Nonlinear MPC for Motion Control and Thruster Allocation of Ships

Alexander Bärlund
Abstract

Critical automated maneuvers for ships typically require a redundant set of thrusters. The motion control system hierarchy is commonly separated into several layers using a high-level motion controller and a thruster allocation (TA) algorithm. This allows for a modular design of the software where the high-level controller can be designed without comprehensive information on the thrusters, while detailed issues such as input saturation and rate limits are handled by the TA. However, for a certain set of thruster configurations this decoupling may result in poor control performance due to the limited knowledge in the high-level controller about the physical limitations of the ship and the behavior of the TA.

This thesis investigates different approaches of improving the control performance, using nonlinear Model Predictive Control (MPC) as a foundation for the developed motion controllers due to its optimized solution and capability of satisfying constraints. First, a decoupled system is implemented and results are provided for two simple motion tasks showing problems related to the decoupling. Thereafter, two different approaches are taken to remedy the observed drawbacks. A nonlinear MPC controller is developed combining the motion controller and thruster allocation resulting in a more robust control system. Then, in order to keep the control system modularized, an investigation of possible ways to augment the decoupled system so as to achieve similar performance as the combined system is carried out. One proposed solution is a nonlinear MPC controller with time-varying constraints accounting for the current limitations of the thruster system. However, this did not always improve the control performance since the behavior of the TA still is unknown to the MPC controller.
Acknowledgments

First of all, I would like to thank my examiner Martin Enqvist at Linköping University and the people at ABB for providing me with the opportunity to write this thesis. All parties were agile making my initial tardiness a non issue.

Then, I would like to direct my deepest gratitude to my supervisors. First, Fredrik Ljungberg at Linköping University for answering my sometimes diluted questions and always providing feedback and inputs. Second, Jonas Linder and Hamid FeyzMahdavian at ABB Corporate Research for their never-ending support in all matters. A special thank you to Jonas Linder for reading my thesis more times than one single person should have to. Without all of you, this thesis would not have been possible.

Finally, I would like to acknowledge all the people who made my stay at Linköping University and ABB so enjoyable. In big or small, you have kept me going during these years.

Linköping, June 2019
Alexander Bärlund
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Purpose</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>Goals</td>
<td>3</td>
</tr>
<tr>
<td>1.4</td>
<td>Delimitation</td>
<td>3</td>
</tr>
<tr>
<td>1.5</td>
<td>Methodology</td>
<td>3</td>
</tr>
<tr>
<td>1.6</td>
<td>Outline</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Selected Topics in Marine Modeling and Control</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Ship Dynamics</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Kinematics</td>
<td>6</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Kinetics</td>
<td>7</td>
</tr>
<tr>
<td>2.1.3</td>
<td>Summary</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Thrusters</td>
<td>8</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Modeling</td>
<td>9</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Power and Efficiency</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Thruster Allocation</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>Model Predictive Control</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>Decoupled Thruster Allocation and Motion Control</td>
<td>15</td>
</tr>
<tr>
<td>3.1</td>
<td>Control Hierarchy</td>
<td>16</td>
</tr>
<tr>
<td>3.2</td>
<td>MPC Formulation</td>
<td>16</td>
</tr>
<tr>
<td>3.3</td>
<td>Implementation</td>
<td>17</td>
</tr>
<tr>
<td>3.4</td>
<td>Simulation Results and Discussion</td>
<td>17</td>
</tr>
<tr>
<td>3.5</td>
<td>Conclusions</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>Combined Thruster Allocation and Motion Control</td>
<td>23</td>
</tr>
<tr>
<td>4.1</td>
<td>MPC Formulation</td>
<td>24</td>
</tr>
<tr>
<td>4.2</td>
<td>Implementation</td>
<td>25</td>
</tr>
<tr>
<td>4.3</td>
<td>Simulation Results and Discussion</td>
<td>25</td>
</tr>
<tr>
<td>4.4</td>
<td>Conclusions</td>
<td>28</td>
</tr>
</tbody>
</table>
5 Augmented Decoupled Motion Control System 31
  5.1 Delimitation 32
  5.2 Motivation 32
  5.3 Time-Varying Input Constraints 32
  5.4 Simulation Results and Discussion 33
  5.5 Conclusions 38

6 Conclusions and Future Work 39
  6.1 Future Work 39

A Simulation Environment 43

Bibliography 47
## Notation

### Notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\eta$</td>
<td>Position and orientation of ship</td>
</tr>
<tr>
<td>$\eta_r$</td>
<td>Reference position and orientation</td>
</tr>
<tr>
<td>$\psi$</td>
<td>Heading of ship</td>
</tr>
<tr>
<td>$\nu$</td>
<td>Linear and angular velocities of ship</td>
</tr>
<tr>
<td>$\nu_r$</td>
<td>Reference velocities</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Generalized force (forces and moment)</td>
</tr>
<tr>
<td>$\tau_c$</td>
<td>Control forces</td>
</tr>
<tr>
<td>$\tau_c^d$</td>
<td>Desired control forces</td>
</tr>
<tr>
<td>$\tau_{env}$</td>
<td>Environmental forces</td>
</tr>
<tr>
<td>$J(\eta)$</td>
<td>Transformation matrix</td>
</tr>
<tr>
<td>$R(\psi)$</td>
<td>Rotation matrix</td>
</tr>
<tr>
<td>$M$</td>
<td>Mass matrix</td>
</tr>
<tr>
<td>$C$</td>
<td>Coriolis matrix</td>
</tr>
<tr>
<td>$D$</td>
<td>Damping matrix</td>
</tr>
<tr>
<td>$g(\eta)$</td>
<td>Restoring forces</td>
</tr>
<tr>
<td>{n}</td>
<td>Earth-fixed coordinate system</td>
</tr>
<tr>
<td>{b}</td>
<td>Body-fixed coordinate system</td>
</tr>
<tr>
<td>$u$</td>
<td>Control input to thrusters</td>
</tr>
<tr>
<td>$n$</td>
<td>Propeller angular velocity</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Thruster orientation</td>
</tr>
<tr>
<td>$T(\alpha)$</td>
<td>Thruster orientation matrix</td>
</tr>
<tr>
<td>$f(n)$</td>
<td>Thruster force function</td>
</tr>
<tr>
<td>$l_x$</td>
<td>Moment arm in $x_b$</td>
</tr>
<tr>
<td>$l_y$</td>
<td>Moment arm in $y_b$</td>
</tr>
<tr>
<td>$T_s$</td>
<td>Sample time</td>
</tr>
<tr>
<td>$T_c$</td>
<td>Time constant</td>
</tr>
<tr>
<td>$N$</td>
<td>Prediction horizon</td>
</tr>
<tr>
<td>$Q$</td>
<td>Penalty matrix stage cost</td>
</tr>
<tr>
<td>$R$</td>
<td>Penalty matrix final cost</td>
</tr>
</tbody>
</table>
### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOF</td>
<td>Degrees Of Freedom</td>
</tr>
<tr>
<td>MPC</td>
<td>(Nonlinear) Model Predictive Control</td>
</tr>
<tr>
<td>TA</td>
<td>Thruster Allocation</td>
</tr>
<tr>
<td>MCS</td>
<td>Motion Control System</td>
</tr>
<tr>
<td>OCP</td>
<td>Optimal Control Problem</td>
</tr>
<tr>
<td>RTI</td>
<td>Real-Time Iteration</td>
</tr>
<tr>
<td>DP</td>
<td>Dynamic Positioning</td>
</tr>
<tr>
<td>RPM</td>
<td>Revolutions Per Minute</td>
</tr>
<tr>
<td>RPS</td>
<td>Revolutions Per Second</td>
</tr>
<tr>
<td>QP</td>
<td>Quadratic Programming</td>
</tr>
<tr>
<td>SQP</td>
<td>Sequential Quadratic Programming</td>
</tr>
<tr>
<td>NED</td>
<td>North-East-Down (coordinate system)</td>
</tr>
</tbody>
</table>
1

Introduction

This is the master’s thesis Nonlinear MPC for Motion Control and Thruster Allocation of Ships performed at ABB Corporate Research in Västerås in collaboration with Linköping University.

1.1 Background

Automation of ships has been the aim of many engineers since the first autopilot system following the invention of the gyrocompass. One such application is Dynamic Positioning (DP). It is used to keep a ship or rig stationary in the horizontal plane, or moving at low constant speed, using only the available thrusters. This is a well established research area, first developed in the 1960s to keep the position of a drilling ship fixed, where other methods like anchors and jack up barges would fail due to significant water depth. The first DP-capable ship was the American drill ship Cuss 1 produced in 1961 (Fay, 1990). Later on in the 1970s significant improvements were made utilizing model based design (Breivik and Sand, 2009). Balchen et al. (1980) stated the goals of DP as: "A dp-system should be designed to keep the given vessel within specified position limits, with a minimum fuel consumption and with minimum wear and tear on the propulsion equipment". After the commercialization of satellite navigation systems in the 1990s, the development of DP technology increased in speed. In time, with further advances, the lines of what DP meant blurred as it came to involve low constant-speed path-following, for instance, for pipe laying.

DP-capable ships are typically over-actuated, meaning that they have more thrusters than what might be necessary for station-keeping and path-following. This allows for precise control and redundancy in case of failure of one or more thrusters. In favorable conditions this may render the control task trivial, reducing the prob-
lem of DP to compensation in the three degrees of freedom (DOF) of the horizontal plane using (three) independent PID controllers (Fossen, 2011).

By extending the workspace of traditional DP-applications, e.g. to include automated docking for tankers or low speed path-following of passenger ferries, a whole new set of ships need to be considered. These ships are typically not designed with the same kind of redundancy and may not be as agile as a DP-capable ship. This implies that the control system needs to utilize the available performance better, allowing for both a higher level of automation for a wide range of ships and increased performance of already DP-capable ships. The extended workspace in which the developed controllers operate will be called Low Speed Motion Control, extending traditional DP-applications. Recent advances in numerical optimal control together with an ever increasing level of computational power allows for real-time implementation of advanced controllers, such as Model Predictive Control (MPC) (Quirynen et al., 2015). This grants the investigation of near-optimal control of the ship and thrusters.

The algorithmic structure seen in Figure 1.1 is common for many automated ships. The focus in this thesis is the Motion Control System (MCS). The design of the MCS has historically been divided into several layers. First, a high-level motion controller calculates the total forces and moments that should be exerted on the ship. Secondly, a thruster allocation (TA) algorithm calculates a set-point in orientation and propeller speed for the individual thrusters in order to obtain the forces requested by the motion controller (Fossen, 2011). This type of decoupled design will be referred to as a decoupled MCS. Decoupling allows for a more flexible and modular design since the high-level controller may remain the same for ships with different actuator configurations while only the TA is updated with the new configuration. However, this might also be an important disadvantage. For instance, the decoupling of the algorithms impose that the high-level motion controller do not consider physical limitations of the ship and thrusters. This could pose a problem regarding the performance of the MCS.

![Diagram](oversikt.png)

**Figure 1.1:** Overview of a common system hierarchy for automated ships.
1.2 Purpose

The purpose of this master’s thesis has been to investigate how the performance of a decoupled MCS using a high-level MPC controller may be improved for low-speed motion control in the horizontal plane.

1.3 Goals

The goals of this thesis are implementing a high-level motion controller for a traditional decoupled MCS in order to evaluate the performance, establishing a benchmark controller combining the high-level motion controller and TA algorithm, and finally, to modify the decoupled system to perform near the same level as the combined controller.

1.4 Delimitation

The most prominent limitation of this thesis is that the workspace of the developed controllers is the horizontal plane. Further, only low speed motion control has been considered, i.e. control of a ship moving at speeds less than 2 m/s, in order to reduce the impact of nonlinear effects.

Measuring and estimating the position and velocities of a ship is crucial for accurate motion control. This task is made increasingly complicated by the induced motion caused by waves. Thus, during the design of a controller, all states have been presumed to be estimated by an existing algorithm or measured directly.

The work only includes development of MPC controllers, thus no other methods are explored. Furthermore, the algorithmic side of solving the (nonlinear) MPC problem has not been considered.

Finally, to reduce modeling and tuning efforts, one specific ship with a given thruster configuration has been considered.

1.5 Methodology

At first, a theoretical study of ship dynamics and dynamic positioning was performed in order to become familiar with the traditional methods, modeling and terminology used in the marine field. Also, theory on model predictive control and previous studies utilizing MPC in marine applications were reviewed.

Three different control structures were implemented and tested. To begin with, a high-level nonlinear MPC controller was developed and implemented in a decoupled MCS together with an existing (black-box) TA-algorithm. This configuration was extensively tested and provoked in order to find shortcomings arising due to the decoupling. Thereafter, the MPC controller was extended to also include thruster allocation, thus solving the combined problem of motion control and TA. This was seen as a benchmark controller for the decoupled MCS. Learning from previous results, efforts were made to improve the performance of the decoupled system by incorporating more information in the high-level MPC.
All MPC controllers have been developed using the MATLAB interface for the open-source ACADO toolkit (Houska et al., 2011) with the optimization problem solved by QPOASES (Ferreau et al., 2014). The online solver was code generated, see Quirynen et al. (2015), and implemented in a simulation environment built in Simulink using a MATLAB S-function, see Appendix A.

1.6 Outline

This thesis is organized as follows. Chapter 2 includes a brief theoretical introduction to the ship- and thruster modeling used when developing the controllers. Additionally, Chapter 2 cover basics on MPC and the thruster allocation problem. Chapter 3 includes the implementation and testing of a high-level nonlinear MPC motion controller used together with an existing implementation of a well-known low-speed TA algorithm. Thereafter, Chapter 4 comprises of an implementation and evaluation of a single nonlinear MPC controller solving the combined problem of position control and thruster allocation, which in theory should yield near optimal control of the thrusters. Chapter 5 contains an investigation on the possibility to augment the decoupled MCS from Chapter 3 with an information link to improve the performance. In Chapter 6 conclusions are made and areas for further development are proposed.
This chapter will briefly describe the models of marine craft and thrusters together with some introductory theory on MPC and thruster allocation used as basis in the motion control systems developed in this thesis. Readability is the focus and no models will be derived but only explained shortly. The interested reader is referred to Fossen (2011) and Perez (2005) for thorough derivations of all models, while Maciejowski (2002) provides in-depth theory on MPC. The chapter starts by introducing a general dynamic model of a ship in 6 DOF for completeness and continues to make motivated simplifications of it, most importantly reducing it to 3 DOF. Thereafter, the modeling of a general thruster configuration is presented followed by an introduction to the thruster allocation problem. Finally, the basic idea of MPC and the advantages and disadvantages of the method will be discussed.

2.1 Ship Dynamics

A ship exhibits motion in 6 DOF. The notation in this section, and subsequently in the thesis, will be adopted from SNAME (1950) and is given in Table 2.1. In Fossen (2011), a general model describing the dynamics is

\[ \dot{\eta} = J(\eta)\nu \] 
\[ M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau \]

where (2.1a) describes the kinematics of the ship while (2.1b) describes the kinetics. The matrix \( J(\eta) \in \mathbb{R}^{6\times6} \) is a transformation matrix while the matrices \( M \in \mathbb{R}^{6\times6}, C(\nu) \in \mathbb{R}^{6\times6}, \text{ and } D(\nu) \in \mathbb{R}^{6\times6} \) describe the mass-inertia, Coriolis forces and damping of the ship, respectively. The vector \( g(\eta) \in \mathbb{R}^{6} \) describes the restoring forces acting on the ship due to buoyancy and gravitation. On the right-hand side of (2.1b), \( \tau \in \mathbb{R}^{6} \) is a vector of forces and moments exerted on the ship by
Table 2.1: Notation used for the different DOF of a ship or submersible (SNAMEN, 1950).

<table>
<thead>
<tr>
<th>DOF</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>( \eta = [x;y;z;\phi;\theta;\psi]^T \in \mathbb{R}^6 )</td>
<td>Orientation of the vessel in 6 DOF given by position and Euler angles in an inertial frame.</td>
</tr>
<tr>
<td>6</td>
<td>( \nu = [u;v;w;p;q;r]^T \in \mathbb{R}^6 )</td>
<td>Linear- and angular velocities in a body-fixed coordinate system.</td>
</tr>
<tr>
<td>6</td>
<td>( \tau = [X;Y;Z;K;M;N]^T \in \mathbb{R}^6 )</td>
<td>Forces and moments decomposed in a body-fixed coordinate system.</td>
</tr>
<tr>
<td>3</td>
<td>( \eta = [x;y;\psi]^T \in \mathbb{R}^3 )</td>
<td>Orientation in the horizontal 3 DOF given by Cartesian position ((x, y)) and heading angle (\psi).</td>
</tr>
<tr>
<td>3</td>
<td>( \nu = [u;v;r]^T \in \mathbb{R}^3 )</td>
<td>Velocities of the vessel in a body-fixed coordinate system in surge ((u)), sway ((v)) and yaw ((r)), respectively.</td>
</tr>
<tr>
<td>3</td>
<td>( \tau = [X;Y;N]^T \in \mathbb{R}^3 )</td>
<td>Forces and moments in a body-fixed coordinate system in surge ((X)), sway ((Y)) and yaw ((N)) respectively.</td>
</tr>
</tbody>
</table>

Actuators and environmental disturbances (wind, waves, current), such that

\[
\tau = \tau_c + \tau_{\text{env}} \tag{2.2}
\]

where \(\tau_c\) are the control forces and moments while \(\tau_{\text{env}}\) are the ones originating from environmental disturbances.

However, this thesis regards controlling the position and heading of a ship on the ocean surface and only the horizontal motion will from here on be considered. The model given by (2.1) will therefore be reduced to 3-DOF below and motion in heave \(z\), pitch \(\theta\) and roll \(\phi\) will neither be monitored nor compensated for.

2.1.1 Kinematics
Two different coordinate systems will be used in the modeling, a body-fixed coordinate system attached to the ship and an Earth-fixed coordinate system. The body-fixed coordinate system, denoted by \(\{b\}\), is commonly attached to a ship by having the origin in the centre of gravity and with the \(x_b\)-axis pointing forward.
2.1 Ship Dynamics

(a) Relationship between the NED and body-fixed coordinate system, $\psi$ is the heading angle.

Figure 2.1: Visual description of the coordinate systems used.

Towards the bow, the $y_b$-axis pointing starboard and the $z_b$-axis pointing downwards (Fossen, 2011). The Earth-fixed coordinate system will be the Cartesian local tangent plane coordinate system NED (North-East-Down), denoted by $\{n\}$. The origin is fixed to a point on the surface of the Earth with the $x_n$-axis pointing north, $y_n$-axis pointing east and the $z_n$-axis pointing towards the center of the Earth. See Figure 2.1a for the relationship of the two coordinate systems seen from above. In this thesis, the NED-system is assumed to be inertial which is reasonable considering the low velocities involved in low speed motion control (Fossen, 2011).

The position and heading $\eta$ of the ship is measured in $\{n\}$ while velocities $\nu$ and forces $\tau$ will be decomposed in $\{b\}$, see Figure 2.1b. This is the reason for the transformation matrix in (2.1a). The relationship between them is purely geometric and (2.1a) reduces to

$$\dot{\eta} = R(\psi)\nu$$

in 3 DOF, where $R(\psi) \in \mathbb{R}^{3 \times 3}$ is a rotation matrix given by

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

2.1.2 Kinetics

Kinetics describe the motion of a body caused by forces and moments. A model of the kinetic motion for ships can be derived using rigid-body mechanics and
theory of hydrodynamics (Fossen, 2011). When considering motion in 3-DOF, (2.1b) together with (2.2) reduces to

\[ M \ddot{\nu} + C(\nu)\nu + D(\nu)\nu = \tau_c + \tau_{env} \]  

(2.5)

where \( M, C(\nu) \) and \( D(\nu) \) ∈ \( \mathbb{R}^{3\times3} \). For the purpose of controller design it is often convenient to work with linear models (Glad and Ljung, 2000). Under the assumption of low speed and due to the fact that the non-constant terms in \( C(\nu) \) and \( D(\nu) \) are quadratic in \( \nu \), (2.5) can be simplified (Fossen, 2011). This yields the linear dynamic equation

\[ M \ddot{\nu} + D\nu = \tau_c + \tau_{env} \]  

(2.6)

Assuming that the ship is symmetric in the \( x_bz_b \)-plane with the center of origin of \( \{b\} \) coinciding with the center line of the vessel, the matrices generally have the structure

\[
M = \begin{bmatrix}
\times & 0 & 0 \\
0 & \times & \times \\
0 & \times & \times
\end{bmatrix}
\]

\[
D = \begin{bmatrix}
\times & 0 & 0 \\
0 & \times & \times \\
0 & \times & \times
\end{bmatrix}
\]

such that the surge motion is decoupled from the sway- and yaw motion. The matrix elements in \( M \) stem from the mechanical properties of the ship such as mass and inertia, as well as from hydrodynamics describing the inertia of the water being pushed by the ship, while the elements in \( D \) are purely derived from hydrodynamics.

### 2.1.3 Summary

To summarize the content in the above sections, a simplified model for the dynamics of a ship in 3-DOF is found by combining (2.3) and (2.6) to be

\[ \dot{\eta} = R(\psi)\nu \]  

(2.7a)

\[ M \ddot{\nu} + D\nu = \tau_c + \tau_{env} \]  

(2.7b)

with states \([\eta^T \ \nu^T]^T\) and input \( \tau_c \). The model can be written as a state-space model by rearranging (2.7b) and multiplying both sides with the inverse of \( M \).

### 2.2 Thrusters

Marine vessels can be equipped with a range of different actuators depending on the intended use. These include propellers, water jets, sails and rudders to name a few (Molland et al., 2011). The purpose of the actuator is to produce a controlled force on the vessel to obtain the desired movement. In low speed motion control, a commonly used actuator is the azimuth thruster (Lewandowski, 2004). It comprises of a propeller mounted on a hub able to rotate (azimuth) freely in the horizontal plane. The modeling in this section assumes a quadratic relationship between thrust and control variable, thus making it general for most
propeller type actuators. However, in this thesis only azimuth thrusters will be considered.

2.2.1 Modeling
The control forces and moments $\tau_c$ created by a thruster are dependent on its location and orientation on the keel and on the force (thrust) produced. Thus, in the general case of $M$ thrusters, $\tau_c$ can be written as

$$\tau_c = h(\alpha, n)$$

where $\alpha \in \mathbb{R}^M$ is a vector of thruster angles and $n \in \mathbb{R}^K$ is a vector of propeller speeds. For low speed, $h$ typically takes the form (Fossen and Johansen, 2006)

$$\tau_c = T(\alpha)f(n) \quad (2.8)$$

where $f(n) \in \mathbb{R}^M$ is a vector of thrust magnitude for each thruster, and

$$T(\alpha) = [t_1, \ldots, t_N] \in \mathbb{R}^{n \times M}$$

describes the geometry of the thruster configuration. In $n = 3$ DOF, the columns of $T(\alpha)$ are given by

$$t_i(\alpha_i) = \begin{bmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \\ l_{x,i} \sin \alpha_i - l_{y,i} \cos(\alpha_i) \end{bmatrix}, \quad i = 1, \ldots, M \quad (2.9)$$

where $l_{x,i}$ and $l_{y,i}$ are the moment arms given in a body-fixed coordinate system and $\alpha_i$ describe the orientation, taken positive clock-wise from the $x_b$-axis, see Figure 2.2.

For low speed motion control, the thrust $f$ produced by a thruster is assumed to be proportional to the square of the rotational velocity of the propeller. More pre-

![Figure 2.2: Definition of moment arms and orientation for a ship equipped with a thruster.](image-url)
cisely, under bollard-pull condition (stationary vessel), a symmetrical propeller’s steady-state axial thrust $f_i$ of the $i^{th}$ thruster is given by

$$f_i = k_i n_i |n_i|, \quad i = 1, \ldots, M$$  \hspace{1cm} (2.10)

where $k_i$ is a constant and $n_i$ is the propeller angular velocity (Whitcomb and Yoerger, 1999). Subsequently, the thrust vector $f(n)$ in (2.8) can be written as

$$f(n) = \begin{bmatrix} f_1(n_1) \\ \vdots \\ f_M(n_M) \end{bmatrix} = K \begin{bmatrix} n_1 |n_1| \\ \vdots \\ n_M |n_M| \end{bmatrix}$$  \hspace{1cm} (2.11)

where $K \in \mathbb{R}^{M \times M}$ is a diagonal matrix with $[k_1, k_2, \ldots, k_N]$ on the diagonal. However, note that in general, the thrust produced depend on the fluid velocity around the propeller, which in turn relate to the velocity of the ship (Whitcomb and Yoerger, 1999).

### 2.2.2 Power and Efficiency

It has been stated that the thrust produced by a propeller under some assumptions is proportional to the square of the rotational velocity. This also holds true for the torque produced (Lewandowski, 2004). From mechanics it is known that the power $P$ required to produce the torque $T$ at the rotational speed $\omega$ follows as

$$P = T \omega$$

Thus, a common approximation for the power required to rotate a propeller is

$$P \propto n^3$$  \hspace{1cm} (2.12)

where $n$ is the rotational speed. Further, the above relation typically holds for open-water conditions, that is the propeller receives an undisturbed and uniform water flow. For podded azimuth thrusters this might not be the case when the propeller is running in reverse, pushing the pod it is mounted on, see Appendix A. Also, propellers are sometimes designed to be more efficient in one direction than the other. Thus, the efficiency of a thruster might be different depending on if it is reversing or not. Such a thruster is said to be asymmetric.

### 2.3 Thruster Allocation

Control allocation is a method often used in the aerospace and marine industry for managing actuator redundancy when designing control systems for overactuated dynamic systems (Johansen and Fossen, 2013). Then, the control system is separated in to a control law specifying the total control effort to be produced and a control allocation algorithm distributing it between the actuators. In the scope of this thesis, control allocation will be called thruster allocation, referring to the task of distributing a desired generalized force $\tau^d$ between the thrusters. Thus, the main goal of the TA is to realize $\tau^d$ at all times. However, due to the redundancy in actuation there is freedom in choosing how to distribute the forces, that is, choosing control input $u = (n, \alpha)$. The more thrusters the ship is equipped
2.3 Thruster Allocation

with, the more combinations of inputs can be used to produce \( \tau^d_c \). The problem of choosing the input \( u \) is naturally handled by formulating it as an optimization problem where the cost function typically involves minimization of fuel or power consumption, while the constraints consider thruster limitations and wear (Johansen and Fossen, 2013). Perhaps the simplest form of thruster allocation can be found by studying (2.8) and (2.11). In Fossen and Johansen (2006), the variable change \( v_i = n_i |n_i| \) with the unique inverse \( n_i = \text{sign}(v_i) \sqrt{|v_i|} \) is introduced. This results in the relation

\[
\tau^d_c = T(\alpha)Kv
\]

(2.13)

between the desired force by the motion controller \( \tau^d_c \) and the actuator controls \( n \) and \( \alpha \). Now, if \( \alpha \) is constant, i.e. the thrusters have fixed orientations, \( T(\alpha)K \) is constant. If no physical limitations are considered, the optimization problem can then be formulated as a weighted least square problem

\[
\min_v \quad vWv^T
\]

s.t. \( \tau^d_c - Hv = 0 \)

The solution is readily found by differentiating and setting to zero, being

\[
v = W^{-1}H^T(H^TW^{-1}H)^{-1}\tau^d_c
\]

(2.14)

If \( W = I \), then (2.14) reduces to the Moore-Penrose pseudo inverse.

However, when considering azimuth thrusters, \( \alpha \) is not constant. Furthermore, it is not guaranteed that the TA is able to fulfill the desired force \( \tau^d_c \) if it requires a force beyond the capability of the thrusters, e.g. due to saturation. A more general problem formulation is

\[
\min_{u,s} \quad p(\eta, \nu, u, s, t)
\]

s.t. \( \tau^d_c - h(\eta, \nu, u, t) = s \)

\( g(\eta, \nu, u, t) = 0 \)

(2.15a)

(2.15b)

(2.15c)

where \( p \) is some cost function of the states \( (\eta, \nu) \), inputs \( u = (n, \alpha) \), slack variables \( s \) and the time \( t \). The constraint (2.15b) represents the main priority of the thruster allocation but with the addition of \( s \) in case it is not feasible. The slack variable is usually weighted much higher in \( p \) than the other objectives to reflect this priority (Johansen and Fossen, 2013). For low speed, the function \( h \) is typically represented by the right hand side of (2.8). The constraint (2.15c) represents the physical limitations of the thruster system, such as saturation or power limits.

In general, the thruster allocation problem (2.15) is non-convex (Fossen and Johansen, 2006). This means that the optimization solver might end up stuck in some local minimum. For asymmetric thrusters that are designed for maximum efficiency in one direction while not being as efficient when running in reverse, this means that the thruster might end up stuck producing thrust in the non-optimal way. This problem can be resolved in different ways and a common ap-
Selected Topics in Marine Modeling and Control

approach is to have an exogenous algorithm evaluating if it is beneficial to rotate the thruster (Veksler et al., 2016).

2.4 Model Predictive Control

Model Predictive Control is an advanced control strategy commonly found in the process industry. Prominent features include the possibility to control multiple input multiple output systems, usage of an internal model to predict future states, and the ability to explicitly handle constraint on states and inputs, such as the voltage sent to a DC-motor or the fluid level in a tank (Maciejowski, 2002).

The difference between linear and nonlinear MPC is as the name suggests, that nonlinear MPC can handle nonlinear dynamics and constraints. In either case, the control input is calculated by solving a finite horizon open-loop optimal control problem (OCP) at each sampling interval. In continuous time, the OCP can be formulated as

$$\min_{u(t)} \int_{t_0}^{t_0+T} f(t, x(t), u(t), x_r(t), u_r(t)) dt + f_N(x(t_0 + T), x_r(t_0 + T))$$

(2.16a)

s.t. $x(t_0) = x_0$ \hspace{1cm} (2.16b)

$\dot{x}(t) = f(x(t), u(t)), \forall t \in [t_0, t_0 + T]$ \hspace{1cm} (2.16c)

$g(t, x(t), u(t)) \leq 0, \forall t \in [t_0, t_0 + T]$ \hspace{1cm} (2.16d)

where $x(t) \in \mathbb{R}^{n_x}$ are the states and $u(t) \in \mathbb{R}^{n_u}$ are the control inputs. The inputs to the OCP are the current state estimate $x_0$ and the reference trajectories $x_r(t)$ and $u_r(t), \forall t \in [t_0, t_0 + T]$. The objective of the controller is defined by the cost function (2.16a), here constructed by two terms, the stage cost $f(\cdot)$ and the final cost $f_N(\cdot)$. The function of the final cost is to estimate the cost of ending up in the final state $x(t_0 + T)$ while the stage cost represents the objective and the cost of obtaining it. A common way to construct the cost terms is to use a least squares objective function, weighting the difference between states and inputs and the respective reference (Glad and Ljung, 2000)

$$f(\cdot) = \|x(t) - x_r(t)\|_{Q_x}^2 + \|u(t) - u_r(t)\|_{Q_u}^2$$

(2.17a)

$$f_N(\cdot) = \|x(t) - x_r(t)\|_{R_x}^2$$

(2.17b)

The dynamic model used for prediction is given by (2.16c) while (2.16d) contains the constraints of the system. Design variables include the prediction horizon $T$, which determines how far into the future the controller will look, and the weight matrices $Q_x \geq 0, Q_u > 0$ and $R_x \geq 0$ weighting the entries in the cost function.

However, in order to solve (2.16) on a computer it must be discretized. There are a range of different methods for discretization of continuous systems, such as Euler sampling and Runge-Kutta methods. The discrete version of (2.16) using
(2.17) sampled with frequency $1/T_s$ is

$$
\min_{U_i} \sum_{k=0}^{N-1} \|x_{i+k} - x_{i+k}^\text{ref}\|^2_{Q_x} + \|u_{i+k} - u_{i+k}^\text{ref}\|^2_{Q_u} + \|x_{i+N} - x_{i+N}^\text{ref}\|^2_{R_x} 
$$

\begin{align*}
\text{s.t.} & \quad x_i = x_0 
\quad x_{i+k+1} = f_d(x_{i+k}, u_{i+k}), \quad k = 0, \ldots, N - 1 
\quad g_{d,i}(x_{i+k}, u_{i+k}) \leq 0, \quad k = 0, \ldots, N - 1
\end{align*}

where $N = T/T_s$. The solution that minimizes (2.18) at time $i$ is a trajectory of control inputs $U_i^* \in \mathbb{R}^{N \times n_u}$. This trajectory is calculated in open-loop, and to achieve feedback only the first element in $U_i^*$ is used. Then the system is propagated and the problem is solved again in the next sampling interval $i+1$. Since an optimization problem is solved at every sampling interval, MPC tends to be computationally heavy. The complexity of the problem increases with the number of states and inputs, as well as the prediction horizon $N$. Thus, there is a trade off between a long prediction horizon and a fast controller.

In the case of a linear model and constraints

$$
\begin{align*}
x_{k+1} &= Ax_k + Bu_k \\
F_x x_k &\leq b_x \\
F_u u_k &\leq b_u
\end{align*}
$$

it is possible to reformulate (2.18) as a Quadratic Programming (QP) problem (Maciejowski, 2002). The optimization problem is then convex, resulting in a globally optimal solution. Solving the nonlinear problem (2.18) is more difficult since the optimization problem in general becomes non-convex. One common approach is to linearize the system around some point, for instance the reference, and then formulate a standard QP. This allows for the solver to find a global solution to the approximated problem. Other methods, like Sequential Quadratic Programming (SQP), iteratively solves multiple QP problems approximating the nonlinear program until convergence (Nocedal and Wright, 2006). The ACADO toolkit will instead export a Real Time Iterations (RTI) scheme for the optimization aimed at providing an approximate but fast solution. The RTI scheme essentially works by linearizing the problem around the current state estimate and solving one QP in each iteration, thus making it only marginally slower than linear MPC (Gros et al., 2016).

In subsequent chapters, all MPC controllers will be formulated in continuous time with the dependence on $t$ for states and controls omitted for readability. Discretization of the problem is done by the ACADO toolkit where the user can choose to export both tailored explicit and implicit integration methods (Quirynen et al., 2015). Further, all controllers will use nonlinear MPC, thus the word nonlinear will be omitted for brevity.
Ships capable of low-speed maneuvering and DP are typically configured with enough actuators to make them overactuated, meaning that there are more than or equally many control inputs $n_u$ as DOF $n$ (Fossen, 2011). This allows for precise control of the ship in the horizontal plane and, in case of redundancy, maintained controllability in case of failure of one or more actuators (Veksler et al., 2016). This chapter will begin by describing the control hierarchy commonly found in the marine industry, followed by an implementation of this hierarchy using nonlinear model predictive control. Finally, some results and drawbacks of the method will be presented.

Figure 3.1: Overview of a decoupled MCS.
3.1 Control Hierarchy

The traditional MCS for ships is divided into multiple layers, see Figure 3.1 (Fossen, 2011). First, a high-level motion controller takes the measured, or estimated, state of the ship \((\eta, \nu)\) and a reference signal \((\eta_r, \nu_r)\) as inputs. The reference signal may be a set-point, path or trajectory. The task is then to calculate a desired generalized force input \(\tau_c^d\) to the ship in order to follow the reference. Several different algorithms have been developed for this purpose, ranging from decoupled PID to linear quadratic and nonlinear controllers (Fossen, 2011). In Section 3.2, a MPC controller will be presented. Second, a thruster allocation algorithm is then tasked with realizing the desired force \(\tau_c^d\) by controlling the thruster speeds and orientations \(u = (n, a)\), see Section 2.3.

3.2 MPC Formulation

In this thesis, a MPC was developed as the high-level motion controller. The dynamic model (2.7) is used for prediction, although with the notation \(\tau_c^d\) instead of \(\tau_c\) since it does not affect the ship directly, see Figure 3.1 and Section 3.1.

In order to try to fulfill the physical limitations on the ship and thruster system the input \(\tau_c^d\) will be subject to both amplitude- and rate constraints. The rate constraints are implemented by augmenting the model with

\[
\dot{\tau}_d = T_c(-\tau_c^d + \tau_a) \quad \text{(3.1a)}
\]

\[
\dot{\tau}_a = u_T \quad \text{(3.1b)}
\]

where \(T_c \in \mathbb{R}^{n \times n}\) is a diagonal matrix of time constants and \(u_T\) now is the control variable. The rate constraints on \(\tau_c^d\) can thus be formulated as magnitude constraints on \(u_T\).

By combining the dynamic model (2.7) with (3.1) together with the constraints, a nonlinear continuous-time OCP was formulated as

\[
\min_{u_T} \int_0^{T_s N} \left( \|\eta - \eta_r\|_{Q_\eta}^2 + \|\nu - \nu_r\|_{Q_\nu}^2 + \|\tau_c^d\|_{Q_{\tau}}^2 + \|u_T\|_{Q_{u_T}}^2 \right) dt + \text{final cost} \quad \text{(3.2a)}
\]

s.t.

\[
\dot{\eta} = R(\psi)\nu \quad \text{(3.2b)}
\]

\[
M \dot{\nu} + D\nu = \tau_c^d \quad \text{(3.2c)}
\]

\[
\dot{\tau}_d = T_c(-\tau_c^d + \tau_a) \quad \text{(3.2d)}
\]

\[
\dot{\tau}_a = u_T \quad \text{(3.2e)}
\]

\[
\tau_c^d \leq \tau_c^d \leq \bar{\tau}_c^d \quad \text{(3.2f)}
\]

\[
u_T \leq u_T \leq \bar{u}_T \quad \text{(3.2g)}
\]

where the final cost contains similar terms as the stage cost. The cost function (3.2a) allows a time-varying reference on \(\eta\) and \(\nu\) while the magnitude of the generalized force and its rate is penalized. The weight matrices \(Q_x\) may be changed
depending on the control objective. For instance, if it is desired to follow a path, the heading \( \psi \) and surge velocity \( u \) could be weighted more while weighting \( \tau^d_c \) and \((x, y)\) less. The constraints (3.2b)-(3.2c) define the dynamic model while (3.2f) and (3.2g) constrain the magnitude and rate of the generalized force \( \tau^d_c \).

### 3.3 Implementation

The control performance was tested in simulation using a model of a supply ship equipped with two Azipod\textsuperscript{®} thrusters, one in the stern and one in the bow, both mounted on the center line of the vessel. See Appendix A for a description of the simulation environment, the Azipod thrusters and important model parameters. The high-level MPC was connected to an existing implementation of a well-known low-speed TA algorithm incorporating one variant of (2.15). Further, the TA also models asymmetric thrusters. Thus, while trying to instantaneously fulfill the desired force, it will if necessary simultaneously rotate the thrusters so that the propellers may run with positive RPM in order to minimize the power consumption.

By studying Figure A.1, it is evident that the ship can produce thrust in all three DOF since the thrusters may rotate. However, it might be momentarily unable to produce thrust in a certain direction depending on the orientation of the thrusters. This makes the bounds on the constraints (3.2f) and (3.2g) non-trivial to decide. In the simpler case, that the thrusters were stationary (\( \alpha \) constant), one could simply use (2.8) together with the thruster specifications to calculate the constraints in amplitude, and rate, on \( \tau^d_c \). Thus, they were treated more as tuning parameters than actual physical limitations.

### 3.4 Simulation Results and Discussion

The decoupled MCS was tested for a wide range of maneuvers. Out of these, the result for two different cases are presented, both highlighting key features. In both cases, the velocity reference \( \nu_r \) was set to zero while a step in position reference \( \eta_r \) occurred at \( t = 8 \). The differences between the cases are the position reference and the initial thruster orientations, see Table 3.1. Tuning parameters were kept constant, chosen rather aggressive in order to provoke problems. The environmental disturbances \( \tau_{env} \) were set to zero in all simulations in order to highlight the behavior due to the decoupled nature of the MCS. The results are summarized in Figure 3.2, visualizing the motion of the ships for both tests.

In the first case, the ship is commanded to move to a position in front of it and

<table>
<thead>
<tr>
<th>Case</th>
<th>Initial position ( [0, 0, 0]^T )</th>
<th>Initial thruster angles ( [0, 0]^T )</th>
<th>Reference position ( [80, 0, 0]^T )</th>
<th>Time ( T_s )</th>
<th>Number ( N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( [0, 0, 0]^T )</td>
<td>( [0, 0]^T )</td>
<td>( [80, 0, 0]^T )</td>
<td>1 s</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>( [0, 0, 0]^T )</td>
<td>( \left[ \frac{\pi}{2}, \frac{\pi}{2} \right]^T )</td>
<td>( [0, -50, 0]^T )</td>
<td>1 s</td>
<td>80</td>
</tr>
</tbody>
</table>
stop. The thrusters are initially pointing straight forward, thus in the direction of travel. Results are shown in Figure 3.3. By studying Figure 3.3a, it can be seen that the ship overshoots the target position by close to 10 meters, which can also be seen in Figure 3.2a. This is clearly not desirable, since it might mean that the ship will crash into a dock or other object. In simulation though, the result is that the ship has to back up slightly, being the reason for the negative surge velocity $u$ in Figure 3.3b. A hint to the reason for this behavior can be seen by studying Figure 3.3c and Figure 3.3d. At $t = 40$, both thrusters are pointing straight forward (Figure 3.3d). At the same time the high-level MPC begins to command a negative surge force (Figure 3.3c). The TA then begins to run the thrusters in reverse. However, a few seconds later it is necessary to rotate the thrusters in order to fulfill $\tau_c^d$. This means that the ship will be momentarily unable to generate any surge force for the duration it takes to rotate them. Since the MPC does not account for this, there will be a significant mismatch between $\tau_c$ and $\tau_c^d$, seen in Figure 3.3c for $t \in [50, 70]$ seconds. The length of the duration is coupled to the rotation rate of the thrusters. In fact, a similar mismatch of forces can also be seen for $t \in [100, 120]$ seconds where the MPC again wants to slow the ship down. At this point, the ship has to rotate the thrusters once more, resulting in a slight undershoot, see Figure 3.3a.

One could argue that this behavior is due to tuning errors, and that would be partially correct. As mentioned, the tuning is in this case rather aggressive, making the ship travel close to 2 m/s in surge, bordering the limit when linear damping dominates the nonlinear effects (Fossen, 2011). The overshoot would be smaller, or even removed entirely, for instance, if the elements in $Q_{\nu}$, in (3.2a) were higher, forcing the velocity down. This would make the ship travel slower, thus making a delay in force not as significant. However, the mismatch would still be there due to the inability to model the rotation delay in the MPC. The test case and tuning

Figure 3.2: Illustration of the ship’s movement in the horizontal plane.
3.4 Simulation Results and Discussion

3.4.1 Results and Discussion

(a) Position $\eta$ and reference position $\eta_r$. Note the overshoot in $x$.

(b) Velocity $\nu$ and reference velocity $\nu_r$. The velocity becomes negative since the ship overshot the reference position.

(c) Force exerted on ship $\tau_c$ and control output $\tau_d^i$ from the high-level MPC. Note the mismatch between them caused by the rotation of the thrusters.

(d) Output controls $n$ and $\alpha$ from the TA. The angles are wrapped to $\pm 180^\circ$.

Figure 3.3: Results from the first test case for the decoupled MCS.

were chosen to reveal this. Further, it is not beneficial from an implementation standpoint to have a control system so dependent on proper tuning due to the usually limited time for commissioning on a ship.

In the second case, the ship has to move in a negative $y_b$-direction. Detailed results are collected in Figure 3.4. The thrusters are from the beginning pointing straight in positive $y_b$, see Figure 3.4d. At the time of the position reference the MPC immediately desires a large negative sway force, since it does not have
any knowledge on the orientation and physical limitations of the thruster system, see Figure 3.4c. This results in a significant mismatch between the desired force and the actual force exerted on the ship. Trying to immediately fulfill $\tau^d_c$ the TA starts to run the thrusters in reverse while simultaneously rotating the thrusters to minimize the power consumption, see Figure 3.4d. When the thrusters are pointing to the stern and bow respectively, at $t = 20$, there is no point in running the thrusters in reverse since no sway force can be obtained. Thus, the TA starts to slow the propellers down while continuing to rotate them to point towards port. However, the propellers have not yet slowed down enough, and this will cause the ship to move in positive $y_b$-direction again, away from the target. This can be seen in Figure 3.4a showing the position of the ship and in Figure 3.4b displaying the velocity. This is clearly not desired considering energy consumption. Although, as opposed to the first scenario there will be no overshoot this time since the MPC sees no need to slow down the ship due to the lower speed and greater drag of the ship in the water.

For both cases, one remedy could be to constrain the rate of $\tau^d_c$ to such a low degree that it will account for the delay which is the turn around time of the thrusters. However, this would significantly withhold the available performance of the thruster system.

### 3.5 Conclusions

As mentioned in the beginning of the thesis, the decoupling offers advantages when it comes to implementation, offering a modular software design. However, as seen here, the performance of the decoupled MCS may deteriorate due to an inability in the high-level controller to know and predict what forces can be generated. Two cases were shown when the behavior might not be as desired. The common theme is that the TA algorithm has to rotate the thrusters in order to meet the force requested by the MPC. This is marked by a transition from positive to negative force, in either DOF. Since the MPC has no knowledge of the position and orientation of the thrusters, it does not know that the ship is momentarily unable to produce the requested force, resulting in a time lag between $\tau^d_c$ and $\tau_c$ relating to the rotation time of the thruster. Other cases where one might anticipate the performance of the decoupled MCS to deteriorate are when the reference trajectory requires the ship to produce forces in more than one DOF simultaneously, for instance, sway force and yaw torque. This can happen since the MPC does not model any coupling between the forces and moment. These drawbacks are highlighted by the chosen vessel and thruster configuration, see Figure A.1. While it is not underactuated according to the definition in Fossen (2011), it is not able to independently generate forces and moments in all DOF. Consequently, it might be momentarily unable to produce a force in some direction. In fact, what the dynamic model in the MPC actually models is a ship equipped with enough actuators to produce force and moment in all DOF independently. However, this could incur a significant cost, especially for ships not typically required to use all actuators.
3.5 Conclusions

(a) Position $\eta$ and reference position $\eta_r$. Note that the ship slightly moves back toward $y = 0$ at $t = 40s$.

(b) Velocity $v$ and reference velocity $v_r$. Note that the ship starts moving in negative sway but then stops again.

(c) Force exerted on ship $\tau_c$ and control output $\tau^d$ from the high-level MPC. The mismatch is due to the rotation of the thrusters.

(d) Output controls $n$ and $\alpha$ from the TA. Thrusters are from the start pointing towards starboard. The angles are wrapped to $\pm 180^\circ$.

Figure 3.4: Results from the second test case for the decoupled MCS.
Both the high-level controller and the thruster allocation in Chapter 3 were formulated as separate optimization problems. Deviating from the traditional structure, this chapter will instead present an implementation of a single MPC combining the work of both algorithms. Thus, the structure of the motion control system will be as shown in Figure 4.1, referred to as combined MCS. The chapter starts with the formulation of the MPC followed by the implementation details. Thereafter, simulation results of the same scenarios as in Chapter 3 are provided. Lastly, a discussion on the performance and strengths of the control system with regards to the decoupled system is given.

**Figure 4.1:** Overview of the combined MCS.
4.1 MPC Formulation

The combined problem of motion control and thruster allocation is modeled by combining (2.7) and (2.8) according to

\[ \dot{\eta} = R(\psi)\nu \]  
\[ M\dot{\nu} + D\nu = T(\alpha)f(n) + \tau_{env} \]

with the states \((\eta, \nu)\) and control inputs \((n, \alpha)\). However, the absolute value function in \(f(n)\), see (2.11), was troublesome to implement in ACADO. This can be remedied in a number of ways. One approach, similar to what is done in Veksler et al. (2016), is to define two inputs \((n^+, n^-)\) instead of \(n\), where \(n^+\) denote positive propeller RPM and \(n^-\) the negative, while both are limited to positive values. With this approach the thrust \(f_i(n_i)\) produced by the \(i\)th thruster, see (2.10), may instead be written as

\[ f_i(n_i) = k_i((n_i^+)^2 - (n_i^-)^2), \quad n_i^+, n_i^- \geq 0 \]  

This approach also allows one to prioritize positive RPM by weighting \(n^-\) higher than \(n^+\) in the cost function, modeling asymmetric thrusters. However, one would also need to add the nonlinear constraint

\[ n_i^+ n_i^- = 0 \]

This constraint was found numerically difficult to achieve, also with slack added. Another method is to simply approximate the absolute value with a continuous function that is differentiable, e.g. by fitting a polynomial or by using

\[ |x| \approx \sqrt{x^2 + \epsilon} \]  

where \(\epsilon\) is a small scalar, see Figure 4.2.

![Figure 4.2: The difference between the absolute value function and the approximation given by (4.4) for \(\epsilon = 0.01\).](image)
Thus, $f_i(n_i)$ is approximated by

$$\tilde{f}_i(n_i) = kn_i\sqrt{n_i^2 + \epsilon} \quad (4.5)$$

Both inputs ($n, \alpha$) are subject to physical constraints. The propeller speeds $n$ are both limited in magnitude and rate while the thruster angles $\alpha$ are only limited in rate. The rate constraints can be implemented by adding extra states, see Section 3.2, omitted here for brevity. Combining (4.1) and (4.5) with the constraints, the continuous-time nonlinear optimization problem was formulated as

$$\min_{\dot{n}, \dot{\alpha}} \int_0^{T,N} \left( \|\eta - \eta_r\|_{Q_{\eta}}^2 + \|\nu - \nu_r\|_{Q_{\nu}}^2 + \|n\|_{Q_n}^2 + \|\dot{\alpha}\|_{Q_{\dot{\alpha}}}^2 + \|\dot{n}\|_{Q_{\dot{n}}}^2 \right) dt + \text{final cost} \quad (4.6a)$$

s.t. \hspace{1cm} \dot{\eta} = R(\psi)\nu \quad (4.6b)

$$M \dot{\nu} + D\nu = T(\alpha)\tilde{f}(n) \quad (4.6c)$$

$n \leq n \leq \bar{n} \quad (4.6d)$

$|\dot{\alpha}| \leq \bar{\dot{\alpha}} \quad (4.6e)$

$\dot{n} \leq \bar{n} \leq \bar{n} \quad (4.6f)$

where the final cost contains similar terms as the stage cost. Similar to (3.2) the cost function allows for a time-varying reference on $\eta$ and $\nu$. According to (2.12) the power is proportional to the cube of $n$. Minimizing power would be desirable, in (4.6a) however the square of $n$ is penalized in order to formulate the cost function as in (2.17). The last two terms in the stage cost penalize the rate of the control inputs. This is desirable to reduce fast changes in the inputs, reducing wear and tear on the propulsion equipment. The constraints (4.6b) and (4.6c) defines the dynamic model while (4.6d)-(4.6f) constrain the control inputs.

### 4.2 Implementation

The combined MCS was tested in the same environment as mentioned in Section 3.3, see Appendix A. The limits on the rate constraints (4.6e) and (4.6f) were chosen according to specifications on the thrusters, see Table A.2. The upper bound on (4.6d) were chosen according to specifications as well, however, the lower bound was chosen to a fraction of the specified lower bound, thus reducing the force achievable when running in reverse. This was done to model the asymmetry of the thrusters, making it more beneficial to turn the thruster around when needing to create an opposite force.

### 4.3 Simulation Results and Discussion

Here, results for the combined MCS will be shown for the two cases presented in Section 3.4 to allow a natural comparison of the systems. The prediction horizon and sampling time were chosen to be the same as for the high-level MPC
in Section 3.4 for a fair comparison. The worst case execution time of the MPC constituting the combined MCS for both test cases was around 0.5 seconds on a laptop with an Intel i7 processor running at 2.9 GHz with 16 GB of RAM.

The resulting motion in the horizontal plane can be seen in Figure 4.3. More detailed information for the first case is found in Figure 4.4. Figure 4.4a shows the position of the ship over time. The ship reaches the target position without overshoot. The velocities are found in Figure 4.4b. Here, it is worth noting that the ship reaches similar speeds as for the decoupled MCS seen in Figure 3.3b. In Figure 4.4c, the corresponding forces are found. Instead of plotting \( \tau_c \) and \( \tau_d \) as in Section 3.4, \( \tau_c \) has been plotted against the thruster model (2.8), where \( n \) and \( \alpha \) are the outputs from the combined MCS. Since an almost identical model is used in (4.6c) this may be interpreted as an internal desired force in comparison to Figure 3.3c. They follow each other closely with only a slight mismatch due to unmodeled thruster dynamics and rudder effects. That is, when the AZIPOD thruster is in a 90 degree angle to the velocity of the ship significant drag will be obtained, resulting in the negative force seen at \( t = 40 \), see Appendix A. The key to how the combined MCS manages this trajectory is found in Figure 4.4d. When the step occurs, the MCS starts creating forward thrust using both thrusters. After a while, at \( t = 28 \), it begins to rotate thruster 2, while simultaneously decreasing \( n_2 \) in order to not create too much yaw moment. When the thruster is beginning to point in a useful direction, towards the stern, it accelerates the propeller again to brake the ship. Meanwhile, the other thruster reverses slightly to help reduce the speed and correct for yaw- and sway movement.

The results of the second test case are similar, see Figure 4.5. The position of the ship converges to the reference point without overshooting it, see Figure 4.5a. As for the first test case, \( \tau_c \) closely follows the calculated force by the combined MCS since it knows the orientations and limitations of the thrusters, see Figure 4.5c.
**Figure 4.4:** Results from the first test case for the combined MCS.

It is interesting to compare Figure 4.5b with Figure 3.4b. Instead of reversing the thrusters in order to quickly accelerate towards the target point, the ship will here first start accelerating some 15 seconds later when the thrusters have been rotated to allow for positive RPM. This is seen in Figure 4.5d showing the output from the MCS. When the step enters, it will start rotating the thrusters in the opposite direction and increase the propeller speeds in such a way that significant thrust is produced first when $\alpha_1, \alpha_2 < 0$, that is when both thrusters are pointing towards port.
4.4 Conclusions

It appears that the combined MCS offers significant improvements in control performance compared to the decoupled MCS from Chapter 3. Since the MPC here has full knowledge on the state and limitations of the thrusters, it is able to coordinate them more efficiently, also with the predicted future in mind. Thus, it can account for the delay caused by the rotation time of the thrusters when planning the motion. This makes it more robust to different tuning and aggressive maneuvers.

If this is the optimal behavior with respect to the objective is hard to tell. Convergence of the solver is not guaranteed and care should be taken to ensure that
it does not get stuck in local minima as mentioned in Section 2.3. Future tests should be made using for instance an SQP solver to see if the behavior is different. However, when using the RTI scheme the test result and execution time on the test computer indicate that this controller can be run in real-time on today’s hardware.
In Chapter 3, a decoupled MCS was implemented and tested. It was argued that the decoupled nature of the system has advantages when it comes to development and implementation, but also observed that it has some drawbacks with regards to performance and robustness. Later, in Chapter 4, a new type of motion control system was developed combining the TA and high-level MPC controller into a single algorithm, yielding theoretically the optimal control performance. This chapter will instead investigate the possibility to augment the decoupled MCS developed in Chapter 3 with an information link between the TA and MPC, referred to as augmented MCS. See Figure 5.1 for an illustration. The goal is to achieve similar performance and robustness as the combined MCS while keeping the benefits of the decoupling.

![Figure 5.1: Proposed structure of the augmented MCS using an information link.](image-url)
5.1 Delimitation

To begin with, some limitations of the proposed structure were made. First, in order to keep the benefits of the decoupled system, namely that the high-level controller should be implementable on a variety of ships with only a difference in tuning and model parameters, no thruster configuration specific information could exist in the high-level MPC controller. This means that the MPC may not know the number of thrusters or their position on the hull. Second, since the TA algorithm was seen as a black-box, no changes to how it operates could be made. This would also give the possibility to augment the system of already existing ships, without having to reprogram the TA. See Figure 5.1 for an illustration of the proposed system structure. Further, the implementation should be based on physical limitations and not depend on an excess of ad-hoc tuning.

5.2 Motivation

From the simulation results and the related discussion in Chapter 3 it is apparent that the control performance deteriorates due to the mismatch between $\tau_{dc}^d$ and the actual control force exerted on the ship. That is, the high-level MPC does not know the physical limitations of the ship in the current moment in time. Meanwhile, the MPC constituting the combined MCS has full knowledge of the thruster system. This difference can be further visualized by plotting the trajectories calculated at each time point by the two alternative MPCs, see Figure 5.2. In Figure 5.2a, the trajectories from the decoupled system are shown. The same mismatch that can be seen in Figure 3.3c is here seen as the difference between the green and blue-crossed lines. From $t = 60$, the new trajectories calculated by the high-level MPC are significantly different from the initial one, shifted right and down. More precisely, the MPC demands a stronger negative surge force since the initial desired force was not fulfilled. In Figure 5.2b, the result from the combined MCS is shown. First, it takes a few iterations for the RTI scheme to converge. Further on, around $t = 40$ the trajectories stay around zero for a while. This may be since the MPC constituting the combined MCS knows that it cannot produce any large negative surge force immediately at that time. The mismatch between the calculated force and actual force is, as mentioned in Section 4.3, due to unmodeled rudder forces. The outcome is that the MPC does not have to use as much negative surge force as initially planned to stop the ship.

5.3 Time-Varying Input Constraints

To try to replicate the behavior of the combined MCS, an attempt was made to provide information to the high-level MPC controller in the decoupled MCS about the physical limitations of the thrusters. That is, translating the limitations on $(n, \alpha)$ to limitations on $\tau_{dc}^d$. This computation was carried out in an exogenous algorithm roughly outlined in Algorithm 1. It takes as input the current $(n, \alpha)$, either provided by the TA or measured directly. Together with the limits on rate and magnitude it is then possible to calculate all possible combinations of $(n, \alpha)$ in the near future. Thereafter, using (2.8) as a mapping between $(n, \alpha)$ and $\tau_{dc}^d$, the
maximum and minimum achievable force in each DOF is calculated. By iterating this sequence $N$ times, the upper- and lower bound on the magnitude of $\tau^d_c$ for each step in the prediction horizon is found. This algorithm is then executed with the same frequency as the high-level MPC controller. In a continuous formulation, this would result in the time-varying constraint

$$\tau^d_c(t) \leq \tau^u_c(t) \leq \bar{\tau}^d_c(t)$$

replacing the constant box constraint (3.2f). In Figure 5.3, an example of the constraints over a prediction horizon of $N = 80$ seconds are shown, calculated at a time when $n = 0$ and $\alpha = 0$.

### 5.4 Simulation Results and Discussion

Here, results for the augmented MCS constructing the time varying constraints are presented for the first test case from Section 3.4, see Figure 5.4. The parameters for the high-level MPC and TA are here identical to those used in for the decoupled MCS in Chapter 3.

In Figure 5.4a, the output of the high-level MPC in the augmented MCS is plotted against the actual force exerted on the vessel. Focusing on the time interval $t \in [40, 60]$ seconds, where the ship starts braking, there is still a visible mismatch. However, in contrast to the results seen in Figure 3.3c, $\tau^d_c$ is here close to zero while $\tau_c$ is more negative, not the other way around. Thus, the constraints kept the high-level MPC from requiring a force not currently achievable. The mis-

**Figure 5.2:** Calculated input trajectories (gray) by the MPC from Chapter 3 and Chapter 4 respectively for the first testcase. Note that the blue crossed line represent the current output by the respective MPC’s while $\tau_c$ is the actual control force exerted on the ship.

(a) Input trajectories from the high-level MPC from Chapter 3.

(b) Input trajectories calculated using (2.8) from the combined MCS in Chapter 4.
Algorithm 1 Algorithm for finding constraints on generalized force given measurements on \((n, \alpha)\)

```plaintext
function CALCULATE_CONSTRAINTS(n, \alpha)
    initialize N, \tilde{n}, \tilde{\alpha}
    for i = 1 to N do
        // MAKE_VECTOR(a, b) returns a uniformly sampled vector in
        // the range \([a - b, a + b]\)
        n_v ← MAKE_VECTOR(n, i \hat{n})
        \alpha_v ← MAKE_VECTOR(\alpha, i \hat{\alpha})

        // MAKE_GRID(a, b) returns the Cartesian grid \((aa, bb)\) from
        // the points specified by vectors a and b
        [nn, \alpha\alpha] ← MAKE_GRID(n_v, \alpha_v)

        // THRUSTER_MODEL(nn, \alpha\alpha) calculates the generalized
        // force for all points in nn, \alpha\alpha
        \tau ← THRUSTER_MODEL(nn, \alpha\alpha)

        // Find maximum and minimum force at given time in
        // prediction horizon
        \tilde{\tau}(i) ← MAX(\tau)
        \tau(i) ← MIN(\tau)
    end for
    return \tilde{\tau}, \tau
end function
```

match here is instead due to unmodeled thruster dynamics and rudder effects. The rudder effects indeed aid the control system for this test case, reducing the speed of the ship. However, as seen in Figure 5.4b the ship still does a significant overshoot of the target point. To see the reason for this, the input trajectories calculated by the high-level MPC are shown in Figure 5.5. In the beginning the MPC seems to converge to a solution close to the initial red trajectory. By the constraints it is a feasible trajectory since it satisfies the cone-like constraints seen in Figure 5.3. The motion is thus planned according to this trajectory. However, it is quite similar to the initial trajectory in Figure 5.2a, which was observed to be infeasible with respect to the physical limitations and behavior of the TA. Around \(t = 40\), the time-varying constraints restrict the initial trajectory, forcing the output close to zero since the thrusters are rotating. New input trajectories thus have to be calculated, significantly different from the previous ones. This delays the planned braking of the ship by about the amount of time it takes to rotate the thrusters, causing it to overshoot the target. Thus the behavior is not much different from that of the decoupled MCS without the time-varying constraints.

Using the time-varying constraints, the high-level MPC has information on what forces may be generated in the very near future. This could be useful for the
5.4 Simulation Results and Discussion

Figure 5.3: Constraints on $\tau_c^d$ over the prediction horizon $k = 1, \ldots, N$ for $T_s = 1$, calculated at a time $t$ when $n = 0$ and $\alpha = 0$. The asymmetry in surge is due to the limit on negative propeller RPM being set lower than the positive. Note that the constraints are not zero at $k = 0$, only small.

(a) Desired force $\tau_c^d$ and actual force $\tau_c$ exerted on the vessel. The mismatch is due to unmodeled thruster dynamics and rudder effects.

(b) The ships movement in the horizontal plane, note the overshoot of the target position (upper bold outline).

Figure 5.4: Results for the first testcase for the augmented MCS.

second test case. However, the problem here is that further on in the calculated trajectory, the constraints are simply box constraints. They only resemble the possible forces that could be achieved in the future but do not say anything about
what trajectories are possible to obtain, that is, how the control force at time $k + 1$ has to relate to the control at time $k$. As for the high-level MPC in the decoupled MCS, this is only limited by the (constant) rate constraints. Thus, the initially planned input trajectory is similar to the one generated with only box constraints, see Figure 5.2a. A number of different variations of the constraints were tried in order to remedy this, without significant improvement.

In order to test if the decoupled MCS would perform better by, in some way, constraining the high-level MPC to only output physically feasible force trajectories, an experiment was carried out. The combined MCS in Chapter 4 was set as the high-level controller together with the black-box TA. The output from the combined MCS, $n$ and $\alpha$, were transformed to a desired force $\tau_d^c$ using (2.8) and then sent to the TA, see Figure 5.6. Simulation results are shown in Figure 5.7. As in Figure 4.4c, Figure 5.7a shows the output from the combined MCS transformed to a desired force plotted against the actual force exerted on the vessel. Although the desired force is close to zero for a while around $t = 40$, allowing time to rotate the thrusters, there is a clearly visible mismatch in surge force at around $t = 60$, when the high-level combined MCS wants to brake the ship. The resulting motion of the ship is seen in Figure 5.7b, showing a significant overshoot of the target point. Some explanation is given by Figure 5.8 showing the output of the TA. The TA starts rotating the thrusters roughly 10 seconds later than the stand-alone combined MCS did in Figure 4.4d.
5.4 Simulation Results and Discussion

Figure 5.6: Experimental setup using the combined MCS as a high-level motion controller together with the TA.

(a) Force exerted on ship $\tau_c$ and calculated force to be exerted on the ship using (2.8). Note the mismatch between them in surge.

(b) The ship's movement in the horizontal plane, note the overshoot of the target position (upper bold outline).

Figure 5.7: Results for the first testcase using the combined MCS as the high-level controller together with the black-box TA, see Figure 5.6.

This result shows that even though the high-level motion controller is able to plan a physically feasible trajectory, the performance might still deteriorate for the decoupled system. The problem is, compared to using only the combined MCS as in Chapter 4, that the TA does not know about the future. Thus, in this case, it can only rotate a thruster when the high-level controller actually demands a negative surge force, instead of rotating one thruster in advance to be prepared to brake. It does not matter for how long the desired force is close to zero, respecting the physical limitations, if the TA does not anticipate the eventual negative force. Due to this, the delay in force as a result of the rotation of the thrusters will still be present. This indicates that when using MPC as the high-level controller in a decoupled MCS, significant gains in performance and robustness could be obtained if there is a stronger coupling between the high-level controller and TA.
Figure 5.8: Output controls $n$ and $\alpha$ from the TA. Note that the angles are wrapped to $\pm180^\circ$.

5.5 Conclusions

To conclude the chapter, the results indicate that it is difficult to obtain a better performing and more robust motion control system by augmenting the decoupled system using only physical limitations. They signify that the high-level MPC would either have to model the behavior of the TA, possibly by formulating more advanced constraints, or information would have to be sent both ways. The high-level MPC could then be provided with knowledge on what is possible at the moment, as with the time-varying constraints, while the TA could get additional information from the high-level MPC. This could allow the TA to be better prepared for fulfilling the current desired force.
Conclusions and Future Work

Several implementations of a control system for low speed motion control have been investigated. Extensive testing with different motion tasks has been carried out, with results presented for two test cases showcasing important behaviors. Problems for the decoupled MCS were observed in Chapter 3, provoked by aggressive tuning, owing to the unmodeled physical limitations and unknown behavior of the TA. This resulted in overshoot of the target point and unnecessary movement. The combined MCS from Chapter 4 arguably proved superior to the decoupled system while still being implementable in real-time. In Chapter 5, an investigation aimed at improving the decoupled system was carried out using an information link from the TA to the high-level controller. The results indicated that it is not enough to model the physical limitations of the thruster system. To achieve similar performance as the combined MCS it was argued that the high-level MPC would have to know about the behavior of the TA, or that the TA gets some additional information.

6.1 Future Work

In this thesis the investigation aimed at improving the performance of the decoupled MCS has focused on changing the high-level controller. Considering the results obtained in Chapter 5, it would be interesting to conduct a similar investigation aimed more toward changing the TA instead.

Another area of future work is to implement a reference generator. This could remedy the problems found regarding the decoupled MCS by providing the high-level MPC with a feasible reference trajectory, possibly regarding the physical limitations and behavior of the TA. In fact, this would be beneficial for all presented implementations of the MCS. The tuning parameters, and thus the behav-
ior of the control system, are currently quite sensitive to the choice of target point since the square of the position error is weighted in the cost function. Moreover, it is likely that the reference generator would be able to plan the motion over time, for instance given a target point. Then, the whole predictive power of the MPC framework could be used if the reference generator also provides the MPC controller with the future references over the prediction horizon.

The workspace of the developed motion control systems have been low speed motion control. Thus, a linear kinetic model describing the ship was argued to be sufficient. Also, the force produced by the thrusters are dependent on the fluid velocity, where the quadratic relationship between thrust and RPM generally only holds close to zero speed. Further, as observed and discussed in previous chapters, unmodeled rudder forces already make a noticeable impact. Future development could include more accurate ship and thruster models in order to allow higher speeds, or an investigation of when the modeling error results in a significant deterioration of the control performance.

In all simulations, the environmental disturbances $\tau_{\text{env}}$ were set to zero, including forces from wind, current and waves. This is in general not the case in reality. Also, compensating for the environmental forces is crucial for achieving a well functioning dynamic positioning system. The disturbances may be roughly divided into high- and low-frequency components. The high frequency components are typically discarded by a wave filter while the low frequency components can be handled by integral action in the controller (Fossen, 2011). Further work should investigate applying these methods in order to compensate for the disturbances and make the motion control more robust in real situations.
Appendix
Simulation Environment

Evaluation of the different motion control systems was done in a simulation environment built in Simulink. The environment was based on the supply vessel model available in the MSS toolbox (Fossen and Perez, 2004), built on the nonlinear model (2.1) in 6 DOF. Also, an existing implementation of a well-known thruster allocation algorithm was used. In Figure A.1 the ship is illustrated from above and in Table A.1 some important parameters for the vessel are found.

Further, the environment included a dynamic thruster model describing both propeller and rudder forces created by two Azipod® thrusters, one mounted in the stern and the other in the bow, see Figure A.1 for an illustration. The AZIPOD thruster is a type of azimuthing thruster. It consists of a propeller mounted on a pod housing an electric engine. The whole pod is then attached to the ship via a shaft that may rotate freely, see Figure A.2. Due to the shape of the pod it will not only exert force due to rotation of the propeller, but it will also have rudder effects at non zero speed, removing the need for a conventional rudder. Furthermore, due to the design of the propeller and hydrodynamic effects, the AZIPOD unit is generally more efficient when the propeller is pulling the pod, rather than pushing it. See Table A.2 for some important parameters used for the AZIPOD thrusters in simulation.
Table A.1: Important parameters for the supply ship model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length $l$</td>
<td>82.80 m</td>
</tr>
<tr>
<td>Beam $b$</td>
<td>19.20 m</td>
</tr>
<tr>
<td>Draft</td>
<td>6.00 m</td>
</tr>
<tr>
<td>Displacement</td>
<td>$6.3622 \times 10^6$ kg</td>
</tr>
</tbody>
</table>

Table A.2: Important parameters for the thruster models.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Thruster 1</th>
<th>Thruster 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>l_x</td>
<td>$, $</td>
</tr>
<tr>
<td>Turn around time</td>
<td>50 s</td>
<td>50 s</td>
</tr>
<tr>
<td>Maximum propeller speed</td>
<td>±2 RPS</td>
<td>±2 RPS</td>
</tr>
<tr>
<td>Maximum propeller acceleration</td>
<td>±0.08 RPS/s</td>
<td>±0.08 RPS/s</td>
</tr>
</tbody>
</table>

Figure A.1: Supply ship model used in simulation.
Figure A.2: Large Azipod® thrusters, note the shape above and below the podded section similar to a rudder. Image courtesy of ABB.


SNAME. Nomenclature for treating the motion of a submerged body through a fluid. Technical and research bulletin 5(1), Society of Naval Architects and Marine Engineers, 1950.
