**NTNU – Trondheim**
Norwegian University of
Science and Technology

# Dynamic Thrust Allocation

Espen Joris Gottschal

Marine Technology
Submission date: June 2014
Supervisor: Roger Skjetne, IMT

Norwegian University of Science and Technology
Department of Marine Technology

# Abstract

The purpose of this Thesis is to research the literature of thrust allocation, thruster modelling and maneuvering design, as well as to design and simulate various dynamic thrust allocation algorithms. Vessels with dynamic positioning (DP) systems are high in demand in multiple industries due to their good abilities to keep position. An important component of the DP system is thrust allocation; algorithms which transform desired forces in surge, sway and yaw into thruster setpoints and rotation angles.

Two dynamic thrust allocation algorithms are developed, and a simulation model of a model vessel is developed and used to test the thrust allocation algorithms.

Findings indicate that the more complex the algorithms are, the more oscillatory the transients become. In addition, numerical instability is a significant concern because of the sensitivity of the barrier function used. Nonetheless, dynamic thrust allocation represents a powerful way to handle saturations.

# Sammendrag

Formålet med rapporten er å undersøke litteraturen om thrust allokering, thruster modellering og manøvrering design, samt å designe og simulere ulike dynamiske thrustallokeringsalgoritmer. Fartøy med dynamisk posisjonering (DP) systemer er høyt etterspurt i flere bransjer på grunn av sine gode evner til å holde posisjonen. En viktig komponent i DP-systemet er thrustallokering, algoritmer som forvandler ønskede krefter i jag, svai og gir til input til thrusterne.

To dynamiske thrustallokeringsalgoritmer utvikles. For det tredje blir en simuleringsmodell for en modellfartøy utviklet og benyttet for testing av thrustallokeringsalgoritmene.

Funn tyder på at jo mer kompliserte algoritmene blir, jo mer oscillerende blir transientene. I tillegg er numerisk ustabilitet en signifikant bekymring på grunn av den anvendte barrierefunksjonens følsomhet. Likevel representerer dynamiske thrustallokering en effektiv måte å håndtere begrensninger på.

# Acknowledgements

This report is written during the spring 2014 as part of achieving my Master of Science. The research is done for the Norwegian University of Science and Technology (NTNU), and is not affiliated with industry.

I would like to thank my supervisor, Roger Skjetne, for showing me how beautifully mathematical tools can describe and shape our world.

I would like to thank my professor, Asgeir Sørensen, for encouraging me to keep going.

And I would like to thank my partner, Astrid Helene Brodtkorb, for love and support along the way.

Trondheim, June 30, 2014

Espen Joris Gottschal

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Major contributions

The main contributions of this thesis are the two dynamic thrust allocation algorithms. In addition, insight gained during simulation of said algorithms may be of use to readers.

## 1.2 Organization of the thesis

First, the theory behind thrust-allocation, low-level thruster control and maneuvering theory is shortly explained with references to relevant publications. Then, mathematical models of thrusters are developed for simulation and control design purposes. After that, two dynamic thrust allocation algorithms are designed. Finally, the thrust allocation designs are simulated and compared to a benchmark test case.

# Chapter 2

# List of Symbols

| | |
|---|---|
| The Maneuvering Variable | $\xi$ |
| The State Variable | $x$ |
| The Desired State | $x_d$ |
| The Desired Generalized Forces | $\tau_d$ |
| Thruster Configuration Matrix | $H$ |
| Azimuthing thruster direction | $\alpha$ |
| Time | $t$ |
| Lyapunov Function | $V$ |
| Thruster Time Constants | $T$ |
| Thrust loss coefficient | $K_{T0}$ |
| Torque loss coefficient | $K_{Q0}$ |
| Propeller diameter | $D$ |
| Position in NED | $\eta$ |
| Input to the thrusters | $u$ |

# Chapter 3

# Theory

In this section previous work conducted on the thrust allocation problem is introduced.

## 3.1 Thrust Allocation

When designing a Dynamic Positioning system, it is usully preferable to make the design modular. This allows for separate testing and reuse of the modules. Figure 3.1 shows the major modules of a DP-control system.

Thrust allocation is one of the modules of a DP-system, and is the mapping between generalized forces in the relevant degrees of freedom and the actual signals to the thrusters which produce said forces, see Figure 3.2. As such, it can be viewed as medium-level, with the overall DP-controller being the high-level part and the local control of the thrusters being the low-level part.

Consider the nonlinear system

$$\dot{x} = f(t, x, \tau) \tag{3.1}$$
$$\tau = h(t, x, u), \tag{3.2}$$

where $t$ is the time, $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^n$ is the control input vector, containing the control input to the thrusters, and $\tau \in \mathbb{R}^p$ is a vector of virtual controls from the high level-controller, typically moments and forces in a mechanical system. During a high-level control design, the virtual control $\tau$ is treated as an available input, although it can only be manipulated indirectly via the input $u$ through (3.2). Mapping the requested $\tau$ to an input $u$ is the control allocation task.

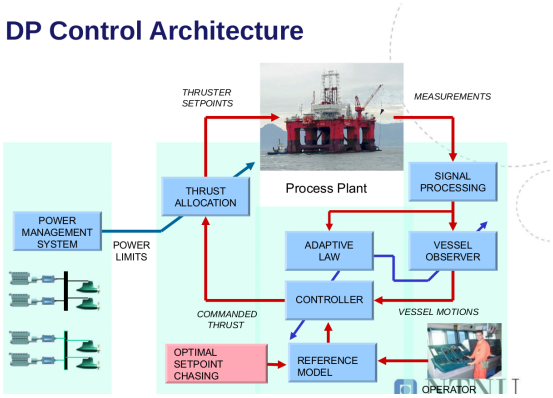


Figure 3.1: Block Diagram of DP-control system. Courtesy: Asgeir Sørensen

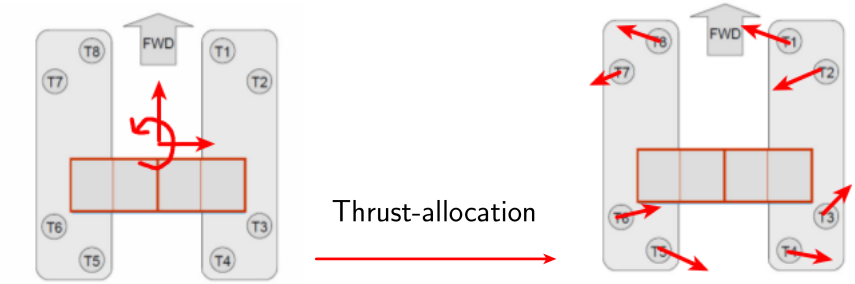


Figure 3.2: Thrust allocation

Normally the control allocation problem is solved as a static problem $\tau = H(\alpha)u$, where $\alpha = [\alpha_1, \alpha_2, ..., \alpha_r]$ are the angles of azimuthing propellers or rudders and $H \in \mathbb{R}^{n \times r}$ is the thruster configuration matrix, which specifies where on the hull the thrusters are located. Each thruster will also have dynamics $\dot{u} = g(u, v)$, with $v$ as input, to determine how fast each thruster can respond to thruster commands.

In this work, only over-actuated systems will be considered, meaning $n > p$.

A range of control allocation methods is presented in Fossen and Johansen [2006], including Linear Quadratic Unconstrained methods, Linear Quadratic Constrained methods and Nonlinear Constrained methods of control allocation.

## Linear Quadratic Unconstrained least-squares solution

Here, the optimization problem

$$\begin{aligned} &\min_f \{J = f^T W f\} \\ &subject\ to: \quad \tau - Hf = 0_{n \times 1}, \end{aligned} \tag{3.3}$$

where $W$ is a positive definite matrix, $J$ is a measure of cost, $f$ is the amount of force produced by each thruster, $\tau$ are the generalized forces, and $H$ is the thruster configuration matrix. This yields the explicit solution

$$f = H^\dagger \tau \quad H^\dagger = W^{-1} H^T (H W^{-1} H^T)^{-1}, \tag{3.4}$$

where $H^\dagger$ is the generalized inverse. Note that this solution is valid for all thrust-directions $\alpha$, but not optimal for all $\alpha$. If, however, the thrust-vectors are decomposed into a surge and sway-component:

$$F_{xi} = f_i \cos \alpha_i, \quad F_{yi} = f_i \sin \alpha_i, \tag{3.5}$$

then equation (3.3) becomes linear and the optimal angles and thrust commands can be found as

$$\alpha_i = \arctan(\frac{F_{yi}}{F_{xi}}) \tag{3.6}$$

$$u_i = \frac{1}{k_i} \sqrt{F_{xi}^2 + F_{yi}^2} \tag{3.7}$$

where $k_i$ is the maximum amount of thrust that this thruster can produce.

This solution is optimal, but does not take constraints into account. For industrial systems optimal solutions are desired in order to reduce power consumption, take into account actuator limits, reduce wear and tear and avoid forbidden sectors and overload to the power system.

## Linear Quadratic Constrained Least-Squares solution

As described by Tøndel, Johansen, and Bemporad [2003], the constrained optimization problem can be formulated as

$$\begin{aligned} \min_{f,s,\bar{f}} \{J = f^T W f + s^T Q s + \beta \bar{f}\} \\ subject\ to: \\ Hf = \tau + s \\ f_{min} \leq f \leq f_{max} \end{aligned} \tag{3.8}$$

where $s \in \mathbb{R}^n$ is a vector of slack variables, $\bar{f} = \max\{f\}$ is the largest individual force, $J$ is a measure of cost, $W$ and $Q$ are weight matrices, and $H$ is the thruster configuration matrix. The first term corresponds to the least squares criterion, while the third term is introduced to minimize the largest single force and $\beta \geq 0$ controls the relative weighting of the criteria. Since $W > 0$ and $Q > 0$, the problem is convex, ensuring existence of a global solution. However, this solution is only valid for nonrotatable actuators. If azimuthing thrusters are included, the problem becomes nonlinear, and methods such as nonlinear constrained solutions can be applied.

## Nonlinear Constrained solution

When azimuthing thrusters are added to the control-allocation problem, it becomes, in general, nonconvex and hard to solve. The primary constraint is

$$\tau = T(\alpha) f. \tag{3.9}$$

The azimuth angles $\alpha$ and control inputs must be computed at each sample instant while both being subject to rotation-rate and maximum-thrust constraints. In addition, the generalized inverse $H^\dagger$ may not always exist due to singularities. The practical meaning of this is that no thrust can be delivered in certain directions, posing challenges for the maneuverability of the vessel. The criterion to be minimized is:

$$\begin{aligned}\min_{f,\alpha-\alpha_0,s}\{J = &\sum_{i=1}^{r}\bar{P}_i|f_i|^{3/2} + s^T Q s \\ &+(\alpha-{\alpha_0}^T)\Omega(\alpha-\alpha_0) \\ &+\frac{\rho}{\epsilon+\det(T(\alpha)W^{-1}T^T(\alpha))}\} \\ &\quad subject\ \ to \\ &T(\alpha)f = \tau + s \\ &f_{min} \leq f \leq f_{max} \\ &\alpha_{min} \leq \alpha \leq \alpha_{max} \\ &\Delta\alpha_{min} \leq \alpha - \alpha_0 \leq \Delta\alpha_{max}\end{aligned} \tag{3.10}$$

There are several ways to solve such a problem. In Johansen [2004], a control-Lyapunov approach has been used to develop an optimal, asymptotically stable dynamic control allocation solution. In Johansen, Fossen, and Berge [2004], the problem is approximated as a convex QP problem by assuming quadratic power consumption and that singularity avoidance can be achieved by linear extrapolation from the last azimuth angle. Then, an iterative approach is used to compute the control inputs and azimuth angles for each timestep.

## Conference Article

In Johansen [2004], a dynamic approach is taken to thrust-allocation. Using a Lyapunov design method, and based on a range of assumptions, it is shown that the derivatives of the Lagrangian converge to zero exponentially fast, meaning an optimal solution has been found.

This is done by considering (3.2), and assuming there exists a virtual control $\tau_c$ that exponentially stabilizes the equation. Then, the optimization problem is defined as

$$\min_u J(t,x,u) \quad subject\ to\ \ \tau_c = h(t,x,u). \tag{3.11}$$

This can in turn be formulated as a Lagrangian

$$\mathcal{L}(t,x,u,\lambda) = J(t,x,u) + (\tau_c - h(t,x,u))^T\lambda \tag{3.12}$$

with a limiting optimal set $E^*$:

$$E^* = \{(x, u, \lambda \in \mathbb{R}^{n+r+p} \mid x = 0, \\ \lim_{t\to\infty} \frac{\partial \mathcal{L}}{\partial u}(t, x, u, \lambda) = 0, \ \lim_{t\to\infty} \frac{\partial \mathcal{L}}{\partial \lambda}(t, x, u, \lambda) = 0\} \tag{3.13}$$

Then a Lyapunov function is defined as:

$$V(t, x, u, \lambda) = V_0(t, x) + \frac{1}{2}(\frac{\partial \mathcal{L}^T}{\partial u}\frac{\partial \mathcal{L}}{\partial u} + \frac{\partial \mathcal{L}^T}{\partial \lambda}\frac{\partial \mathcal{L}}{\partial \lambda}) \tag{3.14}$$

where $V_0$ is the Lyapunov function showing that $\tau_c$ exponentially stabilizes the system in (3.2). Then, (3.14) is differentiated with respect to time and shown to be negative definite. Since the local minima of (3.11) satisfy the first order optimality conditions, the control effort is minimized dynamically, in a separated update law running in parallel with the rest of the control system.

An advantage of the dynamic approach is computational efficiency. If one were to solve (3.11), using for example quadratic programming, at each time instant in a discrete system, a significant amount of computing power is required. However, static or quasi static models have an advantage in modularity.

## 3.2 Thruster Control

### Torque and Power-Control

Traditionally, low level thruster control has been accomplished by using speed-control. This works fine for ships in transit, but causes unfortunate load variations with increased wear and tear and increased risk of blackout when operating in DP or low-speed maneuvering. An improved scheme based on power and torque control is presented in Sørensen, Ådnanes, Fossen, and Strand [1997], reducing significantly the power and torque-peaks experienced using speed-control.

### Anti-Spin Control

Building on the work done in [Sørensen et al., 1997], [Øyvind N. Smogeli, Sørensen, and Minsaas, 2006] addresses the problems encountered with the power and torque control scheme when the propeller experiences ventilation and/or in-and-out-of-water effects. These effects cause the torque to drop significantly, and since the torque is being controlled, the rotational speed

will be increased. These large variations in propeller speed cause increased wear-and-tear due to the large accelerations. Motivated by anti-spin for car-wheels, anti-spin thruster control is used to prevent this. Ventilation is detected using one of several available methods, triggering a switch to the traditional speed-control for the duration of the ventilation-event.

## 3.3 Maneuvering Theory

The Maneuvering Control Problem is presented in Skjetne [2005], and is a combination of path-following and path-tracking with two subtasks. The geometric task seeks to ensure that the path is followed, while the dynamic task states how the path should be followed. The maneuvering design was generalized from a one dimensional path to a $q$ dimensional manifold in [Skjetne, Jorgensen, and Teel, 2011].

For a system output $y = h(x)$ where $h : \mathbb{R}^n \to \mathbb{R}^m$, the desired manifold is all points represented by the set

$$\mathcal{Q} := \{x \in \mathbb{R}^n : \exists \xi \in \mathbb{R}^q | h(x) = h_d(\xi)\} \tag{3.15}$$

where $q \leq m$ and the map $\xi \mapsto h_d(\xi)$ is sufficiently smooth. Given the parametrization $h_d(\xi)$ of the manifold an a dynamics assignment on the manifold, the **Maneuvering Problem** is comprised of the two tasks:

1. **Geometric task:** For some absolutely continuous function $\xi(t)$, force the output $y$ to converge to the desired manifold $h_d(\xi)$,

$$\lim_{t\to\infty} |y(t) - h_d(\xi(t))| = 0. \tag{3.16}$$

2. **Dynamic task:** Force $\dot{\xi}$ to converge to a desired dynamic assignment $f_d(\xi, y, t)$,

$$\lim_{t\to\infty} |\dot{\xi}(t) - f_d(\xi(t), y(t), t)| = 0. \tag{3.17}$$

In order to address the geometric task, (3.16) is reformulated as

$$\mathcal{A} = \{(\xi, x, p) \in \mathbb{R}^q \times \mathbb{R}^n \times \mathbb{R}_{\geq 0} : h(x) = h_d(\xi)\}, \tag{3.18}$$

where $p$ is included to represent a possible time-variation in $f_d(\xi, y, t)$ with dynamics $\dot{p} = 1$, $p(0) = t_0$. The geometric task then comprises of stabilizing (3.18), see [Skjetne, 2005] or [Teel, Panteley, and Loría, 2002] for information on set-stability.

Consider the dynamic system

$$\dot{x} = f(x, u, t), \quad y = h(x) \tag{3.19}$$

where for each $t \geq t_0$, $x(t) \in \mathbb{R}^n$ is the state vector, $u(t) \in \mathbb{R}^p$ is the control, $y(t) \in \mathbb{R}^m$ is the output, and $f : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions.

**Proposition 3.3.1** *Suppose there exists a control law*

$$u = \alpha(\xi, x, t), \tag{3.20}$$

*a smooth Lyapunov function $V : \mathbb{R}^q \times \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $K_\infty$-functions $\alpha_1$, $\alpha_2$, and a continuous positive definite function $\alpha_3$, such that for all $(\xi, x, p) \in \mathbb{R}^q \times \mathbb{R}^n \times \mathbb{R}_{\geq 0}$,*

$$\alpha_1(|(\xi, x, p)|)_{\mathcal{A}} \leq V(\xi, x, p) \leq \alpha_2(|(\xi, x, p)|)_{\mathcal{A}} \tag{3.21}$$

$$\begin{aligned} V^\xi(\xi, x, p) f_d(\xi, x, p) + V^x(\xi, x, p) f(x, \alpha(\xi, x, p), p) \\ + V^p(\xi, x, p) \leq -\alpha_3(|\xi, x, p|_{\mathcal{A}}). \end{aligned} \tag{3.22}$$

*Then, assuming that the closed-loop system*

$$\dot{\xi} = f_d(\xi, x, t) \tag{3.23}$$

$$\dot{x} = f(x, \alpha(\xi, x, t), t) \tag{3.24}$$

*is forward complete, the noncompact set (3.18) is UGAS, and this solves the maneuvering problem.*

The proof follows from Lyapunov arguments for noncompact sets, see [Teel, Panteley, and Loría, 2002] and [Skjetne, 2005].

**Proof** Let $\beta_v$ be a class-$\mathcal{KL}$ function such that

$$V(\xi(t), x(t), t) \leq \beta_v(V(\xi_0, x_0, t_0), t - t_0), \tag{3.25}$$

which follows from (3.21) and (3.22), where $x_0 = x(t_0)$ and $\xi_0 = \xi(t_0)$. Letting $\beta_{\mathcal{A}}(y, t) := \alpha_1^{-1}(\beta_v(\alpha_2(y), t))$ this gives

$$|(\xi(t), x(t), t)|_{\mathcal{A}} \leq \beta_{\mathcal{A}}(|(\xi_0, x_0, t_0)|_{\mathcal{A}}, t - t_0) \quad \forall t \geq t_0. \tag{3.26}$$

*Q.E.D.*

Using the update law (3.23), the dynamic task (3.17) is satisfied identically. However, since the dynamic assignment needs to be satisfied only in the limit, more possibilities exist.

**Proposition 3.3.2** *Suppose the conditions of Proposition 3.3.1 are satisfied. Let* $\omega : \mathbb{R}^q \times \mathbb{R}^n \times \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^q$ *be a continuous tuning function that for all* $(\xi, x, p) \in \mathbb{R}^q \times \mathbb{R}^n \times \mathbb{R}_{\geq 0}$ *satisfies*

1. $|\omega(\xi, x, p)| \leq \alpha_4(|(\xi, x, p)|_{\mathcal{A}})$ *where* $\alpha_4$ *is a class-*$\mathcal{K}$ *function, and*
2. $V^{\xi}(\xi, x, p)\omega(\xi, x, p) \geq 0$.

*Then, under the assumption that the system*

$$\dot{\xi} = f_d(\xi, x, t) - \mu\omega(\xi, x, t) \tag{3.27}$$
$$\dot{x} = f(x, \alpha(\xi, x, t), t) \tag{3.28}$$

*is forward complete, renders the set (3.18) UGAS and solves the maneuvering problem.*

The proof follows from Lyapunov arguments for noncompact sets.

There are several options for the design of $\omega(\xi, x, t)$. One option is a filtered gradient feedback as presented in Skjetne [2005]. This can be used to improve performance when noise or disturbances degrade the quality of the measurments.

Another option is a gradient tuning function as presented in [Skjetne, Teel, and Kokotovic, 2002] and [Skjetne et al., 2011].

# Chapter 4

# Mathematical Modelling

## 4.1 Propeller hydrodynamics

The relationship between propeller thrust $T_a$, torque $Q_a$, shaft speed n is given by:

$$T_a = f_T(\cdot) = K_T \rho D^4 n|n| \tag{4.1}$$

$$Q_a = f_Q(\cdot) = K_Q \rho D^5 n|n| \tag{4.2}$$

where $K_T$ and $K_Q$ are loss coefficients, $\rho$ is the density of water and $D$ is the propeller diameter [Sørensen, 2011, p. 300]. An open loop, quasi-static mapping is used here because measurements of the thrust or torque are not easily obtained.

The expressions for $K_T$ and $K_Q$ for deeply submerged propellers are found experimentally and can be described by the following parameters:

$$K_T = f_1(\theta_p, x_p) = f_1(J_a, \frac{P}{D}, \frac{A_E}{A_O}, Z) \tag{4.3}$$

$$K_Q = f_2(\theta_p, x_p) = f_2(J_a, \frac{P}{D}, \frac{A_E}{A_O}, Z, R_n, \frac{t}{c}) \tag{4.4}$$

where $J_a = V_a/(nD)$ is the advance ratio, $V_a$ is the inflow velocity to the propeller, $P/D$ is the pitch ratio, $A_E/A_O$ is the expanded-area ratio, $Z$ is the number of blades, $R_n$ is the Reynolds number, $t$ is the maximum thickness of the blade section and $c$ is the chord length of the blade section. $\theta_p$ and $x_p$ represent static and dynamic input parameters, respectively. [Sørensen, 2011, p. 306].

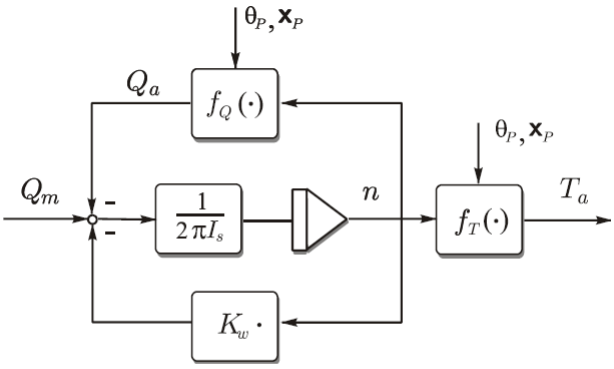


Figure 4.1: Propeller shaft block diagram with linear friction. Courtesy: Sørensen [2011]

For DP and low-speed maneuvering applications it is common to assume a constant zero inflow velocity, meaning $K_T$ and $K_Q$ can be expressed by the constants $K_{T0}$ and $K_{Q0}$.

## 4.2 Modelling of Thruster Dynamics

The rotational dynamics of the propeller shaft can be described by:

$$I_s\dot{\omega} = Q_m - Q_a - K_\omega\omega \tag{4.5}$$

where $Q_m$ is the input torque from the motor, $Q_a$ is the load torque, found through the mapping in equation 4.2, and $K_\omega$ is a linear friction coefficient. A block diagram of equation 4.5 is found in figure 4.1.

The torque build-up in electrical motor drives can be modelled through a 1st order model:

$$\dot{Q}_m = \frac{1}{T_m}(Q_c - Q_m) \tag{4.6}$$

where $Q_c$ is the commanded torque and $Q_m$ the produced torque [Sørensen, 2011, p. 315].

## 4.3 Low-level thruster control

For low-level thruster control there are various schemes available. The conventional way of doing this is by shaft-speed feedback control. A PI-controller

uses measurements of the shaft speed $n$ and compares it with the commanded shaft speed $n_c$, which can be found by inverting the relationship in equation 4.1:

$$Q_c = K_p(n_r - n) + K_i \int_0^t (n_r(\tau) - n(\tau))d\tau. \tag{4.7}$$

Alternatively, a torque-feedforward control scheme can be used, where the commanded torque $Q_c$ can be derived from the commanded thrust $T_r$ by combining equations 4.1 and 4.2:

$$Q_c = \frac{K_{Q0}}{K_{T0}} D T_r. \tag{4.8}$$

Lastly, a power feedback control scheme can be used. Here, the electric motors power consumption $P$ is estimated by measuring the voltage and current and multiplying them together. This is then compared to a commanded power $P_c$, which is found by using the relationship

$$P_r = Q_r 2\pi n_r \tag{4.9}$$

and inserting for $Q_r$ and $n_r$ from equations 4.1 and 4.2 to give

$$P_r = |T_r|^{3/2} \frac{2\pi K_{Q0}}{\sqrt{\rho} D K_{T0}^{3/2}}, \tag{4.10}$$

see Sørensen [2011, p. 322] for more details.

## 4.4 Control Plant Model

The control plant model

$$\dot{x} = -\frac{1}{T_{1x}}(x - x_r) \tag{4.11}$$

will be tuned to fit adequately with the process plant model. The control plant model will be used for control design while the process plant model will be used for simulations.

# Chapter 5

# Thrust Allocation algorithms

As a benchmark case, the pseudoinverse described in [Fossen and Johansen, 2006] and [Fossen, 2011] is used. The psudoinverse convert the output from the DP-controller, $\tau_d$ into thrust commands for the individual thrusters. These thrust commands are then saturated both in rate change and absolute value. When the other two algorithms are tested, the saturations are left as they are.

## 5.1 Dynamic with Constrained Pseudoinverse

Using the maneuvering control formulation described in Section 3.3, a dynamic thrust allocation algorithm is formulated. Using the control plant model

$$\dot{x} = -T^{-1}(x - u), \qquad x \in \mathbb{R}^n,\, u \in \mathbb{R}^n \tag{5.1}$$

$$\tau = Hx, \qquad \tau \in \mathbb{R}^3,\, H \in \mathbb{R}^{3\times n} \tag{5.2}$$

where $x$ is the thrust vector, $u$ is the input to the local thruster control, $T$ is a diagonal matrix of time constants and $H$ is the thruster configuration matrix. In order to provide a mapping from the desired rigid body forces $\tau$ and desired thrust $x_d$, the Moore-Penrose Pseudoinverse is used, that is

$$x_d(t) = H^{\dagger}\tau_d(t). \tag{5.3}$$

Let the control objective be maneuvering, with a geometric task to stabilize the set

$$\mathcal{T}_A := \{(x, \xi) \in \mathbb{R}^n \times \mathbb{R}^n : x - \xi = 0\} \tag{5.4}$$

subject to the dynamic task

$$f_d(\xi, t) := -\Gamma(\xi - x_d(t)) + \dot{x}_d(t) \tag{5.5}$$

$$\lim_{t\to\infty}(\dot{\xi}(t) - f_d(\xi(t), t)) = 0, \tag{5.6}$$

where $x_d(t)$ is given by (5.3) and $\dot{x}_d(t) = H^\dagger \dot{\tau}_d(t)$, and $\Gamma$ is a tuning matrix. The nature of the Geometric and Dynamic tasks is taken from a worknote written by Roger Skjetne.

The control Lyapunov function

$$V(s, x) = \frac{1}{2}(x - \xi)^T(x - \xi) \tag{5.7}$$

is then used to design the input $u$. Differentiating

$$\begin{aligned}\dot{V}(x, \xi, \dot{\xi}) &= (x - \xi)^T(\dot{x} - \dot{\xi}) \\ &= (x - \xi)^T(-T^{-1}(x - u) - \dot{\xi}).\end{aligned} \tag{5.8}$$

Then, $u$ is designed to stabilize (5.4), that is

$$u = \xi - TK(x - \xi) + Tf_d. \tag{5.9}$$

Inserting $u$ in (5.8)

$$\begin{aligned}\dot{V}(x, \xi, \dot{\xi}) = (x - \xi)^T(-T^{-1}(x - \xi)) \\ -V^\xi(T^{-1}f_d - \dot{\xi}).\end{aligned} \tag{5.10}$$

Now all that remains is designing an update law for $\dot{\xi}$ to satisfy the dynamic task in (5.6)

$$\dot{\xi} = f_d - \Omega V^\xi \tag{5.11}$$

## 5.2 Dynamic Thrust Allocation with Maneuvering Behavior

Let

$$\begin{aligned}\dot{x} &= -T^{-1}(x - u) \\ \tau &= Bx\end{aligned} \tag{5.12}$$

where $x \in \mathbb{R}^n$ is the thruster vector, $u \in \mathbb{R}^n$ is a control input, $T \in \mathbb{R}^{n \times n}$ is a matrix of timeconstants and $\tau \in \mathbb{R}^r$ is a vector of resultant forces.

Let the control-objective be maneuvering thrust allocation. Let the geometric task be for $x$ to follow an auxilliary variable $\xi \in \mathbb{R}^n$ such that

$$\lim_{t \to \infty} |x - \xi| = 0. \tag{5.13}$$

Let the dynamic task be to solve the optimization problem

$$\min_{\xi} J(\xi) \quad \text{subject to} \quad \tau_d(t) - B\xi = 0, \tag{5.14}$$

in the limit, where $\tau_d$ is a vector of desired resultant forces and $\xi$ is the maneuvering variable. This is done by designing a feedforward-term $f_d(\xi, t, \lambda)$ that satisfies

$$\lim_{t \to \infty} |\dot{\xi} - f_d(\xi, t, \lambda)| = 0, \tag{5.15}$$

where $t$ is time and $\lambda$ will be defined as a Lagrange multiplier shortly.

Reformulating (5.14) by introducing a Lagrange-multiplier $\lambda \in \mathbb{R}^r$, the Lagrangian according to (5.14) becomes

$$\mathcal{L}(\xi, \lambda, t) = J(\xi) + (\tau_d(t) - B\xi)^T \lambda. \tag{5.16}$$

Then, using the control Lyapunov function

$$W(\xi, \lambda, \tau_d) = \frac{1}{2} \frac{\partial \mathcal{L}}{\partial \xi}^T \frac{\partial \mathcal{L}}{\partial \xi} + \frac{1}{2} \frac{\partial \mathcal{L}}{\partial \lambda}^T \frac{\partial \mathcal{L}}{\partial \lambda}, \tag{5.17}$$

desired dynamics for $\xi$ and $\lambda$ are designed. $\dot{W}$ becomes

$$\begin{aligned} \dot{W} =& W^{\xi} \dot{\xi} + W^{\lambda} \dot{\lambda} + W^{\tau_d} \dot{\tau}_d(t) \\ =& + \Big( \frac{\partial \mathcal{L}}{\partial \xi}^T \frac{\partial^2 \mathcal{L}}{\partial \xi^2} + \frac{\partial \mathcal{L}}{\partial \lambda}^T \frac{\partial^2 \mathcal{L}}{\partial \lambda \partial \xi} \Big) \dot{\xi} \\ &+ \Big( \frac{\partial \mathcal{L}}{\partial \xi}^T \frac{\partial^2 \mathcal{L}}{\partial \xi \partial \lambda} \Big) \dot{\lambda} \\ &+ \Big( \frac{\partial \mathcal{L}}{\partial \lambda}^T \frac{\partial^2 \mathcal{L}}{\partial \lambda \partial \tau_d} \Big) \dot{\tau}_d. \end{aligned} \tag{5.18}$$

Let

$$\alpha = \frac{\partial^2 \mathcal{L}}{\partial \xi^2}\frac{\partial \mathcal{L}}{\partial \xi} + \frac{\partial^2 \mathcal{L}}{\partial \lambda \partial \xi}\frac{\partial \mathcal{L}}{\partial \lambda} \tag{5.19}$$

,

$$\beta = \frac{\partial^2 \mathcal{L}}{\partial \xi \partial \lambda}\frac{\partial \mathcal{L}}{\partial \xi} \tag{5.20}$$

and

$$\delta = \frac{\partial \mathcal{L}}{\partial \lambda}^T \frac{\partial^2 \mathcal{L}}{\partial \lambda \partial \tau_d}\dot{\tau}_d = (\tau_d - B\xi)^T \dot{\tau}_d \tag{5.21}$$

and rewrite (5.18) to

$$\dot{W} = \alpha^T \dot{\xi} + \beta^T \dot{\lambda} + \delta. \tag{5.22}$$

*Assumption 1:*

$$\frac{\partial h}{\partial \xi}\frac{\partial h}{\partial \xi}^T \geq \rho I_n, \quad \rho > 0 \tag{5.23}$$

*Assumption 2:*

$$\kappa_1 I_n \leq \frac{\partial^2 \mathcal{L}}{\partial \xi^2} \leq \kappa_2 I_n, \quad \kappa_2 > \kappa_1 > 0 \tag{5.24}$$

*Lemma 1:* Suppose assumtions 1 and 2 hold. Then $\alpha = 0$ and $\beta = 0$ is equivalent to $\frac{\partial \mathcal{L}}{\partial \xi} = 0$ and $\frac{\partial \mathcal{L}}{\partial \lambda} = 0$.
Proof: See [Johansen, 2004].

Choosing the update law for $\xi$ as

$$\dot{\xi} = f_d = -\Gamma\alpha + \zeta, \tag{5.25}$$

where the first term is to make $\alpha$ negative definite and the second term is to deal with the indefinite term $\delta$ in (5.22), more on this later. Then, choosing an update law for $\lambda$ as

$$\dot{\lambda} = -\Lambda\beta + \phi, \tag{5.26}$$

where the first term is to make $\beta$ negative definite and the second term to deal with the last indefinite term in (5.22).

(5.22) can now be rewritten as

$$\begin{aligned} \dot{W} = &- \alpha^T \Lambda \alpha - \beta^T W \beta \\ &+ \alpha^T \zeta + \beta^T \phi + \delta. \end{aligned} \tag{5.27}$$

*Proposition 1:* It is always possible to find signals $\zeta(t) \in \mathbb{R}^n$ and $\phi(t) \in \mathbb{R}^r$ such that the scalar algebraic equation

$$\alpha^T \zeta + \beta^T \phi + \delta = 0 \tag{5.28}$$

holds.

*Proof:* To find a solution to (5.28), a least-squares problem subject to (5.28) is solved, leading to a Lagrangian

$$\mathbb{L}(\zeta, \phi, \nu) = \frac{1}{2}(\zeta^T \zeta + \phi^T \phi) + \nu(\alpha^T \zeta + \beta^T \phi + \delta), \tag{5.29}$$

where $\nu \in \mathbb{R}$ is a Lagrange multiplier. First order optimality conditions leads to $\zeta$ and $\phi$ being given by the solution to the time-varying linear system of equations

$$\begin{bmatrix} I_n & 0 & \alpha \\ 0 & I_r & \beta \\ \alpha^T & \beta^T & 0 \end{bmatrix} \begin{bmatrix} \zeta \\ \phi \\ \nu \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\delta \end{bmatrix}. \tag{5.30}$$

Given assumptions 1 and 2, (5.30) always has a unique solution, see Johansen [2004] for proof.

Then, since (5.28) holds, (5.27) can be written as

$$\dot{W} = -\alpha^T \Gamma \alpha - \beta^T \Lambda \beta, \tag{5.31}$$

showing that the update laws (5.25,5.26) solve the dynamic task exactly.

In order to solve the Geometric Task, (5.13) is rewritten as a noncompact set

$$\mathcal{A} = \{(\xi, x) \in \mathbb{R}^n \times \mathbb{R}^n \,|\, x = \xi\}, \tag{5.32}$$

which is stabilized by using Lyapunov arguments for noncompact sets, see [Teel et al., 2002] and [Skjetne, 2005]. The control Lyapunov function

$$V(x, \xi) = \frac{1}{2}(x - \xi)^T (x - \xi) \tag{5.33}$$

is used to stabilize (5.32). Differentiating $V$ gives

$$\dot{V}(x, \xi) = (x - \xi)^T (\dot{x} - \dot{\xi}). \tag{5.34}$$

(5.25) solves the Dynamic Task (5.14) exactly. However, since the Dynamic Task needs to be solved only in the limit, design flexibility exists, see [Skjetne, Jorgensen, and Teel, 2011]. Defining the error state

$$\omega = f_d - \dot{\xi} \tag{5.35}$$

and inserting (5.12) and (5.35) into (5.34) gives

$$\dot{V}(x,\xi) = (x-\xi)^T(-T^{-1}(x-u) - f_d + \omega) \tag{5.36}$$

Choosing

$$u = \xi + T(-K_p(x-\xi) + f_d) \tag{5.37}$$

and inserting $u$ into (5.36) results in

$$\begin{aligned}\dot{V}(x,\xi) =& -(x-\xi)^T(T^{-1}(x-(\xi+T(-K_p(x-\xi)+f_d))) + f_d - \omega) \\ =& -(x-\xi)^T(T^{-1}+K_p)(x-\xi) \\ & -(x-\xi)^T(f_d - f_d + \omega) \\ =& -(x-\xi)^T(T^{-1}+K_p)(x-\xi) \\ & -V^{\xi}(x,\xi)\omega\end{aligned} \tag{5.38}$$

In order for (5.35) to still solve the Dynamic Task, $\omega$ needs to satisfy

1. $|\omega(\xi,x)| \leq \alpha(|\xi,x|_{\mathcal{A}})$ where $\alpha$ is a class-$\mathcal{K}$ function
2. $V^{\xi}(\xi,x)\omega(\xi,x) \geq 0$.

Choosing

$$\omega = \Omega(x-\xi) = \Omega V^{\xi}(\xi,x)^T, \tag{5.39}$$

where $\Omega = \Omega^T > 0 \in \mathbb{R}^{n\times n}$ is a tuning matrix, and inserting in (5.38) gives

$$\dot{V}(x,\xi) = -(x-\xi)^T(T^{-1}+K_P)(x-\xi) - V^{\xi}(x,\xi)\Omega V^{\xi}(\xi,x)^T, \tag{5.40}$$

which solves the Geometric and Dynamic Tasks in the limit.

Summary:

| | |
|---|---|
| $u$ | $= \xi + T(-K_p(x-\xi) + f_d)$ |
| | $= \xi + T\left(-K_p(x-\xi) - \Gamma\left(\frac{\mathrm{d}^2 J}{\mathrm{d}\xi^2}(\frac{\mathrm{d}J}{\mathrm{d}\xi} - B^T\lambda) - B^T(\tau_d(t) - B\xi)^T\right) + \zeta\right)$ |
| $\dot{\xi}$ | $= f_d - \omega$ |
| | $= -\Gamma\alpha + \zeta - \Omega V^{\xi}(\xi, x)^T$ |
| | $= -\Gamma\left(\frac{\mathrm{d}^2 J}{\mathrm{d}\xi^2}(\frac{\mathrm{d}J}{\mathrm{d}\xi} - B^T\lambda) - B^T(\tau_d(t) - B\xi)\right) + \zeta - \Omega V^{\xi}(\xi, x)^T$ |
| $\dot{\lambda}$ | $= -\Lambda\beta + \phi$ |
| | $= -\Lambda\left(-B^T(\frac{\mathrm{d}J}{\mathrm{d}\xi} - B^T\lambda)\right) + \phi$ |

# Chapter 6

# Simulation Setup

## 6.1 CSE1

To simulate the various thrust allocation algorithms, the CyberShipEnterprise1 (CSE1) is used as a test case. The CSE1 is a model ship used in the Marine Cybernetics laboratories at the NTNU, built and tested by Håkon Skåtun [Skåtun, 2011].

As indicated in figure 6.1, CSE1 is fitted with two Voith-Schneider propellers astern and and one bow-thruster. Since the Voith-Schneider propellers can produce thrust in any direction, each of these will be treated as two separate thrusters.

Using the theory presented in Chapter 4, a simulation model of the thrusters is implemented in Simulink. Table 6.1 lists the parameter values for the thruster models. Figure 6.2 shows a comparison of the process plant model and the control plant model.

| Parameter | Value | Unit |
|---:|:---:|:---:|
| $K_{T0}$ | 0.36 | - |
| $K_{Q0}$ | 0.044 | - |
| Diameter $D$ | 0.07 | $m$ |
| Rotational inertia $I_s$ | 0.075 | $kg \cdot m^2$ |
| Shaft friction $K_w$ | 0.0075 | $N \cdot m \cdot s$ |
| Induction Motor Time Constant $T_m$ | 0.1 | $s$ |
| Density of Fresh Water $\rho$ | 1000 | $kg/m^3$ |
| Thrust control P-gain $K_{pn}$ | 0.5 | $N \cdot m \cdot s$ |
| Thrust control I-gain $K_{in}$ | 0.05 | $N \cdot m$ |
| Control Plant Model Time Constant $T_{1x}$ | 1 | $s$ |

Table 6.1: Thruster Model Parameters

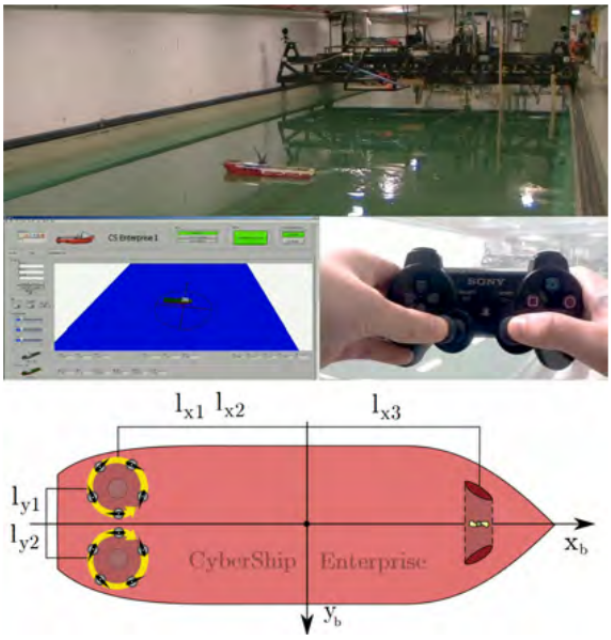


Figure 6.1: The CSE1 in the MC-Lab. Courtesy: Skåtun [2011]

Using the 3-DOF vessel model

$$\begin{aligned} \dot{\eta} =& R(\psi)\nu \\ M\dot{\nu} =& -D_L\nu + R^T(\psi)b + \tau \end{aligned} \tag{6.1}$$

presented in Fossen [2011], where $\eta \in \mathbb{R}^3$ is the position in NED, $\nu \in \mathbb{R}^3$ is the BODY-centered velocity vector, $R \in \mathbb{R}^{3\times 3}$ is an orthogonal rotation matrix, $\psi \in \mathbb{R}$ is the heading angle, $M \in \mathbb{R}^3$ is the inertia matrix, $b$ is a slowly varying bias term and $\tau$ is the thrust-force vector. $\tau$ is computed using the thruster configuration matrix

$$\tau = Hx, \tag{6.2}$$

where $x$ is the thrust output from each thruster.

In order to fulfill the stationkeeping objective, a nonlinear PID DP-controller

$$\tau_d = -R^T(\psi)(K_p(\eta - \eta_d) + K_d\dot{\eta} - K_i \int_0^t (\eta(p) - \eta_d(p))\,\mathrm{d}p \tag{6.3}$$

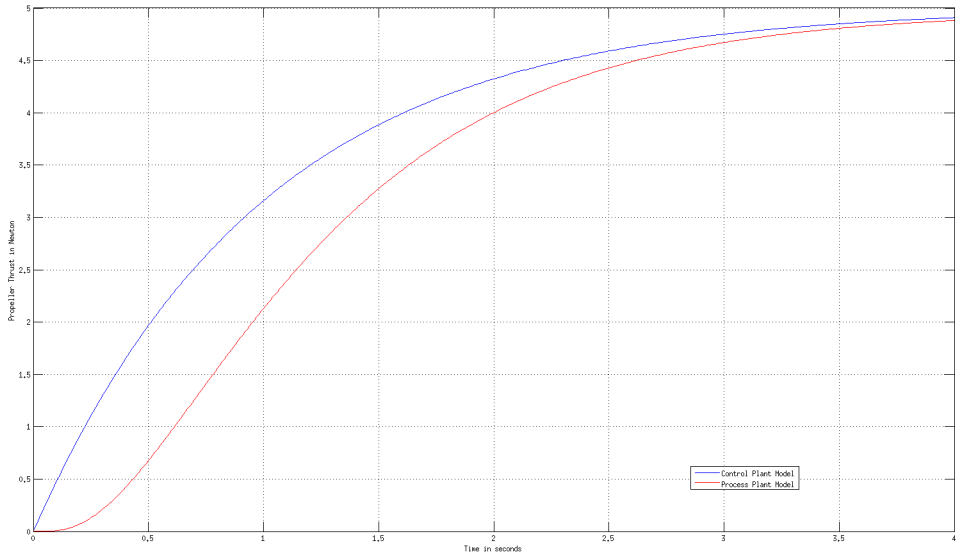


Figure 6.2: Comparison of Control Plant Model and Process Plant Model

as described in Fossen [2011] is used. $\tau_d$ is then the input to the various thrust allocation algorithms.

Table 6.2 lists the parameter values.

| Parameter | Value |
|---|---|
| $M$ | $\begin{bmatrix} 10 & 0 & 0 \\ 0 & 20 & 5 \\ 0 & 5 & 50 \end{bmatrix}$ |
| $D_L$ | $\begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 40 \end{bmatrix}$ |
| $K_p$ | $\begin{bmatrix} 5 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 25 \end{bmatrix}$ |
| $K_d$ | $\begin{bmatrix} 25 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 125 \end{bmatrix}$ |
| $K_i$ | $\begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$ |

Table 6.2: Ship Model Parameters

## 6.2 Scenarios

In all the simulation scenarios, the overall objective will be stationkeeping. Since the simulations are meant to show the limits of the system, this may not always be achieved.

1. Steadily increasing current from the front and the side until the thrusters barely saturate, see Table 6.3 for details.

2. Turning maneuver with no current. A first order reference generator is used to generate a continuous signal. See Table 6.4 for details.

3. Steadily increasing current from the front and the side until the thrusters saturate a slight while longer, see Table 6.5 for details. The vessel will drift slightly off, causing the DP-controller to demand the absolute maximum of the thruster system in yaw for a short while.

4. Steadily increasing current from the front and the side until the thrusters saturate just a little while longer, see Table 6.6 for details. The vessel will drift a bit more off, causing the DP-controller to demand the absolute maximum of the thruster system in yaw for and rendering the system unstable, probably due to numerical issues.

| Parameter | Value | Unit |
|---|---|---|
| $\eta_0$ | $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ | $\begin{bmatrix} m & m & RAD \end{bmatrix}^T$ |
| $\eta_d$ | $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ | $\begin{bmatrix} m & m & RAD \end{bmatrix}^T$ |
| Slope of the current ramp | $\begin{bmatrix} -0.4 & -0.1 & 0 \end{bmatrix}^T$ | $\begin{bmatrix} N/s & N/s & Nm/s \end{bmatrix}^T$ |
| Maximum current force | $\pm \begin{bmatrix} 5.52 & 8.1 & \infty \end{bmatrix}^T$ | $\begin{bmatrix} N & N & Nm \end{bmatrix}^T$ |

Table 6.3: Scenario 1 Parameters

| Parameter | Value | Unit |
|---|---|---|
| $\eta_0$ | $\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ | $\begin{bmatrix}m & m & RAD\end{bmatrix}^T$ |
| $\eta_d$ | $\begin{bmatrix}0 & 0 & \pi/2\end{bmatrix}^T$ | $\begin{bmatrix}m & m & RAD\end{bmatrix}^T$ |
| Slope of the current ramp | $\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ | $\begin{bmatrix}N/s & N/s & Nm/s\end{bmatrix}^T$ |
| Reference generator gain | 0.04 | [-] |

Table 6.4: Scenario 2 Parameters

| Parameter | Value | Unit |
|---|---|---|
| $\eta_0$ | $\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ | $\begin{bmatrix}m & m & RAD\end{bmatrix}^T$ |
| $\eta_d$ | $\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ | $\begin{bmatrix}m & m & RAD\end{bmatrix}^T$ |
| Slope of the current ramp | $\begin{bmatrix}-0.3 & 0 & 0\end{bmatrix}^T$ | $\begin{bmatrix}N/s & N/s & Nm/s\end{bmatrix}^T$ |
| Maximum current force | $\pm\begin{bmatrix}5.58 & 8.1 & \infty\end{bmatrix}^T$ | $\begin{bmatrix}N & N & Nm\end{bmatrix}^T$ |

Table 6.5: Scenario 3 Parameters

| Parameter | Value | Unit |
|---|---|---|
| $\eta_0$ | $\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ | $\begin{bmatrix}m & m & RAD\end{bmatrix}^T$ |
| $\eta_d$ | $\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ | $\begin{bmatrix}m & m & RAD\end{bmatrix}^T$ |
| Slope of the current ramp | $\begin{bmatrix}-0.3 & 0 & 0\end{bmatrix}^T$ | $\begin{bmatrix}N/s & N/s & Nm/s\end{bmatrix}^T$ |
| Maximum current force | $\pm\begin{bmatrix}5.7 & 8.1 & \infty\end{bmatrix}^T$ | $\begin{bmatrix}N & N & Nm\end{bmatrix}^T$ |

Table 6.6: Scenario 4 Parameters

## 6.3 Thrust Allocation Parameters

The thrust allocation algorithms designed in Chapter 5 are implemented in Simulink. They are then tuned to work satisfactory. Tables 6.7, 6.8 and 6.9 show the details.

| Parameter | Value |
|---|---|
| Maximum values | $\begin{bmatrix} 3 & 3 & 3 & 3 & 3 \end{bmatrix}^T$ |
| Minimum values | $-\begin{bmatrix} 3 & 3 & 3 & 3 & 3 \end{bmatrix}^T$ |
| Rate constraints | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$ |

Table 6.7: Constrained Pseudoinverse Parameters

| Parameter | Value |
|---|---|
| Maximum values | $\begin{bmatrix} 3 & 3 & 3 & 3 & 3 \end{bmatrix}^T$ |
| Minimum values | $-\begin{bmatrix} 3 & 3 & 3 & 3 & 3 \end{bmatrix}^T$ |
| Rate constraints | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$ |
| $\Omega$ | $40 \cdot I_{5\times5}$ |
| $\Gamma$ | $2.5 \cdot I_{5\times5}$ |
| $K$ | $I_{5\times5}$ |
| $\xi_0$ | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ |

Table 6.8: Constrained Pseudoinverse with Maneuvering Behavior Parameters

| Parameter | Value |
|---|---|
| Maximum values | $\begin{bmatrix} 3 & 3 & 3 & 3 & 3 \end{bmatrix}^T$ |
| Minimum values | $-\begin{bmatrix} 3 & 3 & 3 & 3 & 3 \end{bmatrix}^T$ |
| Rate constraints | $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$ |
| $\Lambda$ | $I_{3\times 3}$ |
| $\Gamma$ | $5 \cdot I_{5\times 5}$ |
| $\Omega$ | $I_{5\times 5}$ |
| $K_p$ | $0.1 \cdot I_{5\times 5}$ |
| $\xi_0$ | $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ |
| $J(\xi)$ | $0.001 \cdot \operatorname{arctanh}(\xi)^2$ |

Table 6.9: Dynamic Thrust Allocation with Maneuvering Behavior Parameters

# Chapter 7

# Simulation Results and Discussion

To simplify writing, the benchmark case of the constrained pseudoinverse will be called Model 1, the constrained pseudoinverse with maneuvering behavior derived in Section 5.1 will be called Model 2 and the dynamic thrust allocation with maneuvering behavior derived in Section 5.2 will be called Model 3.

## 7.1 Scenario 1

### 7.1.1 Model 1

Figures 7.1 and 7.2 show the performance of the left azimuth thruster. Notice how $x$ does not increase over 3 even though $u$ does. This is because $u$ is saturated after it is plotted.

The rest of the plots for scenario 1 can be found in Appendix B.

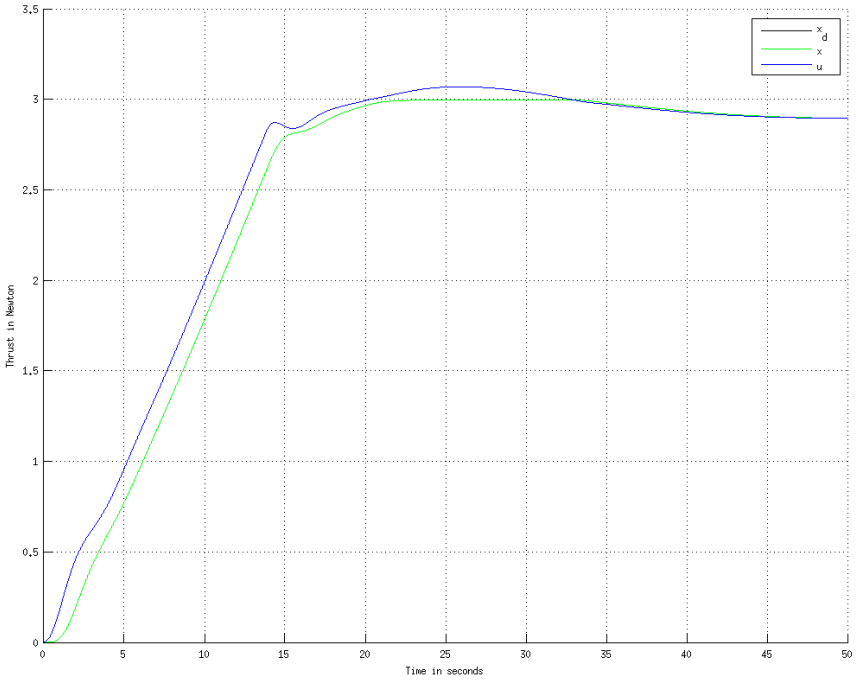


Figure 7.1: Scenario 1, Model 1: Left azimuth in x-direction

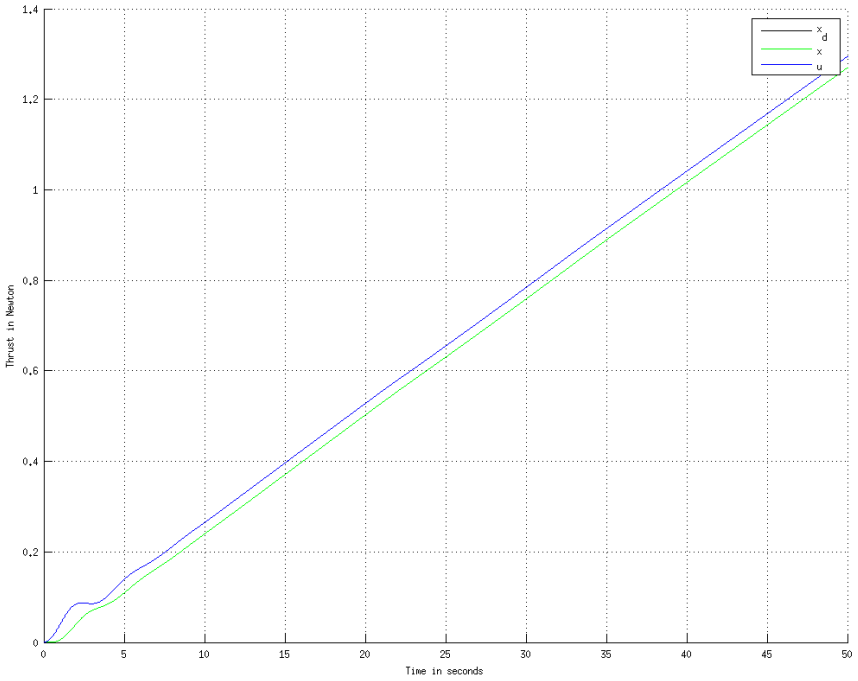


Figure 7.2: Scenario 1, Model 1: Left azimuth in y-direction

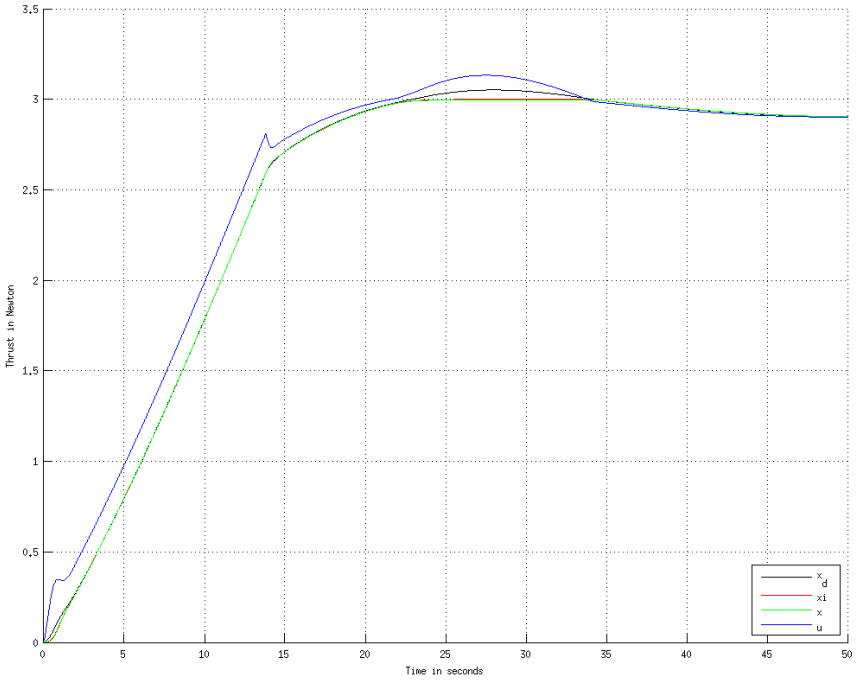


Figure 7.3: Scenario 1, Model 2: Left azimuth in x-direction

### 7.1.2 Model 2

Figures 7.3 and 7.4 show the performance of the left azimuth thruster.

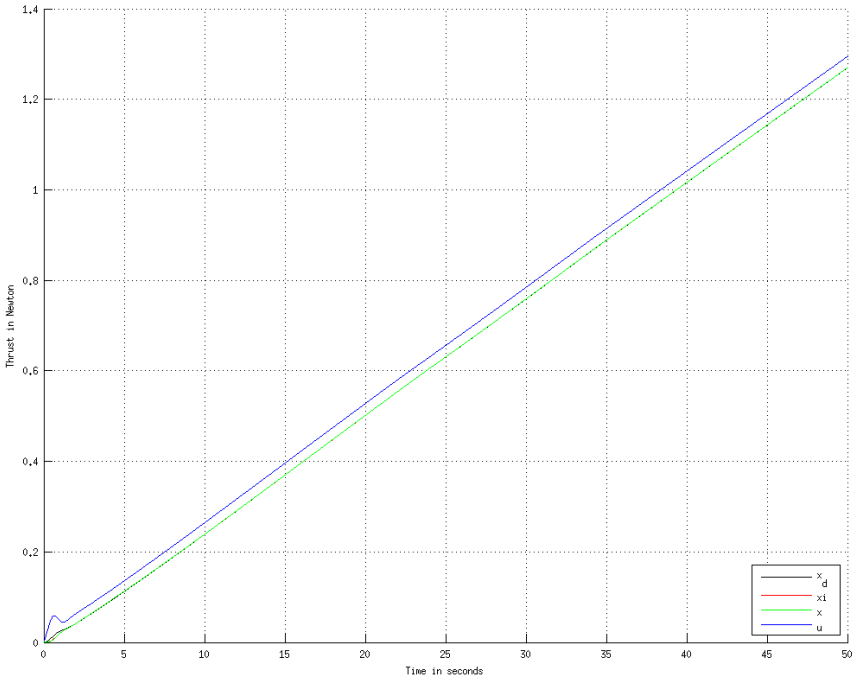


Figure 7.4: Scenario 1, Model 2: Left azimuth in y-direction

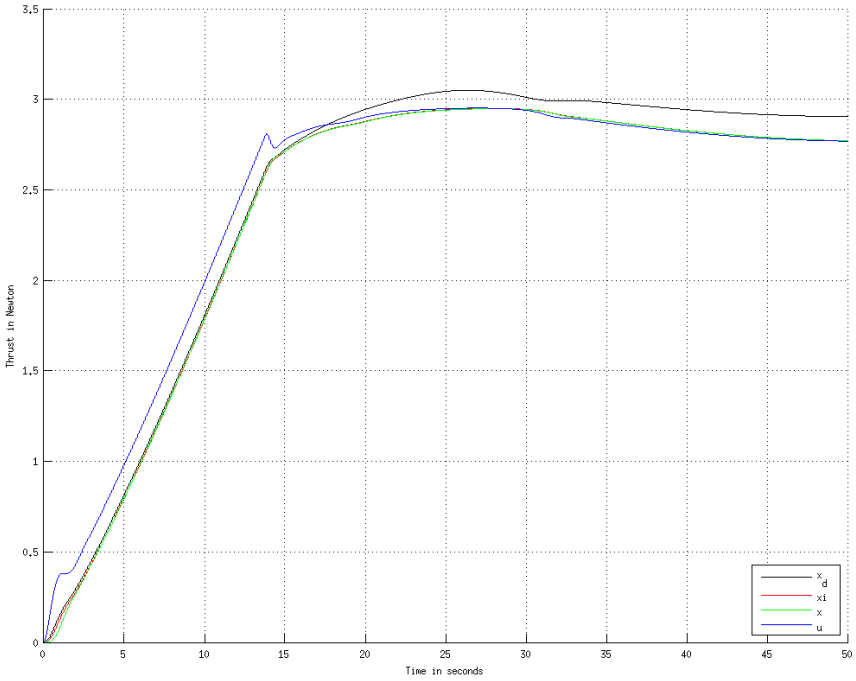


Figure 7.5: Scenario 1, Model 3: Left azimuth in x-direction

### 7.1.3 Model 3

Figures 7.5 and 7.6 show the performance of the left azimuth thruster. Notice how $u$ never goes all the way up to 3, this is because of the barrier cost function. It appears $\xi$ does not quite follow $x_d$ in the later stages of the simulation. Keep in mind though, that $x_d$ in the plot has been calculated using the pseudoinverse, and is not a direct input to the thrust allocation algorithm.

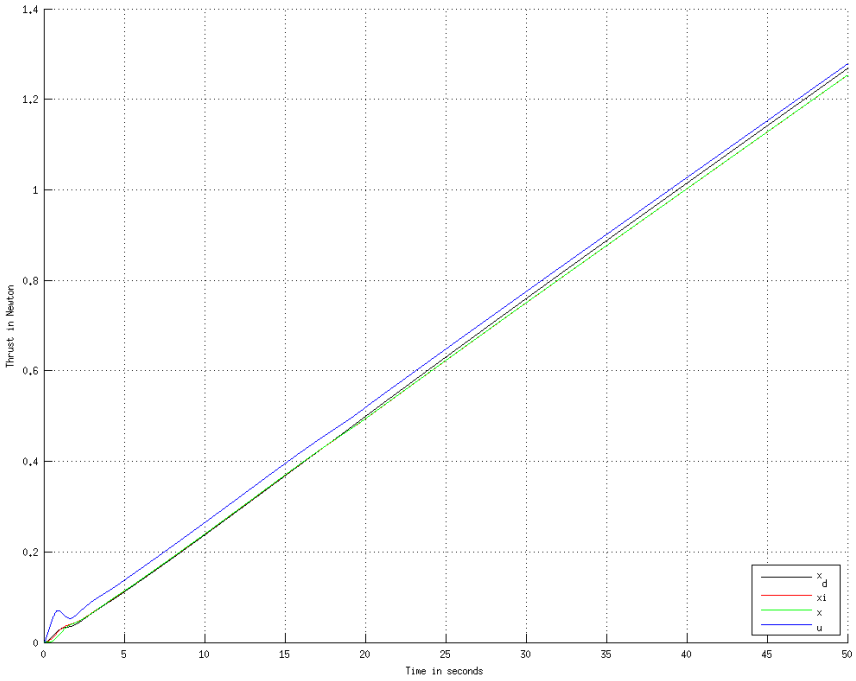


Figure 7.6: Scenario 1, Model 3: Left azimuth in y-direction

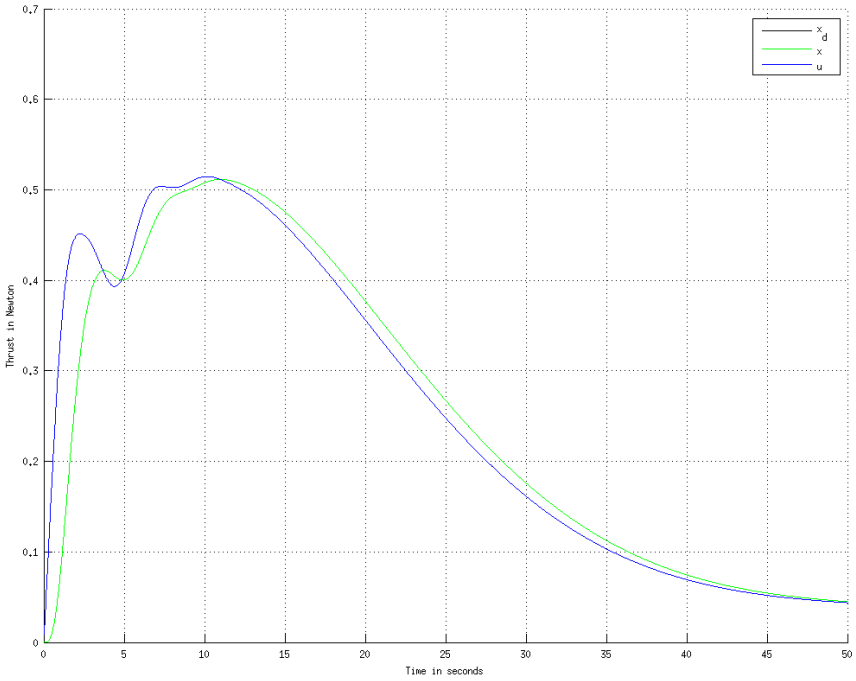


Figure 7.7: Scenario 2, Model 1: Left azimuth in x-direction

## 7.2 Scenario 2

### 7.2.1 Model 1

Figures 7.7 and 7.8 show the performance of the left azimuth thruster. The initial oscillations may be due to aggressive DP-controller tuning.

The rest of the plots for scenario 2 can be found in Appendix C.

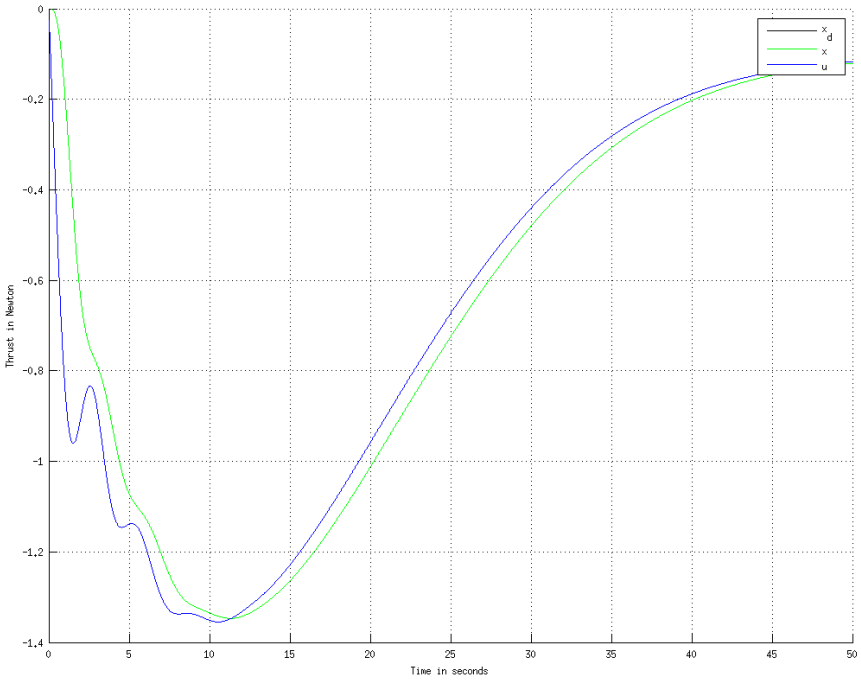


Figure 7.8: Scenario 2, Model 1: Left azimuth in y-direction

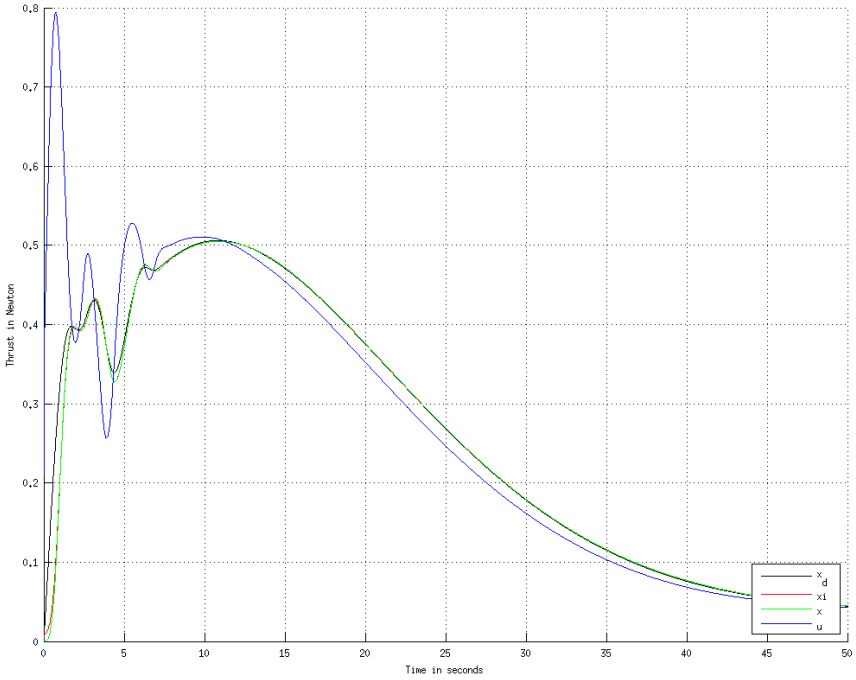


Figure 7.9: Scenario 2, Model 2: Left azimuth in x-direction

### 7.2.2 Model 2

Figures 7.9 and 7.10 show the performance of the left azimuth thruster. The initial oscillations here significantly larger, which may be due to the feedforward term $f_d$.

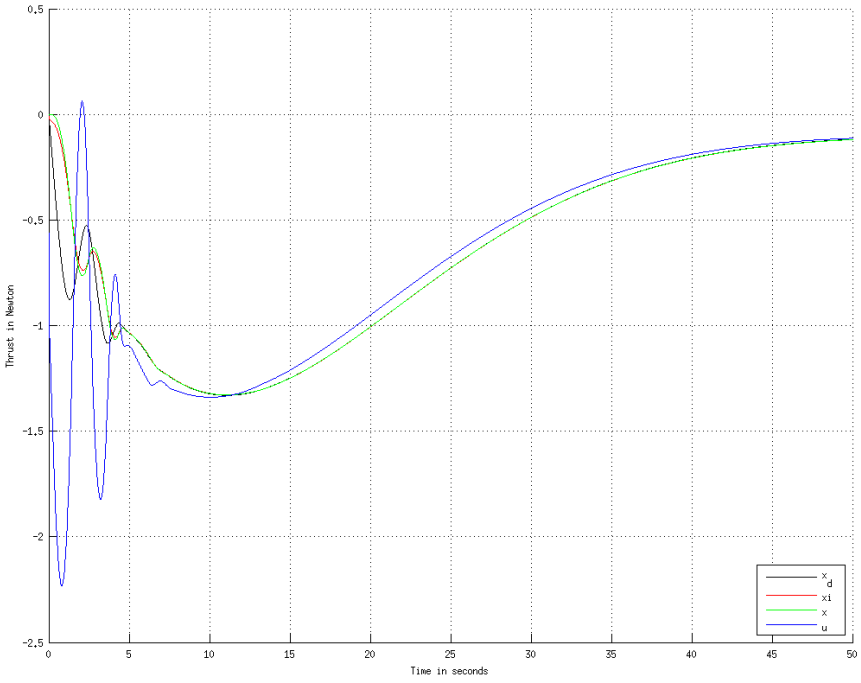


Figure 7.10: Scenario 2, Model 2: Left azimuth in y-direction

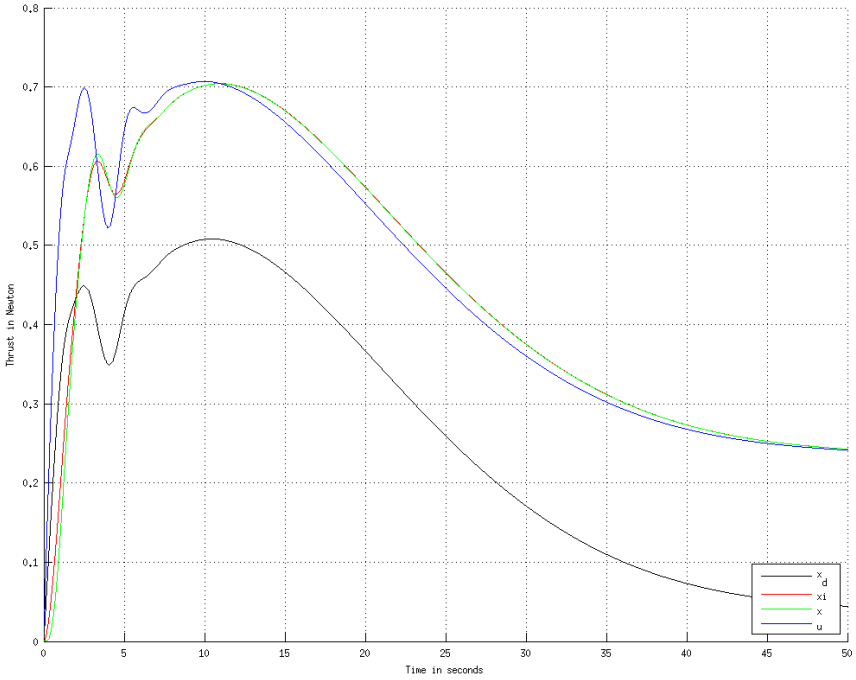


Figure 7.11: Scenario 2, Model 3: Left azimuth in x-direction

### 7.2.3 Model 3

Figures 7.9 and 7.10 show the performance of the left azimuth thruster. The oscillations are significantly less, although still present. Figure 7.9 shows the dynamic algorithm using more thruster force than the $x_d$ computed from the pseudoinverse. This may be because the cost function $J(\xi)$ does not take a quadratic cost function into account, while the pseudoinverse does.

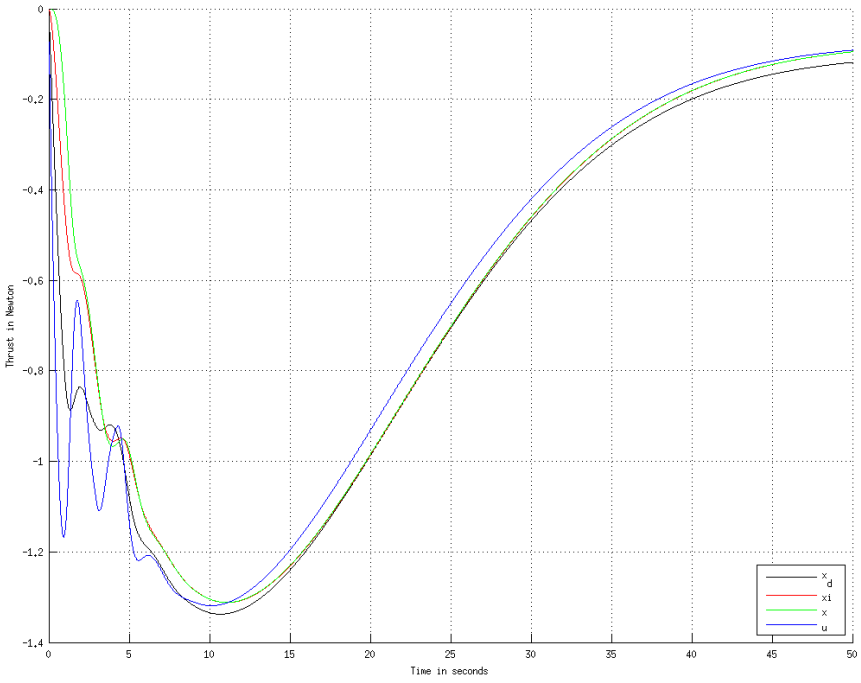


Figure 7.12: Scenario 2, Model 3: Left azimuth in y-direction

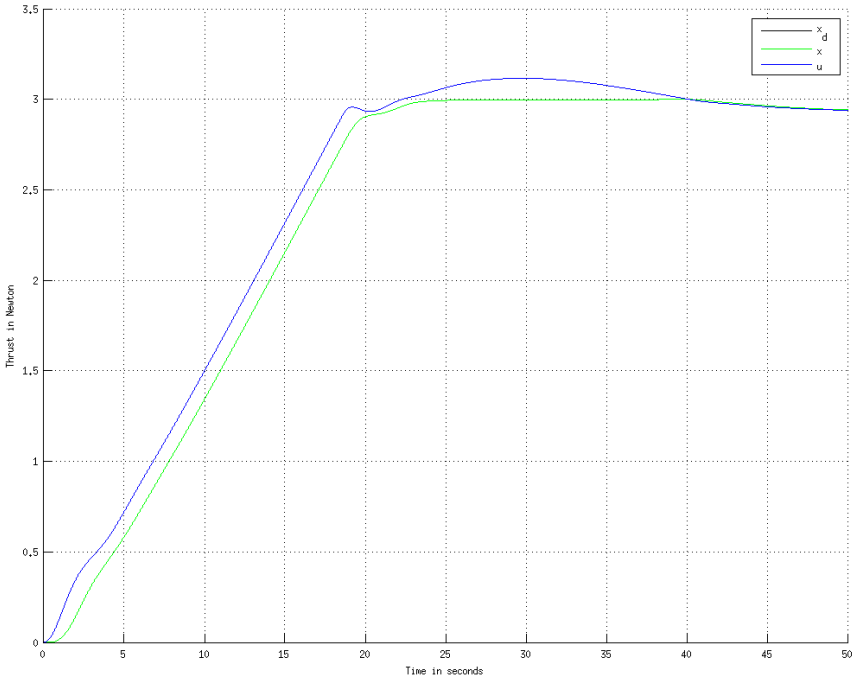


Figure 7.13: Scenario 3, Model 1: Left azimuth in x-direction

## 7.3 Scenario 3

### 7.3.1 Model 1

Figure 7.7 shows the performance of the left azimuth thruster in $x$-direction. This plot is very simular to Figure 7.1, because the scenarios are very similar.

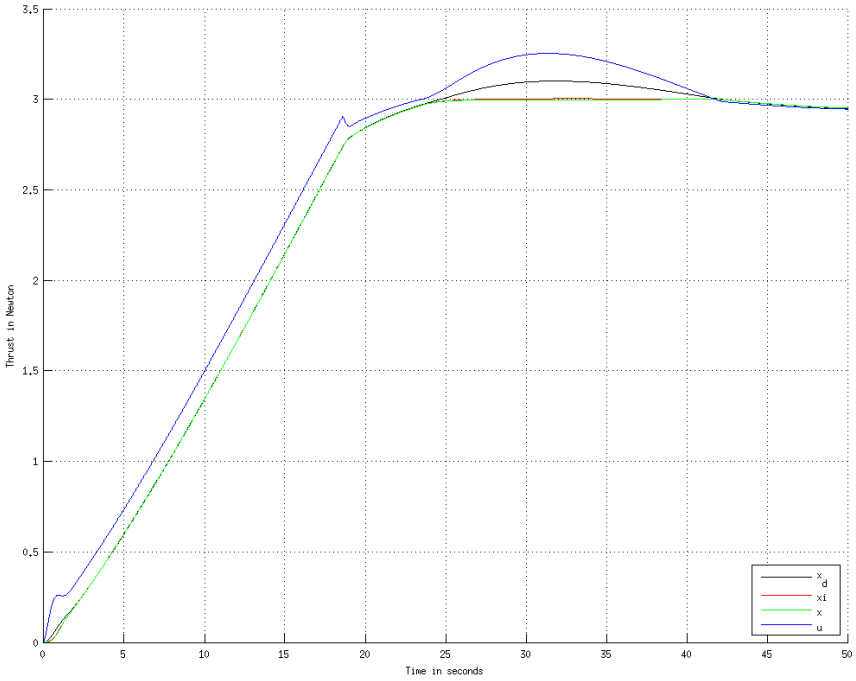


Figure 7.14: Scenario 3, Model 2: Left azimuth in x-direction

### 7.3.2 Model 2

Figure 7.7 shows the performance of the left azimuth thruster in $x$-direction. This plot is very simular to Figure 7.3, because the scenarios are very similar.

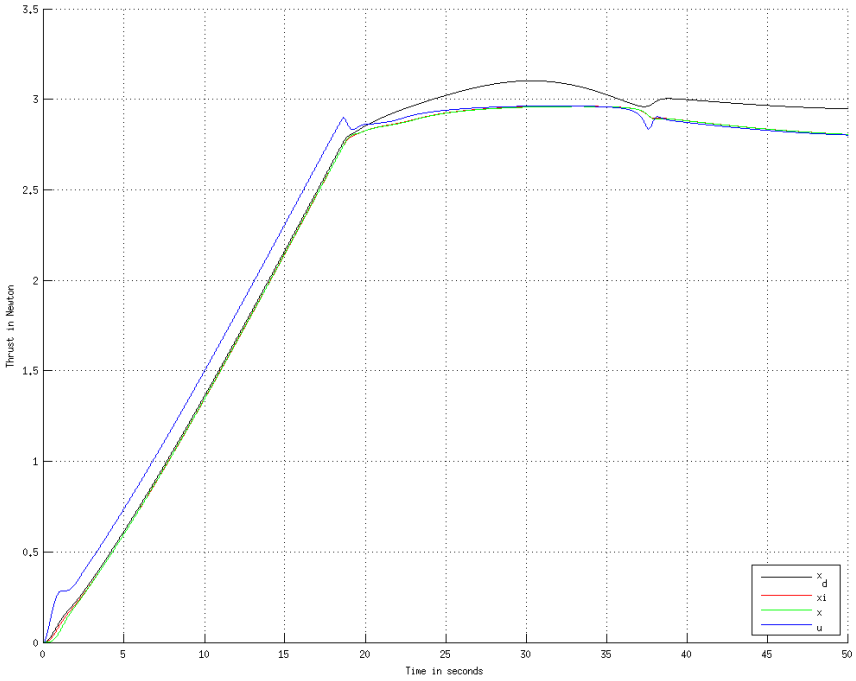


Figure 7.15: Scenario 3, Model 3: Left azimuth in x-direction

### 7.3.3 Model 3

Figure 7.15 shows the saturating behavior of the dynamic thrust allocation algorithm more clearly. The "bump" experienced at about $37s$ is probably explained by Figure 7.16, which plots the derivative of the cost function. The large drop around $37s$ influences the control input.

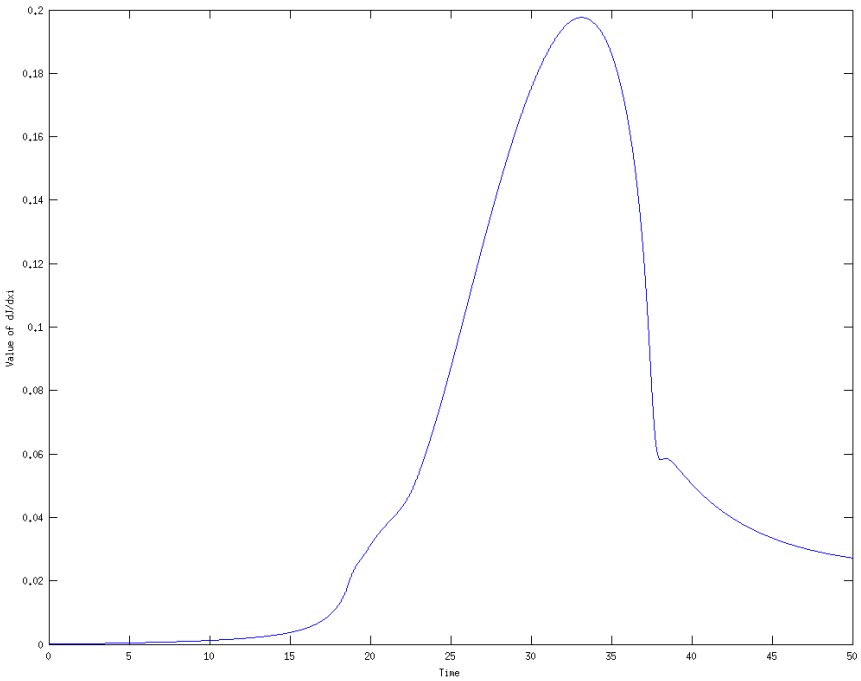


Figure 7.16: Scenario 3, Model 3: $\frac{\mathrm{d}J}{\mathrm{d}\xi}$ for the left azimuth thruster in the x-direction

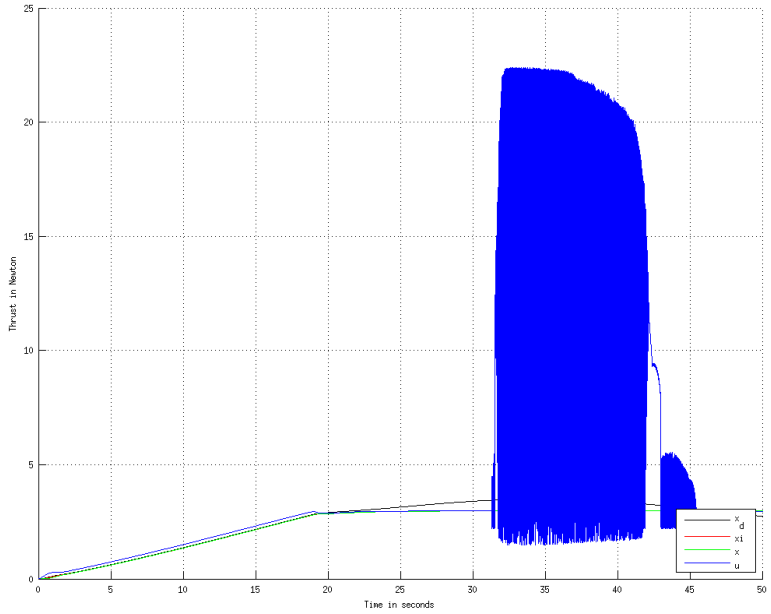


Figure 7.17: Scenario 4, Model 3: Left azimuth in x-direction

## 7.4 Scenario 4

### 7.4.1 Model 3

Although Scenario 4 is only slightly different from Scenarios 1 and 3, unstable behavior can clearly be seen in Figure 7.17. Because the unstable behavior can be made to set in later by reducing step size or increasing solver order, it is assumed that the problem is numerical. If the problem is indeed numerical, this may be because the cost function

$$J(\xi) = 0.001 \cdot \operatorname{arctanh}(\xi)^2 \tag{7.1}$$

is very sensitive to very small changes when $\xi \to \pm 1$.

## 7.5 Final Discussion

The dynamic thrust allocation algorithm beautifully handles contstraints, though only in a very limited amount of cases. The constrained pseudoinverse shows a much more robust behavior.

# Bibliography


Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

Thor I Fossen and Tor A Johansen. A survey of control allocation methods for ships and underwater vehicles. In *Control and Automation, 2006. MED'06. 14th Mediterranean Conference on*, pages 1–6. IEEE, 2006.

Tor A Johansen, Thor I Fossen, and Svein P Berge. Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming. *Control Systems Technology, IEEE Transactions on*, 12(1): 211–216, 2004.

Tor Arne Johansen. Optimizing nonlinear control allocation. In *43rd IEEE Conference on Decision and Control (CDC'04)*, volume 4, 2004.

Håkon Nødset Skåtun. Development of a dp system for cs enterprise i with voith schneider thrusters. Master's thesis, Norwegian University of Science and Technology, Department of Marine Technology, 2011.

R. Skjetne, U. Jorgensen, and A.R. Teel. Line-of-sight path-following along regularly parametrized curves solved as a generic maneuvering problem. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 2467–2474, Dec 2011. doi: 10.1109/CDC.2011.6161364.

Roger Skjetne. *The Maneuvering Problem*. PhD thesis, Norwegian University of Science and Technology, 2005.

Roger Skjetne, Andrew R Teel, and Petar V Kokotovic. Stabilization of sets parametrized by a single variable: Application to ship maneuvering. In *Proceedings of the 15th international symposium on mathematical theory of networks and systems*, 2002.

Asgeir J. Sørensen. *Marine Control Systems*. Department of Marine Technology, Norwegian University of Science and Technology, Trondheim, Norway, 2nd edition, 2011.

Asgeir J Sørensen, Alf Kåre Ådnanes, Thor I Fossen, and Jann-Peter Strand. A new method of thruster control in positioning of ships based on power control. In *4th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC'97), Brijuni, Croatia, 172*, volume 179. Citeseer, 1997.

Andrew Teel, Elena Panteley, and Antonio Loría. Integral characterizations of uniform asymptotic and exponential stability with applications. *Mathematics of Control, Signals and Systems*, 15(3):177–201, 2002.

Petter Tøndel, Tor Arne Johansen, and Alberto Bemporad. An algorithm for multi-parametric quadratic programming and explicit mpc solutions. *Automatica*, 39(3):489–497, 2003.

Øyvind N. Smogeli, Asgeir J. Sørensen, and Knut J. Minsaas. The concept of anti-spin thruster control. *Control Engineering Practice*, 16:465–481, 2006.

# Appendices

# Appendix A

# Thesis Description

**NTNU Trondheim**
**Norwegian University of Science and Technology**
*Department of Marine Technology*

# PROJECT DESCRIPTION SHEET

**Name of the candidate:** Espen Gottschal

**Field of study:** Marine control engineering

**Thesis title (Norwegian):** Dynamisk thrust-allokering for DP fartøy.

**Thesis title (English):** Dynamic thrust allocation for DP vessels.

**Background**

For dynamic positioning (DP) of offshore vessels, the limiting capacity of stationkeeping in heavy environmental conditions is the thruster configuration and the maximum resultant forces and moment that can be produced for surge, sway, and yaw motions. The DP control law calculates a commanded net force/moment to be produced by the propulsion system to compensate the environmental loads. A thrust allocation algorithm distributes this net force/moment in an efficient manner to a commanded force and direction for each individual thruster based on their rated power, locations in the hull, thruster types, and constraints.

Due to a separation of time-scales (slow DP positioning; fast thrust production), the DP control algorithm typically assumes that thrust is produced instantaneously. Hence, thrust allocation is typically solved as a static problem, relating the net force moment to the individual thruster forces, that is, $\tau = Hu$ where $H \in \mathbb{R}^{m \times n}$ is the thruster configuration matrix and $u \in \mathbb{R}^n$ is the forces produced by the individual thrusters. Each thruster will have some dynamics to determine its behavior and how fast it can respond to thruster commands, that is, $\dot{x} = f(t, x, v),\ u = h(t, x)$, where $x$ is the thruster state, $v$ is the thruster control input, and $u$ is the output produced thrust.

In this project the objective is to consider thrust allocation as a dynamic control problem with $v$'s as inputs and the net force/moment $\tau$ as the output to be controlled to the commanded $\tau_{cmd}$(t). As a case study, the C/S Enterprise I (CSE1) model ship shall be utilized as a case study.

**Work description**

1) Perform a literature review to provide background and relevant references on:
   - Thrust allocation and control allocation, incl. constrained optimization techniques.
   - Thruster dynamics, thruster constraints, and local thruster control.
   - The (generalized) maneuvering control problem.
   - Lagrange multiplier techniques to dynamically handle constraints.

   Write a list with abbreviations and definitions, and a section explaining particularly relevant terms and concepts related to DP, thruster control, and thrust allocation in an alphabetic order.
2) Model the thruster dynamics of relevant thrusters for an offshore DP vessel in terms of a process model that contains realistic thruster saturation constraints and dynamic behavior.
3) Using C/S Enterprise I (CSE1) as a case study, develop a simulation model of the thruster system for this vessel. Propose a set of "benchmark" test cases, targeting both unconstrained and constrained scenarios, to test your different thrust allocation algorithms on. Derive and test the unconstrained pseudoinverse thrust allocation for CSE1 as a baseline algorithm.
4) Propose a simplified control plant model of the thruster dynamics, relevant for dynamic thrust allocation methods, and present a *problem formulation* for dynamic thrust allocation. Consider here both unconstrained and constrained allocation, as well as convex vs. nonconvex allocation.
5) Derive a dynamic (generic) maneuvering-based thrust allocation algorithm for the unconstrained convex case. Verify first the algorithm by simulation using the control model. Then test the algorithm based on simulation with the CSE1 process model and your proposed test cases.

6) Study the concept of Lagrange multipliers and how this can be incorporated into the dynamic thrust allocation setup to dynamically handle constraints in the system. Is it possible to update your dynamic thrust allocation method with Lagrange multipliers?

***Tentatively:***

7) Consider the recursive nullspace-based control allocation method based on the pseudoinverse and linear constraints. Combine this method with the dynamic thrust allocation method, verify the resulting algorithm by simulations for the CSE1 model, and discuss the results.
8) Study the *"convex linearly constrained quadratic thrust allocation"* method by Eivind Ruth (2008 PhD thesis). Propose how to update the recursive nullspace-based control allocation method by replacing the pseudoinverse by the linear quadratic optimization method.

**Guidelines**

The scope of work may prove to be larger than initially anticipated. By the approval from the supervisor, described topics may be deleted or reduced in extent without consequences with regard to grading.

The candidate shall present his personal contribution to the resolution of problems within the scope of work. Theories and conclusions should be based on mathematical derivations and logic reasoning identifying the various steps in the deduction.

The report shall be organized in a rational manner to give a clear exposition of results, assessments, and conclusions. The text should be brief and to the point, with a clear language. The report shall be written in English (preferably US) and contain the following elements: Abstract, acknowledgements, table of contents, main body, conclusions with recommendations for further work, list of symbols and acronyms, references, and (optionally) appendices. All figures, tables, and equations shall be numerated. The original contribution of the candidate and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. *natbib* Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct. Such practice is taken very seriously by the university and will have consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed upon.

The thesis shall be submitted with two printed and electronic copies, to 1) the main supervisor and 2) the external examiner, each copy signed by the candidate. The final revised version of this thesis description must be included. The report must appear in a bound volume or a binder according to the NTNU standard template. Computer code and a PDF version of the report shall be included electronically.

**Start date:** 1 February, 2014 **Due date:** As specified by the administration.

**Supervisor:** Prof. Roger Skjetne
**Co-advisor(s):** Prof. Asgeir J. Sørensen

**Roger Skjetne**
Supervisor

# Appendix B

# Scenario 1

## B.1 Model 1

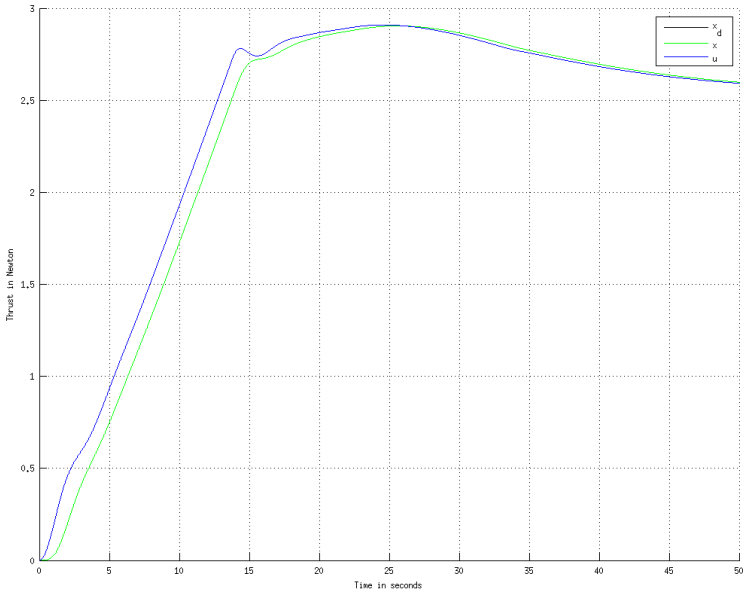


Figure B.1: Right azimuth in x-direction

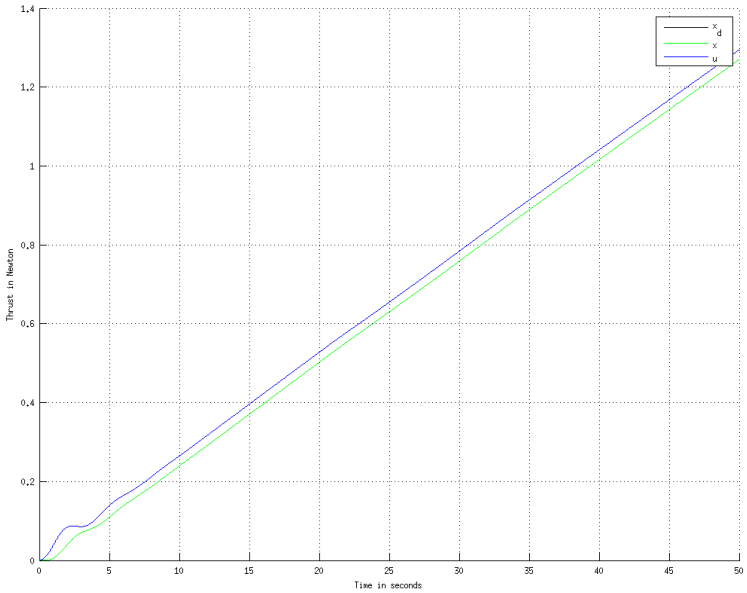


Figure B.2: Right azimuth in y-direction

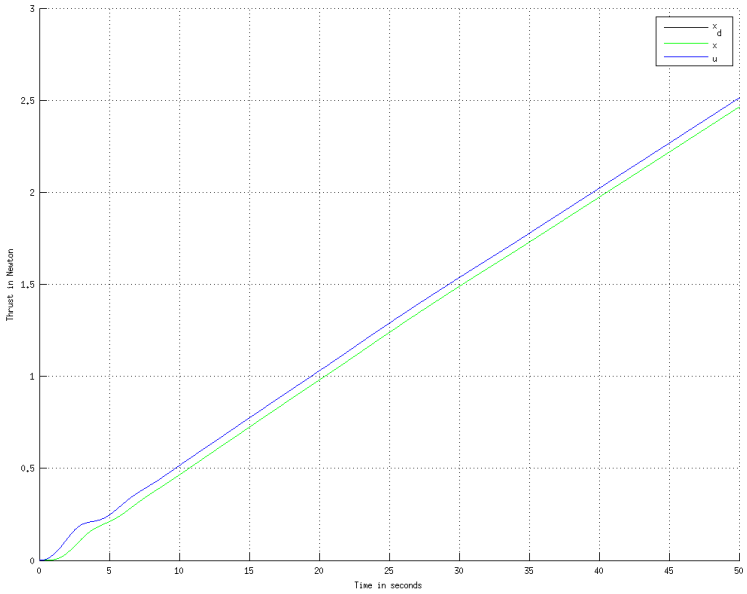


Figure B.3: The bow-thruster

## B.2 Model 2

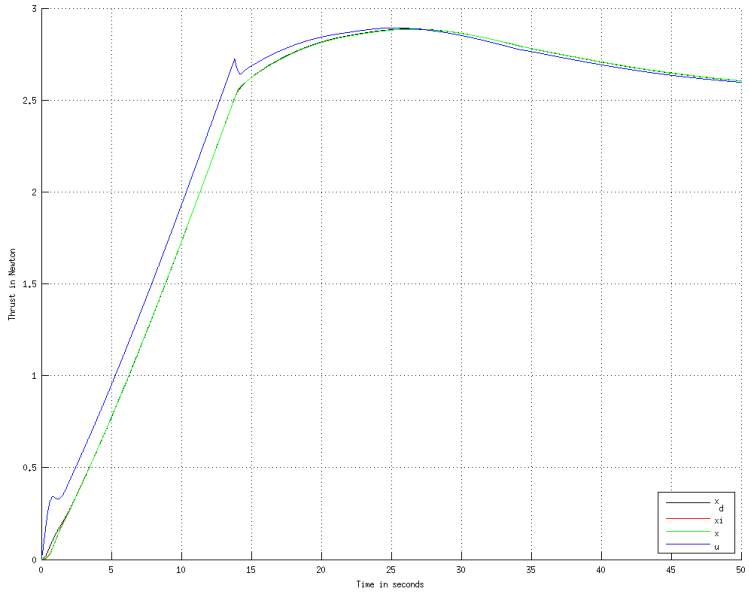


Figure B.4: Right azimuth in x-direction

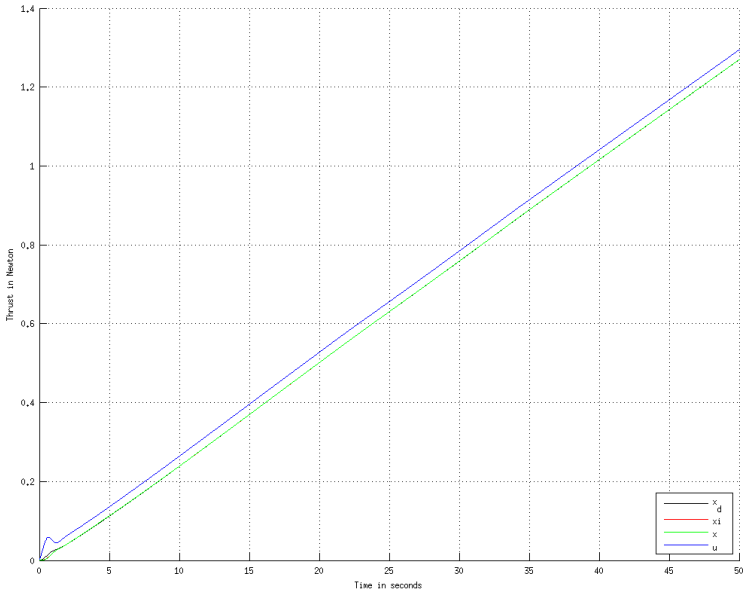


Figure B.5: Right azimuth in y-direction

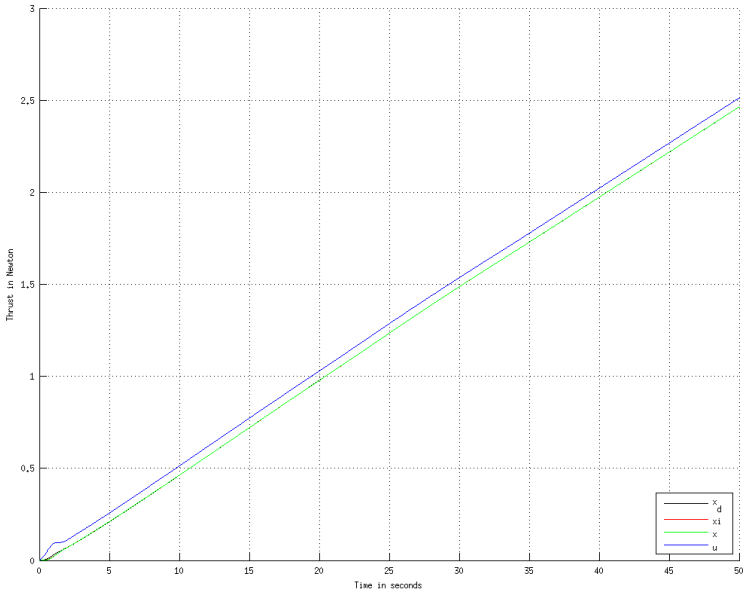


Figure B.6: The bow-thruster

## B.3 Model 3

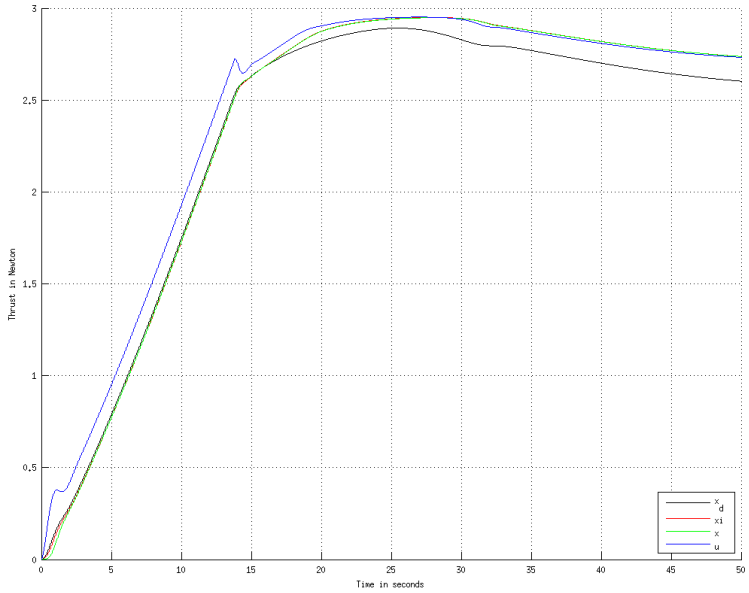


Figure B.7: Right azimuth in x-direction

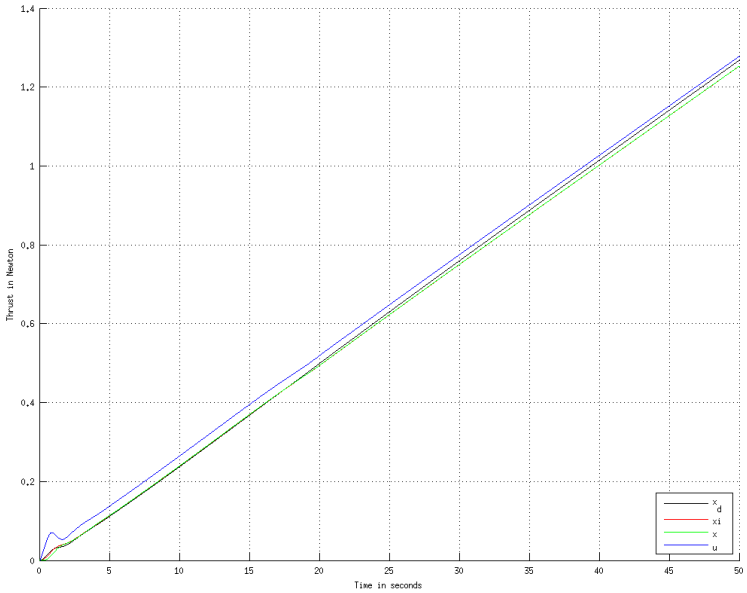


Figure B.8: Right azimuth in y-direction

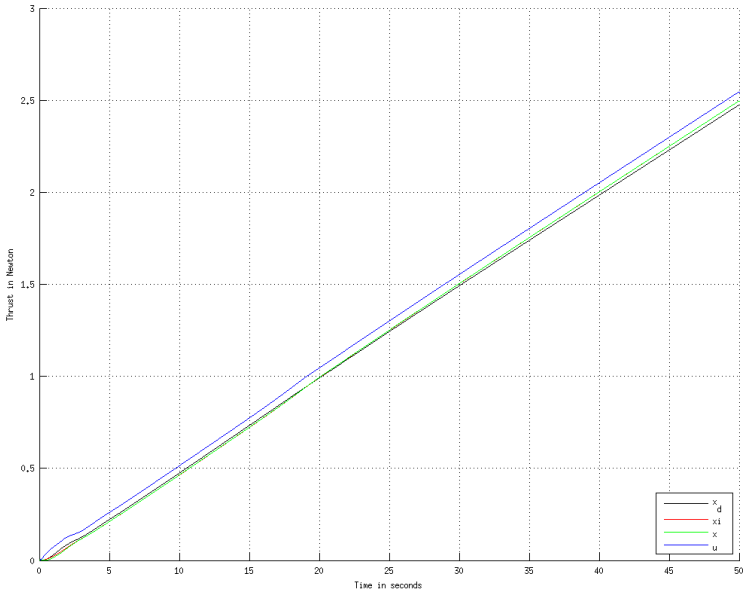


Figure B.9: The bow-thruster

# Appendix C

# Scenario 2

## C.1 Model 1

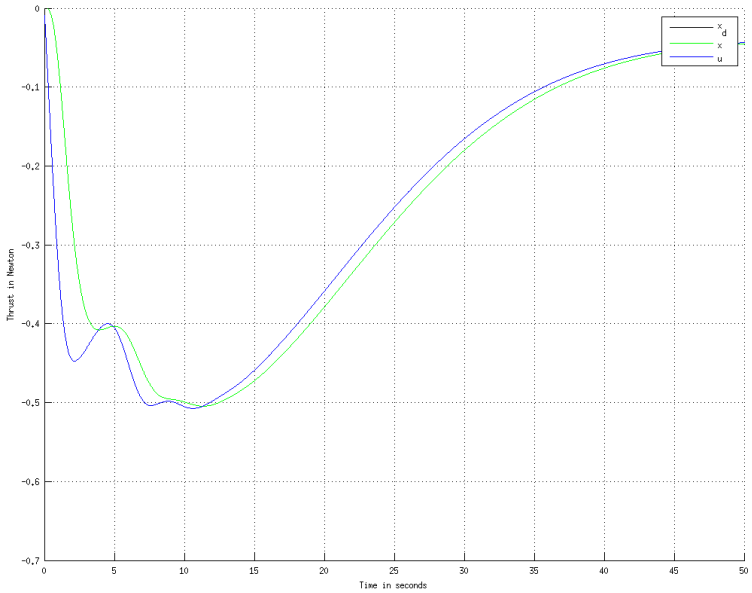


Figure C.1: Right azimuth in x-direction

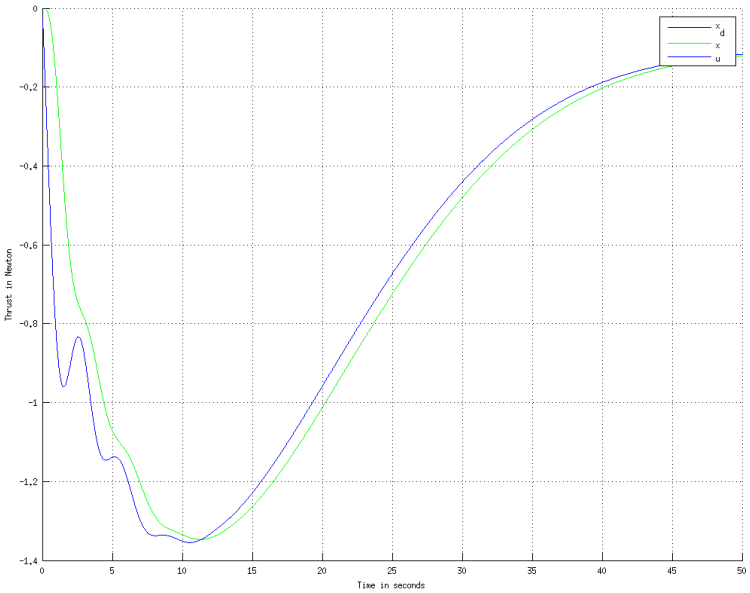


Figure C.2: Right azimuth in y-direction

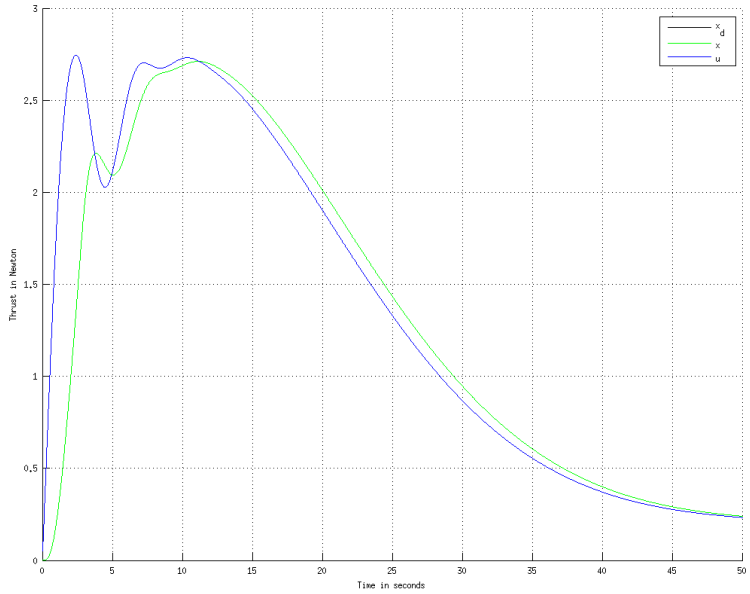


Figure C.3: The bow-thruster

## C.2 Model 2

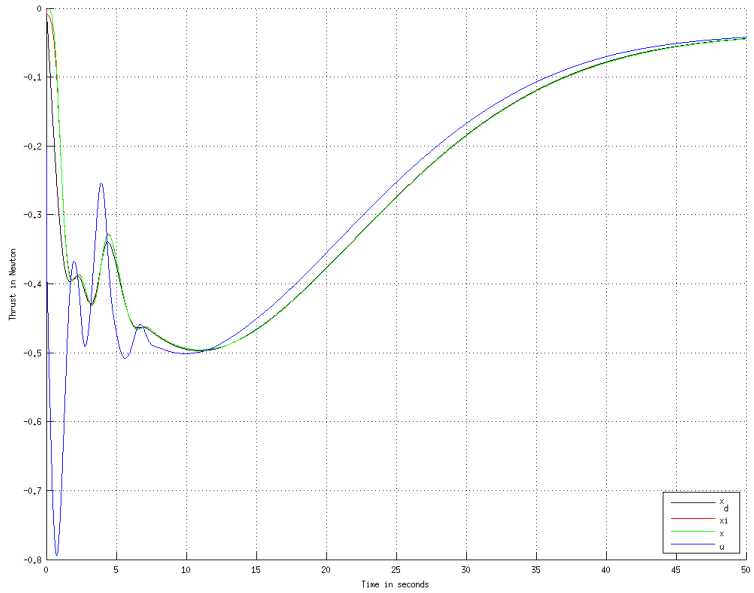


Figure C.4: Right azimuth in x-direction

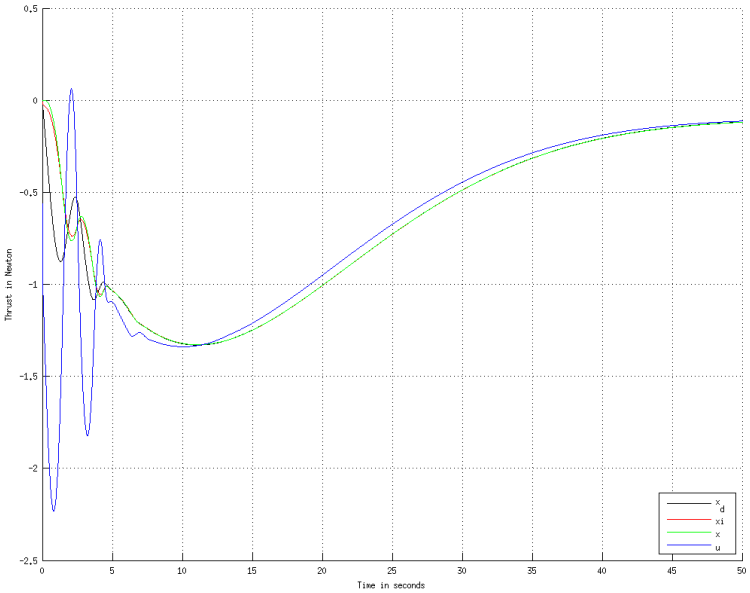


Figure C.5: Right azimuth in y-direction

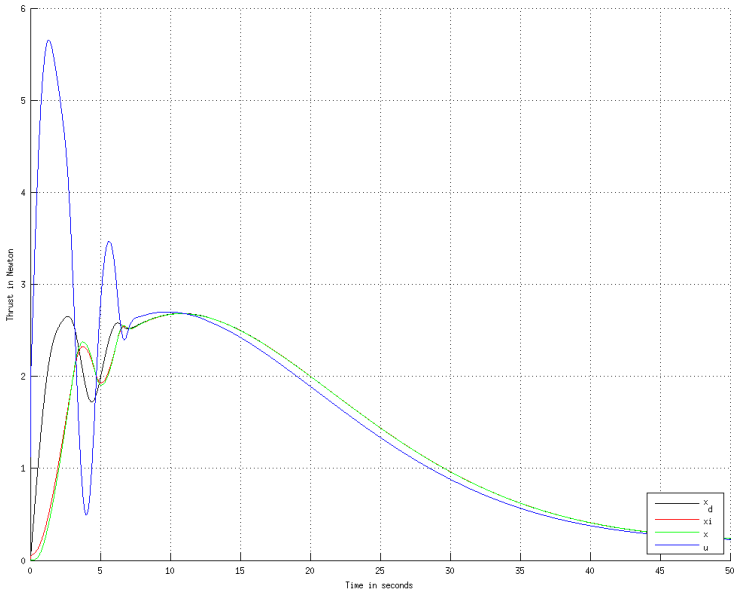


Figure C.6: The bow-thruster

## C.3 Model 3

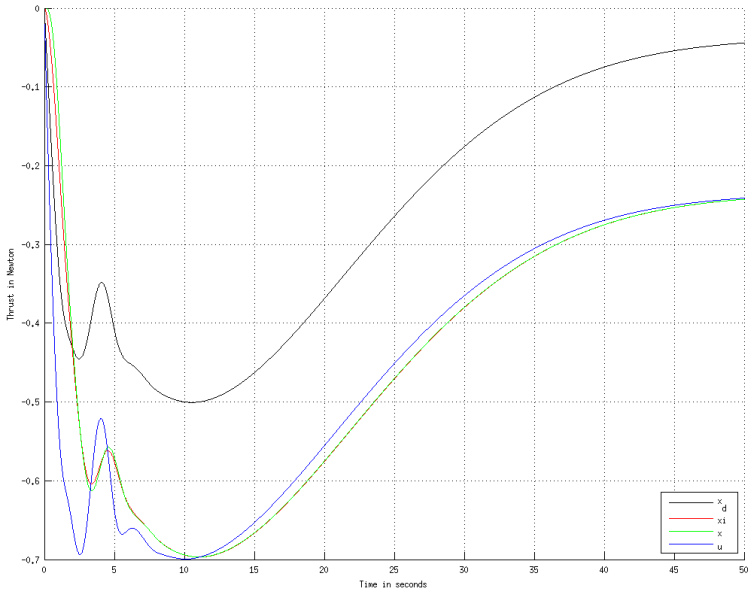


Figure C.7: Right azimuth in x-direction

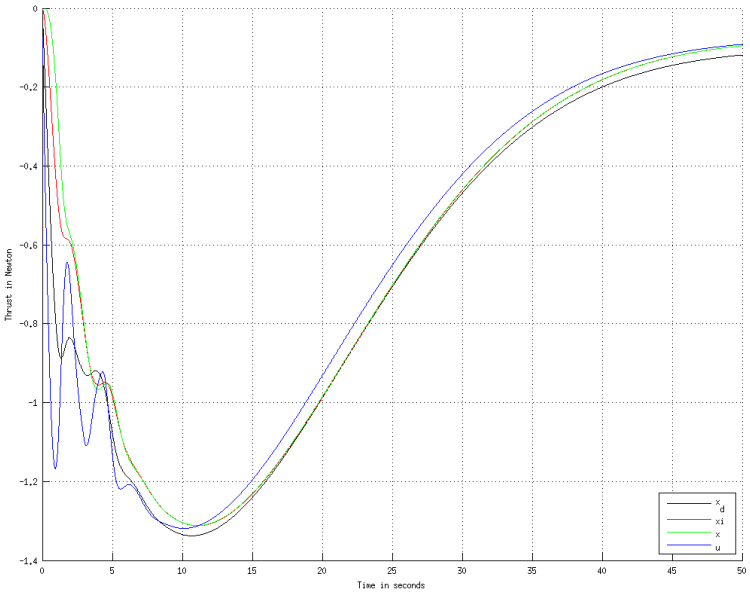


Figure C.8: Right azimuth in y-direction

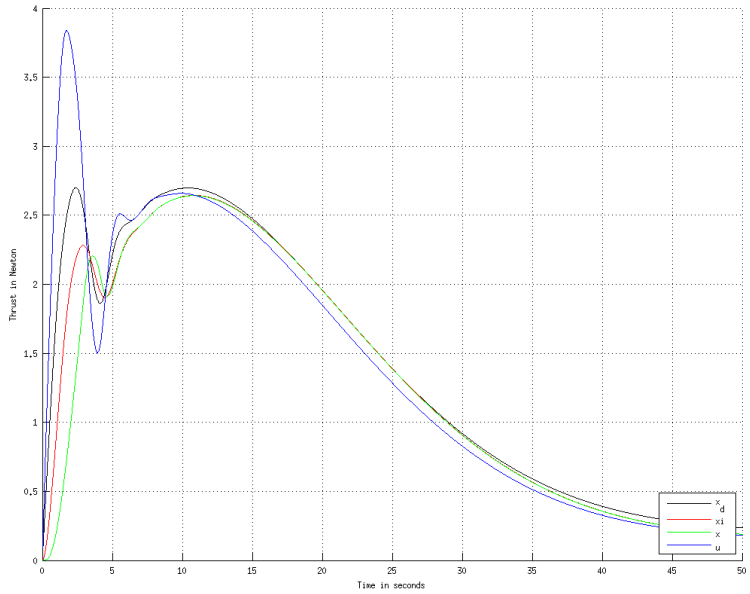


Figure C.9: The bow-thruster

# Appendix D

# Scenario 3

## D.1 Model 1

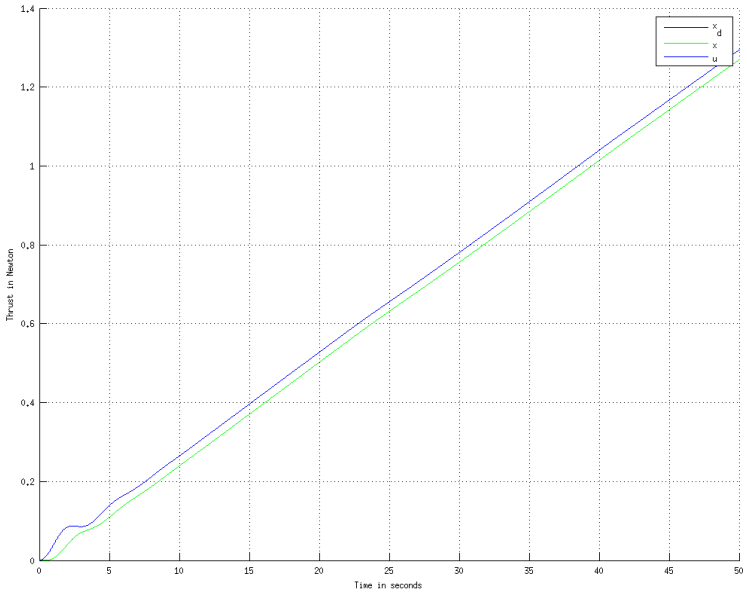


Figure D.1: Scenario 3, Model 1: Left azimuth in y-direction

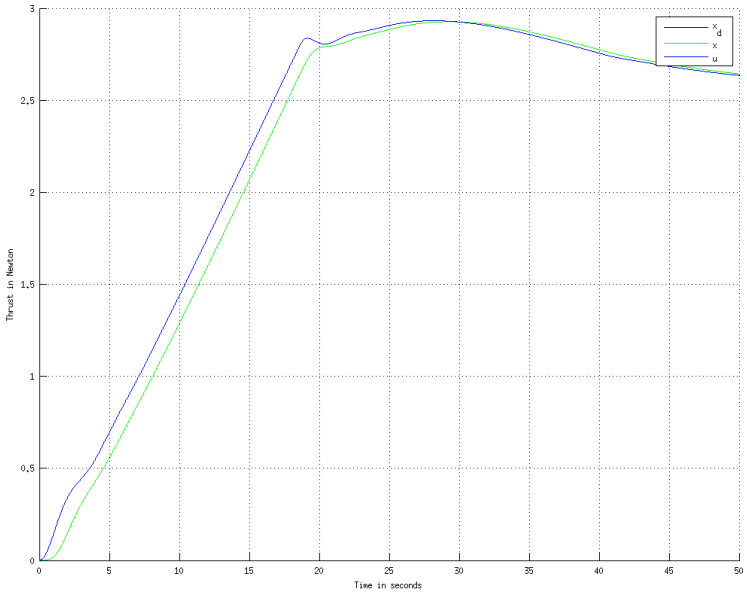


Figure D.2: Right azimuth in x-direction

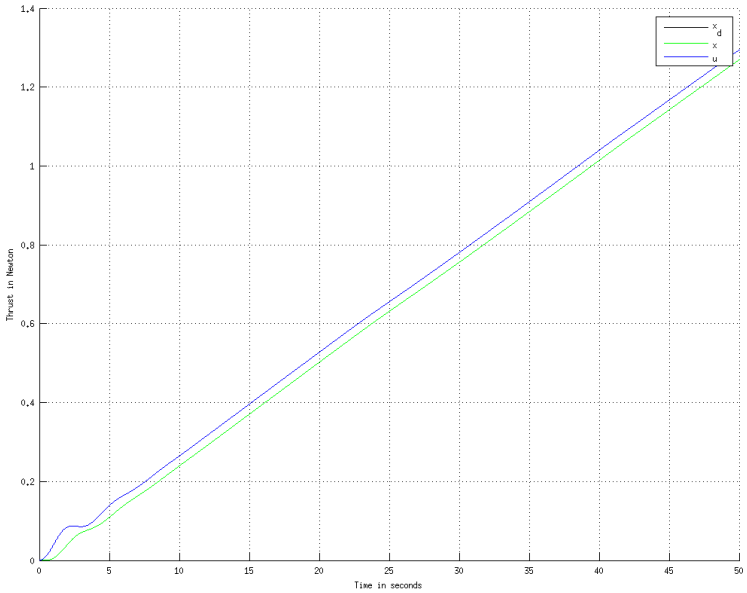


Figure D.3: Right azimuth in y-direction

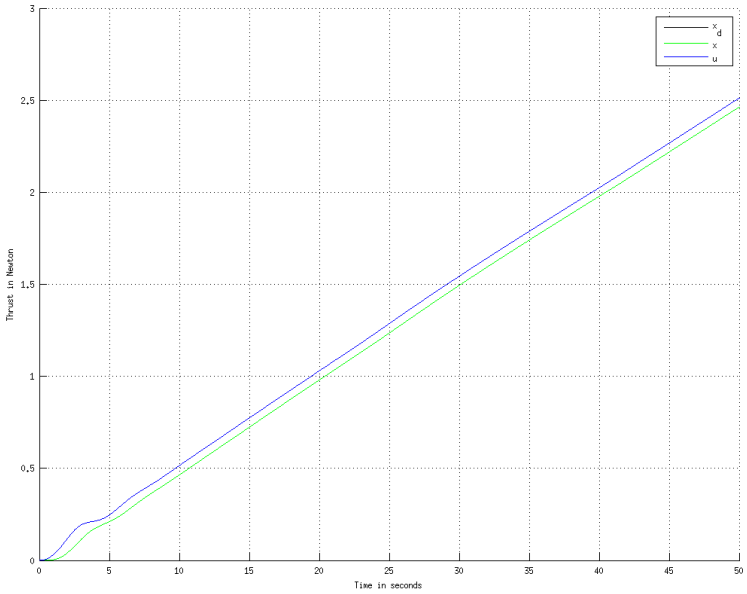


Figure D.4: The bow-thruster

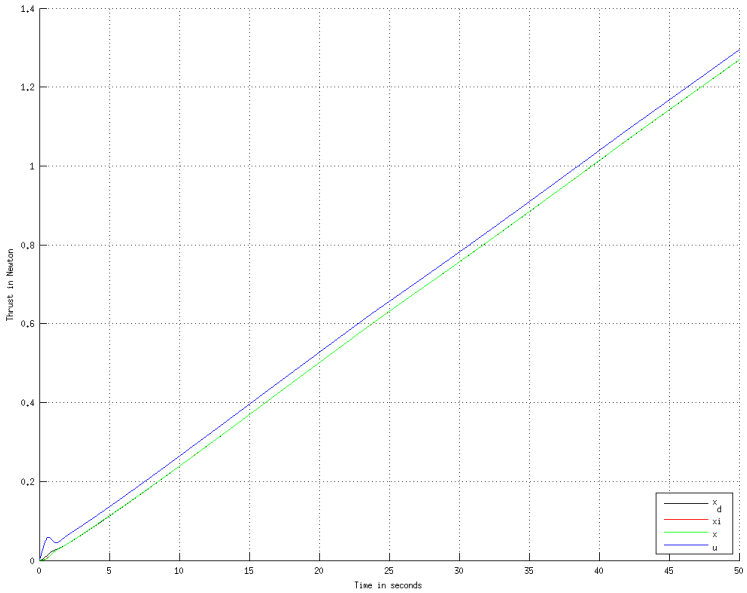


Figure D.5: Scenario 3, Model 2: Left azimuth in y-direction

## D.2 Model 2

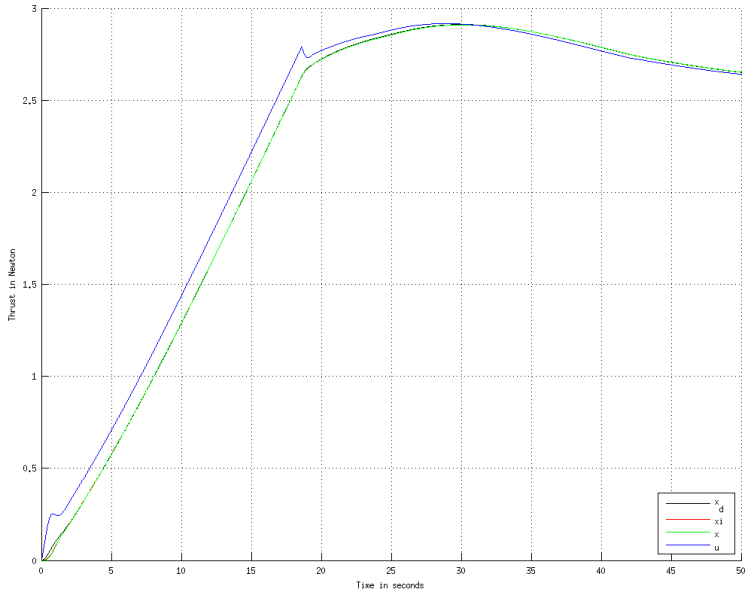


Figure D.6: Right azimuth in x-direction

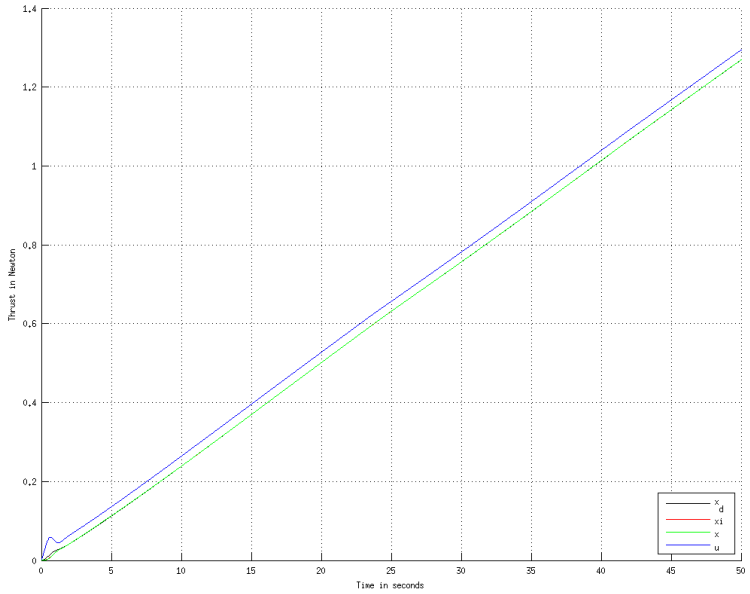


Figure D.7: Right azimuth in y-direction

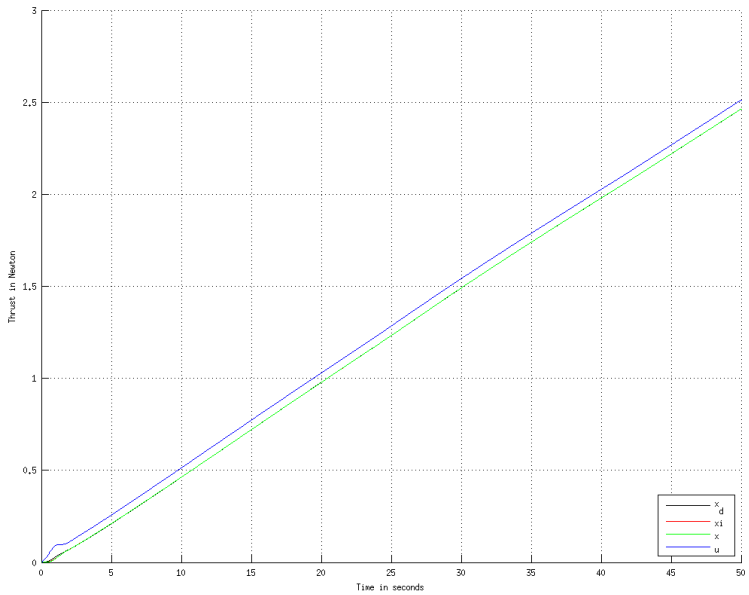


Figure D.8: The bow-thruster

## D.3 Model 3

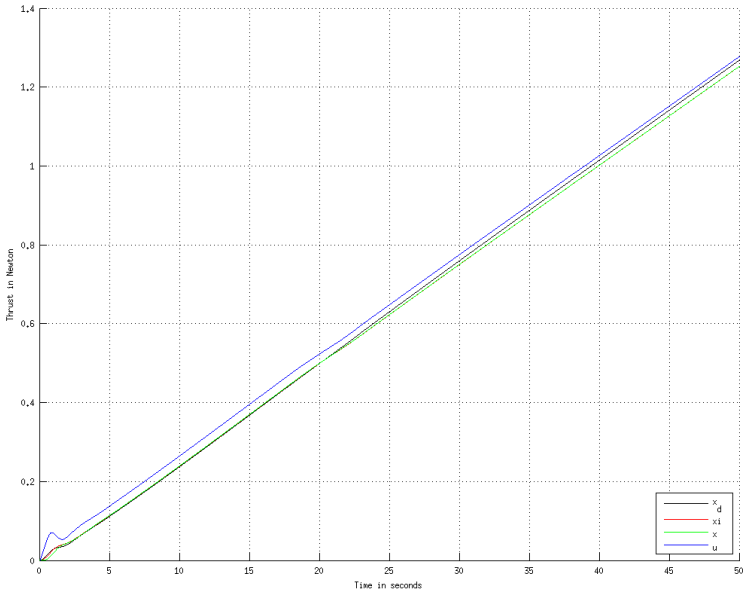


Figure D.9: Scenario 3, Model 3: Left azimuth in y-direction

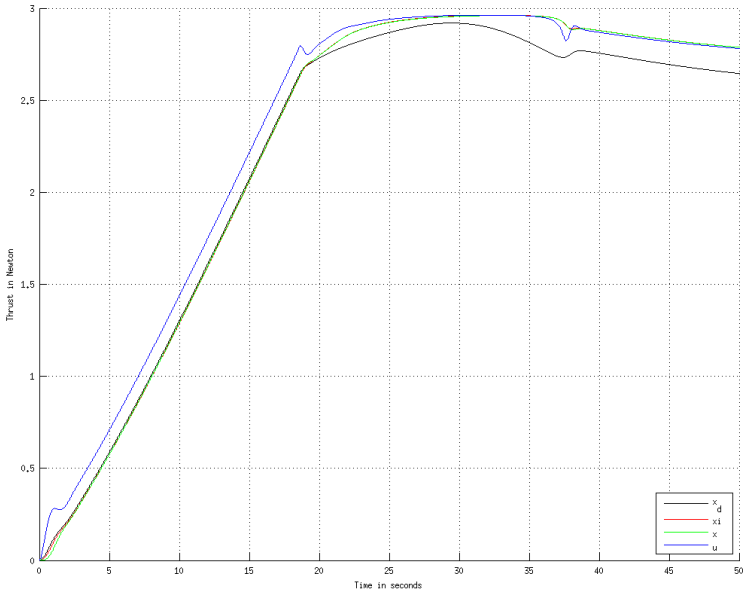


Figure D.10: Right azimuth in x-direction

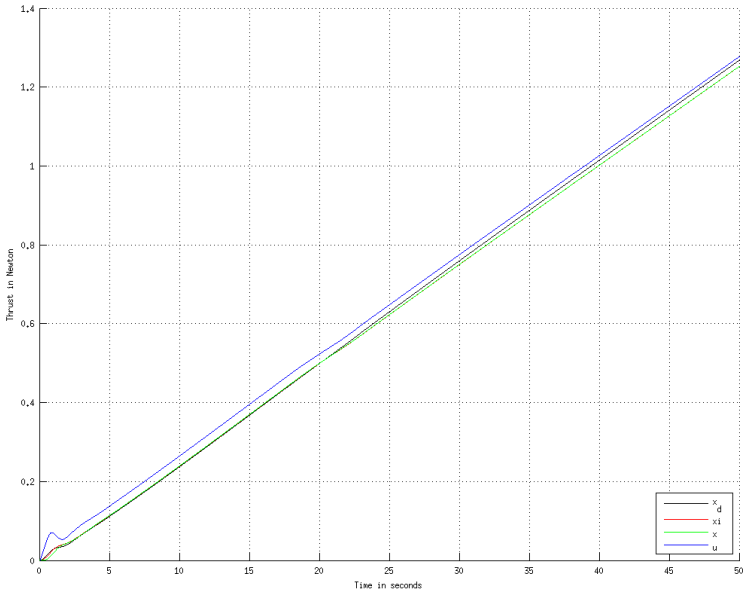


Figure D.11: Right azimuth in y-direction

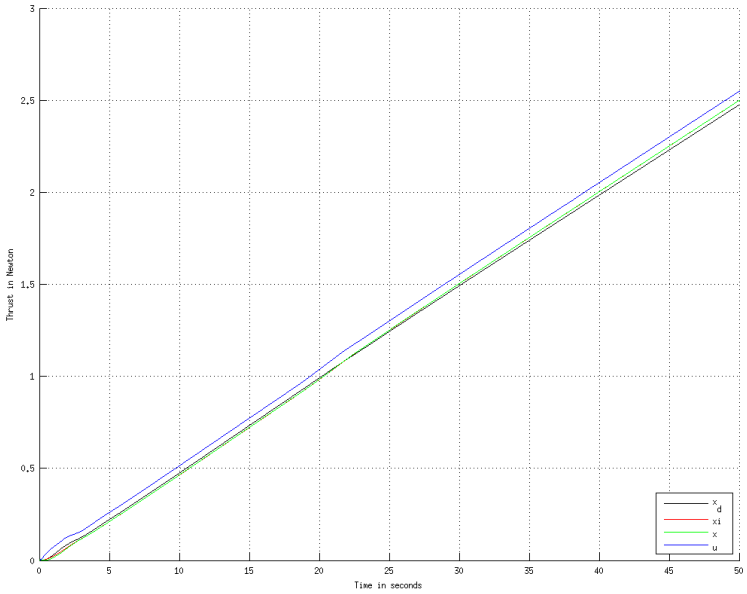


Figure D.12: The bow-thruster