



NTNU – Trondheim
Norwegian University of
Science and Technology

Optimal Thrust Allocation for Dynamic Positioning Systems using Mixed Integer Linear Programming

Mads Erdal

Master of Science in Cybernetics and Robotics

Submission date: June 2015

Supervisor: Lars Imsland, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

PROJECT DESCRIPTION SHEET**Name of the candidate:** Mads Erdal**Thesis title (Norwegian):****Thesis title (English):** Optimal Thrust Allocation for Dynamic Positioning Systems using Mixed Integer Linear Programming**Background**

In Dynamic Positioning systems, Thrust Allocation (TA) is the component responsible to specify set-points for the actuators in order to track a desired generalized force, which is the set of surge, sway and yaw demanded by the control system.

While TA is generally solved by means of non-linear (NL) solvers additional logical/operational constraints cannot be taken directly into account in the standard NL formulation. Typical constraints arising by operational limitation are for instance the forbidden sectors.

Project goal: The thesis should find a good trade-off between actuator model vs optimization problem complexity.

1. The actuator force image can be described using polytopic regions (azimuth thruster [circle] force domain can be approximated by an hexagon for example)
2. The MILP problem formulated using the polytopic regions, will provide a general solution for the operational/behavioral logic [Demand -> Forces].

Work description

1. Recast actuator force image into a finite number of polytopic regions
2. Combine the polytopic regions with MILP, consider using big-M technique
3. Extend the MILP formulation to include a piecewise linear formulation of force response
4. Perform a simulation of the thrust allocation algorithms
5. Discuss and conclude your work and suggest future work where you see that your findings could be useful

Start date: January, 2015**Due date:** 26 June, 2015**Supervisor:** Lars Imsland**Co-advisor(s):** Vincenzo Calabro, Kongsberg Maritime

Trondheim, __ January 2015 _____

**Lars Imsland**

Supervisor

AddressSem Sælandsvei 5
NO-7491 Trondheim**Org.no.** 974 767 880E-mail: postmottak@itk.ntnu.no
<http://www.itk.ntnu.no>**Location**O.S. Bragstads plass 2D
NO-7034 Trondheim**Phone**

+ 47 73 59 43 76

Fax

+ 47 73 59 45 99

Phone: + 47 47 23 19 49

1 Abstract

Thrust allocation is the part in a dynamic positioning system that is responsible for giving set-points to the different actuators on a vessel in order to track a desired generalized force, which is the set of surge, sway and yaw demanded by the control system. While thrust allocation generally is solved by means of non-linear solvers, additional logical and operational constraints cannot be taken directly into account in the standard non-linear formulation. The goal of this thesis was to explore the possibilities of using Mixed Integer Linear Programming (MILP) for thrust allocation and finding a good trade off between model accuracy and problem complexity.

This thesis shows that mixed integer programming, both MILP and Mixed Integer Quadratic Programming (MIQP) are powerful tools in optimization, and by using piecewise linear approximations of non-linear functions anything can be included in the formulation. However, using piecewise linear approximations tend to greatly increase the complexity of the problem, because the approximation often must consist of many lines to be accurate enough and to counteract the effect that the solver will tend to choose solutions in the vertices of the approximation. Using logical constraints is a powerful tool for introducing forbidden sectors, and could also be used as a state variable for e.g. whether an actuator is in forward or reverse mode.

In this thesis actuators has been modelled using MILP formulations, and an LP, a QP, several MILP and a MIQP formulation of the thrust allocation problem has been presented. The thesis suggests a method to include forbidden sectors in the feasible thrust region of an azimuth thruster using big-M technique, only introducing 2 equations and 2 variables to the problem formulation. This method can also be used to model the thrust region of a propeller-rudder pair. Power minimization has been included in the MILP formulation by approximation the thrust-power relationship using a piecewise linear approximation. Asymmetry in tunnel thrusters has also been added to the problem formulation. Simulations are done in MATLAB on a model of a ship to demonstrate the differences in response between the formulations. It is suggested that if the functions in the cost function can be approximated as quadratic functions, MIQP should be used, because approximating them by use of piecewise linear approximations increases the complexity to the extent that the MIQP formulation is quicker to solve.

2 Sammendrag

Thrust allokering, eller kontrolltildeling er den delen av et dynamisk posisjoneringssystem som er ansvarlig for å gi settpunkter til de forskjellige aktuatorene på et skip, slik at skipet kan følge en gitt generalisert kraft bestående av surge sway og yaw, gitt av kontrollsystemet. Kontrolltildelingsproblemet løses generelt ved bruk av ulineære løserer, men logiske og operasjonelle begrensninger kan ikke inkluderes direkte i en standard ulineær problemformulering. Målet med denne masteroppgaven var å utforske mulighetene bruk av Mixed Integer Linear Programming (MILP) gir i forhold til å finne et godt kompromiss mellom nøyaktighet og kompleksitet av kontrolltildelingsproblemet.

I denne oppgaven vises det at både MILP og Mixed Integer Quadratic Programming (MIQP) er kraftige verktøy innen optimalisering, og at de aller fleste ulineære funksjoner kan inkluderes i problemformuleringen ved hjelp av stykkevis lineære tilnærminger. En ulempe med stykkevis lineære tilnærminger er at de øker kompleksiteten til problemet kraftig, fordi tilnærmingen ofte må bestå av veldig mange linjer skal de bli gode nok tilnærminger av det ulineære uttrykket. En annen ulempe er at løsningen har en tendens til å ende opp i hjørnene mellom to linjer i tilnærmingen, hvilket må motvirkes med å bruke flere linjer i tilnærmingen. Logiske begrensninger er et godt verktøy i forbindelse med forbudte sektorer, og kan også bukes som tilstandsvariable, for eksempel for å bestemme om en propell skal gå fremover eller i revers.

I denne oppgaven har aktuatorer blitt modellert ved hjelp av MILP teknikker, og både en LP, en QP, en MIQP og flere MILP formuleringer av kontrolltildelingsproblemet blir presentert. Rapporten foreslår en metode for å inkludere forbudte sektorer i det gyldige området til en azimuth thruster ved bruk av big-M metoden. Dette tilfører kun 2 ligninger og to variable til problemformuleringen. Denne metoden kan også brukes til å modellere en propell med ror. Ved å inkludere forholdet mellom thrust og kraft som en stykkevis lineær tilnærming kan kraft minimeres in MILP formuleringen. Asymmetriske tunnel thruster er modellert og inkludert i problemformuleringen. Det er gjort simuleringer i MATLAB på en modell av et skip for å vise forskjellene i respons ved bruk av forskjellige formuleringer og hvordan den endres når det inkluderes mere kompleksitet. Det foreslås i denne oppgaven at dersom funksjonene i kostfunksjonen kan tilnærmes som kvadratiske funksjoner bør MIQP bukes fremfor MILP. Det er fordi bruken av stykkevis lineære funksjoner i kostfunksjonen øker kompleksiteten av problemet såpass mye at MIQP problemet er raskere å løse enn MILP problemet.

3 Acknowledgements

I would like to thank my main supervisor at NTNU, Professor Lars Imsland for his excellent help and guidance throughout my master thesis. I would also like to thank my supervisor at Kongsberg Maritime, Ph.D. Vincenzo Calabro who proposed the problem to me and who has provided help and guidance through extensive e-mail correspondences.

I would also like to thank Professor Roger Skjetne for his help regarding the model of CyberShip Enterprise 1 and Ph.D. candidate Kristian Gaustad Hanssen for giving me an introduction to the YALMIP modelling language and later helping me with YALMIP related issues along the way.

Last but not least I would like to thank my family and friends, especially my parents for everything they have done and provided for me and their continuous support, and my grandmother for introducing me to mathematics at an early age.

Mads Erdal

Contents

1	Abstract	iii
2	Sammendrag	iv
3	Acknowledgements	v
4	Introduction	1
5	Dynamic Positioning	2
5.1	Propulsion and thruster system	2
5.1.1	Main propellers and rudders	3
5.1.2	Azimuth thrusters	3
5.1.3	Tunnel thrusters	4
5.2	Control system	5
5.3	Thrust allocation in a DP system	7
6	Thrust Allocation	8
6.1	Optimization formulations	9
6.1.1	Linear Programming	9
6.1.2	Quadratic Programming	10
6.1.3	Mixed Integer Linear Programming	10
6.1.4	Mixed Integer Quadratic Programming	12
6.2	Slack variables	12
6.3	Minimizing thrust	13
6.4	Minimizing power	14
6.4.1	Power constraints	15
6.5	Minimizing fuel consumption	16
7	Modelling constraints	17
7.1	Polygon Approximation	17
7.1.1	Polygon Approximation error	17
7.1.2	Calculating the polygon	18
7.2	Forbidden sectors	21
7.2.1	Formulating forbidden zones	21
7.3	Asynchronous tunnel thruster	27

8	Problem formulations	29
8.1	LP formulation	29
8.2	QP formulation	30
8.3	MILP formulation	30
8.4	MIQP formulation	32
9	Implementation	33
9.1	CyberShip Enterprise 1	33
9.2	Polygon Approximation	34
9.3	Explicit constraints on thrust	35
9.4	Forbidden sectors	36
9.5	Main propeller and rudder	38
9.6	Tunnel thruster	40
9.7	Piecewise linear approximation of power	41
10	Simulations and results	44
10.1	Linear Program	44
10.2	Quadratic Program	50
10.3	Mixed Integer Linear Program	54
	10.3.1 Synchronous tunnel thruster	54
	10.3.2 Asynchronous tunnel thruster	57
	10.3.3 Power approximation	58
10.4	Mixed Integer Quadratic Program	62
11	Discussion	66
11.1	Modelling	66
11.2	Simulations	66
12	Conclusion	69
12.1	Future work	70
13	Appendix	73
13.1	Appendix A - Additional plots	73
	13.1.1 LP	73
	13.1.2 QP	74
	13.1.3 MILP synchronous tunnel thruster	75
	13.1.4 MILP asynchronous tunnel thruster	76
	13.1.5 MILP power minimization	77
	13.1.6 MIQP	77

List of Figures

1	Illustration photo of a main propeller with rudder, courtesy of Becker Marine Systems and Kongsberg Maritime	3
2	Illustration photo of an azimuth thruster, courtesy of Thrustmaster.net	4
3	Illustration photo of a tunnel thruster, courtesy of Brunvoll AS	5
4	x vs x^2	11
5	The thrust approximated by an N -sided polygon. Here, $N = 8$	18
6	A plot of 4 polytopic regions constraining a variable x	22
7	The relaxed mixed integer model using $M = 50$	23
8	The relaxed mixed integer model of an optimal big- M model, using the convex hull of the original constraints	24
9	Plot of a forbidden sector constructed of two half circles	25
10	A figure of the thruster configuration on CyberShip Enterprise 1 (CSE1)	33
11	Plot of the polygon approximation using $N = 16$ and $ u_{max} = 1$	35
12	A plot of a thrusters feasible area with 3 forbidden sectors	38
13	A plot of the feasible set of a main propeller-rudder pair, where the rudder can move 45° in both directions	39
14	A plot of the piecewise-linear approximations of $p_{T_i} = T_i ^{3/2}$ using n lines in the approximation	42
15	Plots of the allocated thrust from an LP formulation for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment	46
16	Calculated τ_X for the allocated thrust when simulating an LP formulation for all possible demanded τ_c with 1.0 increments	47
17	Calculated τ_Y for the allocated thrust when simulating an LP formulation for all possible demanded τ_c with 1.0 increments	48
18	Calculated τ_N for the allocated thrust when simulating an LP formulation for all possible demanded τ_c with 1.0 increments	49
19	Plots of the allocated thrust from a QP formulation for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment	51
20	Calculated τ_X for the allocated thrust when simulating a QP formulation for all possible demanded τ_c with 1.0 increments	52
21	Calculated τ_Y for the allocated thrust when simulating a QP formulation for all possible demanded τ_c with 1.0 increments	53

22	Plots of the allocated thrust from a MILP formulation for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment	55
23	Calculated τ_X for the allocated thrust when simulating a MILP formulation, including forbidden sectors on the azimuth thrusters prohibiting flushing, for all possible demanded τ_c with 1.0 increments	56
24	Calculated τ_Y for the allocated thrust when simulating a MILP formulation, including forbidden sectors on the azimuth thrusters prohibiting flushing, for all possible demanded τ_c with 1.0 increments	57
25	Plots of the allocated thrust from a MILP formulation minimizing power for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment	59
26	Calculated τ_Y for the allocated thrust when simulating a MILP formulation minimizing power, including forbidden sectors on the azimuth thrusters prohibiting flushing, for all possible demanded τ_c with 1.0 increments	60
27	A plot of the time needed to solve the thrust allocation problem with the complete MILP formulation	61
28	Plots of the allocated thrust from a MIQP formulation for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment	63
29	Calculated τ_X for the allocated thrust when simulating a MIQP formulation minimizing thrust, including forbidden sectors on the azimuth thrusters prohibiting flushing, for all possible demanded τ_c with 1.0 increments	64
30	A plot of the time needed to solve the thrust allocation problem with the MIQP formulation	65
31	Plot of the allocated thrust for the tunnel thruster in an LP formulation	73
32	Plot of the allocated thrust for the tunnel thruster in a QP formulation	74
33	Calculated τ_N for the allocated thrust from a QP formulation when simulating for all possible demanded τ_c with 1.0 increments	75
34	Plot of the allocated thrust for the tunnel thruster in a MILP formulation	76
35	Calculated τ_Y for the allocated thrust from a MILP formulation when simulating for all possible demanded τ_c with 1.0 increments	77

36	Calculated τ_N for the allocated thrust from a MILP formulation when simulating for all possible demanded τ_c with 1.0 increments	78
37	Plots of the allocated thrust from a MILP formulation for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment	79
38	Plot of the allocated thrust for the tunnel thruster in a MILP formulation	80
39	Calculated τ_X for the allocated thrust from a MILP formulation when simulating for all possible demanded τ_c with 1.0 increments	81
40	Calculated τ_N for the allocated thrust from a MILP formulation when simulating for all possible demanded τ_c with 1.0 increments	82
41	Plot of the allocated thrust for the tunnel thruster in a MILP formulation minimizing power	83
42	Calculated τ_N for the allocated thrust from a MILP formulation minimizing power when simulating for all possible demanded τ_c with 1.0 increments	84
43	Calculated τ_N for the allocated thrust from a MILP formulation minimizing power when simulating for all possible demanded τ_c with 1.0 increments	85
44	Plot of the allocated thrust for the tunnel thruster in a MIQP formulation minimizing thrust	86
45	Calculated τ_Y for the allocated thrust from a MIQP formulation minimizing thrust when simulating for all possible demanded τ_c with 1.0 increments	87
46	Calculated τ_N for the allocated thrust from a MIQP formulation minimizing thrust when simulating for all possible demanded τ_c with 1.0 increments	88

4 Introduction

The purpose of thrust allocation is to give the different actuators on the vessel a set-point in order to track a desired generalized force, which is the set of surge, sway and yaw demanded by the control system. In order to calculate these set-points, an accurate model of the vessel and its actuators are needed. This model will consist of numerous logical and behavioural limitations that the actuators may have.

The goal of this thesis is to create a MILP formulation of the problem in order to include logical constraints. This will introduce some errors associated with linearisation of non-linear behaviour, so a good trade off between accuracy and complexity must be found.

Thrust allocation depends on what the purpose of the allocation is. Thrust allocation is an optimization problem, and thus the formulation will depend on what you want to accomplish with the allocation. You may want to minimize power consumption, wear-and-tear by changing thrust as little as possible, error between demanded force and allocated force, etc. The resulting formulation will differ depending on the purpose. This thesis will focus on a formulation minimizing thrust and power.

Modelling actuators is a big part of the problem, and these models will remain the same regardless of what you choose to minimize in the thrust allocation optimization problem. The focus in this report is how to exploit the MILP framework in order to create accurate models of the actuators. Tunnel thrusters, azimuth thrusters and propeller-rudder pairs are considered, with a main focus on the azimuth thrusters and including forbidden sectors.

MILP is not the general way of solving the thrust allocation problem, because MILP solvers are highly complex, but there has been made significant advances in MILP solvers in recent years, combined with the advances in computer technologies, which together allows for reasonably fast solving of MILP problems of high complexity.

This report will first more thoroughly present the dynamic positioning (DP) problem in section 5 before section 6 concentrates on the thrust allocation part. Section 7 presents different constraints, and how they are included in the problem by linearising them and using the MILP framework. In section 8 the final formulations are presented mathematically before the implementation is explained in section 9. Finally, simulations and the results from the simulations are presented in section 10 following a discussion and a conclusion section. Suggestions for future work are presented in the conclusion. Abbreviations are explained when they are first used.

5 Dynamic Positioning

A vessel at sea is affected by several external forces like wind, waves and currents, as well as the forces generated by the vessels propulsion system. These forces act on the vessel and affects its position and heading. These changes are then measured using sensors like gyroscopes and accelerometers, as well as GPS measurements. A DP system is a computer controlled system that automatically maintains a vessels heading and position using its own propulsion system [Balchen et al., 1980]. Parts of this section and its sub sections are based on the authors project thesis.

A DP system contains a mathematical model of the vessel, including the position and force image of the vessels thrusters, and wind and current drag coefficients relative to the angle of attack of the current and wind, as well as the vessels speed. A DP system will use this information combined with the sensor data and a desired position and heading, to calculate a set of generalized forces, surge sway and yaw, for the vessel in order to put the vessel at a desired position with a desired heading [Balchen et al., 1976]. This allows a ship to maintain its position and heading at sea where mooring is not possible, or anchoring is not feasible due to sea depths, or structures, e.g. pipelines, on the seabed.

The consequences of failures in a DP system depend on the tasks performed during a DP operation. Hence, the required safety level in a DP system will vary [Ruth, 2008]. Therefore, the International Maritime Organization (IMO) has established different safety levels for DP systems, depending on the level of accuracy, called DP-classes [International Maritime Organization, 1994]: Class 1: "loss of position may occur in the event of single fault". Class 2: "loss of position is not to occur in the event of a single failure in any active component or system". Class 3: loss of position is not to occur in the event of a single failure in any component or system, including "All components in anyone watertight compartment, from fire or flooding". A risk analysis of the operation to be performed by the vessel should therefore be performed when choosing a DP class for the vessel.

5.1 Propulsion and thruster system

A vessels propulsion system consists of a set of actuators that can exert forces on the vessel. These actuators usually use a rotating propeller to generate thrust, but there are also some jet propulsion devices. The angle of the force is controlled either by a rudder or by rotating the propeller itself. The amount of thrust can be controlled by adjusting the RPM on the propeller, or the angle

on the propeller blades. Since there are several different thrusters and propeller systems available, the most basic types will be discussed briefly in the following sub sections, explaining their key features.

5.1.1 Main propellers and rudders

The main propellers main task is to generate thrust in the vessels longitudinal direction. The main propellers and rudders are mainly used in transit from one place to another, producing high speed, but can also be included in DP operations. The main propellers are usually not rotatable, so the angle of the thrust is controlled with rudders. One challenge with using the main propulsion system in DP operations, is the non-convexity that arises in the available thrust region, because the rudders can only be used to change the direction of the thrust, when the propellers exert thrust in their direction. Hence, when backing, the rudders will not work the same way as when going forwards. A main propeller with rudder is illustrated in figure 1.



Figure 1: Illustration photo of a main propeller with rudder, courtesy of Becker Marine Systems and Kongsberg Maritime

5.1.2 Azimuth thrusters

Azimuth thrusters are rotatable thrusters that can provide thrust in all directions in the surge/sway-plane. Azimuth thrusters are usually only used for fine

manoeuvring of the vessel and can be kept inside a closed compartment in the hull of the vessel when not in use. Azimuth thrusters are usually constructed with a duct around the propeller, as shown in figure 2, in order to reduce cavitation and increase the efficiency in one direction, and is therefore usually not used in reverse. Although the azimuth thruster can rotate 360 degrees, some angles may be unwanted angles to operate on. There are usually more than one azimuth thruster on a vessel, and forbidden/spill-zones are used to prevent thruster-thruster interactions, which can reduce efficiency. Forbidden zones, or sectors, can also be used to prevent the thrusters from blowing at the skeg, or in other unwanted angles. A challenge with forbidden sectors is that it leads to a non-convex thrust region, which means that a linear program or quadratic program can not be formulated as a single problem, but must be split into several convex sub-problems, or formulated as a mixed integer program, as will be done in this report. Forbidden sectors will be expanded on later in this report.

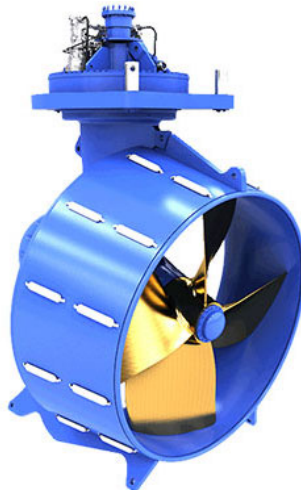


Figure 2: Illustration photo of an azimuth thruster, courtesy of Thrustmaster.net

5.1.3 Tunnel thrusters

Tunnel thrusters are usually placed at the bow or stern of the vessel in order to generate the most momentum possible. They generate transverse thrust and are

primarily used for steering purposes and not for propulsion purposes. Tunnel thrusters usually have rotatable propeller blades, in order to switch quickly from forward to reverse thrust and vice versa. Larger vessels may have multiple tunnel thrusters placed next to each other in order, as shown in figure 3, in order to generate more thrust. Some tunnel thrusters may be constructed such



Figure 3: Illustration photo of a tunnel thruster, courtesy of Brunvoll AS

that they are optimized for thrust in one direction, thus being asynchronous in their thrust response. Asymmetric tunnel thrusters will be expanded on and modelled later in this report.

5.2 Control system

The feedback control system provides the necessary commands in order to track the desired position and heading. It consists of an estimator, typically a Kalman-filter [Balchen et al., 1976], which uses a mathematical model of the vessel to estimate non-measured states and disturbances, like current forces. The control system then uses the sensor inputs, operator inputs and estimator estimates and calculates a 3-dimensional control output vector to keep the vessels desired

position and heading.

$$\tau_c = \begin{pmatrix} F_{surge} \\ F_{sway} \\ M_{yaw} \end{pmatrix}$$

The thruster allocation problem is an optimization problem that takes this 3-dimensional control output vector from the controller and calculates a set-point for each individual thruster

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$$

where

$$u_i = \begin{pmatrix} u_{i,surge} \\ u_{i,sway} \end{pmatrix}$$

u_i is a vector that contains the control input in surge and sway for thruster i . u_i can also be calculated into polar coordinates with a total control input

$$u_i = \sqrt{u_{i,surge}^2 + u_{i,sway}^2}$$

for each thruster, and an angle

$$\alpha_i = \arctan\left(\frac{u_{i,sway}}{u_{i,surge}}\right)$$

for each thruster. u can be calculated to an RPM and/or pitch for the propeller blades of the thruster, and the angle α is the input rudder angle or azimuth angle for the thruster.

Allocating thrust to the different thrusters based on the control output from the DP system is an optimization problem because most vessels are over actuated, hence there may be infinitely many thruster configurations that could produce the desired force. Using an optimization algorithm to solve this problem creates a lot of opportunities. You can minimize with respect to fuel consumption, force, use of thrusters, change in thrust, turn rate, time for better manoeuvrability, wear and tear etc. We can also consider forbidden zones as means of improving fuel consumption, wear and tear etc.

5.3 Thrust allocation in a DP system

The main objective of a thruster allocation algorithm, is to compute control inputs for all thrusters, τ , so that they jointly produce the DP commanded forces and moments, τ_c , at all time. As mentioned in section 5.2, this is an optimization problem, since there could be more than one solution to this problem, or no exact solution at all. In the case of no feasible solution, the thruster allocation algorithm should find a solution that minimizes the error $\epsilon = \tau u_c - \tau$, between its solution and the commanded input. Some kind of priority is usually involved, so that the error of some thrusters are penalized more than others. When the vessel is over actuated, so that no unique solution exists, other objectives and constraints can be introduced. Some examples are listed below.

- minimize change in thrust Δu
- minimize change in azimuth angle $\Delta \alpha$
- minimize fuel or power -consumption
- Avoid forbidden zones or spill zones in order to only use thrusters at their optimal operation angles
- introduce constraints to minimize wear and tare on thrusters and transmissions.

The thrust allocation problem is generally solved by means of non-linear solvers, but additional logical and operational constraints cannot be taken directly into account in the standard non-linear formulation. Typical constraints arising by operational limitations are for instance forbidden sectors. Solving the problem in a non-linear way may be necessary for achieving the desired performance, since linearising introduces some errors. Modelling the problem with piecewise linear functions leads to thruster allocation problem formulations that can be solved using Mixed Integer Linear Programming (MILP). The motivation for doing so, is when you have non-convex constraint sets which can be represented as the union of a small number of polyhedral sets [Johansen and Fossen, 2013]. By introducing forbidden zones, you will get a non-convex set of constraints. When also modelling the effector as piecewise linear and having piecewise linear cost function, MILP could be a good solution [Johansen and Fossen, 2013]. This report will focus on finding a good trade off between actuator model accuracy and optimization problem complexity using MILP.

6 Thrust Allocation

The ultimate goal of the thrust allocation optimization problem is, as mentioned in section 5.3, to ensure that the control forces and moments denoted τ_c calculated by the DP-controller is equal to the generalized forces τ calculated by the thrust allocation algorithm. τ is calculated through some function $f(u)$ where u is a vector of each thrusters force in surge and sway. A common choice for $f(u)$ is

$$f(u) = Bu$$

where B is a $3 \times 2n$ matrix that relates u to forces and moments on the vessel. However, in this thesis a thrust configuration, given in [Fossen, 2011]

$$\tau_c = T(\alpha)Ku \tag{1}$$

will be used.

Here $T(\alpha)$ is the thruster configuration matrix, describing the location and angle of each thruster and K is a diagonal force coefficient matrix describing the amount of thrust given by an input u for a given thruster. In the work done in conjunction with this thesis, K has been chosen such that the input u is given in the interval $u \in [0, 1]$.

This thruster configuration matrix is dependent on the azimuth angles α and is hence non-linear. In order to avoid this, the rotatable thrusters can be treated as two forces. The force F_i produced by thruster number i , then becomes the two forces

$$\begin{aligned} F_{surge,i} &= F_i \cos(\alpha_i) \\ &= K_i u_i \cos(\alpha_i) \end{aligned}$$

$$\begin{aligned} F_{sway,i} &= F_i \sin(\alpha_i) \\ &= K_i u_i \sin(\alpha_i) \end{aligned}$$

where

$$F_i = \sqrt{F_{surge,i}^2 + F_{sway,i}^2}$$

Equation (1) then becomes the extended thrust configuration [Fossen, 2011]

$$\tau_c = T_e K_e u_e \tag{2}$$

where e denotes extended. The control inputs can then be derived from the extended control elements by mapping the pairs $(u_{surge,i}, u_{sway,i})$ according to

$$u_i = \sqrt{u_{surge,i}^2 + u_{sway,i}^2} \quad \alpha_i = \text{atan2}(u_{sway,i}, u_{surge,i})$$

where atan2 is the four-quadrant inverse tangent MATLAB function. This function returns values in the closed interval $[-\pi, \pi]$ which may have to be taken into account in the implementation. In the thrust allocation algorithm implemented in conjunction with this thesis, thrust in cartesian coordinates has been used.

6.1 Optimization formulations

Choosing a formulation of the problem is critical for an optimization problem. You want a formulation that describes the problem accurately enough, but also a formulation that can be solved within a reasonably short time. For thrust allocation, if the model is not accurate enough, wrong inputs will be found for the thrusters, and the vessel will not respond as was intended. However a problem that is accurate but takes too long to solve is no good either, because by the time the problem is solved, the position and heading of the vessel may have changed, and thus the solution would no longer be a viable solution. Some optimization techniques will now be presented briefly.

6.1.1 Linear Programming

Linear programs (LP) have a linear objective function and linear constraints, which may include both equalities and inequalities. The feasible set is a polytope, a convex, connected set with flat, polygonal faces. The contours of the objective function are planar [Nocedal and Wright, 2006]. The standard form of a linear program is

$$\min c^T x, \text{ s.t. } Ax = b, x \geq 0 \quad (3)$$

where c and x are vectors in \mathbb{R}^n , b is a vector in \mathbb{R}^m and A is a $m \times n$ matrix. The solution of a linear program can be a single vertex, which is a unique solution. The solution can also be a line, a face or even the entire feasible set, depending on the problem. The benefit of linear programs is that there are very efficient algorithms for solving them [Nocedal and Wright, 2006].

6.1.2 Quadratic Programming

A quadratic program (QP) has linear constraints, as a linear program, but has a quadratic objective function. A general quadratic program can be stated as

$$\begin{aligned} \min_x q(x) &= \frac{1}{2}x^T Gx + x^T c \\ \text{s.t. } a_i^T x &= b_i, \quad i \in \mathcal{E} \\ a_i^T x &\geq b_i, \quad i \in \mathcal{I} \end{aligned}$$

where G is a symmetric $n \times n$ matrix, \mathcal{E} and \mathcal{I} are finite sets of indices, and c , x and $\{a_i\} \in \mathcal{E} \cup \mathcal{I}$ are vectors in \mathbb{R}^n [Nocedal and Wright, 2006]. If the QP is convex it is often similar in difficulty to a linear program. The difference between QP and LP is the objective function. Here the decision variable(s) are minimized quadratically and not linearly. This results in QP being less aggressive around zero than LP, and more aggressive for larger values, as can be seen from figure 4. This can have an influence when minimizing error, but more importantly, as will be shown later in this report, when minimizing a resource. If two resources equally influence the total system, a QP solution will divide equally between the two, while an LP may not. Taking minimizing thrust as an example, if two thrusters equally contribute to the vessels total force and the solution should be 1 in total, a QP will divide that equally between the two thrusters, while an LP may result in using all on one thruster, since i.e. $1 + 0 = 0.5 + 0.5$ results in the same value when minimizing an LP, while $1^2 + 0^2 > 0.5^2 + 0.5^2$ for a QP, so a QP solver will choose 0.5 for both thrusters while an LP may not. This will be shown and discussed further later in this thesis.

6.1.3 Mixed Integer Linear Programming

A Mixed Integer Linear Program (MILP) is an optimization method that combines continuous and discrete variables. A general MILP formulation can be stated as

$$\begin{aligned} \min_{x,y} x^T c + d^T y \\ \text{s.t. } Ax + By &\geq b \\ (x, y) &\in \mathbb{R}_+^n \times \mathbb{Z}_+^p \end{aligned}$$

where A is an $m \times n$ matrix, B is an $m \times p$ matrix, b is an m -dimensional vector, c is an n -dimensional vector, d is a p -dimensional vector, \mathbb{R}_+^n is the n -dimensional

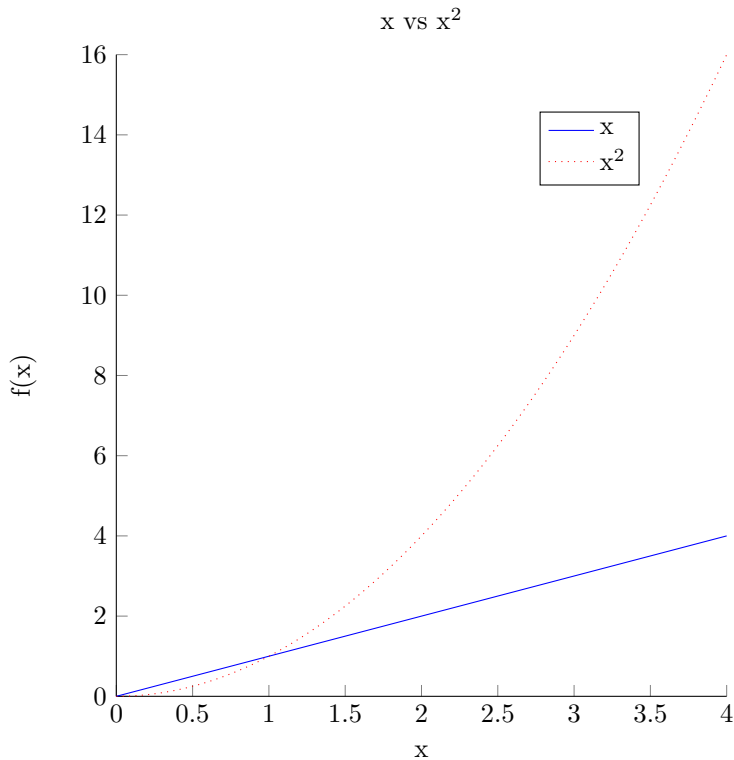


Figure 4: x vs x^2

space of all non-negative real numbers and \mathbb{Z}_+^p is the p-dimensional space of all non-negative integers. The set of feasible solutions, X , are defined as

$$X = (x, y) \in \mathbb{R}_+^n \times \mathbb{Z}_+^p : Ax + By \geq b$$

Integer values can be introduced so that a non-linear function can be approximated as a piecewise linear function, for creating either-or constraints, conditional constraints etc. A MILP equally to an LP minimizes a linear objective function, and thus faces the same challenges. A MILP problem is generally solved using a linear-programming based branch-and-bound algorithm [GUROBI, 2015].

6.1.4 Mixed Integer Quadratic Programming

A Mixed Integer Quadratic Program (MIQP) is similar to a MILP, but with a quadratic objective function. Some solvers also allow quadratic cost functions [GUROBI, 2015] but this report will only consider MIQP with linear constraints. A MIQP minimizing thrust will be presented and simulated for comparison with the full MILP formulation.

6.2 Slack variables

As previously stated, the ultimate goal of thruster allocation is to ensure $\tau_c = \tau$, and hence

$$\tau_c = T_e K_e u_e. \quad (4)$$

Now u_e has to be chosen such that it fulfils equation (4). It could be reasonable to choose a least square solution of (4), $\|\tau_c - T_e K_e u_e\|_2^2$ and choose a u_e that minimizes this equation, but this could give many, or even no solution at all because of physical limitations on the thrusters, thus, an optimization problem must be formulated to choose the best one. Subscript e will from now on not be used for simpler notation.

$$\min_s s^T Q s \quad (5a)$$

s.t.

$$\tau_c = T K u + s \quad (5b)$$

$$u_{min} \leq u \leq u_{max} \quad (5c)$$

$$\Delta u_{min} \leq \Delta u \leq \Delta u_{max} \quad (5d)$$

$$\alpha_{min} \leq \alpha \leq \alpha_{max} \quad (5e)$$

$$\Delta \alpha_{min} \leq \Delta \alpha \leq \Delta \alpha_{max} \quad (5f)$$

This is a typical QP formulation of the thrust allocation problem. Here, s is the decision variable and Q is a square, symmetric and positive definite weighting matrix. s is the slack variable, and the goal is to minimize s . We see that if $s = 0$, $\tau_c = T K u = \tau$, which is the ultimate goal. The reason for introducing slack variables is the physical constraints on the thrusters, which in this case is given by equations (5c) - (5f). By introducing s , some deviations from $\tau_c = \tau$ are allowed in the relaxed constraint (5b) for situations where this is just not possible. With (5b) the optimization problem still has a feasible solution for the situations where $\tau_c = \tau$ is not possible.

While the constraint (5b) is linear, the constraints (5c) - (5f) are not, because they describe physical limitations on the thrusters that are not linear. For instance, for a given thruster u_i , the constraint (5c) becomes

$$u_{i,min} \leq \sqrt{u_{ix}^2 + u_{iy}^2} \leq u_{i,max}$$

and the angle limitation becomes

$$\alpha_{i,min} \leq \arctan\left(\frac{u_{iy}}{u_{ix}}\right) \leq \alpha_{i,max}$$

which are both clearly non-linear with respect to the decision variables. However; they can be linearised, which was done in the authors project thesis.

In this thesis, the constraints 5d and 5f will not be included in the optimization problem. The thrust allocation developed in this thesis will find a new set-point for each thruster, and assume that the controller for each thruster will make sure the constraints 5d and 5f are satisfied. This can also be included in the thrust allocation problem by remembering the previous thrust. Constraints 5d and 5f can then be written as

$$\begin{aligned} \Delta u_{min} &\leq u_i - u_{i-1} \leq \Delta u_{max} \\ \Delta \alpha_{min} &\leq \alpha_i - \alpha_{i-1} \leq \Delta \alpha_{max} \end{aligned}$$

Since part of the objective with this thesis is to find a trade off between model accuracy and formulation complexity, the time needed to solve the allocation problem is one of the parameters considered. Since these constraints only add two constraints to the problem formulation, the additional time the solver needs are negligible compared to the time needed to solve the whole problem, and thus these constraints are not included.

6.3 Minimizing thrust

Minimizing slack variables yields a feasible solution, but you will have no control over the amount of thrust allocated to each thruster. Hence, the solution given by minimizing slack may yield a very suboptimal solution with respect to fuel consumption. The common way of solving this issue, is to include a term with u in the minimization formulation in order to penalize the use of thrust.

The term used to penalize thrust is

$$u^T P u \tag{6}$$

where P is a square, symmetric and positive definite weighting matrix, similar to Q in the $s^T Q s$ term presented in the previous section. How much the thrust is penalized depends on the terms in P . Generally, $u^T P u$ is used together with the $s^T Q u$ term, where the weights in $Q \gg P$ to make sure that the main priority is obtaining the DP-order τ_c [Rindarøy, 2013].

Minimizing thrust is reasonable in order to reduce wear and tare, and to prevent the thrusters from going on full throttle all the time. However, it is not equivalent to minimizing power or fuel consumption. Power has a non-linear relationship with thrust. Hence, minimizing thrust is not equivalent to minimizing power or fuel consumption. This will be expanded on in the following sub sections.

6.4 Minimizing power

As mentioned in the previous sub section, minimizing thrust is not equivalent to minimizing fuel or power consumption, but minimizing thrust will probably reduce power consumption for the vessel, since less thrust will be used. The relationship between thrust and power consumed by a given thruster is

$$p_{T_i} = |T_i|^{3/2} \quad (7)$$

where

$$T_i = \sqrt{u_{i,surge}^2 + u_{i,sway}^2}$$

and p_{T_i} is the power consumed by thruster number i . This term must be included in the objective function for each thruster. Hence, an objective function for minimizing slack and power consumption would have the following form

$$J = s^T Q s + \sum_{i=1}^n p_{T_i}$$

equation 7 is clearly not linear, and has to be linearised in order to be able to use a linear solver. [Ruth, 2008] presents a quadratic approximation of this term as

$$p_{T_i} \approx p_{T_i}^Q = \frac{T_i^2}{\sqrt{|T_{i,prev}|}}$$

which can be used for formulating a QP formulation, since $T_i = \sqrt{u_{i,surge}^2 + u_{i,sway}^2}$. [Rindarøy, 2013] implemented this in his master thesis using the term

$$u^T H u$$

in the objective function where

$$H = \begin{pmatrix} \frac{1}{\gamma_1 \sqrt{T_{1,prev}}} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{\gamma_1 \sqrt{T_{1,prev}}} & 0 & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & \frac{1}{\gamma_n \sqrt{T_{n,prev}}} & 0 \\ 0 & \cdots & \cdots & 0 & \frac{1}{\gamma_n \sqrt{T_{n,prev}}} \end{pmatrix}$$

where γ_i is a weighting factor introduced by [Rindarøy, 2013], allowing for different penalization for different thrusters further explained in [Rindarøy, 2013]. Note that each term appears twice on the diagonal, since

$$u_i = \begin{pmatrix} u_{i,surge} \\ u_{i,sway} \end{pmatrix}$$

In this thesis, power minimization will be included in a MILP formulation using special ordered set of type 2 (SOS2) variables. SOS2 variables is a vector of variables where at most two of the variables are non-zero, and the non-zero variables must be consecutive in their ordering [Löfberg, 2015c]. SOS2 variables will be used to add the thrust-power relationship as a piecewise linear function. By using piecewise linear approximation of the thrust-power relationship, any non-linear relationship can be included in the optimization problem. The power will be approximated as

$$p_{T_i} = |T_i|^{3/2} \quad (8)$$

with an associated weighting vector, R , resulting in the following linear objective function

$$\min_{s, p_{T_i}} Qs + Rp_{T_i} \quad (9)$$

6.4.1 Power constraints

In order to make sure thrusters do not consume more power than what is available, we can put a constraint on power. For electric thrusters driven by power generators, you want to make sure the accumulated power consumed by all the thrusters does not exceed the capacity of the generators [Veksler et al., 2012]. This must be put as a constraint

$$p_{T_i} = |T_i|^{3/2} \leq p_{max}$$

where T_i is the thrust as explained above, and p_{max} is the power generators maximum capacity. This has not been implemented in the simulations done in conjunction with this report, because the power supply on the model ship is able to provide the thrusters with enough power even when they all run on full throttle. This is trivial to implement and has no significant impact on the total complexity of the problem.

6.5 Minimizing fuel consumption

Minimizing fuel consumption as supposed to minimizing power is a problem involving much more than the power used by the different thrusters. A typical vessel has one or more power generators providing power not only to the propulsion and thruster system, but to the rest of the vessel as well [Rindarøy, 2013]. Minimizing fuel instead of power therefore heavily rely on other consumers than the thrust and propulsion system. Other consumers may include hotel loads, drilling units, cranes, pumps, heave compensation systems and many more [Veksler et al., 2012]. If you only have electrically driven thrusters and propellers and one electrical board, minimizing fuel and power will probably yield the same result. However, if you have diesel driven propellers or thrusters, or more than one electrical board, minimizing fuel instead of power may benefit the fuel consumption. In this thesis, minimizing fuel consumption will not be implemented, but a thrust to fuel relationship could be implemented as a piecewise linear function using special ordered set of two (SOS2) variables.

7 Modelling constraints

In this section the modelling of the different actuators are presented. The big-M technique used to include forbidden sectors are also explained in detail, as understanding big-M is a key part of the modelling of forbidden sectors done in this thesis.

7.1 Polygon Approximation

The Polygon Approximation is a method for approximating a circular area as an N-sided polygon made out of a set of N linear inequalities. Parts of this section is taken from the authors project thesis.

The thrust produced by an azimuth thruster is given by

$$T \leq \sqrt{u_{surge}^2 + u_{sway}^2} \quad (10)$$

This thrust, T , is bounded by some physical limitation in the thruster, such that $T \leq T_{max}$, which draws a circular area in the surge-sway plane. Equation 10 is clearly non-linear, but can be linearised using the Polygon Approximation. With the Polygon Approximation, we wish to approximate the circle with an N-sided polygon fitted inside the circle.

7.1.1 Polygon Approximation error

Given a circle with radius $T_{max} = R > 0$, the polygon approximation fits a N-sided polygon within this circle, as shown in figure 5. Fitting another circle inside the polygon will give the maximum approximation error. The inner circle will have a radius

$$r = R \cos\left(\frac{\pi}{N}\right) \quad (11)$$

Hence, the maximum approximation error, ϵ , is given by the difference between the two radii R and r ,

$$\epsilon = R - r = R \left(1 - \cos\left(\frac{\pi}{N}\right)\right) \quad (12)$$

Thus, the maximum approximation error depends on the maximum thrust, $T_{max} = R$ and the number of sides, N , of the polygon. The minimum number of sides for a given maximum thrust and allowed error is therefore given by

$$N \geq \left\lceil \frac{\pi}{\arccos\left(1 - \frac{\epsilon}{R}\right)} \right\rceil \quad (13)$$

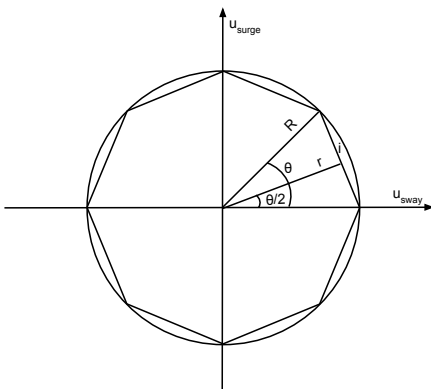


Figure 5: The thrust approximated by an N -sided polygon. Here, $N = 8$.

$R > 0$. As given by 12, the more sides used to approximate the circle, the more accurate it will be. However, the complexity of the problem regarding the number of equations the solver has to check also grows accordingly. Hence, a good trade of between model accuracy and complexity of the optimization problem must be found.

7.1.2 Calculating the polygon

Each side of the polygon will be given as a linear inequality constraint in the optimization problem. Letting i be one side of the polygon, r will be perpendicular to i and intersect in the middle of i . θ is the angle that one side, i , spans over, hence, i and r will intersect at $\theta/2$. The slope of r , Δr is given by

$$\Delta r = \frac{\sin(\theta/2)}{\cos(\theta/2)} \quad (14)$$

and the slope of i , Δi is therefore

$$\Delta i = -\frac{1}{\Delta r} = -\frac{\cos(\theta/2)}{\sin(\theta/2)} \quad (15)$$

since i and r are perpendicular. The point where i and r intersect, $p = (x_1, x_2)$ is given by

$$\begin{aligned}x_1 &= r \cos(\theta/2) \\x_2 &= r \sin(\theta/2)\end{aligned}$$

Given the equation for a straight line, we get

$$x_2 = ax_1 + b \quad (16)$$

where a is the slope, Δi , and b is the intersection point with the y-axis, in our case, u_{surge} -axis. By inserting $p = (x_1, x_2)$ and $a = \Delta i$ into 16, we get

$$r \sin(\theta/2) = -\frac{\cos(\theta/2)}{\sin(\theta/2)} \cdot r \cos(\theta/2) + b \quad (17)$$

and thus

$$b = r \cos(\theta/2) + \frac{\cos^2(\theta/2)}{\sin(\theta/2)} \quad (18)$$

Inserting this back into 17, yields the equation for one of the N sides in the polygon

$$x_2 = -\frac{\cos(\theta/2)}{\sin(\theta/2)} \cdot x_1 + r \sin(\theta/2) + \frac{\cos^2(\theta/2)}{\sin(\theta/2)} \quad (19)$$

multiplying with $\sin(\theta/2)$ and rearranging yields

$$x_2 \sin(\theta/2) + x_1 \cos(\theta/2) = r \underbrace{(\sin^2(\theta/2) + \cos^2(\theta/2))}_{=1}$$

and thus, the equation for one of the N sides in the polygon is

$$r = x_2 \sin(\theta/2) + x_1 \cos(\theta/2) \quad (20)$$

Generalizing equation 20 with $u_{surge} = x_1$, u_{sway} and angle θ_k , the approximated polygon thrust region becomes the following set of linear inequalities [Wit, 2009]

$$[\cos(\theta_k), \sin(\theta_k)] \begin{pmatrix} u_{surge} \\ u_{sway} \end{pmatrix} \leq r \quad (21)$$

for every

$$\theta_k = \frac{\pi}{N} + k \frac{2\pi}{N}, \quad k = 0, \dots, N - 1$$

So, for a set of n azimuth thrusters, the linear inequality constraints can be put as

$$Cu \leq r$$

where

$$C = \begin{pmatrix} C_1 & 0 & \cdots & 0 \\ 0 & C_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & C_n \end{pmatrix}$$

$$C_i = \begin{pmatrix} \sin(\theta_0) & \cos(\theta_0) \\ \sin(\theta_1) & \cos(\theta_1) \\ \vdots & \vdots \\ \sin(\theta_{N-1}) & \cos(\theta_{N-1}) \end{pmatrix}$$

$$r = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}$$

$$r_i = \begin{pmatrix} r \\ r \\ \vdots \\ r \end{pmatrix}$$

$$u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}$$

$$u_i = \begin{pmatrix} u_{i,surge} \\ u_{i,sway} \end{pmatrix}$$

The matrix C_i holds the N linear equations describing the approximated polygon that constrains the thrust produced by thruster number i. The vector r_i contains the maximum value thrust, specified by r for thruster number i. Hence, for each i, $C_i u_i \leq r_i$ is equation (21).

7.2 Forbidden sectors

Although an azimuth thruster is able to rotate 360° , it may not be desirable to operate on all angles. For instance, we do not want to flush other thrusters or disturb equipment mounted on the vessel near the thruster. These unwanted operation areas are called forbidden sectors. A way to formulate forbidden sector is that the azimuth angle, α , is only allowed to take certain values, for instance

$$\alpha \in \left[0, \frac{3\pi}{4}\right) \cup \left(\frac{5\pi}{4}, 2\pi\right] \quad (22)$$

where the forbidden zone is $\alpha \in \left[\frac{3\pi}{4}, \frac{5\pi}{4}\right]$.

7.2.1 Formulating forbidden zones

The formulation given in 22 is one way to formulate a forbidden sector, but we need a linear formulation. The linearisation also has to be convex, or a combination of convex sets. In this thesis, this problem is solved by doing a big-M reformulation. In this section, the big-M reformulation technique is first explained, and then used in order to exclude forbidden sectors from the feasible set for a thruster in the thrust allocation problem.

Big-M reformulation

Big-M reformulations are used to convert a logic or non-convex constraint to a set of constraints describing the same feasible set, using auxiliary binary variables and additional constraints [Löfberg, 2015a]. A common approach to big-M modelling, is to add very big constraints in order to make sure the big-M model includes everything it should. This will work, but will give very bad and essentially useless models. The big-M reformulations will feature terrible numerical behaviour, and the relaxations of the big-M model that are used in the mixed integer solver will be very weak, leading to excessive branching and thus increased computation time [Löfberg, 2015a].

As mentioned, a problem with the big-M model is when the big-M constant is chosen to large, and hence creating an unreasonable large feasible set. The consequence is, besides poor numerics, the weakness of the relaxed mixed integer model. To illustrate the problem, take the 4 polytopic regions shown in figure 6. These are 4 random polytopic regions, constraining a variable x , given by

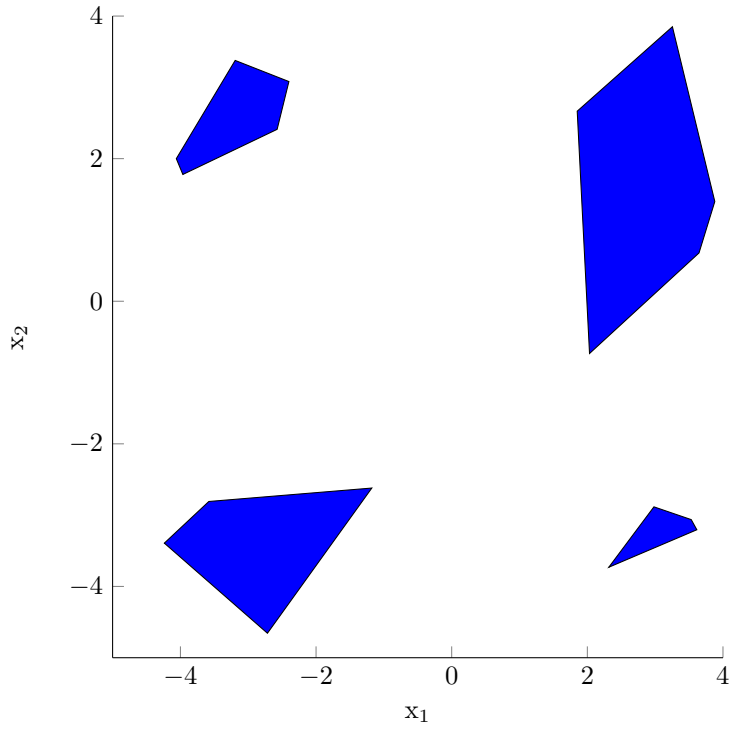


Figure 6: A plot of 4 polytopic regions constraining a variable x

the equations

$$A1 \cdot x \leq b1$$

$$A2 \cdot x \leq b2$$

$$A3 \cdot x \leq b3$$

$$A4 \cdot x \leq b4$$

The big-M model for this case is given by

$$\begin{aligned}
 F &= \text{sum}(d) = 1 \\
 F &= [F, A1 \cdot x - b1 \leq M1 \cdot (1 - d(1))] \\
 F &= [F, A2 \cdot x - b2 \leq M2 \cdot (1 - d(2))] \\
 F &= [F, A3 \cdot x - b3 \leq M3 \cdot (1 - d(3))] \\
 F &= [F, A4 \cdot x - b4 \leq M4 \cdot (1 - d(4))]
 \end{aligned}$$

where d is an array of 4 binomial variables, and $M1, M2, M3, M4$ are the big-M constants. F is the big-M model. Choosing a too large value for the big-M constants, for instance 50, will lead to a poor model, shown in figure 7. This is

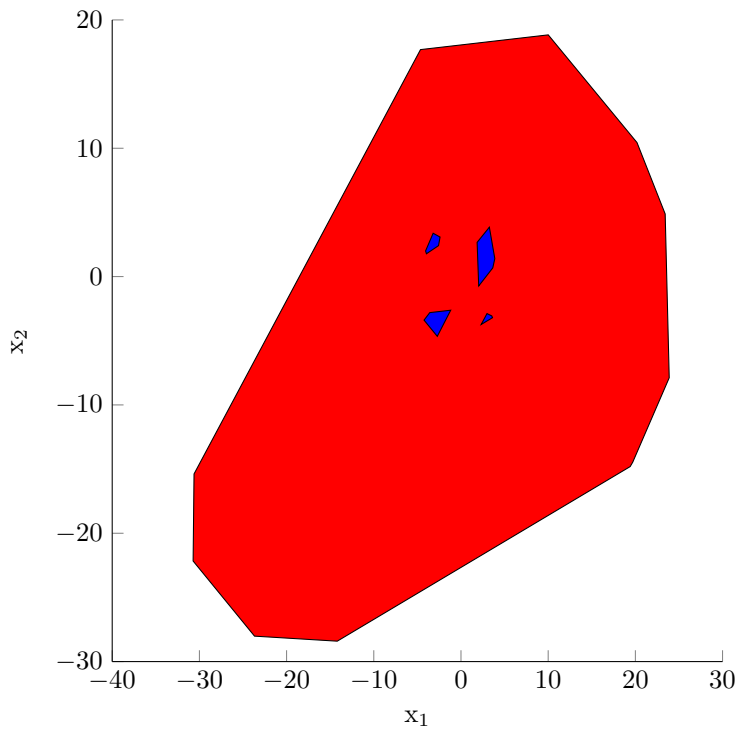


Figure 7: The relaxed mixed integer model using $M = 50$

clearly not a good approximation of the true feasible set. A better big-M model

is obtained by using reasonable and thought through big-M constants. A good big-M constant is based on insight in the original model. In this case it is hard to choose a good big-M constant, since the data is randomly generated, but by using the hull command [Löfberg, 2015b], figure 8 shows the optimal big-M model for this case.

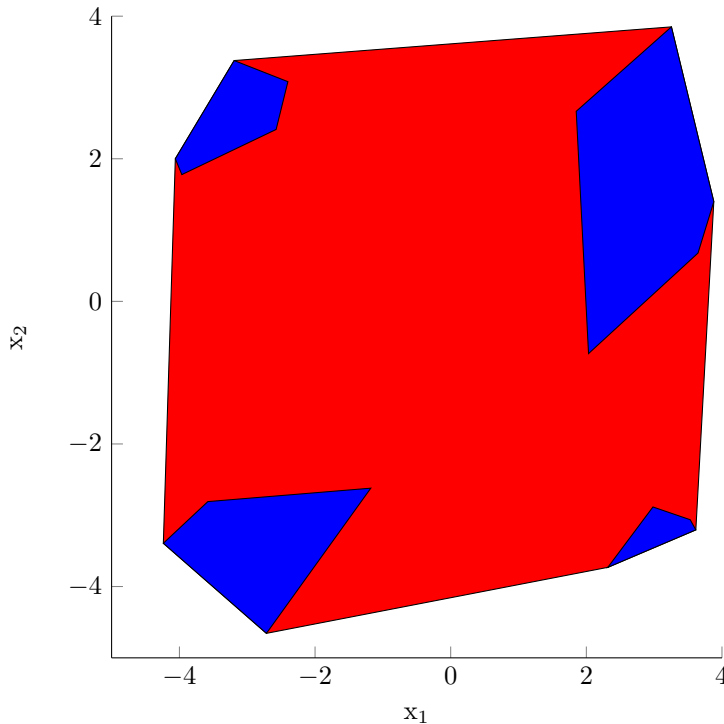


Figure 8: The relaxed mixed integer model of an optimal big-M model, using the convex hull of the original constraints

The goal in creating a big-M model is to create a model whose relaxation is as close as possible to the convex hull of the original constraints. The convex hull of a set of points, vectors or sets, $x = x_1, x_2, \dots, x_n$ is the smallest convex set containing all the points, vertices or sets [Nocedal and Wright, 2006]. Using predefined commands, such as hull in YALMIP, will introduce more variables and constraints. This may often yield a good model, but the best approach is

defining the model yourself, so that you can control all variables and constraints. This, however, demands a good knowledge of what you are modelling.

Big-M reformulation for a forbidden sector

In this thesis, big-M is used to exclude forbidden sectors from the thrusters feasible set. In addition to the polygon approximation, two linear constraints is added excluding the forbidden sector from the feasible set. This results in two halves partially overlapping on the opposite side of the forbidden sector, as shown in figure 9. these lines are given by the following equations

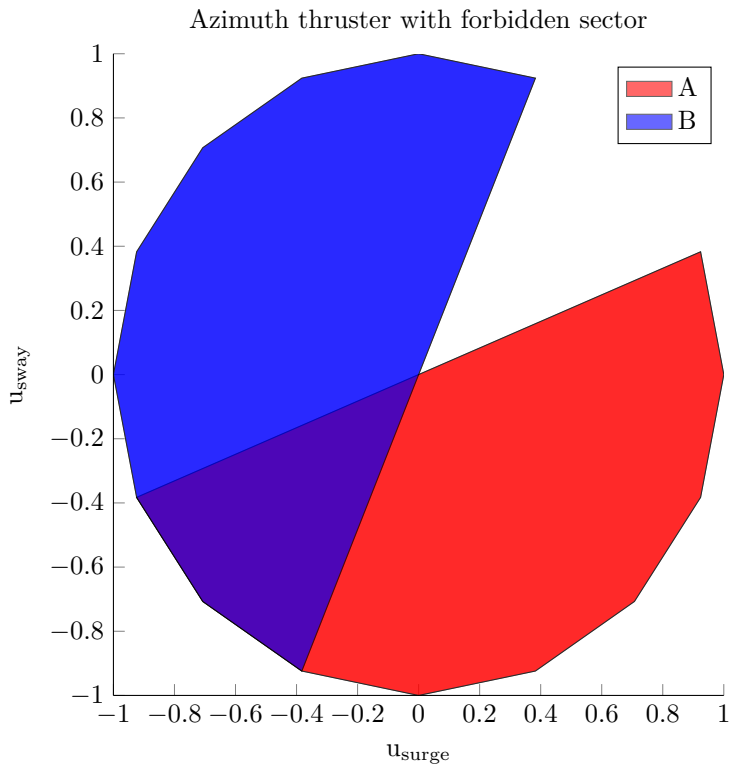


Figure 9: Plot of a forbidden sector constructed of two half circles

$$A = [\cos(\alpha_{start} + \frac{\pi}{2}), \sin(\alpha_{start} + \frac{\pi}{2})] \cdot u \leq 0$$

$$B = -[\cos(\alpha_{end} + \frac{\pi}{2}), \sin(\alpha_{end} + \frac{\pi}{2})] \cdot u \leq 0$$

where u is the thrust

$$u = \begin{pmatrix} u_{surge} \\ u_{sway} \end{pmatrix}$$

and α_{start} and α_{end} are the angles at which the forbidden sector starts and ends, given counter clockwise from the positive u_{surge} axis. This is implemented in MATLAB with the following code.

```
A = [[cos(alphaStart + pi/2), sin(alphaStart + pi/2)]*u <= 0];
B = -[[cos(alphaEnd + pi/2), sin(alphaEnd + pi/2)]*u <= 0];
A = [A, polygonApproximation];
B = [B, polygonApproximation];
```

where `polygonApproximation` is the set of constraints constructing the polygon approximation as explained in section 7.1. The implementation of this technique will be expanded on in later sections.

As these two half circles creates a non-convex set, it has to be reformulated in order to be used in a mixed integer linear program. These two half circles are in this thesis combined using a big-M formulation. Two binary variables are introduced to switch between the two half circles, and a big-M constant is added to shift the non-active constraint so that includes the active half circle. This is done for each forbidden sector, hence introducing 2 variables and 2 constraints per forbidden sector. The relaxed set will always be the polygon approximation using this technique. That will, for many configurations of forbidden sectors, not make the minimum relaxed set, but very close to it, and you keep control of all constraints and variables creating the constraint set.

The lines creating the forbidden sector are shifted, as explained above, by introducing the following term.

$$A = [\cos(\alpha_{start} + \frac{\pi}{2}), \sin(\alpha_{start} + \frac{\pi}{2})] \cdot u \leq 0 + M(b(1) - 1)$$

$$B = -[\cos(\alpha_{end} + \frac{\pi}{2}), \sin(\alpha_{end} + \frac{\pi}{2})] \cdot u \leq 0 + M(b(2) - 1)$$

where b is a vector of two binary variables, and M is the big-M constant. This big-M constant must be chosen large enough that it includes what should be feasible, but not more than that. As [Löfberg, 2015a] explains it, it should rather be named the "sufficiently-large-small-M" rather than big-M. We want the big-M constant to be sufficiently large so that it includes the other half circle.

The smallest big-M we can choose is $M = u_{max}$, where u_{max} is the radius of the feasible thrust circle. Hence, the equations for the two lines becomes

$$\begin{aligned} A &= \left[\cos \left(\alpha_{start} + \frac{\pi}{2} \right), \sin \left(\alpha_{start} + \frac{\pi}{2} \right) \right] \cdot u \leq 0 + u_{max}(b(1) - 1) \\ B &= - \left[\cos \left(\alpha_{end} + \frac{\pi}{2} \right), \sin \left(\alpha_{end} + \frac{\pi}{2} \right) \right] \cdot u \leq 0 + u_{max}(b(2) - 1) \end{aligned}$$

Combined with the equations making the polygon approximation, we now have a set of constraints for a thruster with one or more forbidden sectors. Explained in logics, a point is feasible if it lies in the polygon approximation, and in A or B. For more forbidden sectors, 2 more constraints and variables are added for each forbidden sector. For n forbidden sector the constraint set becomes

$$Constraints = Cu \leq r \ \& \ \sum_{i=1}^n (A_i | B_i) \quad (23)$$

where $Cu \leq r$ is the polygon approximation, and A_i and B_i are the two halves for forbidden sector number i for n forbidden sectors.

This method will, as mentioned, not make the minimum relaxed set possible for all configurations of forbidden sectors, but will only introduce two variables and two constraints per forbidden sector in addition to the polygon approximation constraints. Using the hull command would probably introduce more constraints and variables [Löfberg, 2015b], and thus mitigating the advantage of a smaller relaxed set.

7.3 Asynchronous tunnel thruster

Some tunnel thrusters may be asynchronous, meaning they can produce more force in one direction than the other. This is due to mechanical design of the thruster, usually to optimize thrust in one direction. This results in

$$|B \cdot u^-| \leq |B \cdot u^+| \quad (24)$$

where u^- is negative thrust, u^+ is positive thrust and B is the matrix that relates u to forces and moments on the vessel from equation 6. The matrix B consists of T and K , as explained in section 6, where T describes the location and angle of each thruster and K describes the amount of thrust given by an input u for a given thruster. As an approximation, the asynchronous tunnel thruster in this thesis is implemented so that the tunnel thruster has a percent wise effect in one direction compared to the other,

$$|B \cdot u^-| = a |B \cdot u^+| \quad (25)$$

where a is an asynchronous-constant $a \in [0, 1]$ where $a = 0$ means the tunnel thruster can not go in reverse and $a = 1$ means the tunnel thruster is synchronous.

8 Problem formulations

In this section the final formulations to be implemented will be presented.

8.1 LP formulation

The final LP formulation to be implemented is

$$\min_{u^+, u^-, s^+, s^-} [Q, \quad Q, \quad P, \quad P] \begin{bmatrix} s^+ \\ s^- \\ u^+ \\ u^- \end{bmatrix} \quad (26a)$$

s.t.

$$[B, \quad B, \quad I, \quad I] \begin{bmatrix} u^+ \\ -u^- \\ s^+ \\ -s^- \end{bmatrix} = \tau_c \quad (26b)$$

$$u^+ \geq 0 \quad (26c)$$

$$u^+ \leq u_{max} \quad (26d)$$

$$u^- \leq 0 \quad (26e)$$

$$-u^- \leq u_{max} \quad (26f)$$

$$s^+ \geq 0 \quad (26g)$$

$$s^- \leq 0 \quad (26h)$$

where Q and P are weighting row-vectors. Constraints 26c - 26f are implemented as written above for tunnel thrusters, and using the polygon approximation for azimuth thrusters. Constraints 26g and 26h are also implemented as written above. u^+ and u^- are for azimuth thrusters also divided into u_{surge}^+ and u_{surge}^- for both surge and sway, so that

$$u_{abs} = \begin{bmatrix} u_{surge}^+ - u_{surge}^- \\ u_{sway}^+ - u_{sway}^- \end{bmatrix}$$

$$u = \begin{bmatrix} u_{surge}^+ + u_{surge}^- \\ u_{sway}^+ + u_{sway}^- \end{bmatrix}$$

for all azimuth thrusters.

8.2 QP formulation

The final QP formulation is similar to the LP formulation, but without having to divide all variables into a positive and negative part.

$$\min_{u,s} s^T Qs + u^T Pu \quad (27a)$$

s.t.

$$[B, \ I] \begin{bmatrix} u \\ s \end{bmatrix} = \tau_c \quad (27b)$$

$$u_{min} \leq u \leq u_{max} \quad (27c)$$

where Q and P are square, symmetric and positive definite weighting matrices. Constraint 27c are implemented as written for the tunnel thruster, and using the polygon approximation for the azimuth thrusters.

8.3 MILP formulation

The MILP formulation is equal to the LP formulation, but with additional logical constraints including forbidden sectors to the problem, with $|B^-| \leq |B^+|$ for the tunnel thruster, making it asynchronous, and with a piecewise-linear function approximating the power consumption which is minimized instead of the thrust. The forbidden sectors are added as

$$a_i^T u_j \leq 0 + u_{j,max}(1 - b_i) \quad (28a)$$

$$b_i^T u_j \leq 0 + u_{j,max}b_i \quad (28b)$$

where 28a creates the line where forbidden sector i for azimuth thruster j starts, and 28b creates the line where forbidden sector i for azimuth thruster j ends.

The final formulation becomes

$$\min_{u^+, u^-, s^+, s^-} [Q, \quad Q, \quad R, \quad R] \begin{bmatrix} s^+ \\ s^- \\ p^+ \\ p^- \end{bmatrix} \quad (29a)$$

s.t.

$$[B, \quad B, \quad I, \quad I] \begin{bmatrix} u^+ \\ -u^- \\ s^+ \\ -s^- \end{bmatrix} = \tau_c \quad (29b)$$

$$u^+ \geq 0 \quad (29c)$$

$$u^+ \leq u_{max} \quad (29d)$$

$$u^- \leq 0 \quad (29e)$$

$$-u^- \leq u_{max} \quad (29f)$$

$$s^+ \geq 0 \quad (29g)$$

$$s^- \leq 0 \quad (29h)$$

$$a_i^T u_j \leq 0 + u_{j,max}(1 - b_i) \quad (29i)$$

$$b_i^T u_j \leq 0 + u_{j,max}b_i \quad (29j)$$

$$p^+ = \sum_{i=1}^n \lambda_i^+ p_i^+ \quad (29k)$$

$$p^- = \sum_{i=1}^n \lambda_i^- p_i^- \quad (29l)$$

$$\sum_{i=1}^n \lambda_i^+ = 1 \quad (29m)$$

$$\sum_{i=1}^n \lambda_i^- = 1 \quad (29n)$$

$$0 \leq \lambda_i^+ \leq 1 \quad (29o)$$

$$0 \leq \lambda_i^- \leq 1 \quad (29p)$$

where p^+ is the power consumed by positive thrusts, and p^- is the power consumed by negative thrusts, positive and negative in the surge-sway plane. p_i

are the power related to λ_i in the piecewise linear function. The SOS2 variables λ_i has the property that only two adjacent variables can be non-zero.

8.4 MIQP formulation

The MIQP formulation is equal to the MILP formulation, but with quadratic cost function

$$\min_{u,s} s^T Qs + u^T Pu \quad (30a)$$

For the MIQP simulations, thrust is minimized instead of power. This is because a piecewise linear approximation introduces a lot of additional equations and variables, and there is a quadratic approximation of power by [Rindarøy, 2013] which would be more efficient to implement. This approximation is not implemented in this thesis, please see [Rindarøy, 2013] for more information about this approximation.

The final formulation is

$$\min_{u,s} s^T Qs + u^T Pu \quad (31a)$$

s.t.

$$[B, \quad I] \begin{bmatrix} u \\ s \end{bmatrix} = \tau_c \quad (31b)$$

$$u_{min} \leq u \leq u_{max} \quad (31c)$$

$$a_i^T u_j \leq 0 + u_{j,max}(1 - b_i) \quad (31d)$$

$$b_i^T u_j \leq 0 + u_{j,max}b_i \quad (31e)$$

$$(31f)$$

9 Implementation

9.1 CyberShip Enterprise 1

The model ship C/S Enterprise 1 is used for model-scale experiments in the NTNU model basin Marine Cybernetics Laboratory. The thruster configuration, shown in figure 10 [Skjetne and Øivind K. Kjerstad, 2013], consists of two aft Voith Schneider thrusters having circular thrust regions, and a bow tunnel thruster producing thrust in the transversal direction [Skjetne and Øivind K. Kjerstad, 2013].

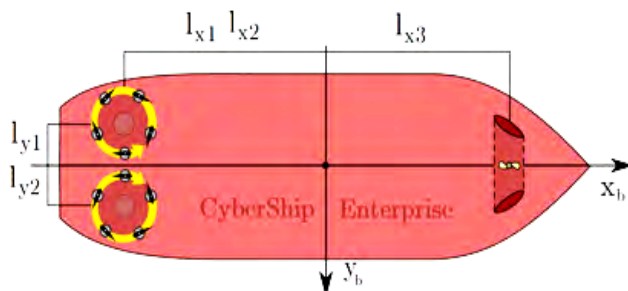


Figure 10: A figure of the thruster configuration on CyberShip Enterprise 1 (CSE1)

The linear effector model is

$$\tau = Tf, f = Ku, \Rightarrow \tau = TKu = Bu \quad (32)$$

$$\begin{bmatrix} \tau_N \\ \tau_X \\ \tau_Y \end{bmatrix} = \begin{bmatrix} -l_{y1} & l_{x1} & -l_{y2} & l_{x2} & l_{x3} \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \\ f_{y3} \end{bmatrix}$$

$$\begin{bmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \\ f_{y3} \end{bmatrix} = \begin{bmatrix} f_{M1} & 0 & 0 & 0 & 0 \\ 0 & f_{M1} & 0 & 0 & 0 \\ 0 & 0 & f_{M2} & 0 & 0 \\ 0 & 0 & 0 & f_{M2} & 0 \\ 0 & 0 & 0 & 0 & f_{M3} \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{y3} \end{bmatrix}$$

where f_i are the individual thruster forces within maximum values f_{Mi} . In this thesis, as in [Skjetne and Øivind K. Kjerstad, 2013], the controls are scaled through the gains in K such that $|u_i| = 1 \Rightarrow |f_i| = |f_{Mi}|$. Here, $y_b = u_{sway}$ and $x_b = u_{surge}$. The thrust allocation algorithm made in conjunction with this thesis is not implemented on the model ship, only simulated in MATLAB using

$$\begin{aligned}
 l_1 &= \begin{bmatrix} -0.425 \\ -0.055 \end{bmatrix} \\
 l_2 &= \begin{bmatrix} -0.425 \\ 0.055 \end{bmatrix} \\
 l_3 &= \begin{bmatrix} 0.425 \\ 0.000 \end{bmatrix} \\
 f_{iM} &= 2.5
 \end{aligned}$$

and, as mentioned, $|u_i| \in [0, 1]$.

9.2 Polygon Approximation

The polygon approximation from section 7.1 is implemented as a function of N and u_{max} where N is the number of linear constraints the circle is approximated of.

```

k = [0:N-1];
thetas = [pi/N + k*(2*pi/N)];
for i = 1:length(thetas) % length(thetas) = N
    Ck(i,1) = sin(thetas(i));
    Ck(i,2) = cos(thetas(i));
    r(i,1) = u_max*cos(thetas(1));
end

```

The function returns a $N \times 2$ matrix C_k and a $N \times 1$ vector r and the resulting constraint set becomes $C_k u \leq r$. In this thesis the approximation is done using $N = 16$ which results in an approximation error of

$$\begin{aligned}
 \epsilon &= R - r = R \left(1 - \cos \left(\frac{\pi}{N} \right) \right) \\
 &= 1 \left(1 - \cos \left(\frac{\pi}{16} \right) \right) \\
 &\approx 0.0192
 \end{aligned}$$

hence at most 1.92% error. The resulting constraint set is shown in figure 11.

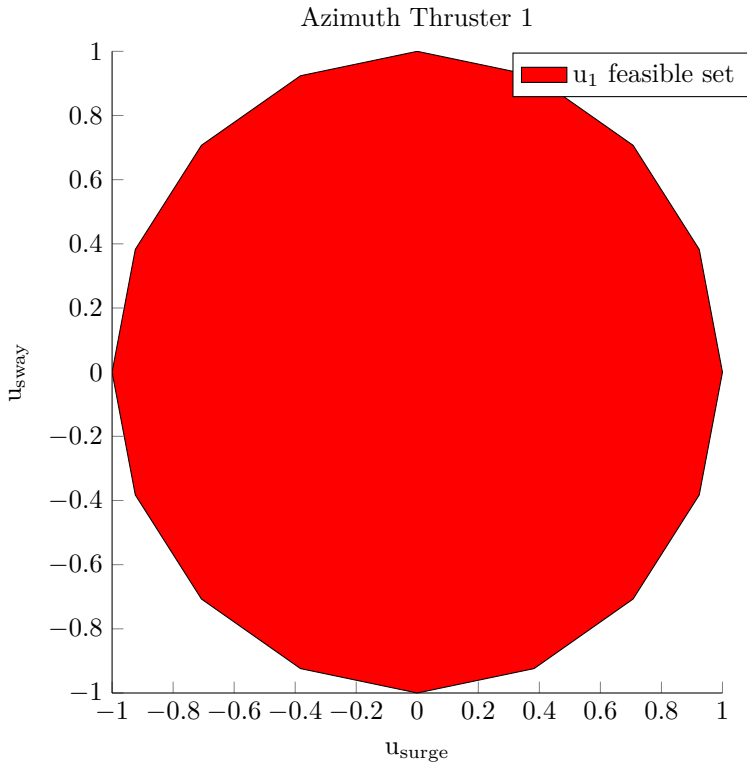


Figure 11: Plot of the polygon approximation using $N = 16$ and $|u_{max}| = 1$

9.3 Explicit constraints on thrust

YALMIP demands explicit constraints on thrust and all other decision variables when formulating a mixed integer problem. Explicit meaning constraints that is not hidden in functions and logic. For an azimuth thruster in a linear problem, the explicit constraints becomes

```

u1_surge_pos = sdpvar(1,1);
u1_surge_neg = sdpvar(1,1);
u1_sway_pos  = sdpvar(1,1);
u1_sway_neg  = sdpvar(1,1);

```

```

Constraints_u1 = [u1_surge_pos >= 0, u1_surge_pos <= u1_max];
Constraints_u1 = [Constraints_u1, u1_surge_neg <= 0];
Constraints_u1 = [Constraints_u1, -u1_surge_neg <= u1_max];
Constraints_u1 = [Constraints_u1, u1_sway_pos >= 0];
Constraints_u1 = [Constraints_u1, u1_sway_pos <= u1_max];
Constraints_u1 = [Constraints_u1, u1_sway_neg <= 0];
Constraints_u1 = [Constraints_u1, -u1_sway_neg <= u1_max];

u1_abs = [u1_surge_pos - u1_surge_neg;...
u1_sway_pos - u1_sway_neg];
u1 = [u1_surge_pos + u1_surge_neg;...
u1_sway_pos + u1_sway_neg];

```

for each azimuth thruster. $u1_abs$ is used for minimization and $u1$ is used in the objective function. For a quadratic formulation there is no need for dividing the decision variable into a positive and a negative part, since the squared thrust is minimized, so the explicit constraints becomes

```

u1 = sdpvar(2, 1); % [u_surge; u_sway]

Constraints_u1 = [u1(1) <= u1_max, -u1(1) <= u1_max];
Constraints_u1 = [Constraints_u1, u1(2) <= u1_max];
Constraints_u1 = [Constraints_u1, -u1(2) <= u1_max];

```

These constraints creates a square which encloses the polygon approximation, and thus has no effect on the feasible set, but is necessary due to the nature of YALMIP.

9.4 Forbidden sectors

The implementation of forbidden sectors was briefly explained in section 7.2.1, where it was mentioned that a forbidden sector is added with the following code

```

A = [[cos(alphaStart + pi/2), sin(alphaStart + pi/2)]*u <= 0];
B = -[[cos(alphaEnd + pi/2), sin(alphaEnd + pi/2)]*u <= 0];
A = [A, polygonApproximation];
B = [B, polygonApproximation];

```

For ease of modelling, a function for making the constraint set of a thruster was made. This function calls the function making the polygon approximation, and then adds forbidden sectors to it, if any. The function takes two vectors of

angles, `alphaStart` and `alphaEnd`, which are the start and end angle of forbidden sectors. The code above is thus expanded to

```
Constraints = [];  
for i = 1:length(alphaStart)  
    A = [[cos(alphaStart(i) + pi/2), ...  
         sin(alphaStart(i) + pi/2)]*u <= 0];  
    B = -[[cos(alphaEnd(i) + pi/2), ...  
          sin(alphaEnd(i) + pi/2)]*u <= 0];  
    A = [A, polygonApproximation];  
    B = [B, polygonApproximation];  
    Constraints = [Constraints, A, B];  
end
```

The function can add an arbitrary amount of forbidden sectors. Figure 12 shows the plot of a thrusters feasible set with 3 forbidden sectors from $\pi/8$ to $7\pi/8$, $7\pi/8$ to $17\pi/18$ and from $11\pi/8$ to $13\pi/8$.

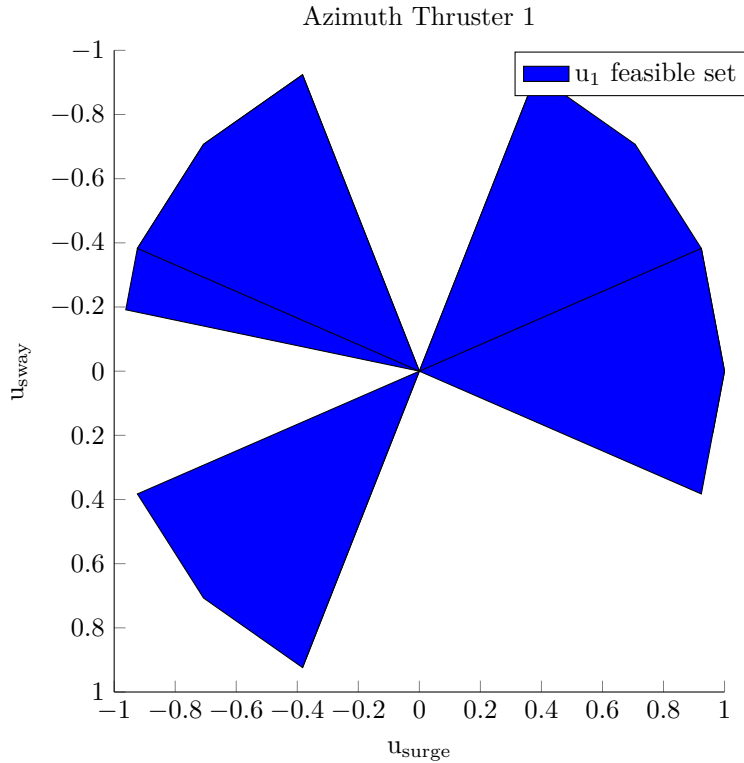


Figure 12: A plot of a thrusters feasible area with 3 forbidden sectors

The lines creating the forbidden sectors are made feasible for two reasons. One, in order to make zero thrust a feasibility, and two, in order to be able to reuse the same function when making the feasible set of a main propeller-rudder pair.

9.5 Main propeller and rudder

The feasible region of a main propeller with a rudder consists of a cone in the direction of the rudder, and a straight line in the opposite direction for going in reverse where the rudder has no effect. The function made for creating the feasible set of an azimuth thruster with forbidden sectors was implemented so that the start and end angles of a forbidden sector was made feasible, which

can easily be implemented, but has not been done in this thesis. Simulations using propeller-rudder pairs has been performed, switching out the azimuth thrusters on C/S Enterprise 1 with propeller-rudder pairs, but plots from these simulations will not be presented in this report in order to reduce the amount of plots, and because this configuration and use of actuators is probably not realistic.

9.6 Tunnel thruster

The constraints for a tunnel thruster is simply

```

u_tunnel_pos = sdpvar(1,1);
u_tunnel_neg = sdpvar(1,1);

Constraints = [u_tunnel_pos >= 0];
Constraints = [Constraints, u_tunnel_pos <= u3_max];
Constraints = [Constraints, u_tunnel_neg <= 0];
Constraints = [Constraints, -u_tunnel_neg <= u3_max];

```

```

u3 = u_tunnel_pos + u_tunnel_neg;
u3_abs = u_tunnel_pos - u_tunnel_neg;

```

for a linear problem and

```

u_tunnel = sdpvar(1,1);

Constraints = [u_tunnel <= u3_max];
Constraints = [Constraints, -u_tunnel <= u3_max];

```

for a quadratic problem. This models a symmetric tunnel thruster which can produce equal thrust in both directions. For an asynchronous tunnel thruster, $|Bu^-| \leq |Bu^+|$, so also the quadratic model must have a positive and a negative part. The asymmetry is introduced by

```

u3 = u_tunnel_pos + asymmetric_tunnel_const*u_tunnel_neg;
u3_abs = u_tunnel_pos - asymmetric_tunnel_const*u_tunnel_neg;

```

where `asymmetric_tunnel_const` is the a in equation 25. Now the thrust of the tunnel thruster can take values from -1 to 1, but the resulting thrust acting on the vessel will only range from - a to 1. Other asynchronous relationships being more non-linear could also be implemented using a piecewise linear function approximating the relationship.

9.7 Piecewise linear approximation of power

For repetition, the relationship between thrust and power is

$$p_{T_i} = |T_i|^{3/2} \tag{33}$$

where T_i is thrust and p_{T_i} is power for thruster number i . In a MILP formulation, the decision variable for an azimuth thruster is already split into u_{surge} and u_{sway} which again are split into a positive and a negative part. All these decision variables must be related to power with a piecewise linear approximation. This results in $4n$ new constraint equations and decision variables per azimuth thruster or propeller-rudder pairs, and $2n$ new constraint equations and decision variables for tunnel thrusters, where n is the number of lines used in the approximation. Although a lot of the equations are removed by the presolver, this is adding a lot of complexity to the problem. Figure 14 shows 3 examples of the piecewise-linear approximation using 3, 5, and 10 lines in the approximation.

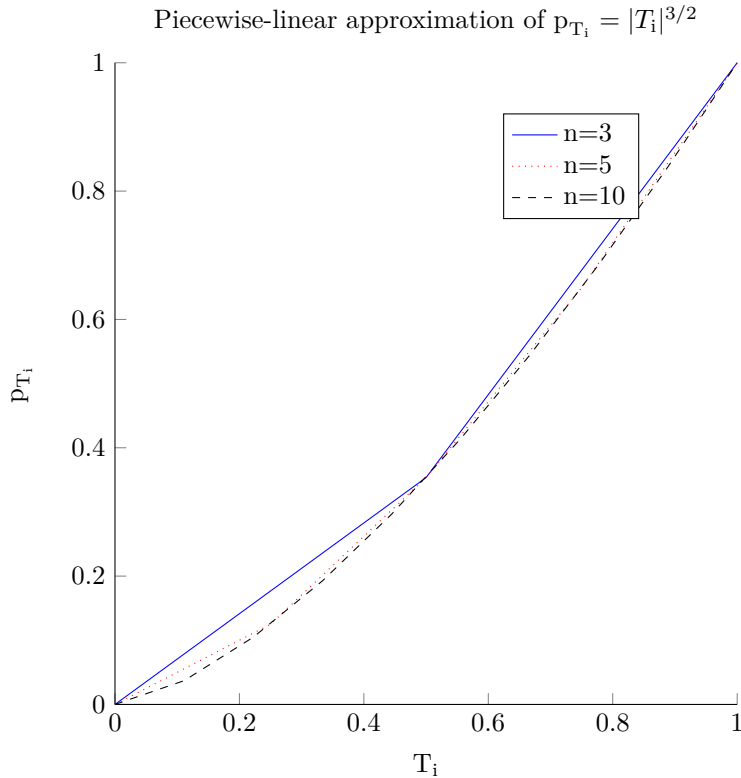


Figure 14: A plot of the piecewise-linear approximations of $p_{T_i} = |T_i|^{3/2}$ using n lines in the approximation

The number of lines used in the approximation will be discussed and expanded on later in this report. Piecewise-linear approximation of power has only been implemented in the MILP formulation, not in the MIQP. This is because e.g. [Rindarøy, 2013] has included a quadratic approximation of power, which will be much more efficient to solve, because it does not introduce more variables and equations to the problem, which a piecewise-linear approximation does. It is important to note that the main objective here is to see if including a thrust-power relationship as a piecewise linear makes the problem to complex to solve within a reasonable time or not, and not the accuracy of the relationship. This relationship will depend greatly on the thruster, and will have to

be approximated for the specific vessel and thruster. Power consumption also depend on how much the thruster is moved between each iteration, which is not considered in this thesis. The relationship chosen to be approximated in this report is from [Rindarøy, 2013].

10 Simulations and results

In this section, several simulations and their results will be presented and commented on. The simulations are done in MATLAB using the YALMIP modelling language and with a CPLEX solver integrated with MATLAB and YALMIP. The simulations will be of increasing complexity, gradually adding on more constraints to the problem. For the first simulation, a lot of plots will be presented, while for the consecutive simulations, a subset of these plots will be presented in order to reduce the amount of plots. Additional plots can be found in the appendix. The simulation is a simple thrust allocation for every possible input from a DP system or joystick. The demanded force, τ_c has been iterated from f_{min} to f_{max} with 1.0 increments for τ_X , τ_Y and τ_N and a set-point thrust allocation has been performed.

```
for tau_X = -f1_max:1:f1_max
    for tau_Y = -f2_max:1:f2_max
        for tau_N = -f3_max:1:f3_max
            runAllocation();
        end
    end
end
```

Not all of these demanded forces are feasible to achieve due to the configuration of thrusters on the ship. The results of the simulations will be presented consecutively, sacrificing some layout norms, in an attempt not to confuse the reader with which results belong to which simulation.

10.1 Linear Program

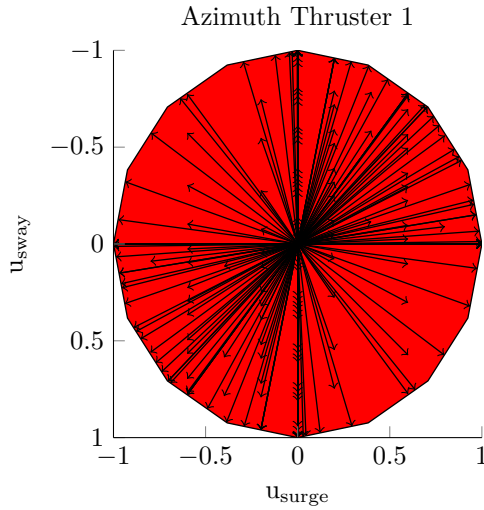
The linear program is the basis of the mixed integer linear program, and will also work as a reference for the additional logic added in the MILP formulation to see how it affects the solution. The linear program is modelled by adding polygon approximation constraints limiting the two azimuth thrusters, and a limit on the tunnel thruster. The weights in this simulation has been chosen to $Q = [1000, 1000, 1000, 1000, 1000]$ and $P = [1, 1, 1, 1, 1]$.

First, all allocated thrusts has been plotted in their feasible area, to see that the constraints are held, and how the thrusts are distributed. As shown in figure 15, the distribution of thrust between azimuth thruster 1 and 2 does not look right.

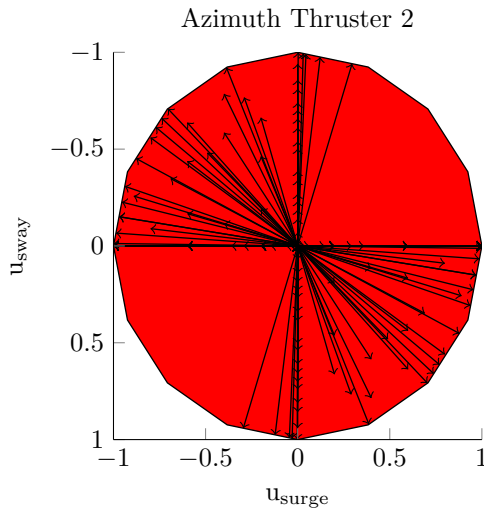
Because azimuth thruster 1 and 2 lies in the same distance from the center of mass on the vessel, they contribute the same force per thrust to the vessel. Since the LP solver tries to minimize a linear thrust, it will start by using azimuth thruster 1 until it reaches its maximum. It will then add the rest of the total thrust needed to azimuth thruster 2 in order to achieve enough total thrust. The more power efficient way would be to distribute the total thrust evenly between the two thrusters. This happens because there is no information in the LP formulation saying that two thrusters at half speed is more power efficient than running one at full speed. This is one of the reasons a thrust-power relationship is important to add in a linear formulation. The plot of the thrust given by the tunnel thruster is not presented in this report, as it is simply arrows distributed on a line from -1 to 1 , but can, for the particularly interested, be found in appendix A.

Figure 16, 17 and 18 shows the demanded τ_c compared to the τ calculated from the allocated thrusts for τ_X , τ_Y and τ_N . As the figures shows, the vessel is saturated for some combinations of demanded force, which is expected. Although τ_X , τ_Y and τ_N separately can range from -2.5 to 2.5 , it depends on the the combination of demanded forces. For instance, $\tau_c = [2.5, 0, 0]^T$ is possible to achieve, while $\tau_c = [2.5, 2.5, 2.5]^T$ is not. We also see that τ_X only has minor numerical errors, while τ_Y and τ_N has larger errors. This is as expected, because τ_X is directly affected by the tunnel thruster, while τ_Y is not. Furthermore, we see from the model equations 32 that thrust from the different thrusters has more impact on τ_X than τ_N , which explains why τ_N has more errors than τ_X , but less than τ_Y because τ_N is affected some by the tunnel thruster, while τ_Y is not affected by the tunnel thruster at all.

The time needed to solve this formulation is negligible, i.e. $\approx 0.02s$.



(a) Allocated thrust from an LP formulation for azimuth thruster 1



(b) Allocated thrust from an LP formulation for azimuth thruster 2

Figure 15: Plots of the allocated thrust from an LP formulation for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment

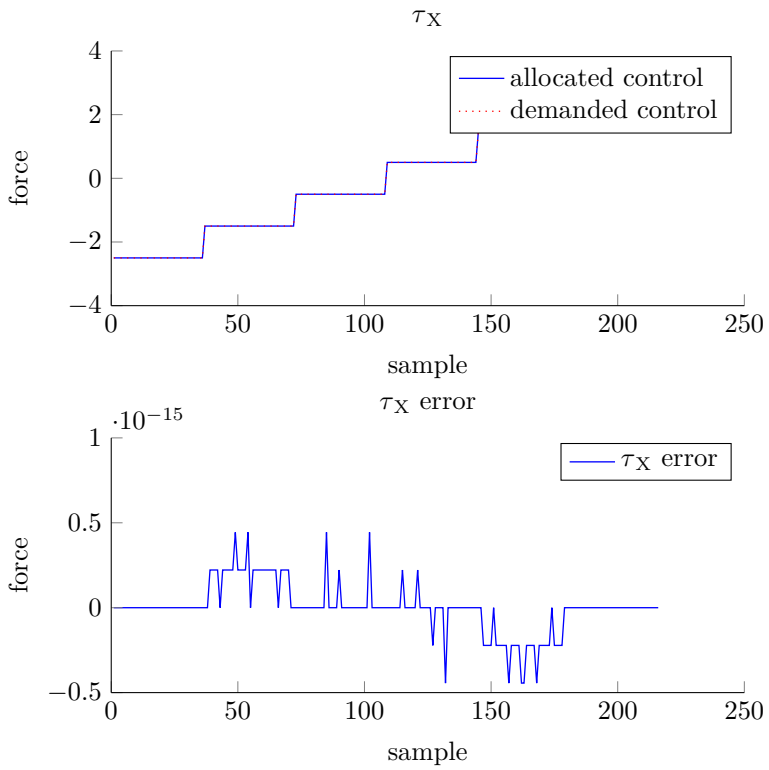


Figure 16: Calculated τ_X for the allocated thrust when simulating an LP formulation for all possible demanded τ_c with 1.0 increments

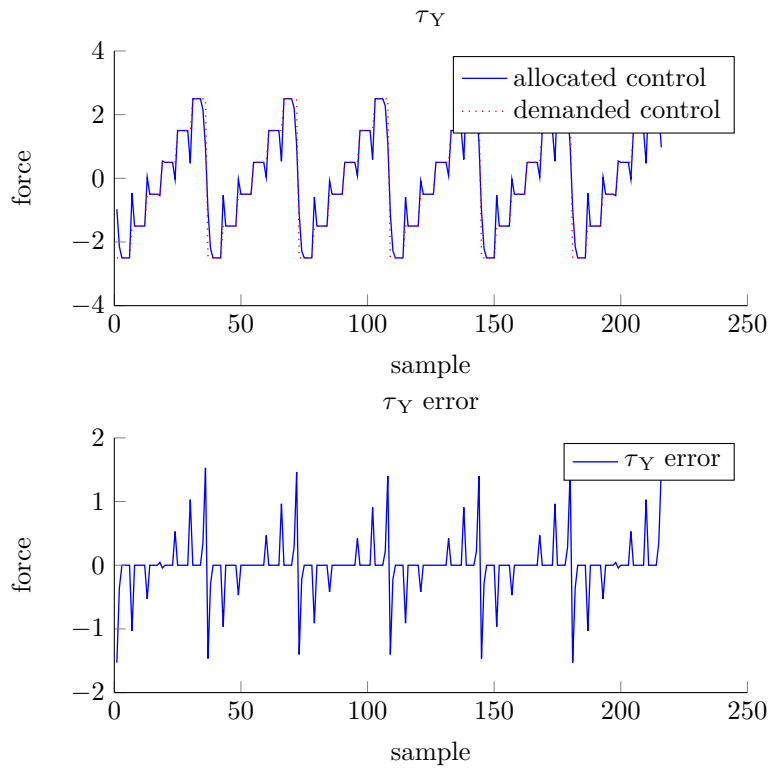


Figure 17: Calculated τ_Y for the allocated thrust when simulating an LP formulation for all possible demanded τ_c with 1.0 increments

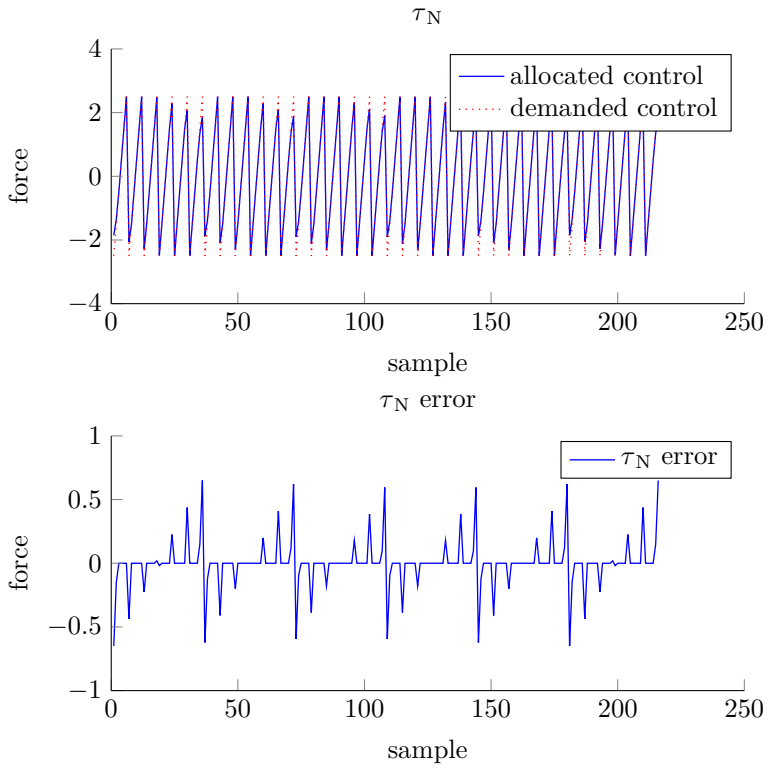


Figure 18: Calculated τ_N for the allocated thrust when simulating an LP formulation for all possible demanded τ_c with 1.0 increments

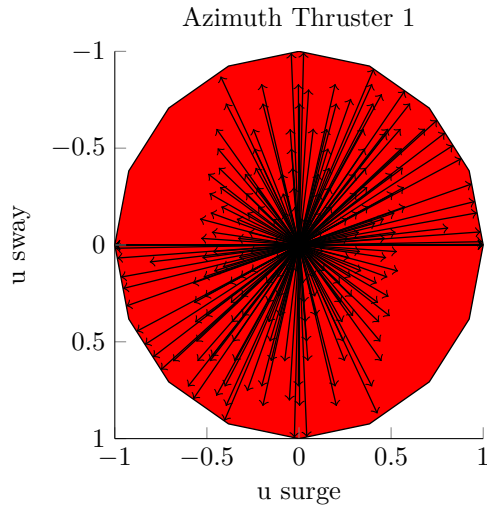
10.2 Quadratic Program

The Quadratic program is equal to the linear program, except that we do not have to separate the positive and negative part, because the squared decision variable is minimized. The weights in the cost function in the simulation has been chosen to $Q = \text{diag}([10000, 10000, 10000, 10000, 10000])$ and $P = \text{diag}([1, 1, 1, 1, 1])$. The constants in the weighting matrix Q is larger than in the LP formulation, this is because a QP tends to be less aggressive around small errors than an LP, so this is done to compensate for that. Large errors are more penalized in a QP than an LP, so this will enhance that effect.

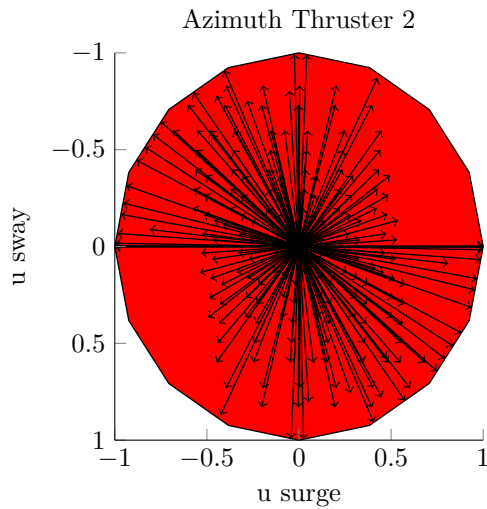
When the squared thrust is minimized, the solver will divide thrust evenly between azimuth thruster 1 and 2 as explained in section 6.1. This can be seen in figure 19.

This is a more intuitive distribution of thrust between the two thrusters. We see that the thrust is centralised in two slightly tilted squares. These are tilted because of the position of the thrusters relative to the vessels centre of mass. In addition, there are two fans of thrust for both thrusters which are thrust used to produce more torque on the vessel (τ_N). Although power is not included in this formulation, the solver gets information that thrust is expensive with square growth, and thus chooses to divide thrust as evenly as possible between the thrusters. As a result, the solution will consume less power than the LP solution. Since power is still not included in the formulation, the solution gives no information about the amount of power used. Therefore, one can not constrain the total amount of power consumed.

The errors are also more evenly distributed, since the error now is penalised cubed. τ_Y still has the largest errors due to the configuration of the thrusters, but the errors on τ_X and τ_N are more even. This is because the QP will strongly punish large errors, and be less aggressive on small errors, so instead of a correct τ_X and large errors on τ_Y and τ_N as the LP solution gave, we now get smaller errors on all forces. For comparison, a plot of τ_X and τ_Y are shown in figure 20 and 21, while plot for τ_N can be found in appendix A together with the plot of the tunnel thrusters allocated thrust. The time needed to solve this formulation is negligible, i.e. $\approx 0.02s$.



(a) Allocated thrust from a QP formulation for azimuth thruster 1



(b) Allocated thrust from a QP formulation for azimuth thruster 2

Figure 19: Plots of the allocated thrust from a QP formulation for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment

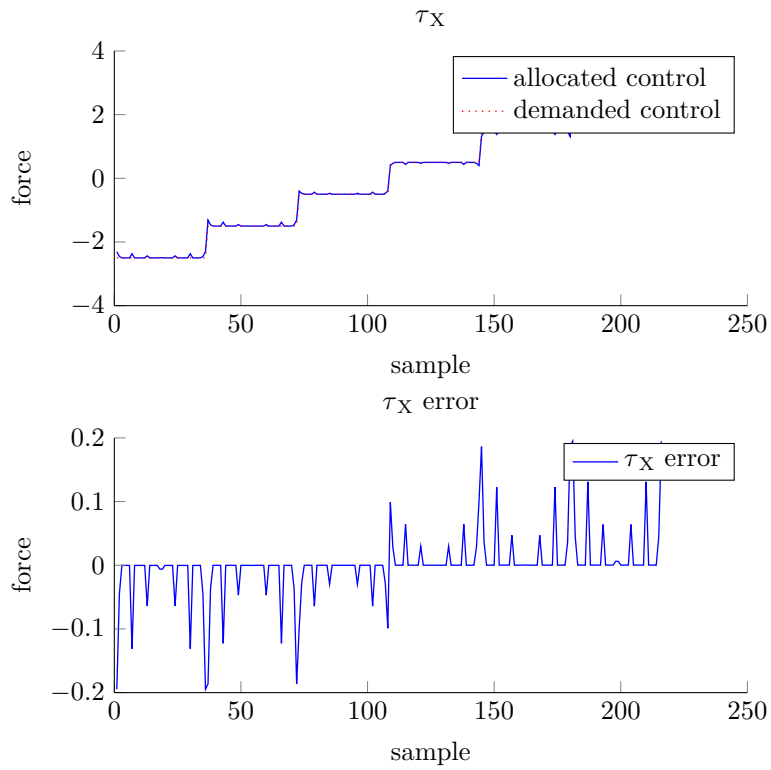


Figure 20: Calculated τ_X for the allocated thrust when simulating a QP formulation for all possible demanded τ_c with 1.0 increments

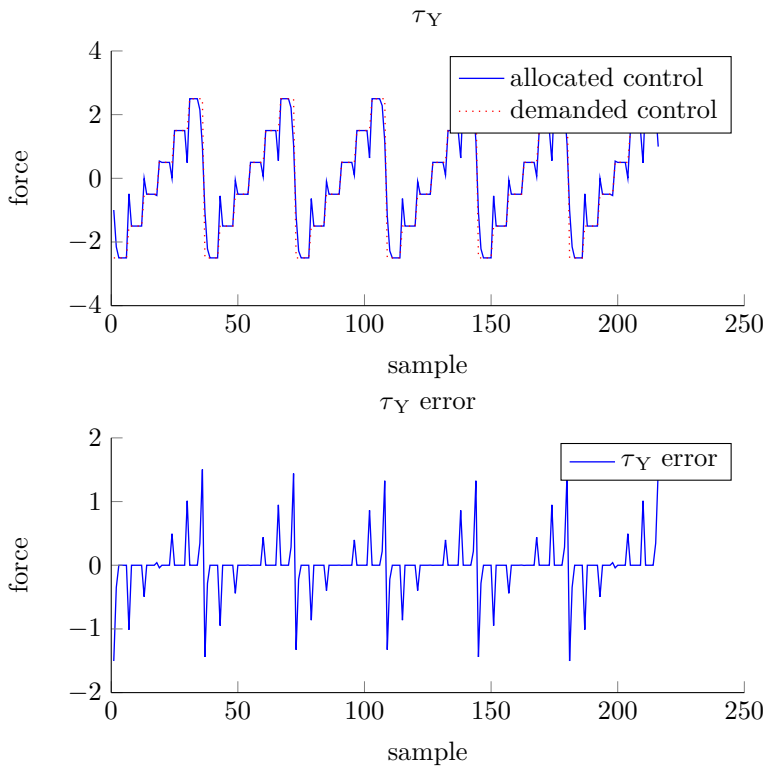


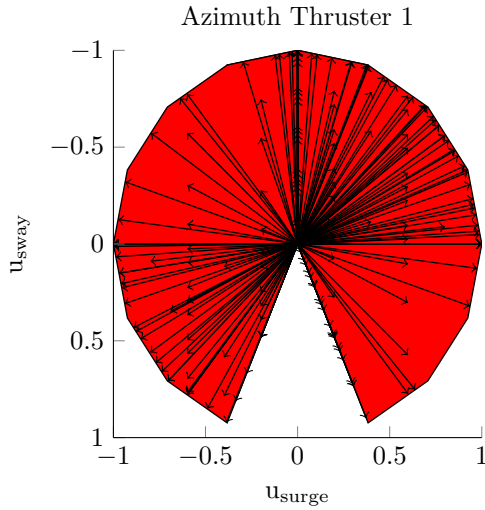
Figure 21: Calculated τ_Y for the allocated thrust when simulating a QP formulation for all possible demanded τ_c with 1.0 increments

10.3 Mixed Integer Linear Program

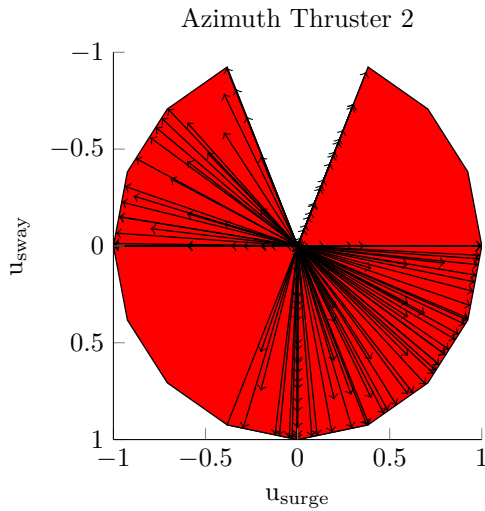
In this section, several simulations will be presented, gradually adding on more complexity. The first simulation will be equal to the LP formulation, only adding on a forbidden sector on each azimuth thruster prohibiting flushing. Second, the tunnel thruster will be made asynchronous before finally the thrust-power relationship is added as a piecewise linear approximation and power is minimized instead of thrust.

10.3.1 Synchronous tunnel thruster

When adding a forbidden sector, we see on figure 22 that no allocated thrusts lie in this sector, as was the intention. On the boarder of the forbidden sector there is an accumulation of allocated thrusts, which could be a problem regarding switching over the forbidden sector. The introduction of forbidden sectors has almost no impact on the resulting force, so it is clear that other solutions exist outside the forbidden sectors. As shown in figure 23 an error is introduced in τ_X when demanding $\tau_c = [2.5, 1.5, -0.5]^T$. This error occurs because the solution in the LP formulation had max thrust in forbidden sectors for both azimuth thruster 1 and 2, which is no longer feasible. This will be further discussed in the discussion section. The time needed to solve this formulation is negligible, i.e. $\approx 0.02s$.



(a) Allocated thrust from a MILP formulation for azimuth thruster 1 with a forbidden sector



(b) Allocated thrust from a MILP formulation for azimuth thruster 2 with a forbidden sector

Figure 22: Plots of the allocated thrust from a MILP formulation for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment

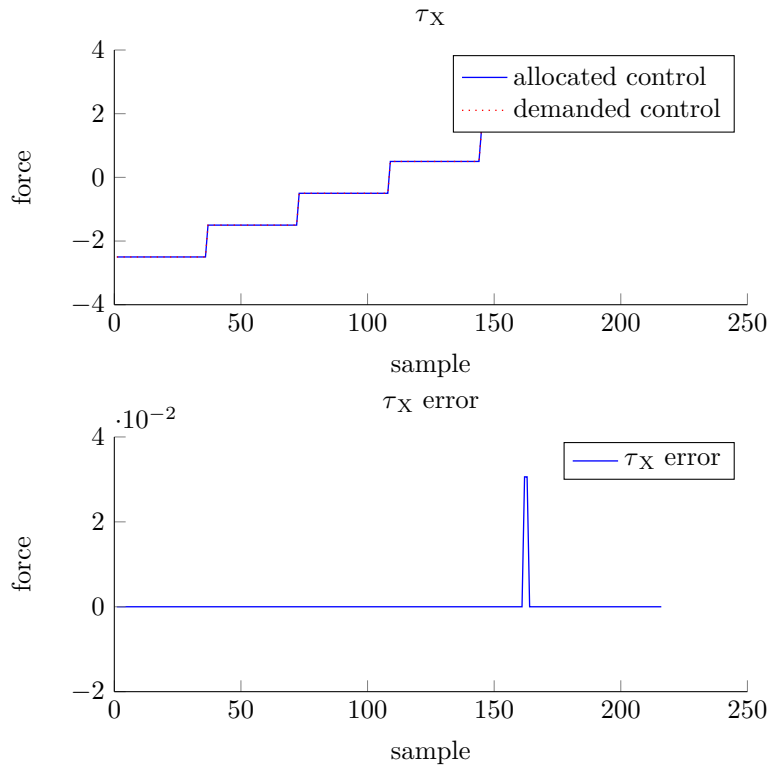


Figure 23: Calculated τ_X for the allocated thrust when simulating a MILP formulation, including forbidden sectors on the azimuth thrusters prohibiting flushing, for all possible demanded τ_c with 1.0 increments

10.3.2 Asynchronous tunnel thruster

The simulation with an asynchronous tunnel thruster was done such that the tunnel thruster had 70% efficiency in reverse, i.e. negative thrust. This does, as expected, introduce a larger error for negative τ_N and negative τ_Y , because negative tunnel thruster thrust strongly influence these forces. This is shown for τ_Y in figure 24. If the vessel is equipped with an asynchronous tunnel thruster, but the tunnel thruster is not modelled as asynchronous in the thrust allocation formulation, this would cause some serious issues, as the thrust allocation problem would think it allocates proper thrust to keep the vessels heading and position, while the vessel would not be able to produce it. The time needed to

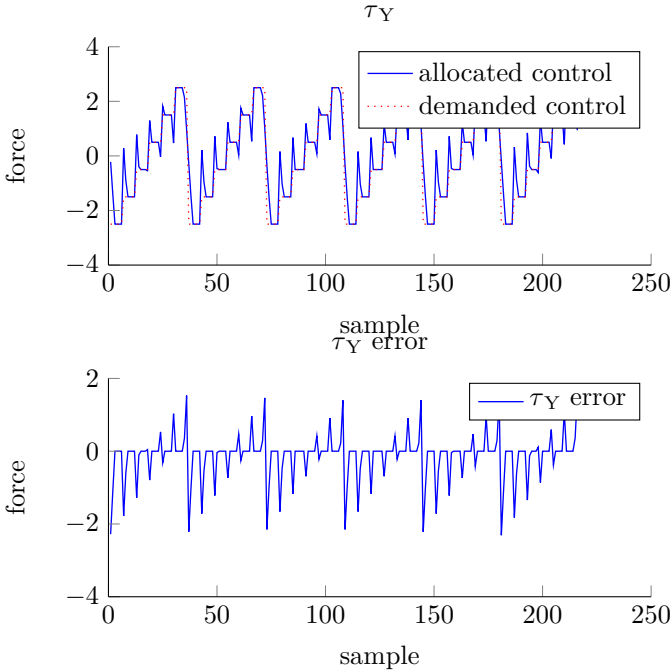


Figure 24: Calculated τ_Y for the allocated thrust when simulating a MILP formulation, including forbidden sectors on the azimuth thrusters prohibiting flushing, for all possible demanded τ_c with 1.0 increments

solve this formulation is negligible, i.e. $\approx 0.02s$.

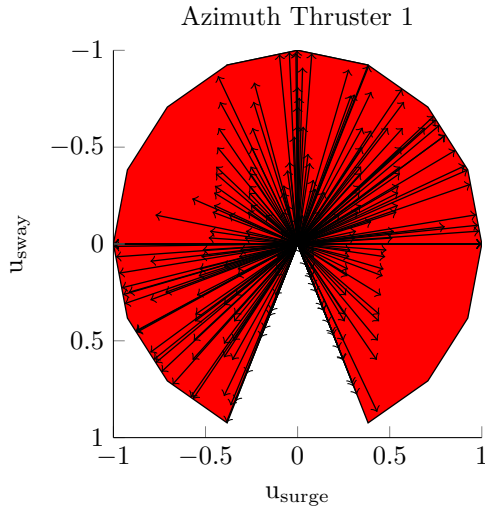
10.3.3 Power approximation

Finally, the relationship between thrust and power was added to the problem as a piecewise linear approximation, and power was minimized instead of thrust. In the simulation shown, the thrust-power relationship is approximated using 1000 lines per decision variable. This may sound a a lot, but the solver will choose the vertices in the approximation to get the most thrust per power, so in reality the decision variable is made discrete by this approximation, and approximation with many lines is necessary to counteract this effect. About half of the equations are cut by the pre-solver, which reduces some of the complexity, but this is still adding a lot of complexity to the problem.

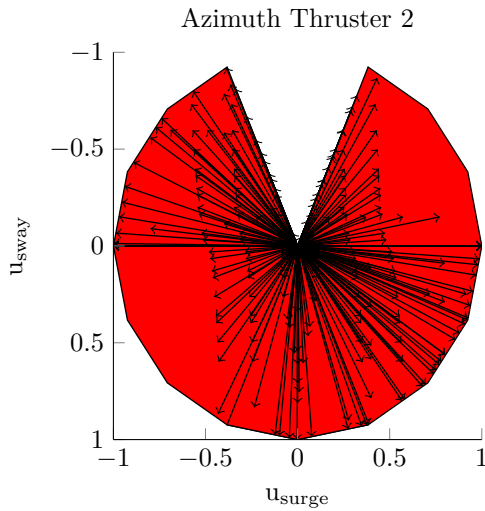
The resulting allocated thrust is now more evenly distributed between the thrusters than without the power minimization, as shown in figure 25a.

The resulting forces, shown for τ_Y in figure 26, are equal to what they were without power approximation, which was expected. Even though the thrust is better distributed between the thrusters, the errors are still penalized linearly.

The solver spends 0.38s solving the problem at an average, but some worst case scenarios takes more than 0.86s to solve. The resulting solving time is shown in figure 27. The time needed to solve the problem depends on the computer the thrust allocation is run on, and what other processes runs on it. It also varies from time to time, depending on scheduling and other variations, so this time can be reduced drastically. It is included here to show that it takes more time compared to the previous formulations.



(a) Allocated thrust from a MILP formulation for azimuth thruster 1 with a forbidden sector



(b) Allocated thrust from a MILP formulation for azimuth thruster 2 with a forbidden sector

Figure 25: Plots of the allocated thrust from a MILP formulation minimizing power for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment

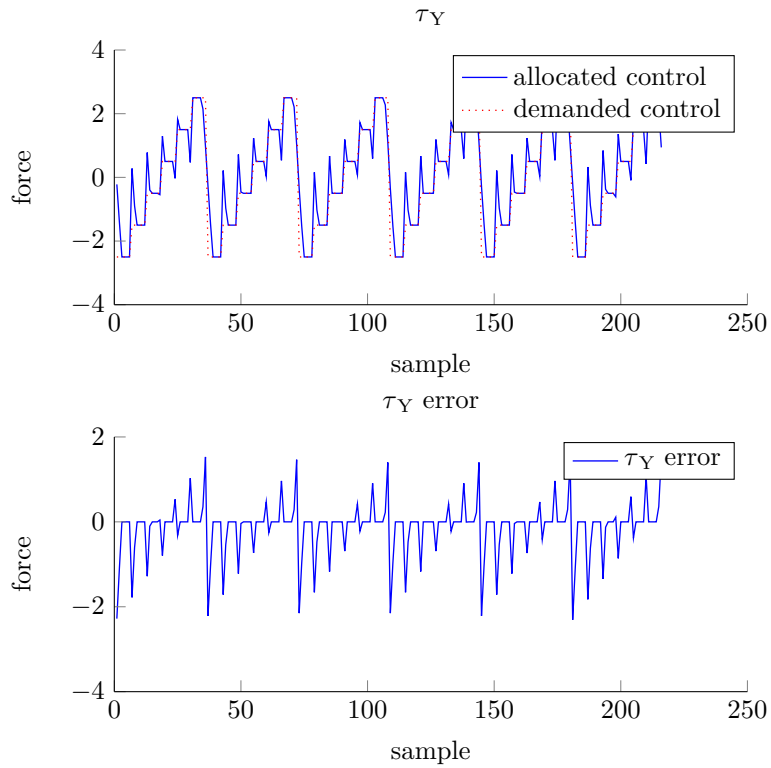


Figure 26: Calculated τ_Y for the allocated thrust when simulating a MILP formulation minimizing power, including forbidden sectors on the azimuth thrusters prohibiting flushing, for all possible demanded τ_c with 1.0 increments

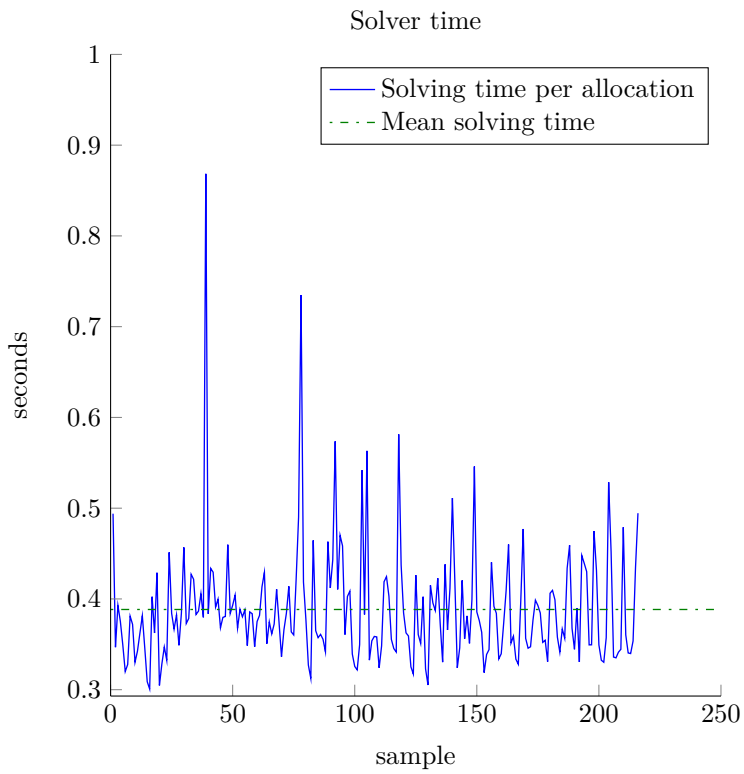


Figure 27: A plot of the time needed to solve the thrust allocation problem with the complete MILP formulation

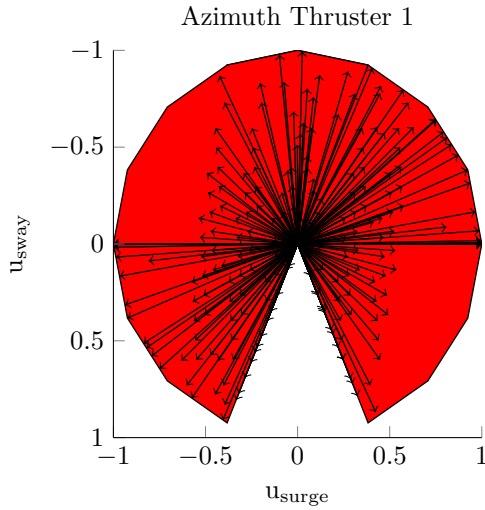
10.4 Mixed Integer Quadratic Program

In addition to the MILP formulation, a MIQP formulation was implemented and simulated. The formulation is the same as the MILP formulation without the power minimization, and switching out the cost function with a quadratic. The distribution of the allocated thrust from the MIQP formulation for the two azimuth thrusters are shown in figure 28.

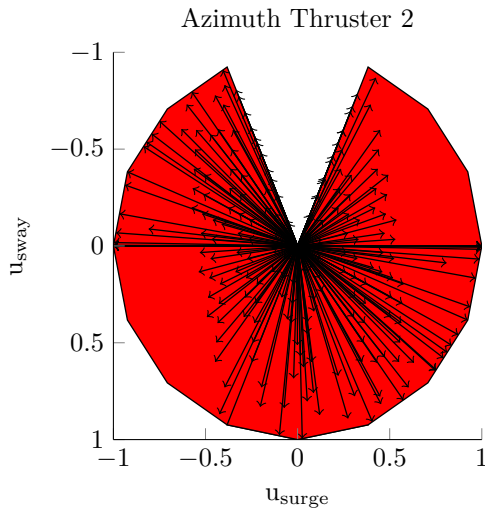
In this formulation, thrust is minimized and not power, but since the squared thrust is minimized, the distribution of allocated thrust is similar to the MILP formulation with power minimization.

As we saw on the LP and QP formulations, the errors was more evenly distributed between the forces with the QP formulation than the LP formulation, because error is penalized squared in the QP formulation. This is also the case with MIQP compared to MILP. Some errors are introduced in τ_X as shown in figure 29, while the errors in τ_Y and τ_N are smaller in magnitude. The plots of τ_Y and τ_N can be found in appendix A.

The MIQP formulation takes less time to solve than the MILP formulation. Although a MIQP solver typically is slower than a MILP solver, the complexity of the MILP formulation is so much greater, due to the piecewise linear approximation of power, that it still takes longer than the MIQP to solve. A plot of the time spent to solve the allocation problem for the MIQP formulation is shown in figure 30 where the mean time was $0.2s$ and the worst case takes $0.37s$. Thus, even the worst case for the MIQP formulation is slightly faster than the MILP formulation.



(a) Allocated thrust from a MIQP formulation for azimuth thruster 1 with a forbidden sector



(b) Allocated thrust from a MIQP formulation for azimuth thruster 2 with a forbidden sector

Figure 28: Plots of the allocated thrust from a MIQP formulation for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment

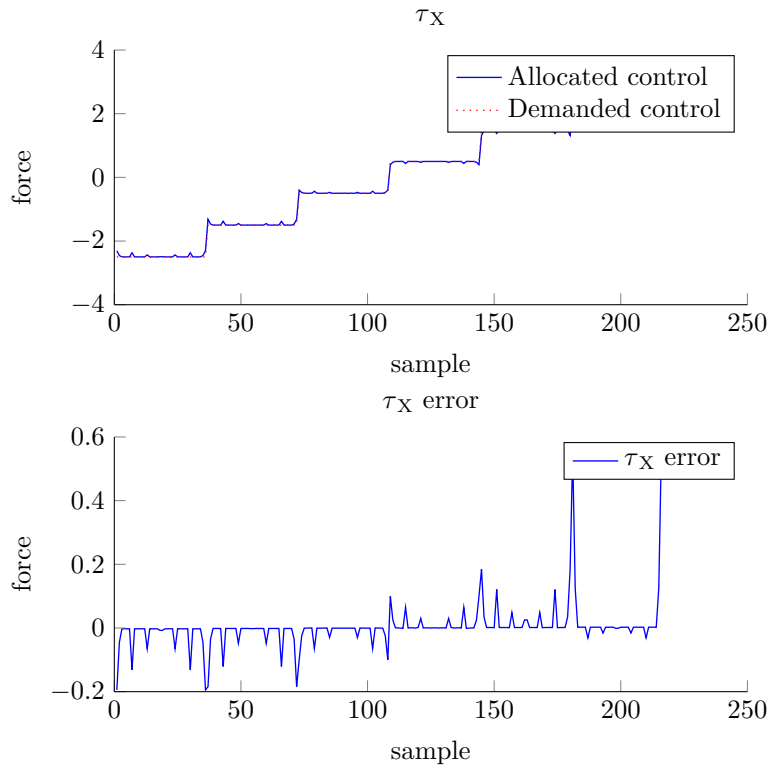


Figure 29: Calculated τ_X for the allocated thrust when simulating a MIQP formulation minimizing thrust, including forbidden sectors on the azimuth thrusters prohibiting flushing, for all possible demanded τ_c with 1.0 increments

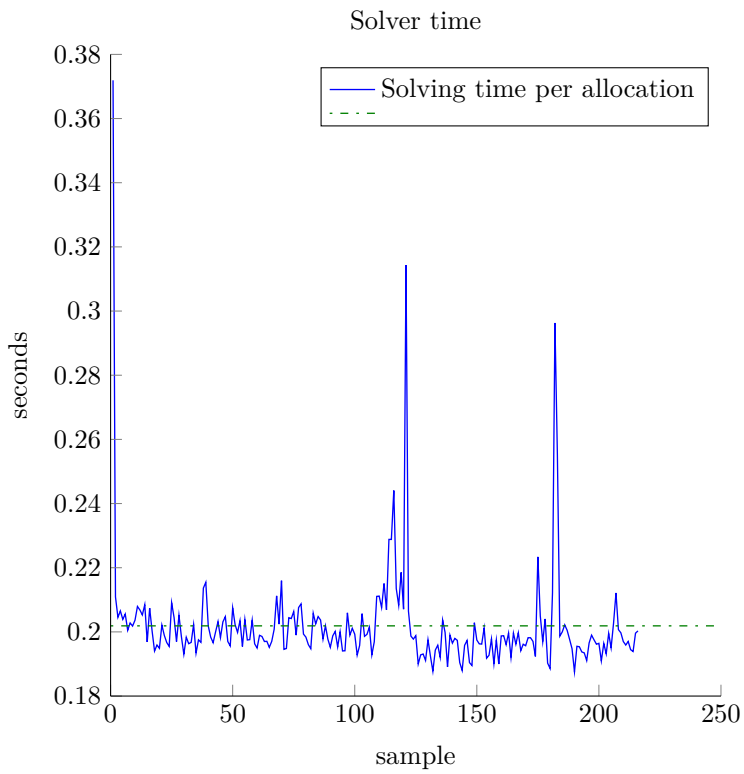


Figure 30: A plot of the time needed to solve the thrust allocation problem with the MIQP formulation

11 Discussion

In this section, the results from the simulations will be discussed and pros and cons for the different formulations will be assessed, in order to suggest what techniques works best for the problem. The modelling of the actuators will also be discussed in regards to their benefits and limitations.

11.1 Modelling

MILP is a powerful tool for modelling an actuator. Being able to represent a non-convex set as the union of several smaller convex sets, enables us to include forbidden sectors to the thrusters feasible set. This, combined with using piecewise linear approximations, provides the opportunity to model most anything. The downside of MILP is that it requires linear constraints. Linearisation of non-linear equations introduces some approximation errors. In order to minimize the error, more lines must be used in the approximation, which again increases the complexity of the problem. The number of lines needed for an accurate enough approximation of a non-linear expression highly depends on the expression. For modelling the thrust image of a thruster, not that many lines was needed to get a good enough approximation, but for the thrust-power relationship, many lines was needed. When using a piecewise linear approximation in the cost function, the solver will tend to choose the vertices in this approximation as solutions, essentially making the decision variables discrete. Many lines in the approximation is needed to counteract this effect.

11.2 Simulations

The LP formulation is clearly not an accurate enough description of the thrust allocation problem. The solutions given by this formulation will run azimuth thruster 1 at full speed most of the time, while azimuth thruster 2 is only used when azimuth thruster 1 is saturated. This is because thrust from these thrusters contribute equally to the vessels force because of their positioning on the vessel, and since thrust is penalized linearly, only one of them are used until it is saturated. The errors are unevenly distributed between the generalized forces when they are weighted equally. In the simulation presented in this report, τ_X is tracked correctly, sacrificing accuracy in τ_Y and τ_N . If this is good or bad is hard to say and dependent on the situation, but often one would prefer small errors in all forces compared to huge errors in some. This effect will also

occur for other types of errors where equally distributed errors may be more important.

When formulating the problem as a QP, the thrust and errors are penalized squared. The result is a more evenly distributed thrust between the thrusters as shown in figure 19 for the two azimuth thrusters. This will probably result in a more power efficient thrust allocation, although power is not explicitly minimized. The errors are also evenly distributed in a QP solution, which may be a desired property to have in a thrust allocation algorithm.

In order to include forbidden sectors, logical constraints must be added to the formulation, making it a mixed integer program. The first MILP formulation had forbidden sectors prohibiting flushing between the azimuth thrusters. The resulting distribution of thrust is uneven, equal to the LP formulation, except that no allocated thrust lies in the forbidden sectors. By including forbidden sectors prohibiting flushing, you avoid allocating thrust in areas that will affect the vessel differently than expected due to flushing. A challenge with this distribution of allocated thrust is switching back and forth over a forbidden sector. A way to deal with this problem would be to add switching over a forbidden sector into the problem formulation and minimize it.

Secondly, the response of the tunnel thruster was made asynchronous. Many tunnel thruster has asynchronous response, as they are constructed optimally for thrust in one direction. When an asynchronous tunnel thruster is mounted on the ship, it is crucial that also the model is asynchronous. The resulting errors is as expected larger in the direction the tunnel thruster has less effect, which it should be if the tunnel thruster is asynchronous. This effect could also be added to other thrusters, e.g. azimuth thrusters. Switching an azimuth thruster to reverse in stead of turning it 180° would be faster, and may be desired considering response time and wear and tare, even though you would have to use more power to achieve the same force in reverse compared to forward motion.

Finally, the relationship between thrust and power was added as a piecewise linear approximation in order to minimize power in stead of thrust. This results in a more accurate model of the problem, resulting in a better distribution of thrust between the thrusters. However, the errors are still unevenly distributed between the forces, because these are still linearly penalized. Again, if this is good or bad depends on the situation, and if one force is more important to track than the others, that error can be penalized more in the weighting vector. The errors in this report are a result of saturation of thrust due to the configuration of thrusters on the vessel, but this uneven distribution of error will also occur for other types of errors.

By having access to the power consumption, a limit can be put on the total power consumed, in order to not exceed the maximum output from the power generators. This has not been implemented in the thesis, but is trivial to do once power is included as a parameter in the problem. This has not been implemented because the thrusters on CSE1 is electrically driven and have access to more power than they can consume and hence, the limit of the total power used is the same as using each thruster at full speed, which is already included as a constraint. This would be relevant for a vessel with a power generator generating power for all the thrusters and propulsion systems on the vessel, not being capable to produce as much power as the thrusters and propulsion system are capable to consume. This would add one constraint to the problem, which is a negligible increase in the complexity of the problem.

In addition to the MILP formulation, a MIQP formulation was simulated, not including the power approximation. This formulation results in a distribution of allocated thrust similar to the MILP formulation, and the errors are more evenly distributed than the response of the MILP formulation. This formulation is also less complex compared to the MILP formulation, and even though a MIQP formulations typically takes longer to solve than a MILP formulation, the difference in complexity is so great that the MILP takes longer to solve than the MIQP.

12 Conclusion

The purpose of this thesis was to explore the possibilities of Mixed Integer Linear Programming (MILP) and, by using MILP, finding a good trade off between actuator model and problem formulation complexity. Thrust allocation is generally solved with non-linear solvers, which can not directly include logical and operational constraints in the formulation, which is the purpose of using MILP instead.

In this thesis it has been shown that MILP formulation can be used to include logical constraints like forbidden sectors in the problem formulation. By using a piecewise linear approximation on the thrust-power relationship, minimizing power instead of thrust has been made possible, which also makes it possible to constrain the total amount of power used by the thrusters. By using piecewise linear approximations, any non-linearity can be included in a MILP framework. The downside of MILP is some approximation errors introduced when linearising non-linear constraints. When using piecewise linear approximations, the downside is an increased complexity because for many functions, many lines must be used to create an accurate enough approximation. This is especially important for piecewise linear approximations in the cost function, because the solver will choose vertices on the approximation as solutions, essentially making the decision variable discrete.

I would recommend using MIQP over MILP whenever the equations in the cost function can be approximated as quadratic. This is because approximation them as piecewise linear causes the solver to yield discrete solutions, only choosing the vertices in the approximation, which has to be counteracted by using many lines in the approximation. Using many lines in the approximation will increase the complexity of the problem greatly, making the MIQP problem more efficient to solve.

12.1 Future work

In this thesis, an accurate model of a specific vessel is not created and the thrust allocation algorithm is not implemented on an actual vessel. This should be done, together with a traditional non-linear thrust allocation algorithm, in order to compare the two techniques. More constraints can also be included in the formulation, e.g.

- Enable azimuth thrusters to go in reverse with asynchronous response
- Include the dynamic constraints on change in thrust, in order to minimize it
- Minimize switching over a forbidden sector
- Instead of completely forbidding a sector, allow operation in the sector, but at reduced efficiency
- Enable the possibility of switching thrusters on and off
- Adjust the model for propeller-rudder pairs such that the effect of the thrust decays for increasing deflection angle on the rudder
- Use a piecewise linear cost function in the MILP formulation to get a more even distribution of errors, similar to how the thrust-power relationship was included

References

- [Balchen et al., 1980] Balchen, J. G., Jenssen, N. A., Mathisen, E., and Sælid, S. (1980). A dynamic positioning system based on kalman filtering and optimal control. *Modeling, Identification and Control*, 1(3):135–163.
- [Balchen et al., 1976] Balchen, J. G., Jenssen, N. A., and Sælid, S. (1976). Dynamic positioning using kalman filtering and optimal control theory. *Automation in offshore oil field operation*.
- [Fossen, 2011] Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd, West Sussex, UK.
- [GUROBI, 2015] GUROBI (2015). Mip basics. <http://www.gurobi.com/resources/getting-started/mip-basics>.
- [IBM,] IBM. Cplex optimizer. <http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/>.
- [International Maritime Organization, 1994] International Maritime Organization (1994). Guidelines for vessels with dynamic positioning systems. Technical Report MSC/Circ.645, London.
- [Johansen and Fossen, 2013] Johansen, T. A. and Fossen, T. I. (2013). Control allocation - a survey. *Automatica*, 49(1087-1103).
- [Johansen et al., 2008] Johansen, T. A., Fulseth, T. P., Tøndel, P., and Fossen, T. I. (2008). Optimal constrained control allocation in marine surface vessels with rudders. *Control Engineering Practice*, 16(457-464).
- [Löfberg,] Löfberg, J. What is yalmip. <http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Main.WhatIsYALMIP>.
- [Löfberg, 2015a] Löfberg, J. (2015a). Big-m and convex hulls. <http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Tutorials.BigMAndConvexHulls>.
- [Löfberg, 2015b] Löfberg, J. (2015b). Hull. <http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Commands.Hull>.
- [Löfberg, 2015c] Löfberg, J. (2015c). Sos 2. <http://users.isy.liu.se/johanl/yalmip/pmwiki.php?n=Commands.SOS2>.

- [Nocedal and Wright, 2006] Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, USA.
- [Rindarøy, 2013] Rindarøy, M. (2013). Fuel optimal thrust allocation in dynamic positioning.
- [Ruth, 2008] Ruth, E. (2008). *Propulsion control and thrust allocation on marine vessels*. PhD thesis, Norwegian University of Science and Technology.
- [Skjetne and Øivind K. Kjerstad, 2013] Skjetne, R. and Øivind K. Kjerstad (2013). Recursive nullspace-based control allocation with strict prioritization for marine craft. *Control Applications in Marine Systems*, 9.
- [Veksler et al., 2012] Veksler, A., Johansen, T. A., and Skjetne, R. (2012). Thrust allocation with power management functionality on dynamically positioned vessels. *American Control Conference*.
- [Wit, 2009] Wit, C. D. (2009). Optimal thrust allocation methods for dynamical positioning of ships.

13 Appendix

13.1 Appendix A - Additional plots

Additional plots from the simulations.

13.1.1 LP

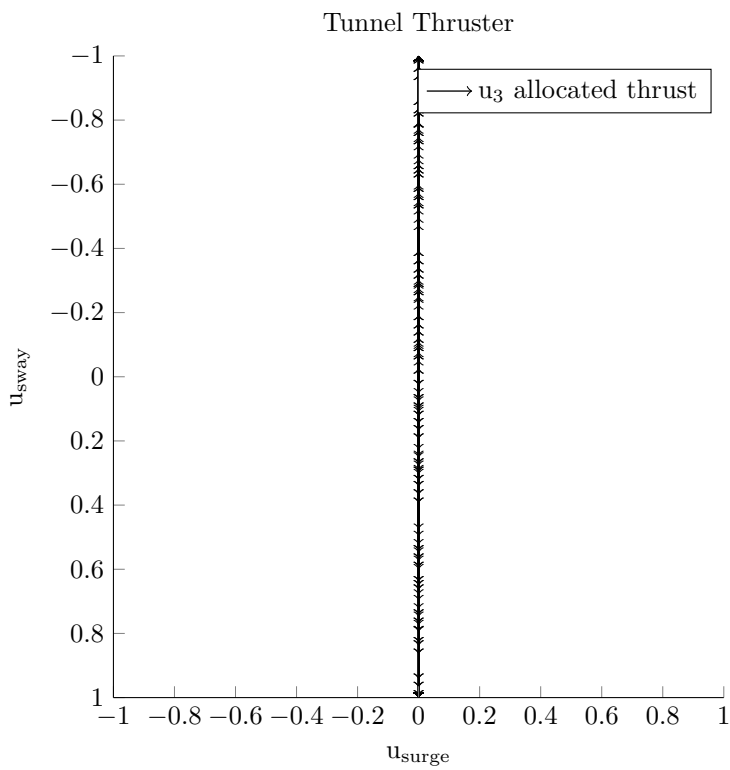


Figure 31: Plot of the allocated thrust for the tunnel thruster in an LP formulation

13.1.2 QP

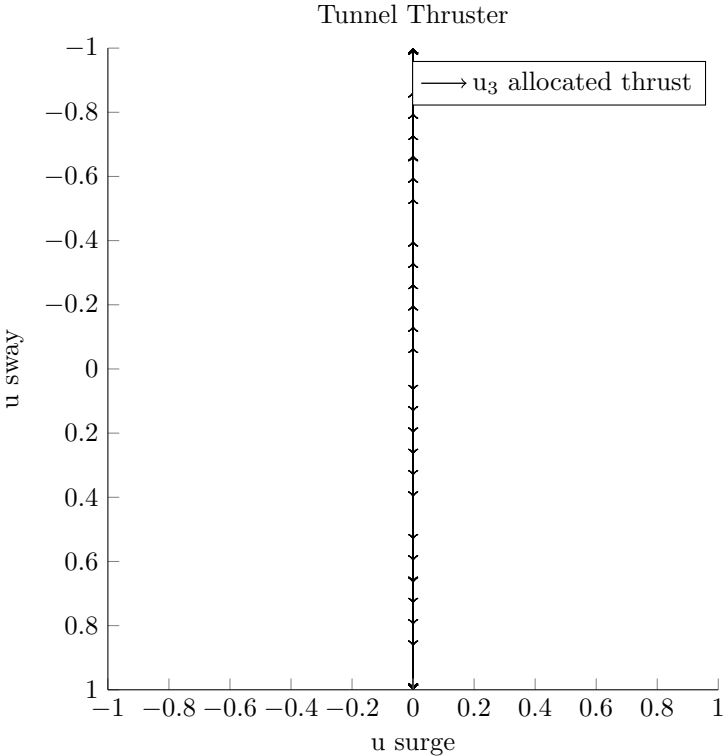


Figure 32: Plot of the allocated thrust for the tunnel thruster in a QP formulation

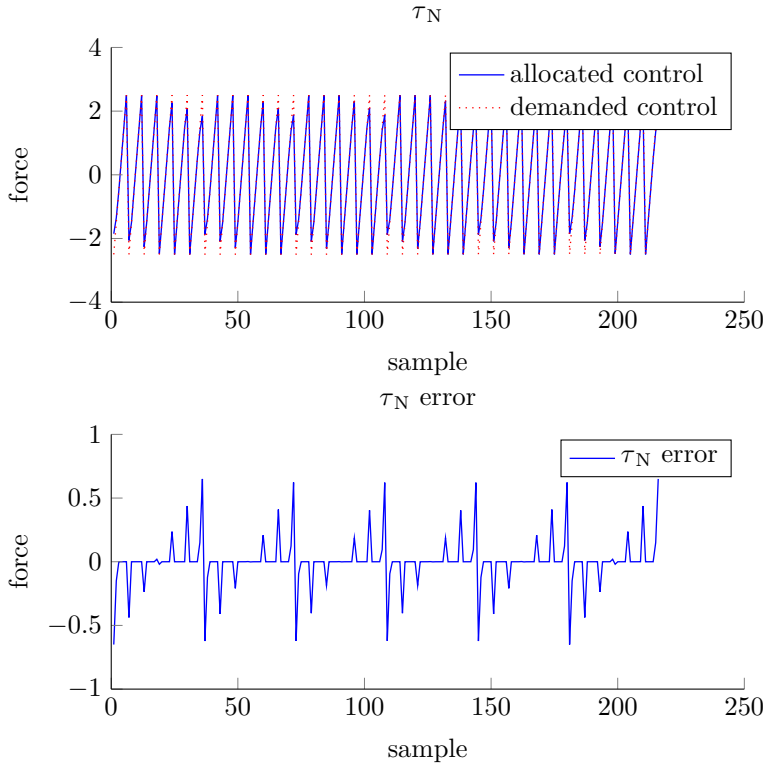


Figure 33: Calculated τ_N for the allocated thrust from a QP formulation when simulating for all possible demanded τ_c with 1.0 increments

13.1.3 MILP synchronous tunnel thruster

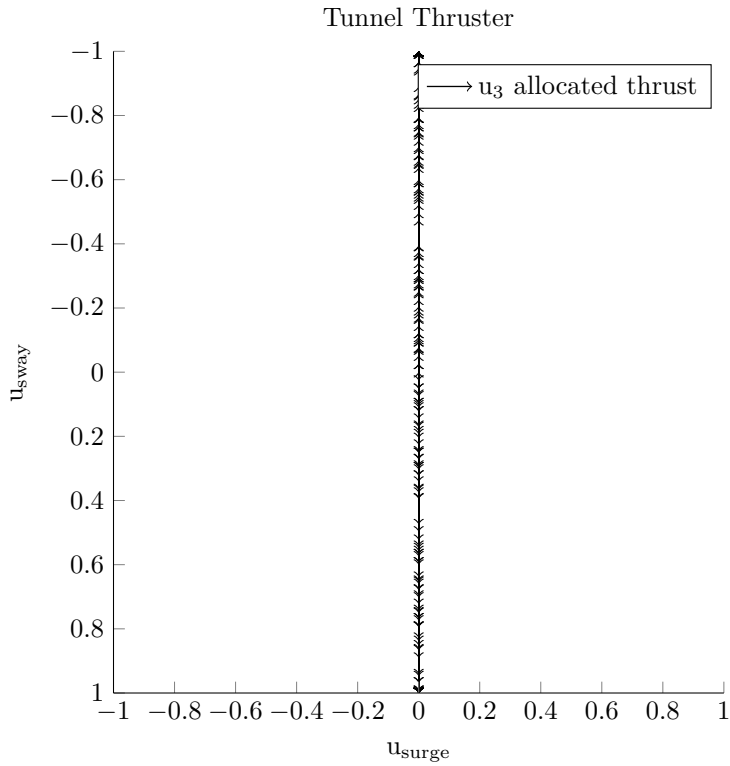


Figure 34: Plot of the allocated thrust for the tunnel thruster in a MILP formulation

13.1.4 MILP asynchronous tunnel thruster

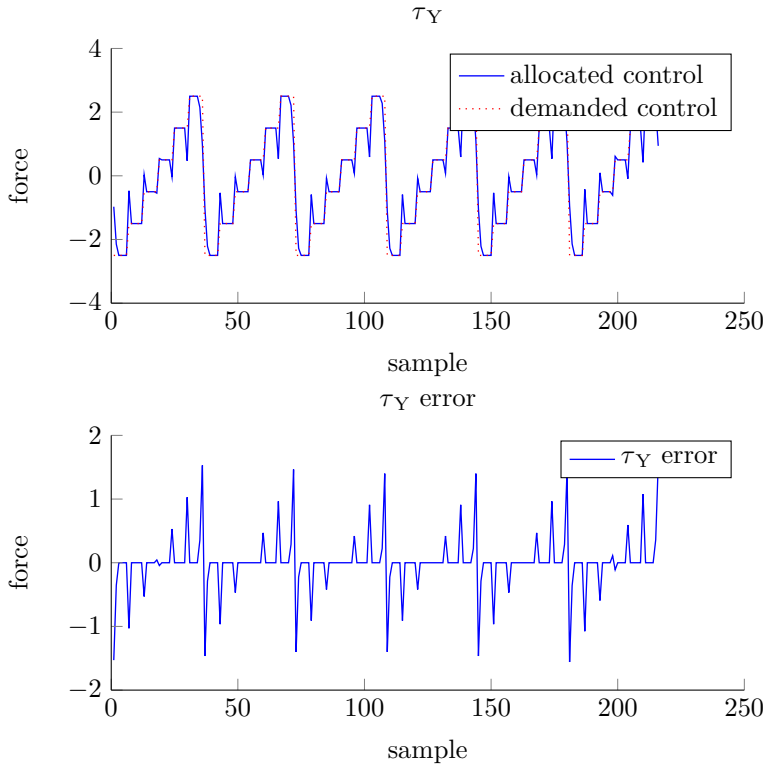


Figure 35: Calculated τ_Y for the allocated thrust from a MILP formulation when simulating for all possible demanded τ_c with 1.0 increments

13.1.5 MILP power minimization

13.1.6 MIQP

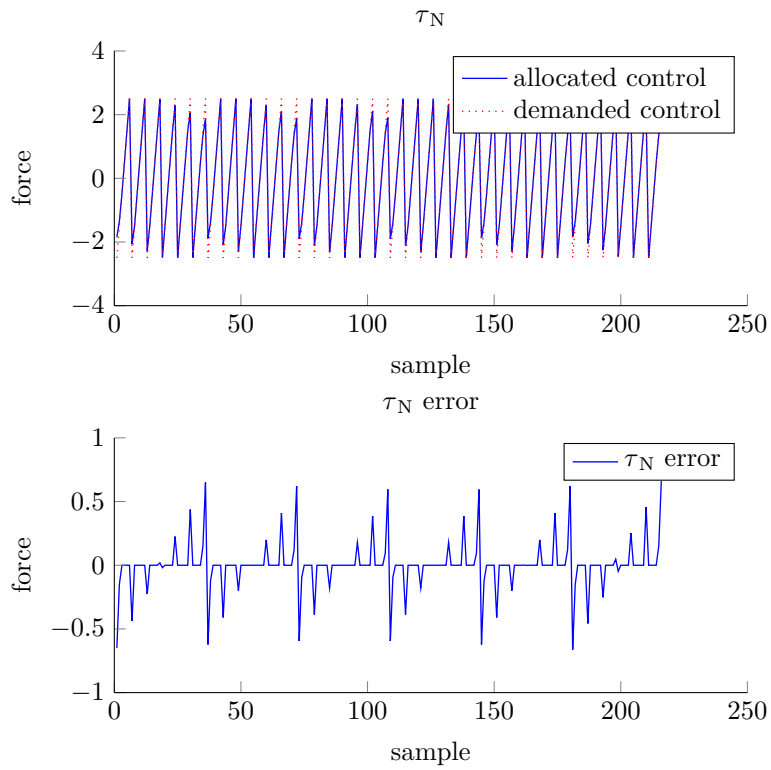
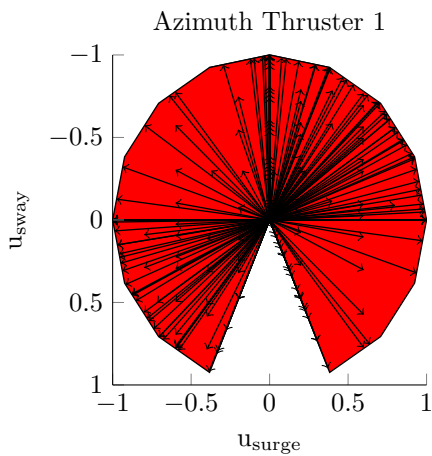
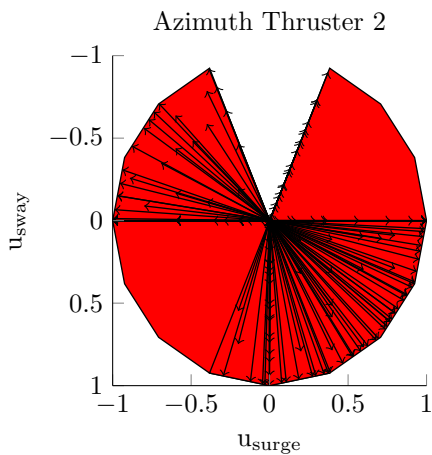


Figure 36: Calculated τ_N for the allocated thrust from a MILP formulation when simulating for all possible demanded τ_c with 1.0 increments



(a) Allocated thrust from a MILP formulation for azimuth thruster 1 with a forbidden sector



(b) Allocated thrust from a MILP formulation for azimuth thruster 2 with a forbidden sector

Figure 37: Plots of the allocated thrust from a MILP formulation for the two azimuth thrusters when iterating over all possible demanded τ_c with a 1.0 increment

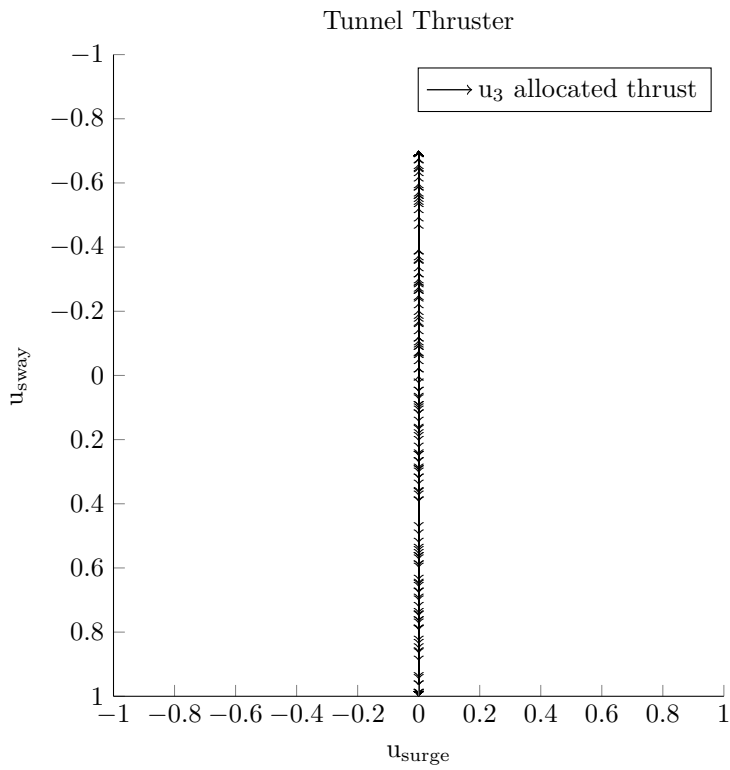


Figure 38: Plot of the allocated thrust for the tunnel thruster in a MILP formulation

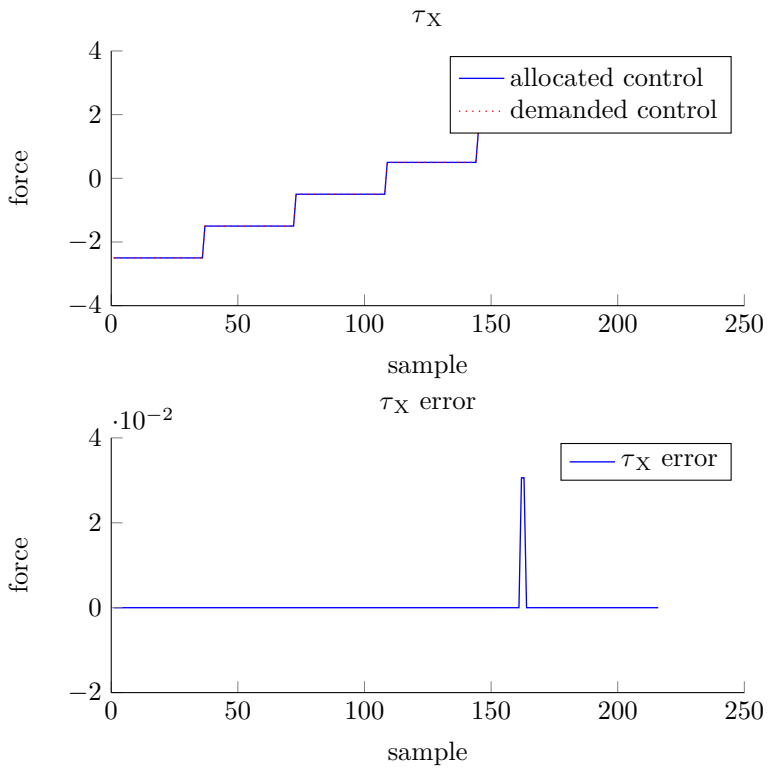


Figure 39: Calculated τ_X for the allocated thrust from a MILP formulation when simulating for all possible demanded τ_c with 1.0 increments

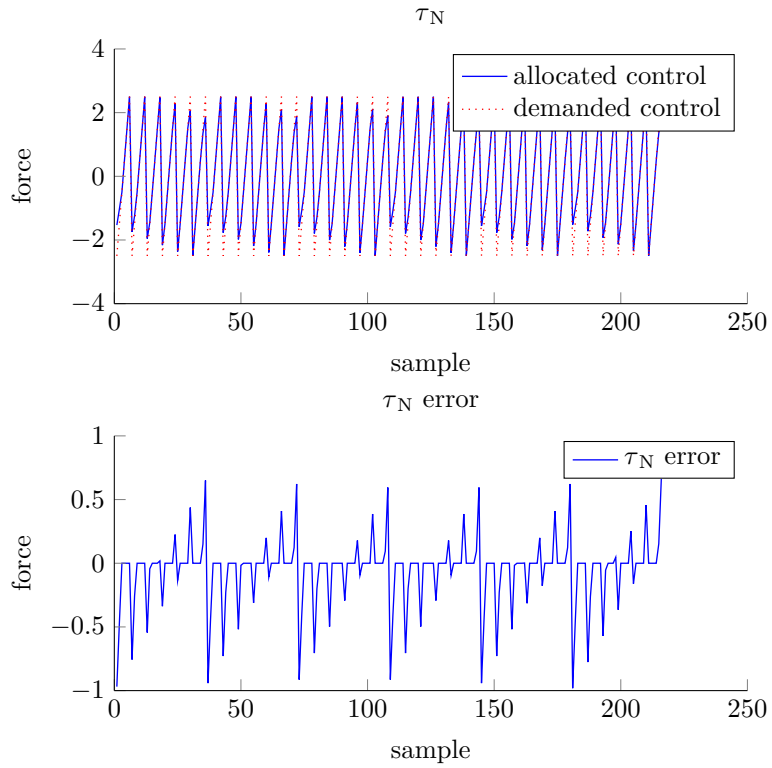


Figure 40: Calculated τ_N for the allocated thrust from a MILP formulation when simulating for all possible demanded τ_c with 1.0 increments

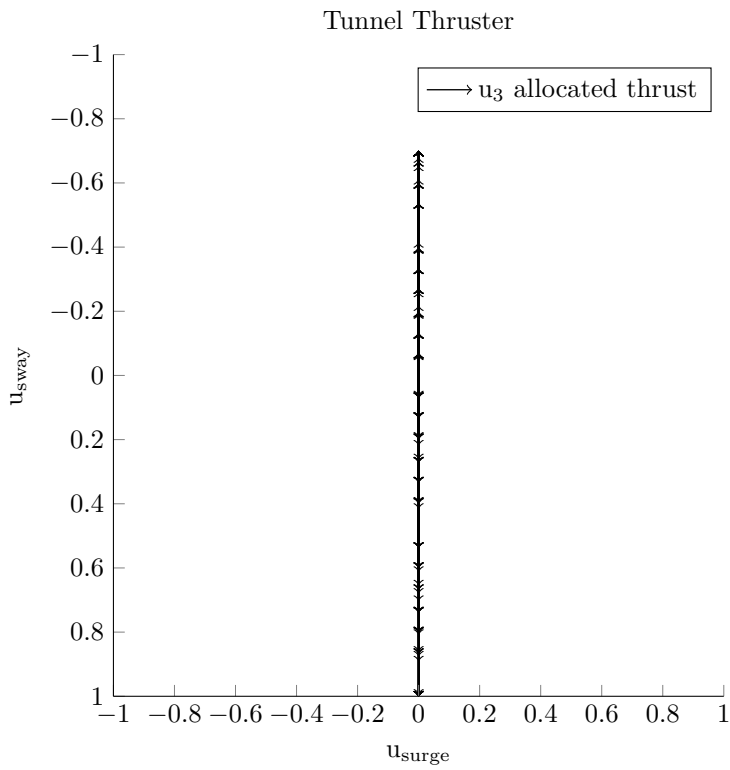


Figure 41: Plot of the allocated thrust for the tunnel thruster in a MILP formulation minimizing power

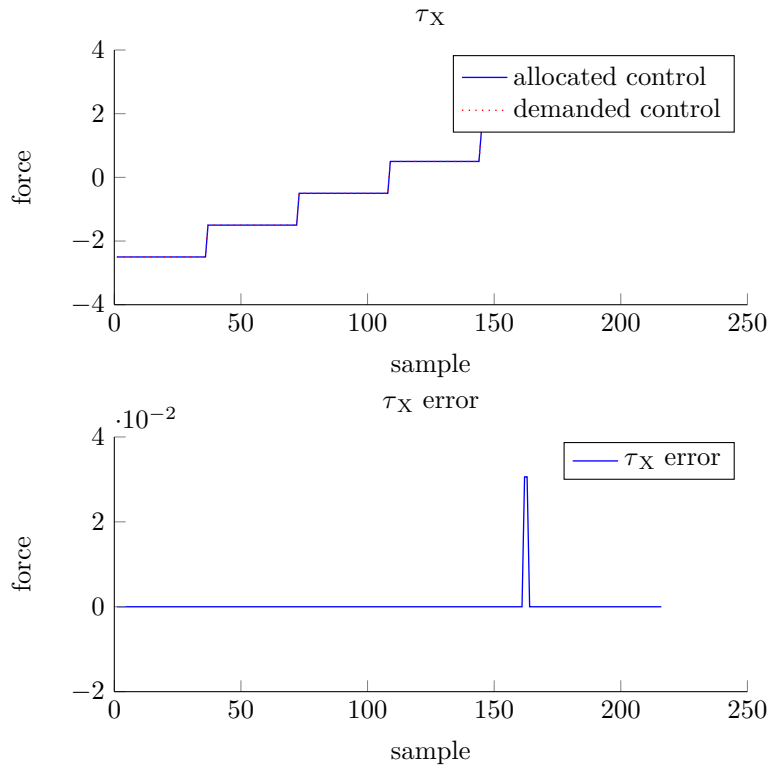


Figure 42: Calculated τ_N for the allocated thrust from a MILP formulation minimizing power when simulating for all possible demanded τ_c with 1.0 increments

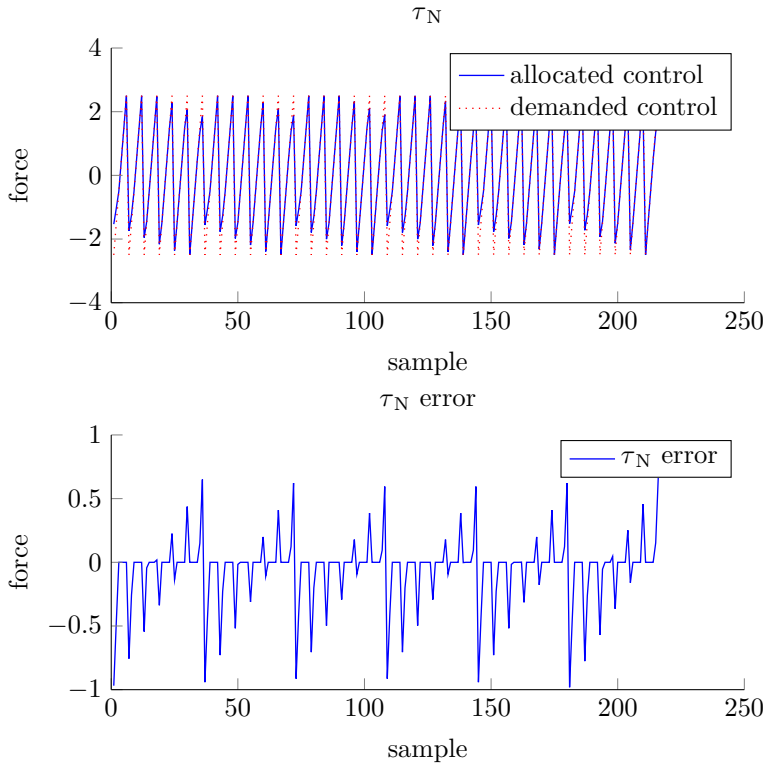


Figure 43: Calculated τ_N for the allocated thrust from a MILP formulation minimizing power when simulating for all possible demanded τ_c with 1.0 increments

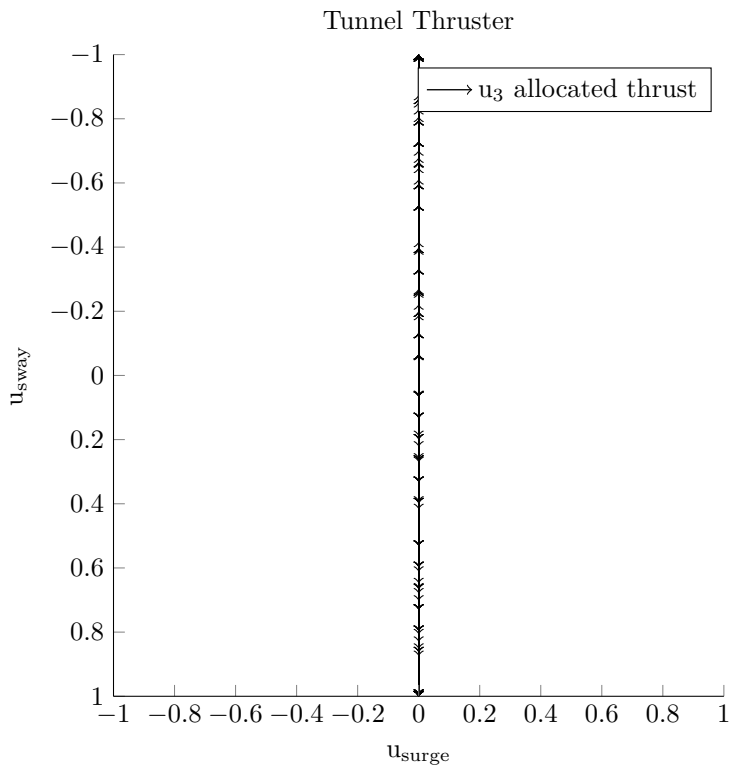


Figure 44: Plot of the allocated thrust for the tunnel thruster in a MIQP formulation minimizing thrust

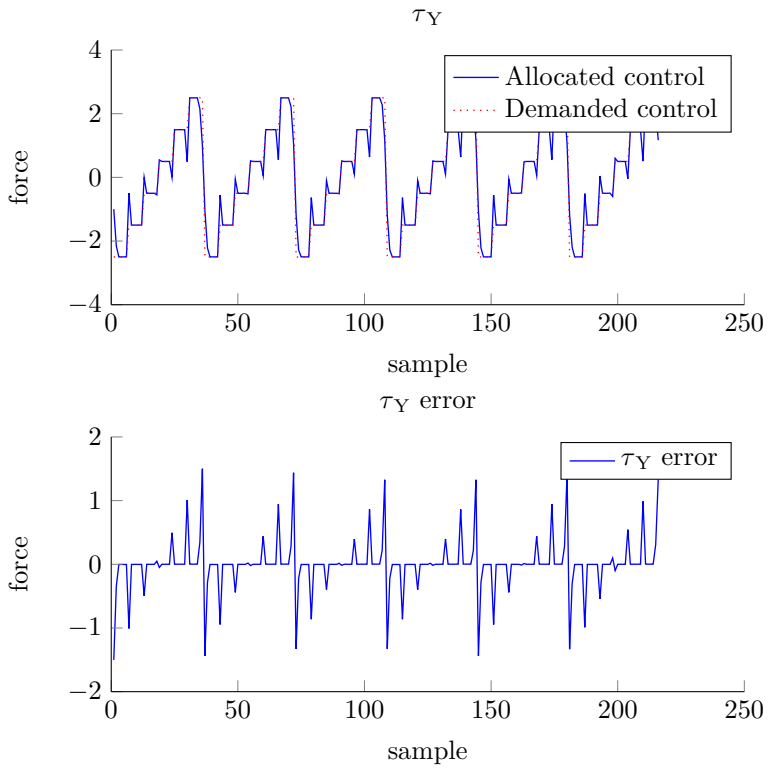


Figure 45: Calculated τ_Y for the allocated thrust from a MIQP formulation minimizing thrust when simulating for all possible demanded τ_c with 1.0 increments

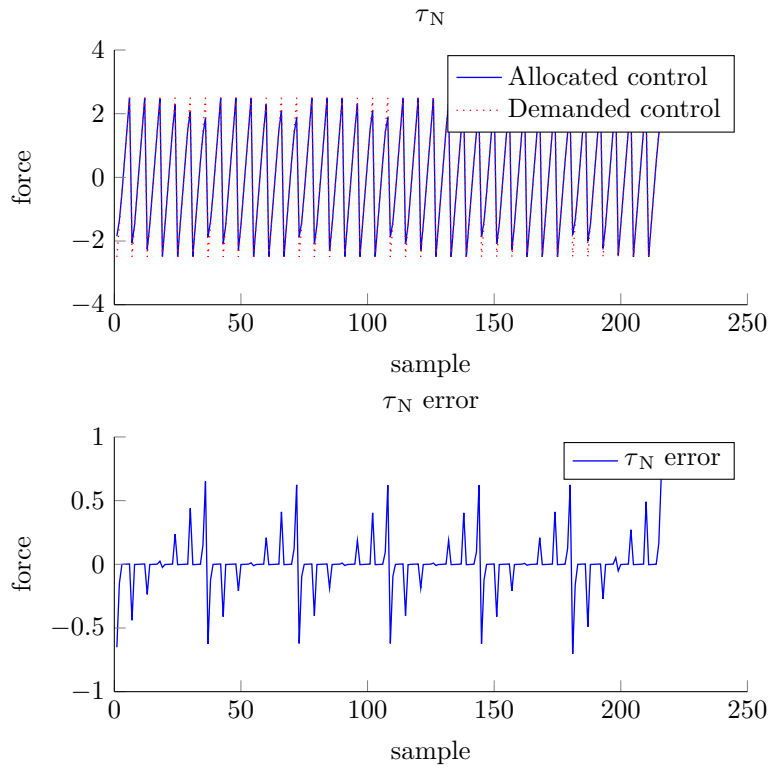


Figure 46: Calculated τ_N for the allocated thrust from a MIQP formulation minimizing thrust when simulating for all possible demanded τ_c with 1.0 increments