{T_{sway}}$ (3.39) and $J_{T_{yaw}}$ (3.40) was delayed 2 minutes, while activating of actuator saturation elements were delayed 1 minute.

The sea state change was assumed automatically detected, and with this change some parameter settings were altered. For instance, the observer was manually pre-tuned to fit each sea state code, and in the test scenario parameters were changed with the sea state code (i.e., the observer was adaptive). Please consult [54] for a observer with similar structure as the observer used here with gain-scheduled wave filtering.

In the first environment, initial inner level settings were

$$\text{level } 1_{surge} = 1 \text{ [m]} \tag{5.14}$$

$$\text{level } 1_{sway} = 1 \text{ [m]} \tag{5.15}$$

$$\text{level } 1_{yaw} = 0.5 \text{ [deg]}, \tag{5.16}$$

which was used in the autotuning in case 1 (i.e., in the autotuning for minimizing deviation in pose in sea state code 4).

After the first autotuning was performed with $\lambda = 1$, inner level of the performance index was set as the largest deviation from the tuning with $\lambda = 1$, and level 2 was set with a γ factor of 0.7 such that

$$\text{level 2} = \text{level 1 } + \gamma(\text{level 3 - level 1}). \tag{5.17}$$

In the second environment, the inner level was set as a function of max average deviation with $\lambda = 1$ and level 2 was set with the same γ factor. Furthermore, the minimum offset parameter κ_{min} (in (3.36)) used for calculating J_{τ} was set to

$$\kappa_{min_i} = \begin{cases} 0 & \text{if } \textit{sea state code } == 4 \\ 0.15 & \text{if } \textit{sea state code } == 5, \end{cases} \tag{5.18}$$

while κ_{max_i} was set to 0.75, for $i \in \{1, 2, 3\}$.

Other parameters used in calculating the performances indices were set as follows:

- The time window (T) used in calculating subperformance indices (3.27), (3.29), (3.34) and (3.35) was set to 5 minutes
- The sampling time (h) used was 0.2 seconds
- The weighting factors ω_{p_i} in (3.30) and ω_{h_i} in (3.31) were set to $\omega_p = \omega_p = [0.2, 0.3, 0.5]$

In order to know what constitute a good controller bandwidth (ω_n), pre-searches in the bandwidth range $\omega_{n_{min}}$ to $\omega_{n_{max}}$ with a step size 0.0125 were carried out. The search order was first surge, then sway and finally yaw. In these searches, the bandwidths were initialized as $\omega_n = \omega_{n_{max}}$ for the search DOFs, where the best ω_n for a DOF was kept to the next DOF, i.e., when searching in sway the result from surge was used for $\omega_{n_{surge}}$ while $\omega_{n_{yaw}}$ was still set to $\omega_{n_{actually}}$. With a step size of 0.0125 and the range $\omega_{n_{max}} - \omega_{n_{min}} = 0.45$, the number of iterations (ς) for each DOF is

$$\varsigma = \frac{0.45}{0.0125} + 1 = 37. \tag{5.19}$$

The ς number of iterations is seen as a competitor for the rule-based algorithm and the GA.

5.3.1 Simulation Results with Rule-Based Tuning

With the rule-based autotuning algorithm in Section 4.3.2, autotuning was first carried out in surge, then in sway and finally in yaw. Initial bandwidth step size was set to 0.1 and minimum step size was set to 0.0125. With the minimum step size equal to the step size used in the pre-search and with the same search order, the algorithm should be able to find the optimum in the pre-search.

The algorithm found the same optimum as in the pre-searches, see examples in Figure 5.3. Besides finding the same optimum, it also found it in a satisfying number of iterations, see Table 5.3 for an iteration log of case 1, 2 and 4. The maximum number of iterations was 14 and the average for case 1, 2 and 4 were respectively 11, 6 and 8, which are less than ς.

DOF	case 1	case 2	case 4
Surge:	14	6	7
Sway:	7	5	10
Yaw:	13	6	7
Total:	34	17	24
Average:	11	6	8

Table 5.3: Number of iterations carried out for the rule-based tuning algorithm in case 1, case 2 and case 4. Note that each iteration takes T minutes.

Examples of iterations for the rule-based algorithm compared with the results from the pre-searches are seen in Figure 5.3.

In the following plots of position and heading performances for each case with the rule-based autotuning algorithm are presented. Furthermore, the performance balance between the pose performances and the force performances are presented for each DOF. A comparison of the performance balance before and after the autotuning is made. For sea state condition 4, the saving in use of forces from case 1 to case 2 is also presented.

Case 1: Autotuning with $\lambda = 1$ in Sea State Condition 4

In sea state condition 4 it was no problem to keep the position inside the operation requirements with good margin, see Figure 5.4. Different initial setting of the inner level led to different solutions for the controller bandwidth ω_n. A strict setting of the inner level typically led to a high bandwidth in surge since this was tuned first and a lower value in sway. A less strict setting led to a more balanced controller bandwidth between the DOFs, which was preferable and used in the results presented here. A more balanced controller bandwidth between the DOFs often reduced the number of iteration carried out in the autotuning from one case to another.

While the position and heading deviations were small, the force performances index values $J_{\tau_{surge}}$, $J_{\tau_{sway}}$ and $J_{\tau_{yaw}}$ were rather poor with this tuning and there should be possibilities for considerable improvements.

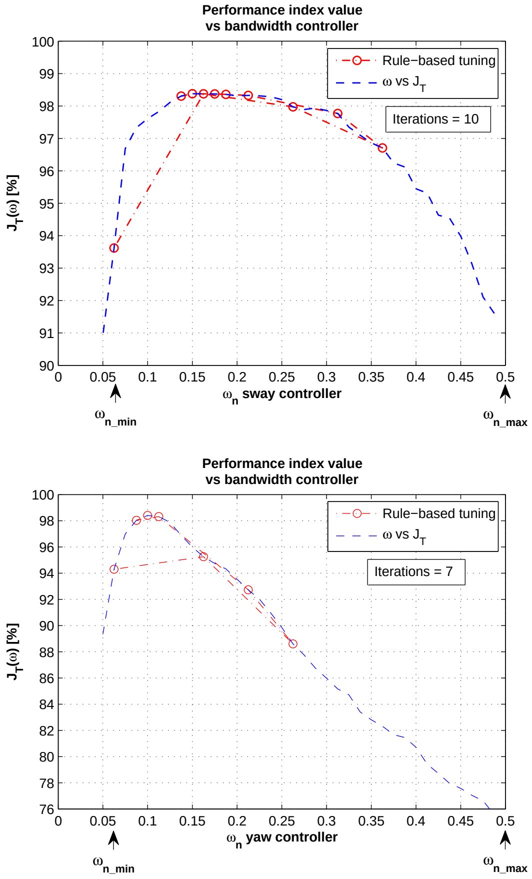


Figure 5.3: Iterations for the rule-based algorithm compared with a search through all bandwidths in the range $\omega_{n_{min}}$ to $\omega_{n_{max}}$ for case 4. *Upper image:* Autotuning in sway, where the rule-based algorithm found the best bandwidth after 10 iterations. *Lower image:* Autotuning in yaw, where the rule-based algorithm found the best bandwidth after 7 iterations.

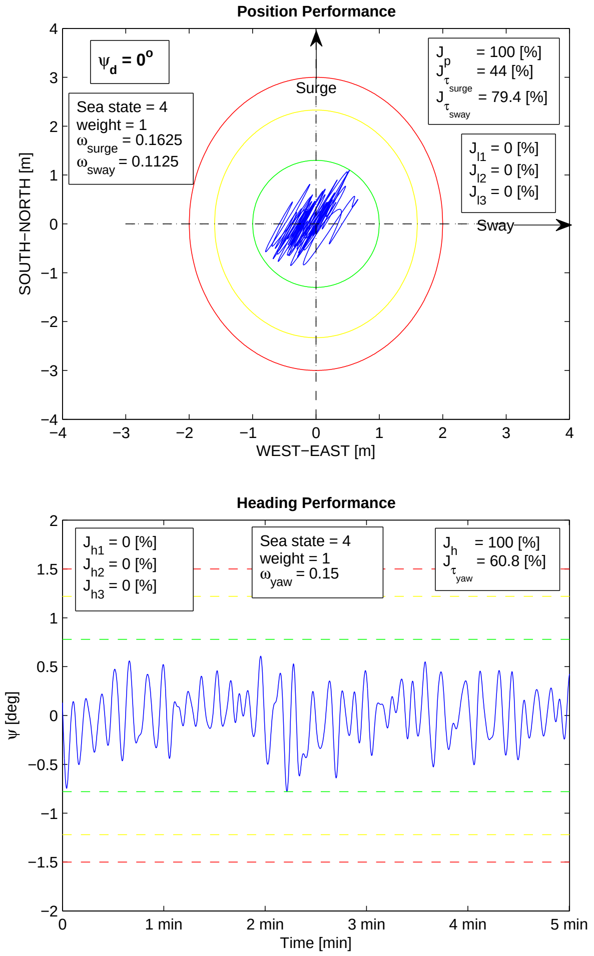


Figure 5.4: Initial performance in sea state condition 4 (case 1), with the controller tuned for minimizing deviation. *Upper image:* Position performances. *Lower image:* Heading performances.

Case 2: Autotuning with $\lambda = 0.5$ in Sea State Condition 4

After an autotuning was carried out there was some growth in the position deviation, but the vessel was still inside the operation requirement level and was only for 3.3 % of a time window of 5 minutes outside level 2, which is approximately 10 seconds. For the heading, there was almost no growth in the deviation, see Figure 5.5.

While there was some growth in the position deviation, a considerable improvement in the force performances were made. The force performance index values $J_{\tau_{surge}}$, $J_{\tau_{sway}}$ and $J_{\tau_{yaw}}$, were much better and almost balanced with the pose performance index values J_p and J_h, see Figure 5.6. The actual effect of these reductions in dynamical force variations is seen in Figure 5.7. For surge, the total movement length for the actuator(s) was reduced with approximately 84 % and the peak range was reduced from 52 % to 12 %. In sway, the reduction was not so large, which is natural since the force performances was basically better. However, the improvements were still considerable. Yaw on the other side had the same reduction in total movement length as surge, and the peak range was reduced with 42 %. With these reductions, actuator wear are reduced and lower peak values results in less energy need. Besides reduced wear and energy need, lower peak values and peak range in actuator positions will often have a second effect in reduced noise and vibrations from the actuators, which increases the comfort of the crew on the vessel.

While peak values and total movement length were reduced for all DOFs, the mean values in forces were not reduced (largest 0.3 % in surge). This fact was expected, since the mean value is necessary for counteracting the static disturbances. In essence obtaining a filtering effect, where those changes that are static in a given time window are counteracted, and those of high frequency neglected due to the thruster weighting.

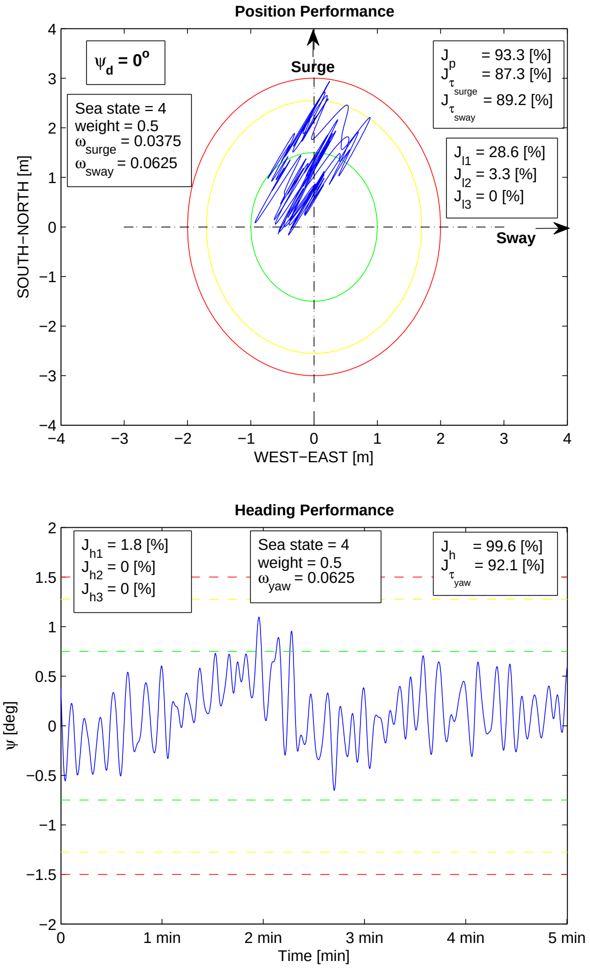


Figure 5.5: Position and heading performances for case 2. *Upper image:* Position performances J_p are almost balanced for both $J_{\tau_{surge}}$ and $J_{\tau_{sway}}$, while the position deviation is still inside the position requirements for the operation. *Lower image:* Heading performances J_h are almost balanced with $J_{\tau_{yaw}}$ and only a small change is seen in J_h from the previous case.

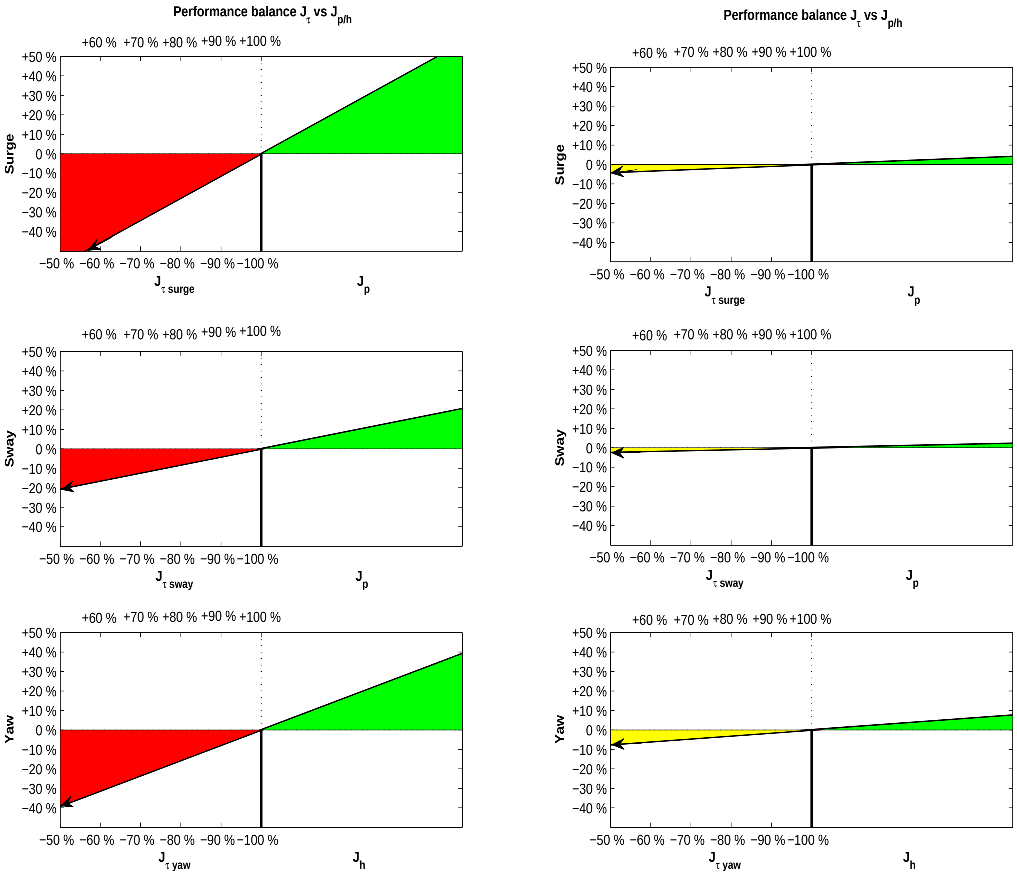


Figure 5.6: Illustration of performance balances in sea state condition 4. *Left image:* For the initial condition (case 1 in the test scenario), where the controller was tuned for only minimizing the pose deviation, it is seen that $J_{p/h}$ has a considerably better performance than J_τ. *Right image:* After an autotuning was carried out (case 2 in the test scenario), the performance balancing was much better in all DOFs.

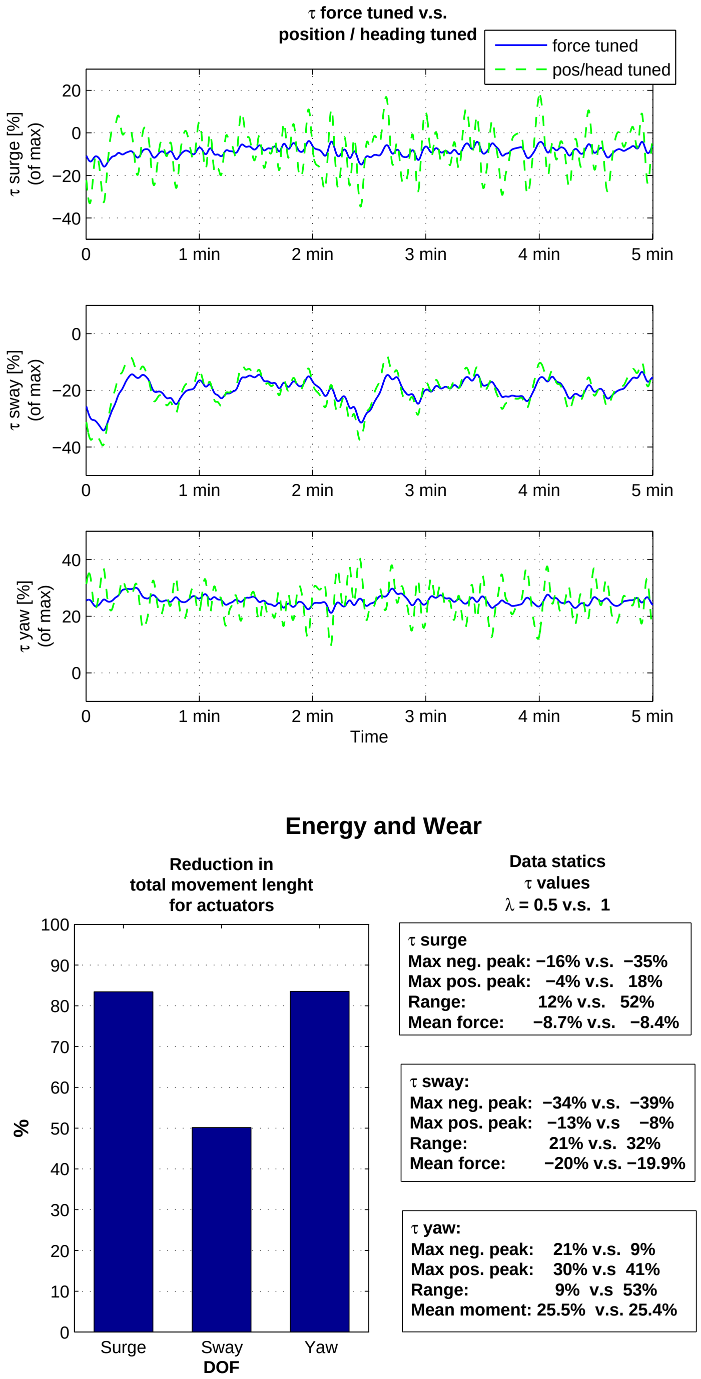


Figure 5.7: Savings in use of forces after an autotuning for balancing performance was carried out in sea state code 4 (case 2 in the test scenario). Notice that largest change in mean force and moment is 0.3 % for surge.

Case 3: Change of Sea State Condition to 5

When the sea state condition changed from upper code 4 to upper code 5, the position performance index J_p decreased to an unacceptable level, which is seen in Figure 5.8. The vessel was breaking the position requirements for over 30 % of the time window and spent almost 50 % of the time window outside the inner level (green sector). The force performance indices $J_{\tau_{surge}}$ and $J_{\tau_{sway}}$ on the other hand were still good (see also Figure 5.13), which indicates that there should be improvement possibilities in the position performance.

The heading performance index J_h also decreased down to an unsatisfying level, see Figure 5.9. The force performance index was good, which means that there are possibilities for improvement.

Note the new inner level (green sector) setting, which also has changed the setting of level 2. The inner level was set as the average maximum deviation after an autotuning with $\lambda = 1$ was carried out. An average maximum was used instead of the maximum deviation, because in the waves generated for this sea state condition there was a wave peak that could not be handled by the actuator limits used, see Figure 5.12. A wave peak that cannot be handled by the actuators is consider as normal in a rough to very rough sea state, see Figure 5.10 for a picture of a vessel in such a sea state condition.

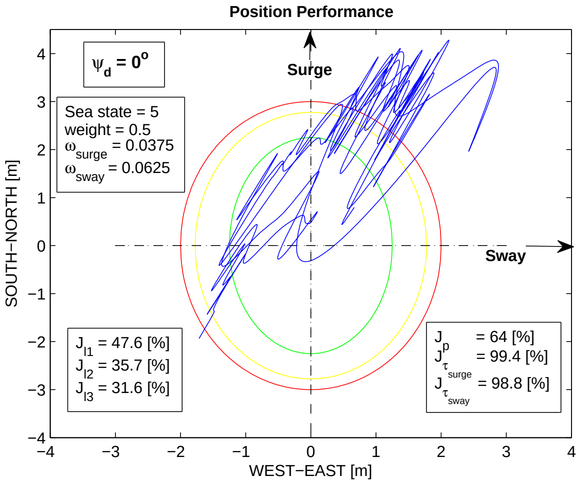


Figure 5.8: Position performances in sea state condition 5 (case 3 in test scenario), with the controller tuned for balancing performances in sea state condition 4.

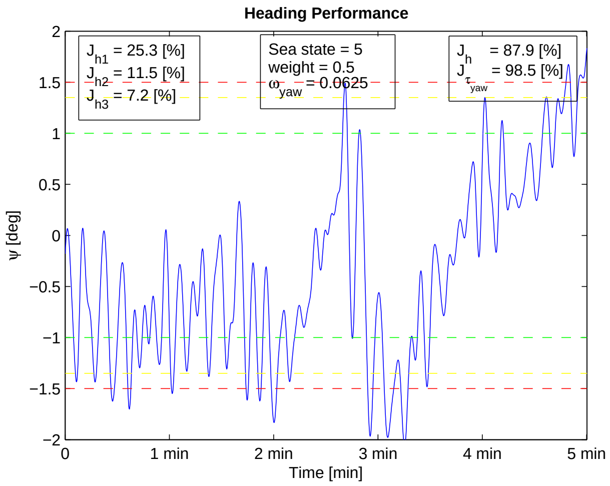


Figure 5.9: Heading performances in sea state condition 5 (case 3 in test scenario), with the controller tuned for balancing performances in sea state condition 4.

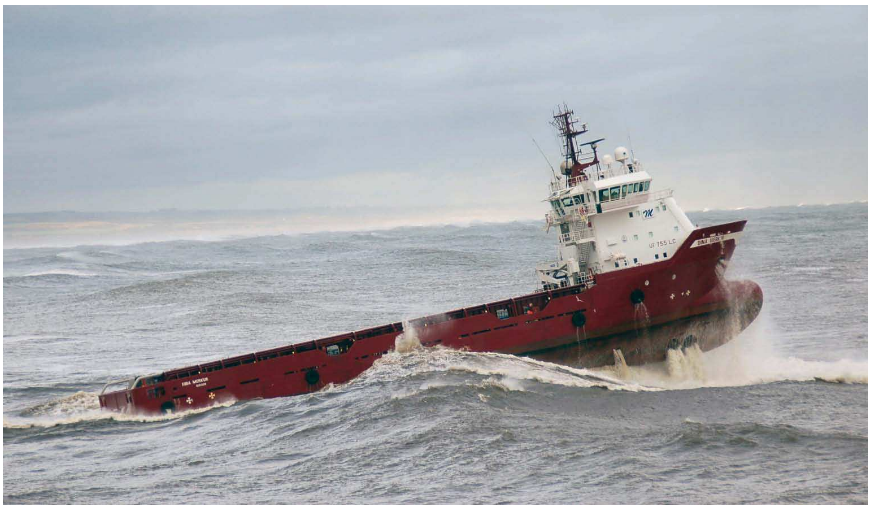


Figure 5.10: A supply vessel in sea state condition upper 5 outside Aberdeen, Scotland. Courtesy of Rolls-Royce Marine AS.

Case 4: Autotuning with $\lambda = 0.5$ in Sea State Condition 5

After an autotuning was carried out for the new sea state condition, the position performance was considerably improved and the vessel position was only outside level 3 for a short peak, see Figure 5.11. This peak occurred because the actuators reached their saturation limits in sway, which can be seen in Figure 5.12. A more aggressively tuned controller would not have helped against this peak, since the actuator saturations are the problem, which was experienced with a $\lambda = 1$ tuning. The question is then rather if the operation can be carried out or such a short peak can be accepted (here it will be assumed that it can be accepted). Note that peaks like this were already selected to be accepted by setting the $w_{p/h} = [0.2, 0.3, 0.5]$ factors in (3.30) and (3.31).

Besides improving the position performance index J_p from 64 % to 97.7 %, the autotuning also kept the force performance indices $J_{\tau_{surge}}$ and $J_{\tau_{sway}}$ at 99 %. This gave an almost perfectly balanced performance, which is seen in Figure 5.14.

Heading performance had approximately the same improvement as the position, where the force performance was kept, and the performance was balanced out between heading and dynamical use of forces, see Figure 5.13 and 5.14.

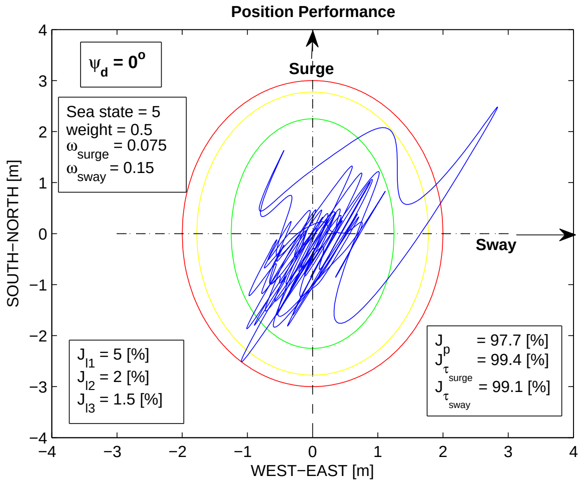


Figure 5.11: Position performances in sea state condition 5 (case 4), after an autotuning for balancing performances was carried out.

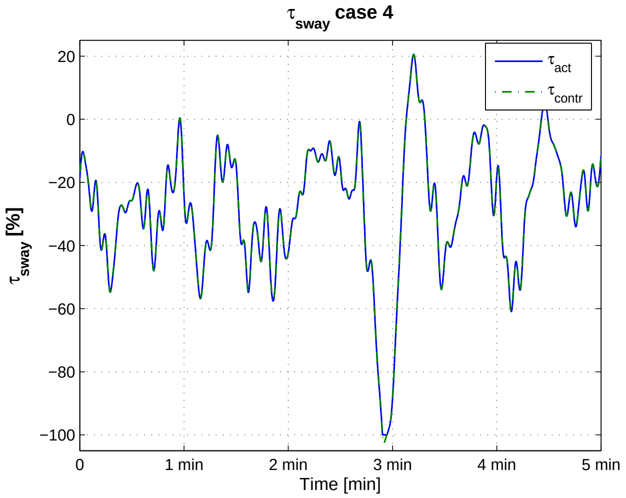


Figure 5.12: Log of controller output and actuator setpoint in sway for case 4. The controller output reached the saturation limit of the actuator and cannot give more force to keep the vessel inside level 3 in the sway DOF.

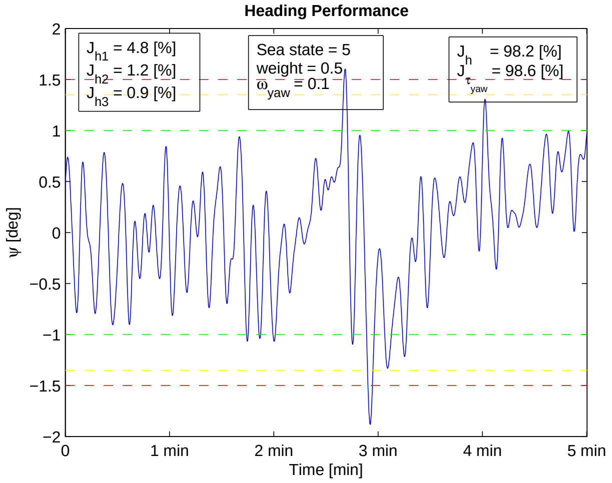


Figure 5.13: Heading performances in sea state condition 5 (case 4), after an autotuning for balancing performances was carried out.

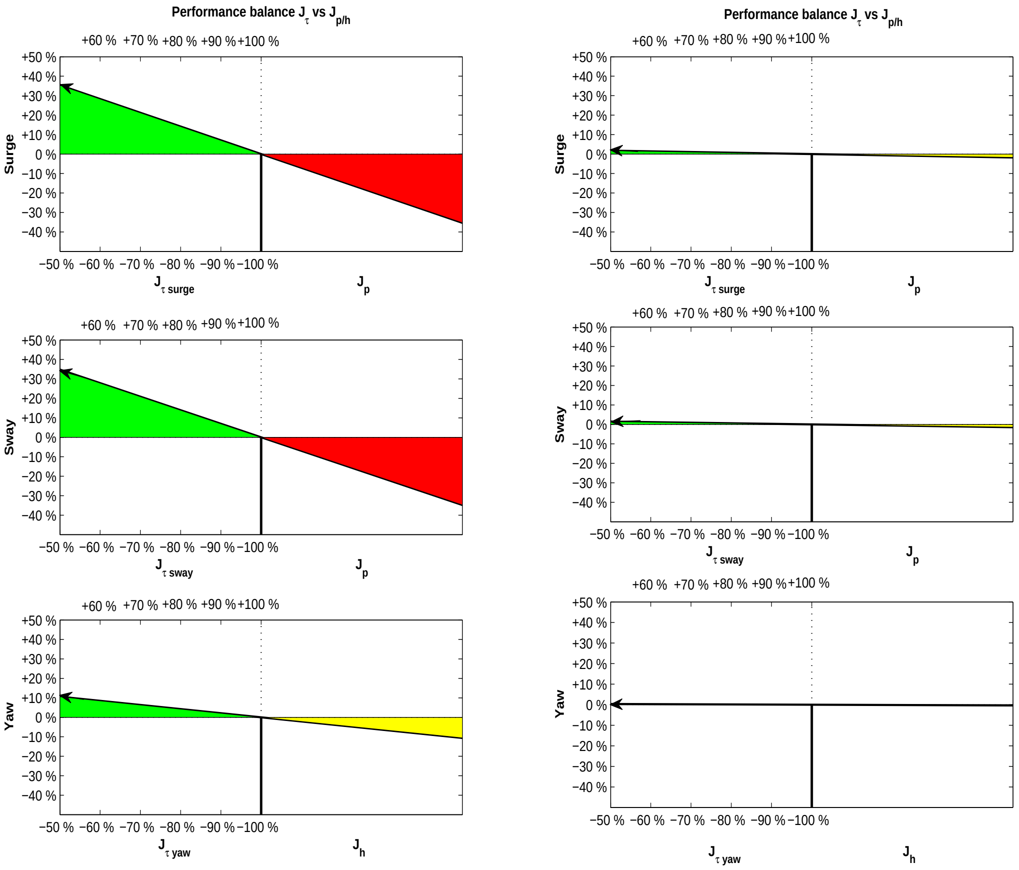


Figure 5.14: Illustration of performance balances in sea state condition 5. *Left image:* Before an autotuning was carried out for the new sea state (case 3), J_τ has a considerably better performance than $J_{p/h}$ for all DOFs. *Right image:* After an autotuning was carried out (case 4), the performance balances were almost perfect.

5.3.2 Simulation Results with GA Tuning

With the GA in Section 4.3.1, the number of iterations ς_{GA} is a function of the population size (N) and the max number of generations (g_{max}) as

$$\varsigma_{GA} = N \cdot g_{max}, \tag{5.20}$$

which should be $\ll \varsigma$ in (5.19) (i.e., $\varsigma_{GA} \ll 37$). There are possibilities that the algorithm terminates before the max number of generations is reached, but worst case must be considered. If the GA shall have a practical function it must have some generations to evolve over, i.e., $g_{max} \geq 3$. In [22], it was suggested that 5 generations is an appropriate choice, which would give a ceiling of 7 chromosomes in a population ($N \leq 7$).

Different combinations were tried out with $g_{max} \in [3, 5]$ and $N \in [5, 7]$. The experiences were that a larger population size gave better results than more generations. Furthermore, a good solution for case 4 close to the solution for the rule-based algorithm was often found, even with N=*5* and $g_{max} = 3$. For case 2, the solution was more random, but more constant and better with a larger *N*. With $N = 7$ and $g_{max} = 3$, the number of iterations became $\varsigma_{GA} = 21$, which is much larger than the maximum number of iterations used for the rule-based algorithm (which was 10), see Table 5.3.1. The conclusion was that the GA could not oust the rule-based algorithm in number of iterations when each DOF was tuned separately.

Since it could not oust the rule-based algorithm in tuning-time, it had to oust it in performances to be a better choice. This might be possible since none of the performance indices in case 2 and 4 equals 100 % for the pre-searches and the rule-based algorithm, a minimum step size was used for the pre-searches and the rule-based algorithm, and there could be other combinations of bandwidths between the DOFs that gives a better performances than the solution found with the pre-searches and the rule-based algorithm.

A sample of 5 autotuning results with the GA ($N = 5$ and $g_{max} = 3$) were taken and compared with the results form the rule-based algorithm, see Table 5.4.

case	RB	GA 1	GA 2	GA 3	GA 4	GA 5
2: $J_T =$	92.5	87.4	87.6	88.7	93.3	82.5
4: $J_T =$	98.5	98.4	97.9	98.4	98.3	98.0

Table 5.4: Performance sample of 5 autotuning simulations carried out with the GA and compared against the rule-based algorithm (RB). The performance index value J_T is calculated according to (3.41).

During the 5 samples, the GA oust the rule-based algorithm in only one of them (sample 4), and with a small margin. The problem was case 2, where the best controller bandwidths were close to the lower bounds. It must be mentioned that with $N = 7$ and $g_{max} = 3$, the scores were often the same as for sample 4, but then with a considerable longer tuning time than the rule based algorithm.

5.4 Discussion

The discussion start with the result of the autotuning functions, where the genetic algorithm (GA) and the rule-based algorithm are compared, and some improvements are suggested. Furthermore, the behaviour and the parameter setting of the performance index are discussed. The advantage of the hybrid controller is compared against a controller with static default parameters. Finally, some concluding remarks are mentioned.

5.4.1 The GA v.s. the Rule-Based Algorithm

The rule-based algorithm found the optimum in the pre-search given by the performance index, and it found it in an acceptable number of iterations. For the GA, the optimum found was not necessary the same as for the pre-search, which is obvious since it has possibilities to use other bandwidths than the pre-search. With possibilities to use other bandwidths, the GA had better performance result than the rule-based for case 2 and it had the same performance result for case 4 (see Table 5.4). On the other hand, to have some certainty in the result the GA needed a minimum population size of 7 and at least 3 generations, which gives 21 iterations for each DOF (i.e., totally 63). For the rule-based algorithm, the largest number of iterations for a DOF was 14 (case 1 surge), while the average for all cases was 8, see Table 5.3. Both the average and the maximum number of iterations are then smaller for the rule-based algorithm than for the GA. However, the number of iterations for the rule-based algorithm depends strongly on the shape of performance index function as well as the start point. For instance, tuning with $\lambda = 1$ had typically more iterations than tuning with $\lambda = 0.5$, since the vertex of the performance index was typical lower with $\lambda = 1$, see Figure 5.15.

The total tuning time (T_{tuning}) is a function of number of DOF, the average number of iterations ($\varsigma_{average}$) and the time window (T), i.e., the average best tuning time for the hole scenario was

$$T_{tuning} = DOFs \cdot \varsigma_{average} \cdot T = 3 \cdot 8 \cdot 5 = 120 \text{ [min]} \qquad (5.21)$$

which is quite some time, but improvements are possible. For instance, the upper bound for the bandwidth was set to $\omega_{n_{max}} = 0.5$ in (5.11), while all optimum found were lower than 0.2. If this was an available knowledge, the upper bandwidth could without loosing solutions be set to 0.25, and the

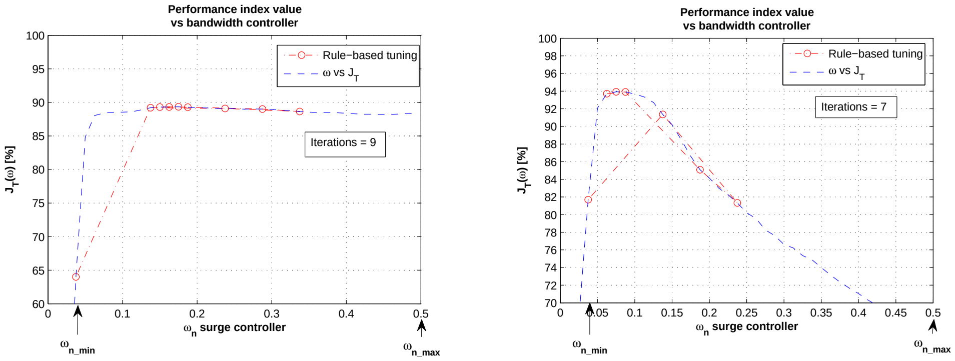


Figure 5.15: Progress in controller bandwidth ω_n search with the rule-based algorithm in sea state condition 5. *Left image:* Search in surge with $\lambda = 1$. *Right image:* Search in surge with $\lambda = 0.5$.

number of iterations would been reduced to less than 10 for all cases and the average would been reduced. Such knowledge is adapted by the training, and the number of iterations will therefore for the rule-based algorithm decrease with number of training performed. Furthermore, a complete time window was run for all iterations in the simulation. If it is already known at an early stage that the current best performance cannot be beaten, reduction in the tuning time can be made by implementing rules which stop the run.

Besides influencing the number of iterations, the shape of the performance index does also have an important role in the the rule-based algorithms ability to find the optimum. If the performance index does not generate a convex function, the rule-based algorithm can stop at a local vertex (optimum) instead of finding the global vertex (optimum).

An advantage with the GA is that it is easier to implement with tuning of the $K_{i_{factor}}$ in (4.10) and the $K_{d_{factor}}$ in (4.11), which gives possibilities for better performance. On the other hand, with more parameters to tune and with interactions between them, the number of generations must be increased and the tuning time will increase considerable. Similar is it for tuning of all 3 DOF simultaneously, which also will have interactions and the number of generations must be increased. Furthermore, with more parameters to tune the population size must be increased to ensure that enough combinations are tested.

Summarized, both autotuning algorithms performed well in finding the optimum given by the performance index, where the rule-based algorithm showed to fined the optimum in less iteration than the GA. On the other hand, the rule-based algorithm is more sensitive to the shape of the performance index function.

5.4.2 The Performance Index

Since the optimum given by the performance index was found for both autotuning methods, the performance result achieved by the autotuning as well as the choice of autotuning method are dependent on the performance index function. The performance index function showed to perform well in the test scenario and the rule-based algorithm would be the obvious choice.

However, how well the performance index behave is depending on the tuning of the parameters in the function, which are:

T	time window
h	sampling time
$L_{p/h}$	levels position/heading
$w_{p/h}$	weighting between levels position/heading
κ_{min}	minimum offset for the $J_{\mathcal{T}}$ performance index
κ_{max}	maximum factor for the $J_{\mathcal{T}}$ performance index
λ	force weighting.

Some of these parameters are dependent on the VOC or more precisely the environmental conditions. Different draught (loading) and water depth conditions would influence the performance of the vessel and thereby demand different controller tuning, but the influence on the tuning of the performance index is considered low and neglected. Among the environmental conditions, the sea state condition is considered as the one that has the largest influence on the tuning of the performance index. This since the sea state condition contributes the largest amount of dynamical use of forces and is difficult to counteract with feedforward terms. Furthermore, how to set these parameters can be seen as to different cases. For the first, if the performance index is used in the training and the controller parameters are supposed to be stored in the hybrid controller in Section 4.1, the setting must be linked to the VOC and the λ weight. On the other hand, if the performance index is used for a single autotuning or as a index for a DP operator, the parameters can be set more freely. Some reflections for both cases follow.

The time window T is set according to the time it takes to detect the average disturbance/performance, and the sampling time h is set equal to the sampling time for the controller. A longer time window (T) will increase the certainty that the disturbance effect is stationary at the detected values, but on the other hand will increase the tuning time. For low to rough seas (i.e., sea state lower than 6) it is common to consider the sea as stationary for periods over 20 minutes. A time window of 20 minutes will give 4 times longer tuning time than used in the simulation and is a considerable long tuning time. On the other hand, the 20 minute rule is used to say that the sea state condition is stationary (i.e., in detecting the sea state condition), which should be performed by the switching function before the autotuning starts. It is believable that the average disturbance can be detected in a shorter time window for the sea state condition and the time window can be set shorter. Furthermore, the autotuning will normally go over some iterations and an accidental disturbance would not affect all iterations.

For the $L_{p/h}$ parameters, the inner level is set as the historical maximum (or maximum average) deviation for the VOC when the controller was tuned for minimum deviation, while the outer level is set by the operation requirements inside a maximum and a minimum limit. These parameters are then dependent both on the VOC and the operation requirement. For storage of controller parameters in the hybrid controller, all level parameters should be fixed to the VOC and the λ weighting factor. An easy setting is that all levels are a factor of the inner level which depends on the VOC, and the factor depends on the λ weighting factor.

The $w_{p/h}$ weighting of the $L_{p/h}$ levels should be set so that outer level is more punished than middle level, and middle level more punished than inner level. To simplify the tuning, it is recommended to keep this weighting static and rather adjust the setting of the outer level and the γ factor (in 5.17), since that will have larger effect and are more logical to set. This means that for storage in the hybrid controller, these parameters are fixed for all VOCs.

The result of the J_τ performance index is depending on the setting of the parameters κ_{min} and κ_{max}. For instance, in case 2 in the test scenario, the position performance index J_p was larger then both force performance indices ($J_{\tau_{surge}}$ and $J_{\tau_{sway}}$). With another setting of the κ_{min} the performance results would been different, e.g., see Figure 5.16 where κ_{min} is changed from 0 to 0.05. The position performance J_p has slightly increased with the new setting, while the force performance $J_{\tau_{surge}}$ and $J_{\tau_{sway}}$ have increased to almost 100 %, but the real use of forces has slightly decreased.

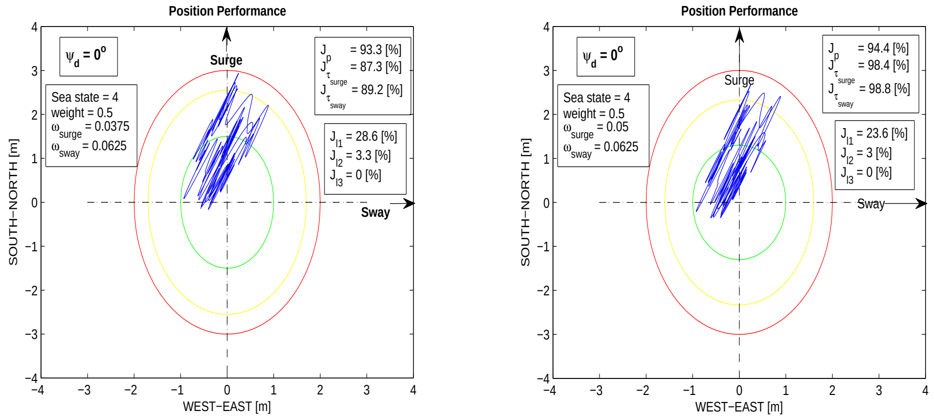


Figure 5.16: Change of position and force performance with different setting of κ_{min} for case 2 in the test scenario. *Left image:* Original setting, $\kappa_{min} = 0$. *Right image:* New setting, $\kappa_{min} = 0.05$.

With the original setting it was not possible to achieve a better result for the force performance, because of the lower bound setting in the autotuning algorithm. While with the new setting the autotuning did not find the real best force performance, but the best performance achieve by the performance index setting, see Figure 5.17. The best of this setting of the κ_{min} offset is dependent on the goal, but the effect of reducing dynamical use of forces was obtained with both settings. Similarly for the heading and yaw moment performance for this case, see Figure 5.18 and Figure 5.19.

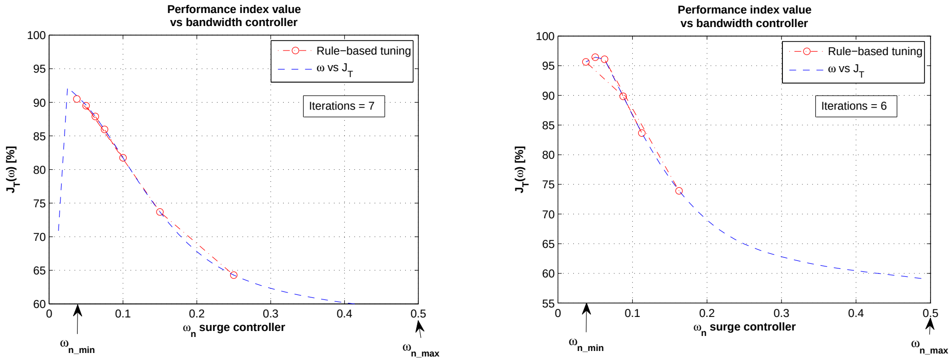


Figure 5.17: Progress in controller bandwidth ω_n search with the rule-based algorithm for case 2 in the test scenario, with different setting of κ_{min}. *Left image:* Search in surge with $\kappa_{min} = 0$. *Right image:* Search in surge with $\kappa_{min} = 0.05$.

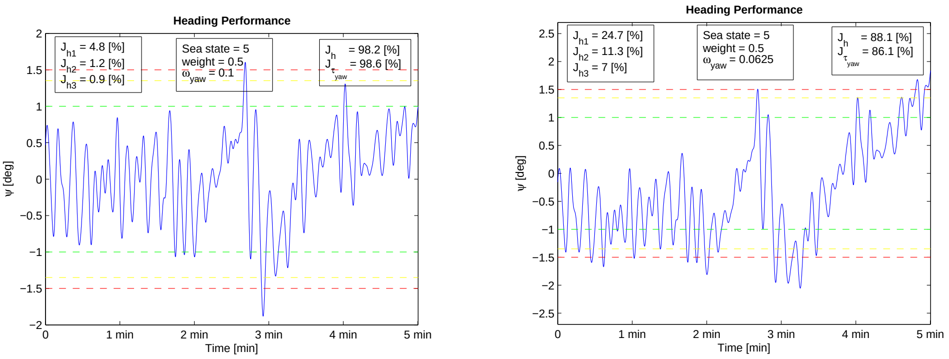


Figure 5.18: Change of heading and force performance with different setting of κ_{min} for case 4 in the test scenario. *Left image:* Heading performance with original setting, $\kappa_{min} = 0.25$. *Right image:* Heading performance with new setting, $\kappa_{min} = 0$.

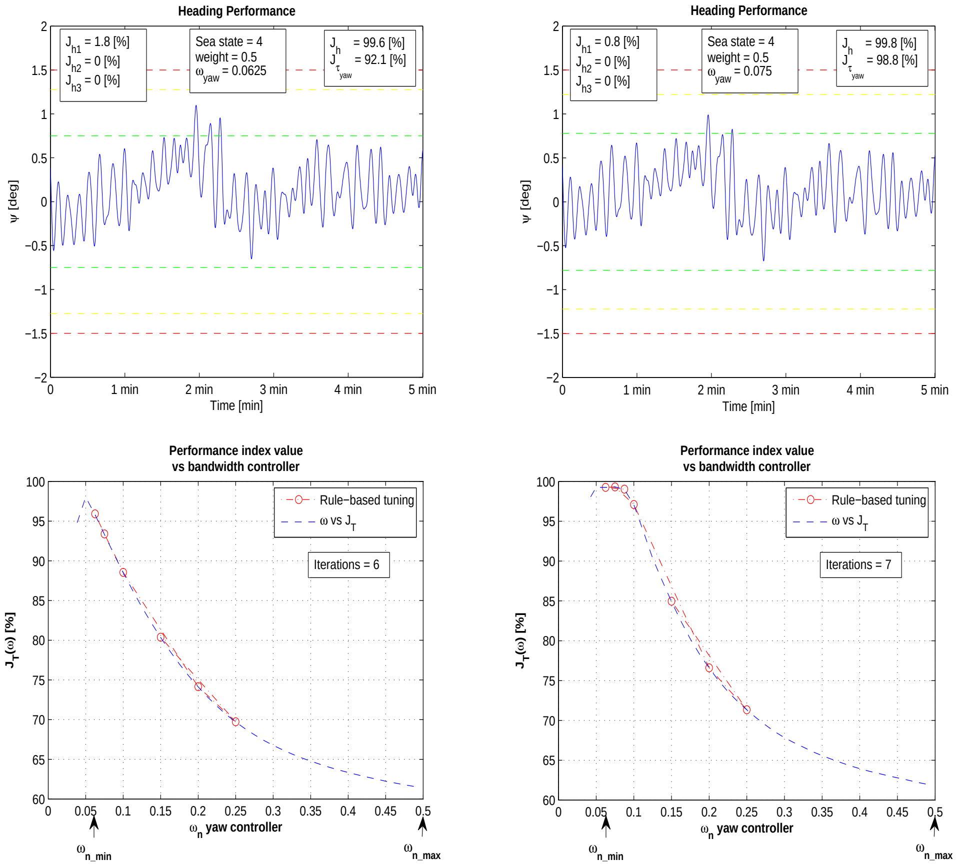


Figure 5.19: Change of heading and force performance with different setting of κ_{min} for case 2 in the test scenario. *Upper left image:* Heading performance with original setting, $\kappa_{min} = 0$. *Upper right image:* Heading performance with new setting, $\kappa_{min} = 0.05$. *Lower left image:* Search in $\omega_{n_{yaw}}$ with original setting, $\kappa_{min} = 0$. *Lower right image:* Search in $\omega_{n_{yaw}}$ with new setting, $\kappa_{min} = 0.05$.

In case 4 of the test scenario, the pose performance was almost 100 % (97.7 % and 98.2 %), and the forces performance was almost 100 % (99.4 %, 99.1 % and 98.6 %), i.e., almost balanced with the pose performance. This was desired, but a larger reduction (saving) in the use of forces could been obtained with a smaller value for the κ_{min} offset at the sacrifice of lower pose performance. Summarized, by setting the λ weight to 0.5 and simultaneously adjust the κ_{min} offset and the κ_{max} factor, the performance index has the opportunity to balance the forces performance with the pose performance at any pose performance level. Unfortunately, this opportunity is not possible for the storage case since the κ_{min} offset and the κ_{max} factor must be fixed to the VOC, where they should be increased with the sea state condition. Furthermore, for the storage case they must be adjusted so the $J_{\mathcal{T}}$ performance is not 100 % with the λ weight = 1.

Keeping of pose should always have highest priority for the controller, which leads to that the λ weighting should never be set lower than 0.5, i.e., balanced between pose performance and use of forces. The range for the λ weighting factor is then 0.5 to 1. A plot of different λ weight settings for the two sea state conditions, and the corresponding optimal controller bandwidth found are seen in Figure 5.20. From the figure, it is seen that a lower λ weight gives a lower controller bandwidth, and a lower sea state gives a lower controller bandwidth, which was expected. For the storage case, the λ weight is also used to adjust the distance between inner and outer level, i.e., with $\lambda = 0.5$ the distance is maximum for the VOC and opposite for $\lambda = 1$. A small weighting on the forces will lead to a more convex function, since the performance index then is a function of two conflicting objectives. For an autotuning it then might be preferable to always have a small weighting on the forces performance (e.g., 0.95), with the cost of might increase the minimum deviation.

5.4.3 The Advantage of the Hybrid Controller

Besides for one DOF (sway from case 2 to 4), different controller parameter setting was found as the optimal for each case during the test scenario, see Table 5.5. This verifies the advantage of the hybrid controller, where the controller parameters changes according to the VOC and the operation requirements.

Case	DOF		
	ω_{n_surge}	ω_{n_sway}	ω_{n_yaw}
1	0.1625	0.1125	0.15
2	0.0375	0.0625	0.0625
4	0.075	0.15	0.1

Table 5.5: Log of optimal bandwidth found for each DOF in the different cases. Notice that besides the parameter setting for sway from case 2 to 4, all parameters has changed from one case to another.

The advantage of the hybrid controller was also confirmed by looking on the performance log for a controller with static controller setting, during the test scenario. With a static default controller set with the initial controller bandwidths $\omega_{n_i} = 0.1$ for $i \in \{surge, sway, yaw\}$, a very good but not optimal pose performance is obtained in sea state 4. On the other hand, the dynamical use of forces are very bad, see Figure 5.21. In sea state 5, the default controller bandwidth is the same as the one found as optimal for the heading/yaw performance and there are of course no change in the

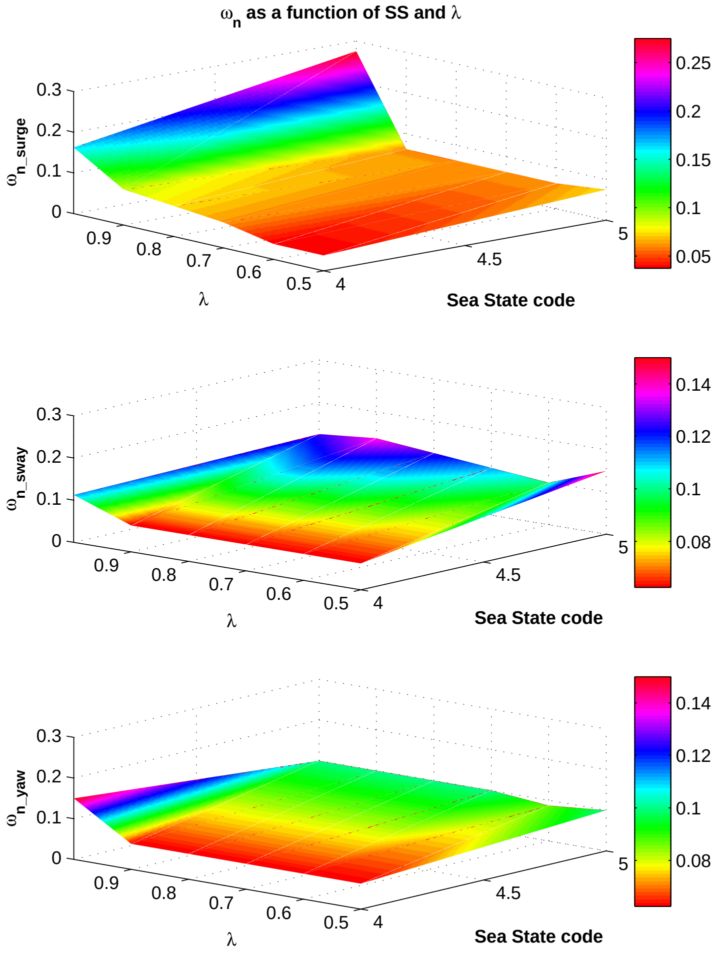


Figure 5.20: Controller bandwidths as a function of different λ weights in sea state upper 4 and upper 5.

performance. For sway, the force performance is also almost equal, while for surge the force performance is considerable reduced, see Figure 5.20. The change in position performance was small with this controller setting (from about 98 % to about 97 %), but again not the optimal setting and both the force performance and the position performance were reduced.

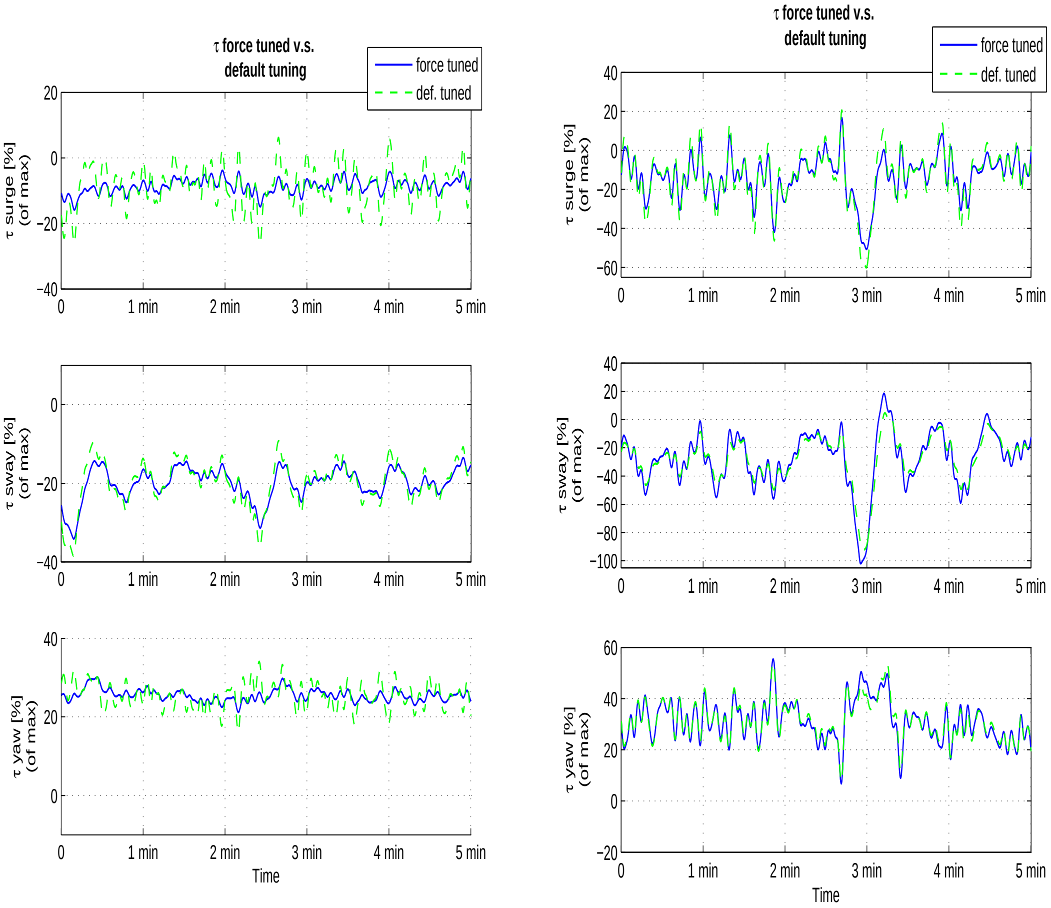


Figure 5.21: Comparison of use of forces between a controller tuned with weighting on the use of forces and a controller with default parameters. *Left image:* In sea state upper 4, it can easily be seen that the force performance is better with the force-tuned controller. *Right image:* In sea state upper 5, for surge is the force performance better with the force-tuned controller, while it is almost equal in sway.

5.4.4 Concluding Remarks

Commercial DP controllers has the advantage of manual gain adjustment, so the comparison made between the hybrid controller and a static controller can be argued to be invaluable. However, such a manual tuning would depend on the experience of the DP operator and the tool he has to verify the performance. Furthermore, it is believable that a DP operator would think in a linear control theory way, when adjusting the controller. In linear control theory it is normal to think that higher gains give less deviation until instability occurs, but for a vessel with actuator saturations and disturbance from waves this is not always the fact. In sea way, the normal fact is that there is a diminishing return where a more active controller would not reduce the pose deviation further. In high sea states, the pose performance can actually decrease with higher gains, which was actually the fact for the heading performance in sea state upper 5, see Figure 5.22.

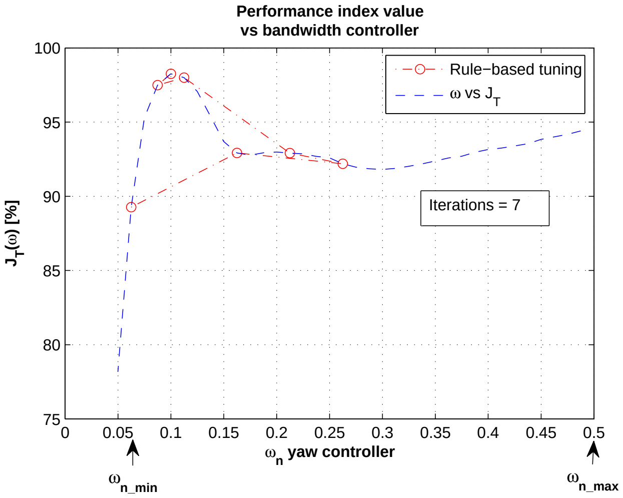


Figure 5.22: Performance index value v.s. controller bandwidth $\omega_{n_{yaw}}$, with $\lambda = 1$ in sea state upper 5.

By tuning the controller based on the performance index with weighting on dynamical use of forces, a filtering effect in the controller was obtained. With this filtering effect where those changes that are static in a time window counteracted and those of high frequency neglected. In the simulation results it was seen that this effect gave a more stationary thruster command, which reduces wear and tear on thrusters, saves energy and increases comfort. Two functions in commercial DP systems that have fuel and wear-and-tear saving effects are mentioned in [3]. The functions are "Relaxed" DP mode (Converteam, formerly known as Alstom) and "Green DP" (Kongsberg Simrad). The differences between these functions and the hybrid controller suggested here (according to the information in [3]), are:

- In the "Relaxed" DP mode, controller parameters are lowered down to a pre-set level depending on the relax factor, and moved up to normal level if the deviation reach a deviation limit ("Relax Limit") set by the operator. With this function, the controller parameters are not optimized for the VOC, and are not using historically adapted information. Furthermore, the function is only suitable for use in light weather conditions.

- In the "Green DP" mode, the operator select an inner area (Working area) and an outer area (Operational area), see Figure 5.23. Inside the Working area the thruster commands are only used to counteract measured disturbance, i.e., the setpoint is following the vessel position, and thruster forces are set equal to measured/predicted disturbance. If the vessel is predicted to violate the limits of the Working or the Operational area, thruster commands are initiated to reduce the overshoot, see also [19]. This function is dependent on the accuracy of the model and the measured/predicted disturbance, which can be difficult to obtain in rough environments. The function is also general and not optimalized for different VOCs. Furthermore, historical data/experience are not used.

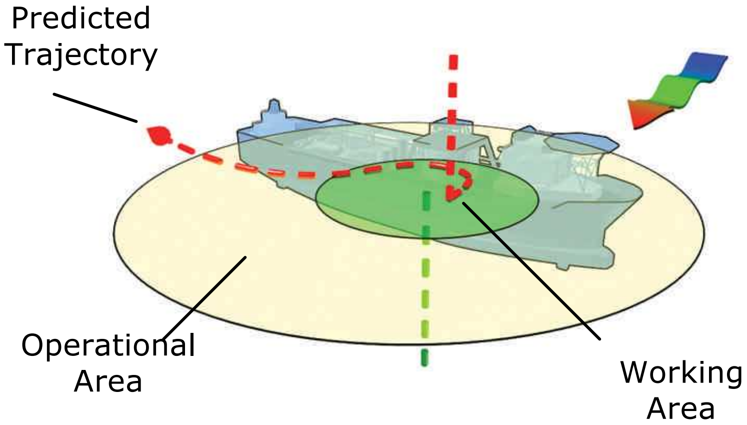


Figure 5.23: Operational limits for green DP. Working area - "soft" inner boundary and Operational area - "hard" outer limit. Courtesy of [19].

For both the previously mentioned industrial functions, thruster commands are increased if the deviation violates a pre-set deviation limit. Violating a pre-set deviation limit can be seen as breaking level 2 or 3 in the performance index suggested in this thesis. Furthermore, in the suggested controller it is assumed that the disturbance would be equal the disturbance in the time window until the VOC change. An instant disturbance that is larger than the disturbance in the time window can of course occur. Such an instant disturbance can make the deviation unacceptable large and must somehow be treated. A solution can be an adaptive controller function, where the controller bandwidth changes as a function of the pose deviation, see Figure 5.24 for an illustration of the shape of the function. The controller bandwidth starts to increase when the deviation reaches level 2 and is at its maximum when the deviation reaches level 3 (this is tuneable with the gradient s_e in (5.22)). In this function, ω_{n_o} is the actual bandwidth according to the VOC, while ω_{max} can be a fixed high gain tuning (e.g., the bandwidth from $\lambda = 1$ tuning for the VOC, or the default bandwidth). A hysteresis function is also implemented where the sign of the error is used, i.e., one value for an increasing error and one value for a decreasing error. The function have three parameters to tune; the gradient (s_e) in

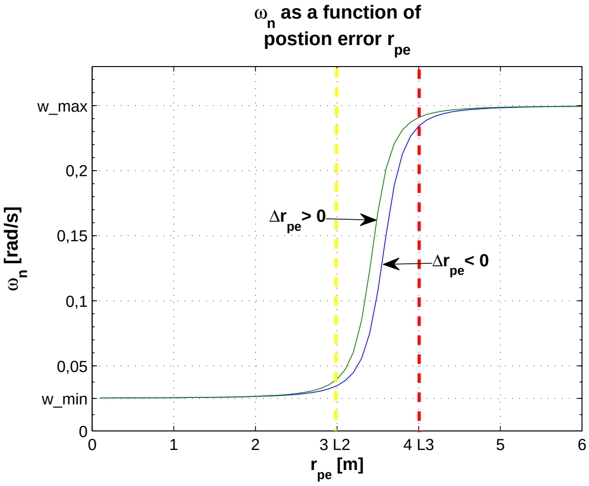


Figure 5.24: The shape of the adaptive bandwidth function, with $s_e = 0.5$ and $\sigma = 0.01$, notice that w_min is the same as ω_{n_o}.

$$\omega_n = \omega_{n_o} + \frac{\omega_{n_{max}} - \omega_{n_o}}{2} \cdot \left(\frac{e_k - e_{sign}(L3 + L2)}{\sqrt{(e_k - e_{sign}(L3 + L2))^2 + (s_e e_{sign}(L3 - L2))^2}} + 1 \right), \tag{5.22}$$

the hysteresis width (σ) and the gradient $s_{\Delta e}$ in

$$e_{sign} = \frac{1}{2} - \sigma \left(\frac{e_k - e_{k-1}}{\sqrt{(e_k - e_{k-1})^2 + (s_{\Delta e})^2}} \right), \tag{5.23}$$

where e_k is the pose error at sample k. Such a function will of course reduce the force performance if the deviation is often larger than level 2. A simulation test was carried out with a step disturbance that affected the vessel in all 3 DOF, see Figure 5.25 and 5.26. In the figures, it can easily be seen that such a function have an effect in reducing the deviation. It must also be mentioned that the idea of the function came late in the study, so the behaviour of the function has not been studied in detail here, and the simulations are too few to draw a conclusion.

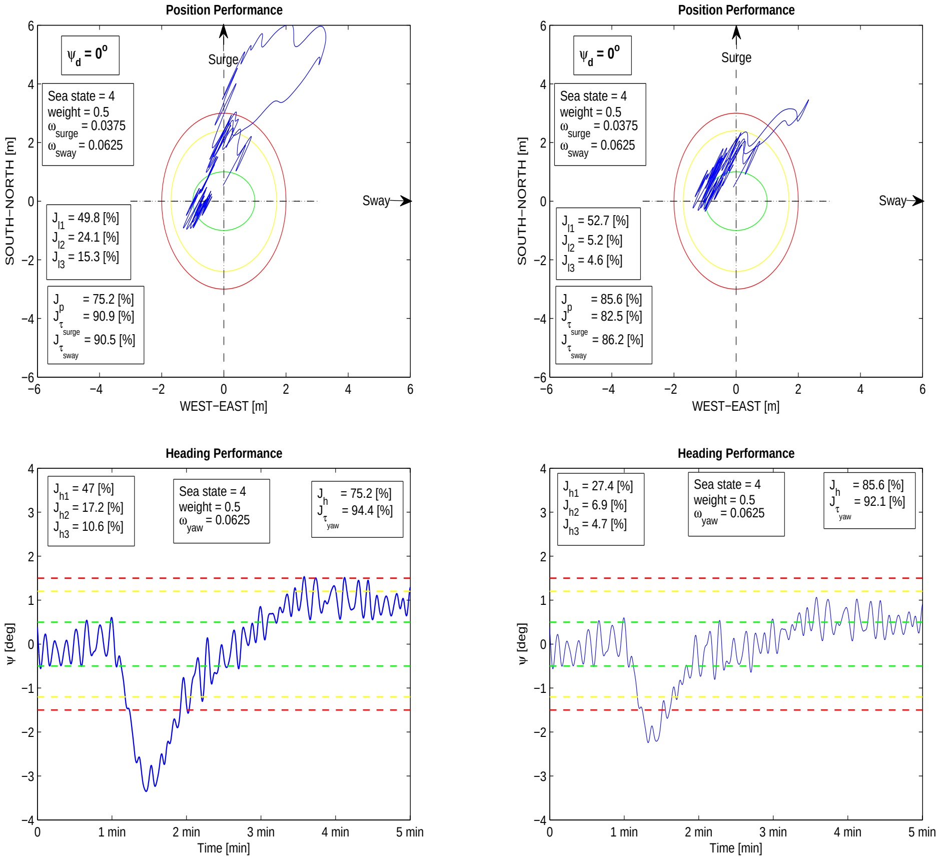


Figure 5.25: Performance of the suggested adaptive controller compared with a static controller, when the vessel was forced with a step disturbance. *Upper left image:* Position performance without the adaptive function. *Upper right image:* Position performance with the adaptive function. *Lower left image:* Heading performance without the adaptive function. *Lower right image:* Heading performance with the adaptive function.

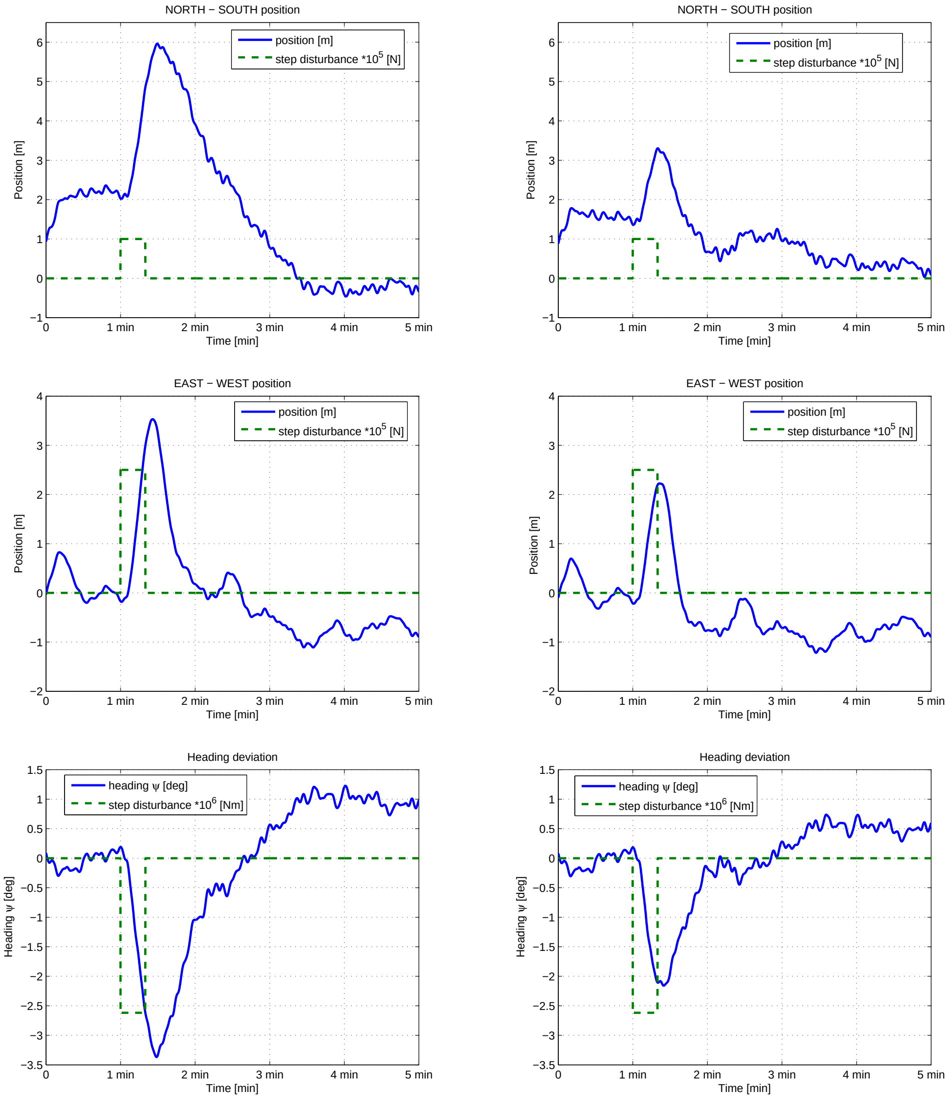


Figure 5.26: Performance of the suggested adaptive controller compared with a static controller, separated into DOF. *Upper left image:* North-South position deviation without the adaptive function. *Upper right image:* North-South position deviation with the adaptive function. *Mid left image:* East-West position deviation without the adaptive function. *Mid right image:* East-West position deviation with the adaptive function. *Lower left image:* Heading deviation without the adaptive function. *Lower right image:* Heading deviation with the adaptive function.

Chapter 6

Conclusion

The suggested performance index for station keeping was behaving as desired, and was separating out unnecessary dynamical use of forces as bad performance and neglecting static use of forces. With this information available, the controller was easily tuned to keep the pose (i.e., position/heading) inside pre-defined limits with minimal use of forces due to weighting on the use of forces. By reducing unnecessary dynamical use of forces, a considerable reduction in the total movement length for actuators were obtained, which will reduce actuator wear. Furthermore, the peak range was also reduced, which will reduce transients in the power supply. Reduced dynamical use of forces will also give more stable thruster commands, thus increasing the comfort of the crew.

Autotuning of the controller is possible with a genetic algorithm if the performance index has the so far mentioned behaviour, but the tuning time is considerably large. If the performance index also generates a convex function of the controller parameters, a rule-based algorithm will perform well and outperform the genetic algorithm in tuning time. The convexity property was fulfilled in all cases tested here and it is expected that it will be fulfilled for all cases were the performance is weighted between use of forces and pose deviation, since these are two conflicting objectives that must be balanced.

The simulation results also showed that with using the suggested performance index and an autotuning algorithm, performance improvements are possible for different vessel operational conditions (VOCs). The performance improvements either result in reduced deviation in pose or reduced use of forces for the actual VOC. The magnitude of the performance improvements depends on the performance index and the tuning of the performance index function. Also, the suggested performance index does have some parameters to tune, but they are logical to set, and by storing adapted information from each autotuning, the behaviour of the performance index will improve by learning.

A practical drawback with the proposed hybrid controller and the training is that adaptive schemes needs information richness to have something to adapt (learn) from, which in some cases involve setpoint changes. Such setpoint changes will naturally lower the comfort for the crew onboard the vessel. However, the need for training would decrease gradually with the amount of knowledge adapted. Another possibility is training with unmanned vessels in safe areas.

Further work

As always when looking into a problem, several other problems arise which leads to a heap of further work. Some of the future work involves:

- **Performance assessment:**
  The performance result will never be better than the weakest link, which was the performance index, and further study should be undertaken.
  - The κ_{min} offset and the κ_{max} factor showed to have good effects, and further study should be made on the use of these factors.
  - The performance weighting was defined as a single index and other definitions might be interesting. For instance, in the discussion it was mentioned that with $\lambda = 0.5$ and adjusting κ_{min} and κ_{max}, the performance could be balanced at all pose performance levels. With a single weighting factor used as index in the look-up table it is not possible to store such data, and extension of the definition might improve the controller function.
  - In the performance index, reduction of dynamical use of force is considered, and another focus is positioning inside power limitations. An interesting approach would be to use the same performance index basis (levels and punishment) for such a focus, or in a combination of power constraints and reduction of dynamical use of forces.
- **The rule-based algorithm:**
  The rule-based algorithm showed to perform well, but improvements are possible. For instance, some improvements of the rule-based algorithm which could reduce the number of iterations was mentioned in the discussion.
- **Adaptive controller bandwidth:**
  In the discussion, an adaptive controller bandwidth function, where controller parameters changes as a function of violating deviation limits is suggested. It would be interesting to see the effect of such a function, and how much the dynamical use of forces increase.

- **Combination with adaptive observer:**
  In the simulation tests performed for the controller parameter optimization (training), only a model with fixed model parameters was considered. An interesting approach would be to combine the proposed controller with the hybrid controller proposed in [34], alternatively with the adaptive observer mentioned in [9].

- **The VOC definition:**
  There can be other terms in the VOC space that has not been considered here which affect the performance of the controller. An example is the assumption that the draught is proportional (in the longitudinal of the vessel, i.e., along the x-axis) with different trim conditions. This is not true, so an implementation to investigate through further work could be to make δ as a function of draught and trim. Another example is that different wave directions will have different effects, so the sea state term in the environmental variable ε should might be extended to also be a function of the wave direction.

- **Other aspects:**

  - Instead of using bandwidth-tuning of a PID-controller, it could be interesting to automate the proposed tuning method in [29].
  - Use neural networks (NN) to estimate the $\boldsymbol{\mu}$ term in equation (2.18) in the unified model introduced in [17, 47]. The NN needs information to learn, and since the $\boldsymbol{\mu}$ term can be considered as 0 when there is no disturbance, one of the drawbacks with the NN is relaxed, and an estimation of the $\boldsymbol{\mu}$ term will be very similar to the NN predictor proposed in [58].
  - Using fuzzy logic, in selecting controller parameters according to the VOC. By using fuzzy logic a smoother switching between different conditions could be obtained.

Bibliography

[1] J. Van Amerongen. Adaptive Steering of Ships - A Model Reference Approach. *Automatica*, 20(1):3–14, 1984.

[2] C. Bohn and D. P. Atherton. An Analysis Package Comparing PID Anti-Windup Strategies. *IEEE Control Systems*, 15(2):34–40, 1995.

[3] D. Bray. *Dynamic Positioning.* OPL, Oilfield Publications Limited, second edition, 2003.

[4] M. Breivik. *Nonlinear Maneuvering Control of Underactuated Ships.* Master thesis, Norwegian University of Science and Tecnology, Trondheim, Norway, 2003.

[5] Y. Cao and T. Lee. Maneuvering of Surface Vessel using a Fuzzy Logic Controller. *Journal of Ship Research*, 47(2):101–130, 2003.

[6] Y. Cao, T. Lee, D. L. Garret, and J. F. Chappell. The Applications of a Time-Domain Fuzzy Logic Controller for Dynamic Positioning of Floating Structures. *Tamkang Journal of Science and Engineering*, 5(3):137–150, 2002.

[7] R. R. Costa, L. Hsu, A. K. Imai, and P. Kokotović. Lyapunov-Based Adaptive Control of MIMO Systems. *Automatica*, 39(7):1251–1257, 2003.

[8] R. R. Costa, L. Hsu, and F. Lizarralde. Lyapunov/Passivity-Based Adative Control of Relative Degree Two MIMO Systems with an Application to Visual Servoing. In *Proceedings of the 2006 American Control Conference*, Minneapolis, Minnesota USA, 2006.

[9] K. D. Do. Global Robust and Adaptive Output Feedback Dynmaic Positioning of Surface Ships. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Roma, Italy, 2007.

[10] H. K. Dong and J. H. Cho. Robust PID Controller Tuning Using Multiobjective Optimization Based on Clonal Selcetion of Immune Algoritm. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 50–56. Springer Berlin / Heidelberg, 2004.

[11] M. A. Duarte-Mermoud and R. A. Prieto. Performance Index for Quality Response of Dynamical Systems. *ISA Transactions*, 43(1):133–151, 2004.

[12] O. M. Faltinsen. *Sea Loads on Ships and Offshore Structures.* Cambridge Univeristy Press, first edition, 1990.

[13] Y. Fang, E. Zergerolu, M. S. de Queiroz, and D. M. Dawson. Global Output Feedback Control of Dynamical Positioned Surface Vessels: An Adaptive Control Approach. *Mechatronics*, 14(4):341–356, 2004.

[14] P. J. Fleming and R. C. Purshouse. Genetic Algorithms in Control Systems Engineering. Technical Report 789, Department of Automatic Control and Systems Engineering, University of Sheffield, 2001.

[15] T. I. Fossen. *Guidance and Control of Ocean Vehicles.* John Wiley & Sons, first edition, 1994.

[16] T. I. Fossen. *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles.* Marine Cybernetics, first edition, 2002.

[17] T. I. Fossen. A Nonlinear Unified State-Space Model for Ship Maneuvering and Control in a Seaway. *International Journal of Bifurcation and Chaos*, 15(9):2717–2746, 2005.

[18] T. I. Fossen and J. P. Strand. Passive Nonlinear Observer Design for Ships Using Lyapunov Methodes: Experimental Results with a Supply Vessel. *Automatica*, 35(1):3–16, 1999.

[19] O. G. Hvamb. A New Concept for Fuel Tight DP Control. In *Proceedings of the Dynamic Positioning Conference*, Houston, Texas, USA, 2001.

[20] I. Ihle. Dynamic Wave Force Model. Internal Report Rolls-Royce Marine AS dep. Control Aalesund, 2008.

[21] P. A. Ioannou and J. Sun. *Robust Adaptive Control.* Prentice Hall, 1996 (out of print in 2003), electronic copy at `http://www.rcf.usc.edu/\textasciitildeioannou/Robust\_Adaptive\_Control.htm`.

[22] M. Jamshidi, L. S. Coelho, R. Krohling, and P. J. Flemming. *Robust Control System with Genetic Algorithms.* CRC Press, first edition, 2002.

[23] M. Jelali. An Overview of Control Performance Assessment Technology and Industrial Applications. *Control Engineering Practice*, 14(5):441–466, 2006.

[24] J. M. J. Journee and W. W. Massie. *Offshore Hydromechanics.* Available online at `http://www.shipmotions.nl`, first edition, 2001.

[25] E. Kristiansen and O. Egeland. Frequency-Dependent Added Mass in Models for Controller Design for Wave Motion Damping. In *Proceedings of the IFAC Conference on Manouevering and Control of Marine Systems (MCMC'03)*, Girona, Spain, 2003.

[26] W. K. Lennon and K. M. Passino. Strategies for Genetic Adaptive Control. In *Proceedings of the Conference on Decision and Control*, San Diego, California, USA, 1997.

[27] K. P. Lindegaard. *Acceleration Feedback in Dynamic Positioning Systems.* Phd thesis, Norwegian University of Science and Tecnology, Trondheim, Norway, 2003.

[28] K. P. Lindegaard and T. I. Fossen. Fuel-Efficient Rudder and Propeller Control Allocation for Marine Craft: Experiment With a Model Ship. *IEEE Transactions on Control Systems Technology*, 11(6):850– 862, 2003.

[29] P. Martin and R. Katebi. Multivariable PID Tuning of Dynamic Ship Positioning Control Systems. *Journal of Marine Engineering and Technology*, 1(A7):11–24, 2005.

[30] E. W. McGookin, D. J. Murray-Smith, and T. I. Fossen. Ship Steering Control System Optimisation using Genetic Algorithms. *Control Engineering Practice*, 8(4):429–443, 2000.

[31] B. Mirkin and P. O. Gutman. Output Feedback Model Reference Adaptive Control for Multi-Input-Multi-Output Plants with State Delay. *System & Control Letters*, 54(10):961–972, 2005.

[32] M. Nakamura, T. Ohmachi, and T. Shima. Model Experiments on Dynamic Positioning System using Neural Network. In *Proceedings of The Thirteenth International Offshore and Polar Engineering Conference*, Honolulu, Hawaii USA, 2003.

[33] M. Negnevitsky. *Artificial Intelligence: A Guide to Intelligent Systems.* Pearson Education, second edition, 2002.

[34] T. D. Nguyen and S. T. Quek A. J. Sørensen. Design of Hybrid Controller for Dynamic Positioning from Calm to Extreme Sea Condition. *Automatica (Journal of IFAC)*, 43(5):768–785, 2007.

[35] 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.

[36] T. Perez, A. J. Sørensen, and M. Blanke. Marine Vessel Models in Changing Operational Conditions - A Tutorial. In *Proceedings of the*

16th IFAC Symposium on System Identification, Newcastle, Australia, 2006.

[37] T. Perez, A. J. Sørensen, and M. Blanke. Vessel Operational Conditions and Formal Specification of Ship Guidance Navigation and Control System Requirements. *Journal of Maritime Research*, 2006. Draft paper.

[38] W. G. Price and R. E. D. Bishop. *Probalistic Theory of Ships Dynamics.* Chapman and Hall, London, third edition, 1974.

[39] S. J. Qin and J. Yu. Multivariable Controller Performance Monitoring. In *International Symposium on Advanced Control of Chemical Processes*, Gramado, Brazil, 2006.

[40] G. N. Roberts, R. Sutton, A. Zirilli, and A. Tiano. Intelligent Ship Autopilots - A Historical Perspective. *Mechatronics*, 13(10):1091–1103, 2003.

[41] M. Sedighizadeh and A. Rezazadeh. Adaptive PID Controller Based on Reinforcement Learning for Wind Turbine Control. *Proceedings of World Academy of Science, Engineering and Tecnology*, 27(1):257–262, 2008.

[42] J. Selkäinaho. Tuning a Dynamic Positioning System. *Automatica*, 29(4):865–875, 1993.

[43] S. L. Shah, T. Yamamoto, and O. Yoshihiro. Design of a Multivariable Self-Tuning PID Controller with an Internal Model Structure. In *Proceedings of the Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Lake Louise, Alta, Canada, 2000.

[44] J. C. Shen. Fuzzy Neural Networks for Tuning PID Controller for Plants with Underdamped Response. *IEEE Transactions on Fuzzy Systems*, 9(2):333–342, 2001.

[45] J. C. Shen. New Tuning Method for PID Controller. In *Proceedings of the IEEE International Conference of Control Applications*, Mexico City, Mexico, 2001.

[46] R. Skjetne. *The Maneuvering Problem.* Phd thesis, Norwegian University of Science and Tecnology, Trondheim, Norway, 2005.

[47] Ø. N. Smogeli and T. I. Fossen. Nonlinear Time-Domain Strip Theory Formulation for Low-Speed Maneouvering and Station-Keeping. *Modelling, Identification and Control*, 25(4):201–221, 2004.

[48] A. Soltoggio. An Enhanced GA to Improve the Search Process Reliability in Tuning of Control Systems. In *Proceedings of the Genetic*

And Evolutionary Computation Conference, Washington DC, USA, 2005.

[49] K. J. Åstrøm and T. Hägglund. The Future of PID Control. *Control Engineering Practice*, 9(11):1163–1175, 2001.

[50] K. J. Åstrøm and B. Wittenmark. *Adaptive Control.* Addieson-Wesley Inc, second edition, 1995.

[51] K. Takao, T. Yamamoto, and T. Hinamoto. Memory-based IMC Tuning of PID Controllers for Nonlinear Systems. *IEEE Transactions on Electrical and Electronic Engineering*, 1(1):364–373, 2006.

[52] E. A. Tannuri, L. K. Kubota, and C. P. Pesce. Adaptive Techniques Applied to Offshore Dynamic Positioning Systems. In *XI DINAME - International Symposium on Dynamic Problems of Mechanics*, Ouro Preto, MG Brazil, 2005.

[53] G. Tao. *Adaptive Control Design and Analysis.* John Wiley & Sons, first edition, 2003.

[54] G. Torsetnes, J. Jouffroy, and T. I. Fossen. Nonlinear Dynamic Positioning of Ships with Gain-Scheduled Wave Filtering. In *Proceedings of the IEEE Conference on Decision and Control*, Paradise Island, Bahamas, 2004.

[55] A. Visioli. Modified Anti-Windup Scheme for PID Controllers. *IEE Proceeding - Control Theory Applications*, 150(1):49–54, 2003.

[56] X. S. Wang, Y. H. Cheng, and W. Sun. A Proposal of Adaptive PID Controller Based on Reinforcement Learning. *Journal of China Univeristy og Mining and Tecnology*, 17(1):40–44, 2007.

[57] S. R. Weller and G. C. Goodwin. Hysteresis Switching Adaptive Control of Linear Multivariable Systems. *Automatica*, 39(7):1360–1375, 1994.

[58] Z. H. Lee Y. Cao and Y. Lin. Application of an On-line Trainning Predictor/Controller to Dynamic Positioning of Floating Structures. *Tamkang Journal of Science and Engineering*, 4(3):141–154, 2001.

[59] T. Yamamoto and S. L. Shah. Design and Experimental Evaluation of a Multivariable Self-Tuning PID-Controller. *IEE Control Theory and Applications*, 151(5):645–652, 2004.

[60] W. Zuo. Multivariable Adaptive Control for a Space Station Using Genetic Algorithms. *IEE Control Theory Applications*, 142(2):81–87, 1995.

Appendix A

CD Contents

- The folder "Report" contains a PDF version of the report.
- The folder "Simulator program" contains Matlab files for running the simulations presented in Chapter 3.
- The folder "References" contains articles and conference papers referred to in this thesis.

Appendix B

Simulator User Guide

Simulator programs used in the simulations contain several Matlab script files, 1 Simulink model file, and in addition some functions. The Matlab script file "Autotuning.m" is the main file that run the test scenario in Section 5.2, where all plots are automatically generated. The vessel model with the nonlinear PID-controller and the passive observer is implemented in the Simulink model file. Calculation of the performance index is also implemented in Simulink, so the progress of the performance index and the subperformance indices can be monitored there. NB! The main file "Autotuning.m" must be run to initialize the parameters in the Simulink model file. The 2 autotuning/training functions are implemented in the script files "RB_Tuning.m" and "GA_Tuning.m" and they use the Simulink model file for running simulation of the vessel model for each iteration. Default tuning is rule-based tuning, and to select GA tuning, the variable "ga" must be set to 1. The function and dependency of training script files are:

- **RB_Tuning.m** is the file used for simulation of rule-based training, the file is calling the Simulink model file "supply.mod" which simulates the vessel model performing station keeping with the selected sea state condition and ocean current. The simulation is stopped after the optimum is found for each DOF.

- **GA_Tuning.m** is the file used for simulation of GA training, this file is calling the Simulink model file "supply.mod" which simulates the vessel model performing station keeping with the selected sea state condition and ocean current. The training is stopped after a predefined number of generations. The file "GA_ Tuning.m" uses the file "initpopNonlin.m" to generate a population with chromosomes, which uses the "rand.m" function in Matlab.