Report 3

Thruster Allocation for Dynamical Positioning

Authors:[1] Koen Poppe[2], Jan Bouwe van den Berg[3], Elisabeth Blank[4], Claude Archer[5], Magnus Redeker[6], Michael Kutter[6], Piet Hemker[7]

Abstract

Positioning a vessel at a fixed position in deep water is of great importance when working offshore. In recent years a Dynamical Positioning (DP) system was developed at Marin [2]. After the measurement of the current position and external forces (like waves, wind etc.), each thruster of the vessel is actively controlled to hold the desired location.

In this paper we focus on the allocation process to determine the settings for each thruster that results in the minimal total power and thus fuel consumption. The mathematical formulation of this situation leads to a nonlinear optimization problem with equality and inequality constraints, which can be solved by applying Lagrange multipliers.

We give three approaches: first of all, the full problem was solved using the MATLAB fmincon routine with the solution from the linearised problem as a starting point. This implementation, with robust handling of the situations where the thrusters are overloaded, lead to promising results: an average reduction in fuel consumption of approximately two percent. However, further analysis proved useful. A second approach changes the set of variables and so reduces the number of equations. The third and last approach solves the Lagrange equations with an iterative method on the linearized Lagrange problem.

3.1 Introduction

In this report, we focus on the allocation part of the full closed loop control system, depicted in Figure 3.1, which is used to keep the vessel in a stationary position. This allocation unit receives the required total force and momentum from the PID-controller and will try to generate these by sending the appropriate control signals to the available actuators of the vessel.

Note that the problem is considered to be 2-dimensional. In fact, any movement in the z-direction (up/down) is ignored due to its periodic behavior. Also, most common actuators do not have the ability to produce trust in the z-direction. This clearly reduces the complexity of the problem.

[1]Other participants: Hans Stigten and Sudhir Jain.
[2]Department of Computer Science, Katholieke Universiteit Leuven
[3]Department of Mathematics, VU University Amsterdam
[4]Department of Mathematics, University of Karlsruhe
[5]Royal Military Academy, Belgium
[6]Department of Mathematics, University Stuttgart
[7]Korteweg-de Vries Institute for Mathematics, University of Amsterdam

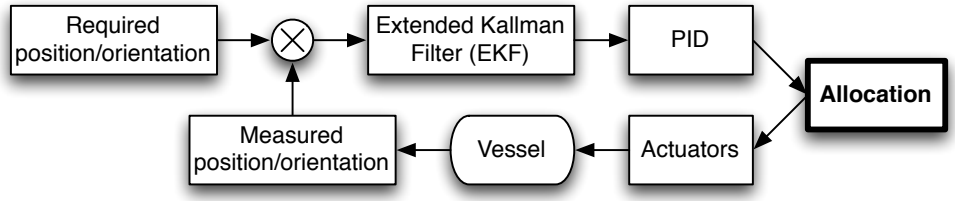


Figure 3.1: The closed loop control system. Measurements of the position of the vessel are compared with the required position. The difference is fed into a Extended Kalman Filter and PID-controller to convert this to the force and momentum required to correct the position. The allocation unit controls the thrusters, which must generate the required force and momentum.

Semi-submersible setup

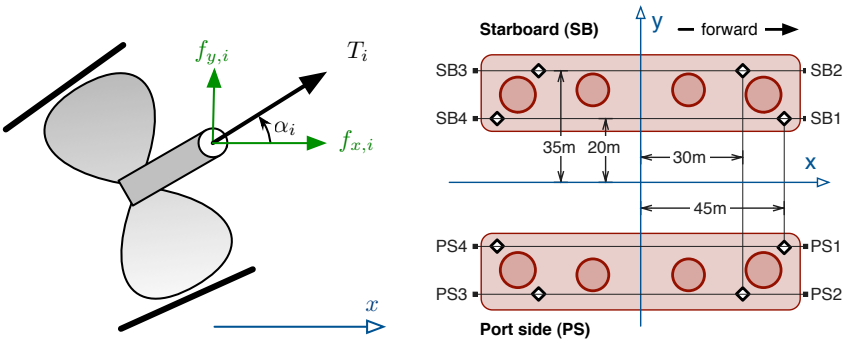
Several types of actuators exist, but we limit ourselves to the azimuth thruster, shown in Figure 3.2(a). These are able to direct their thrust in 360 degrees around the z-axis. They are frequently used in the semi-submersible configuration shown in Figure 3.2(b). This kind of vessel consists of two pontoons which remain under water and eight poles that connect the pontoons with a large rectangular platform above the waters surface, a setup that is often used for offshore drilling in deep water, where anchors cannot be used.

It is necessary to introduce some notations. First of all, the coordinate system is installed in the center of gravity and the x-axis is pointing in the *forward* direction. The z-axis is the upwards direction and so the y axis points towards starboard. Let us denote the total required forces, given in x- and y-direction, by $\mathbb{F}_x$ and $\mathbb{F}_y$ respectively, and $\mathbb{M}_z$, the required momentum in z-direction. These must be generated by the N thrusters that are positioned on the bottom of the ship. We denote the force per thruster by its components $(f_{x,i}, f_{y,i}) \in \mathbb{R}^2$ for $i = 1, ..., N$. An alternative polar notation using $T_i = \sqrt{{f_{x,i}}^2 + {f_{y,i}}^2}$ for the thrust and $\alpha_i \in [-\pi, \pi)$ for the orientation relative to the x-axis, is shown in Figure 3.2(a). Furthermore, we use P for the total power used in a given time step and $\overline{T}_i$ and $\overline{P}_i$ respectively for the maximal thrust and maximal power for thruster i.

Some dimensions and the coordinates (x_i, y_i), indicating the positions of the thrusters on the semi-submersible, are summarized in Figure 3.2(b).

3.2 General problem statement

The allocation problem can be translated to a constrained optimization problem. We introduce the objective function and the constraints of this minimization problem in the following paragraphs.



(a) Azimuth thruster (b) The semi-submersible

Figure 3.2: Detail of an azimuth thruster with its associated variables in 3.2(a) and a top-view of the actual semi-submersible that was used in the simulations. The azimuth thrusters are located on diamonds in 3.2(b).

Objective function

Cost-efficiency is important when considering the dynamical positioning. Therefore, we try to satisfy the requirements with the least energy possible

$$P_\nu = \sum_{i=1}^{N} {T_i}^{2\nu} = \sum_{i=1}^{N} \left({f_{x,i}}^2 + {f_{y,i}}^2 \right)^\nu . \tag{3.1}$$

Here we assume that each thruster has the same specifications. If not, each term in the sum (3.1) must be scaled with a thruster specific constant $\overline{P}_i / {\overline{T}_i}^{2\nu}$.

In this report we use the realistic setting of the power $\nu = \frac{3}{4}$. Previously at MARIN, this optimization problem was only considered for $\nu = 1$, because it leads to linear derivatives of (3.1). Simulations have shown that the energy consumption may be lower by about 2% when using the realistic $\nu = \frac{3}{4}$.

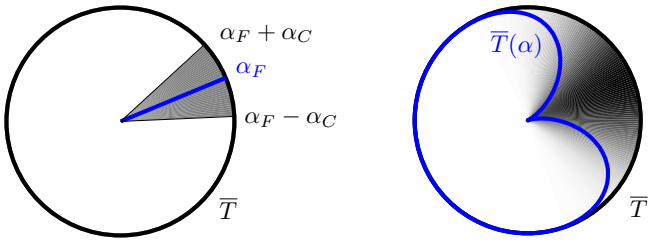
Equality constraints

The first set of constraints follows from the need to generate the required force and momentum. If these would not be satisfied, the vessel can start to drift. To avoid this, the following equality constraints must be satisfied:

$$\mathbb{F}_x = \sum_{i=1}^{N} f_{x,i} \ , \tag{3.2}$$

$$\mathbb{F}_y = \sum_{i=1}^{N} f_{y,i} \ , \tag{3.3}$$

$$\mathbb{M}_z = \sum_{i=1}^{N} x_i f_{y,i} - y_i f_{x,i} \ . \tag{3.4}$$



(a) Forbidden angle approach (3.6). (b) $\overline{T}(\alpha)$ approach.

Figure 3.3: Two ways of dealing with disallowed angles. In 3.3(a), certain angles are forbidden, whilst in 3.3(b), the maximal thrust is variable (depending on α), and the intensity of the gray that indicates the available amount of of available thrust for each angle α.

Note that these conditions assume that the total installed thruster capacity is sufficient. A more elaborate discussion on this is deferred to Section 3.3.

Inequality constraints

The second set of constraints originates from physical limitations. First of all, the thrusters have a limited capacity[8] $\overline{T}_i$. Secondly, thrusters that are close together have an influence on each other: if one is in the stream of the other, its efficiency drops significantly. In order to avoid this, certain angles are prohibited. This results in the following constraints, for $i = 1, \ldots, N$:

$$\overline{T}_i \geq T_i = \sqrt{{f_{x,i}}^2 + {f_{y,i}}^2} \ , \tag{3.5}$$

$$|\alpha_i - \alpha_{F,i}| \geq \alpha_{C,i} \ . \tag{3.6}$$

In this, cf. Figure 3.3(a), $\alpha_{F,i}$ is the center of thruster i's forbidden zone, the angle that is oriented away[9] from the neighboring thruster. The constant $\alpha_{C,i}$ gives the minimal angular distance from $\alpha_{F,i}$ needed to avoid influence on the other thrusters. A typical value used for $\alpha_{C,i}$ is about 10 degrees.

An alternative way to describe the inequalities is by using an angular dependent maximal thrust. In this, the maximum thrust in the forbidden regions can be limited to zero. Constraint (3.5) and (3.6) are combined into

$$\overline{T}_i(\alpha_i) \geq T_i. \tag{3.7}$$

This way of describing forbidden angles provides much more freedom and is easy to adapt for the influence of several near by thrusters. A rough example of such an

[8] This is also an approximation because this is the *open water* thrust capacity. Because of currents and interaction with the hull of the boat, the actual thrust might be far less.

[9] We use a thrust notation that shows the direction of the resulting force, while the water is pushed in the opposite direction. Hence the forbidden angles are opposite to the neighboring thruster.

angular dependent maximum thrust $\overline{T}_i(\alpha_i)$ for two neighboring thrusters is given in Figure 3.3(b). Note that it might be convenient to use a smooth continuous differentiable function for this angular dependence.

Full problem statement

The full allocation problem can be summarized as follows:

$$\begin{aligned} &\text{minimize } (3.1) \\ &\text{subject to } (3.2),\ (3.3),\ (3.4) \text{ and} \\ &\qquad (3.5),\ (3.6) \text{ for } i=1,\ldots,N\ . \end{aligned} \tag{3.8}$$

3.3 Approaches

We are looking for a numerical method to solve the optimization problem described by (3.8) in real time: the computation should be instantaneous compared to the timescales of the vessels actions. Currently, the closed loop control system from Figure 3.1 runs at 4Hz. The method is thus required to take far less than 0.25s, because the allocation is only part of the calculations.

Idea I: Direct nonlinear optimization

With the vast amount of tools available, solving a nonlinear optimization problem is not that difficult. Without considering the efficiency or the computation time, the problem (3.8) can be solved by the MATLAB optimization toolbox. More specifically, fmincon was used to gain insight into the behavior of the problem. For this specific setting, the fmincon routine used a Sequential Quadratic Programming (SQP) approach for the inner, and line search for the outer loop. The Hessian is updated using a quasi-Newton scheme (more details are available in the MATLAB documentation).

In order to provide a reasonable starting point, the solution of the linearized problem was considered at $t=0$. When ignoring the inequality constraints, problem (3.8) with $\nu=1$ has a quadratic Lagrange function. This leads to a much easier linear problem that can be solved directly.

For subsequent time steps, the result of the previous step were taken as initial value. This is reasonable because of the rather large timescales that are involved with the actual movement of the vessel and external influences like wind and water currents.

Alternative penalty approach for overloaded situations

As mentioned before, there may be times where the thrusters can not produce the requested forces. It is necessary to be aware of this situation and to prioritize the requirements. What is most important the drift avoidance or the energy consumption? In practice, the former is considered to be crucial.

No value can be attributed to the output of fmincon when the force and momentum constraints (3.2), (3.3) and (3.4) cannot be met within the thruster capacity

(i.e. (3.5)). It is thus necessary to consider this, undesirable but sometimes unavoidable, case separately. The approach used here is to loosen the constraints on the required forces if the capacity of the thrusters is insufficient. The force/momentum constraints (3.2), (3.3), (3.4) are then dropped, but they are reflected by an additional penalization term in the modified objective function. In particular, we penalize differences between required and provided forces and momentum. Together with the power consumption of the thrusters, this leads to the following modified objective function:

$$\tilde{P}_\nu = w_1 P_\nu + w_2 \left({\Delta_{\mathbb{F}_x}}^2 + {\Delta_{\mathbb{F}_y}}^2 \right) + w_3 {\Delta_{\mathbb{M}_z}}^2 \ , \tag{3.9}$$

where w_1, w_2 and w_3 are weights (to be chosen appropriately), and

$$\begin{aligned} \Delta_{\mathbb{F}_x} &= \mathbb{F}_x - \sum_{i=1}^{N} f_{x,i} \ , \\ \Delta_{\mathbb{F}_y} &= \mathbb{F}_y - \sum_{i=1}^{N} f_{y,i} \ , \\ \Delta_{\mathbb{M}_z} &= \mathbb{M}_z - \sum_{i=1}^{N} (x_i f_{y,i} - y_i f_{x,i}) \, . \end{aligned} \tag{3.10}$$

The associated alternative minimization problem, *that is used only when the thrusters have insufficient capacity*, can be written as

$$\begin{array}{c} \text{minimize (3.9)} \\ \text{subject to (3.5), (3.6) for } i = 1, \ldots, N. \end{array} \tag{3.11}$$

The weights in (3.9) need to be chosen so that the right balance is reached between the energy consumption, drift and orientation changes. Consulting the experts from MARIN led to the use of $w_1 = 0$: there is no intention of minimizing the power in extreme weather conditions. The other weights were chosen relative to the maximal required thrust and moment for a given time series. This somehow equally treats both position and orientation errors.

Idea II: Reduction to a 3×3 nonlinear system

We will show that, even for the more realistic case where $\nu = \frac{3}{4}$, the $3N+3$ Lagrange equations can be reduced to only 3 equations and the 3 unknown multipliers. The equations obtained from the multipliers technique can be simplified using T_i and α_i, the thrust and azimuth angle of thruster i, as main variables. The associated feasible region then reduces to the product of intervals $T_i \notin [0, \overline{T}_i]$ and $\alpha_i \in [\alpha_{F,i} - \alpha_{C,i}, \alpha_{F,i} + \alpha_{C,i}]$. Solutions will be either inside this domain, as stationary points of the Lagrange function $\Lambda(\mathbf{T}, \mathbf{A}, \mathbf{L})$ (where we collect the thrusts $\mathbf{T} = [T_1, \ldots, T_N]^\mathsf{T}$, angles $\mathbf{A} = [\alpha_1, \ldots, \alpha_N]^\mathsf{T}$ and Lagrange multipliers $\mathbf{L} = [\lambda_1, \lambda_2, \lambda_3]^\mathsf{T}$ in column vectors) or on the boundary of the domain. These cases are discussed separately.

Solutions in the inside of the feasible region

Using the notation (3.10), the Lagrange function for problem (3.8), assuming that the inequality constraints are inactive, can be written as

$$\Lambda_{\nu=\frac{3}{4}}(\mathbf{T},\mathbf{A},\mathbf{L}) = \sum_{i=1}^{N} {T_i}^{\frac{3}{2}} - \lambda_1 \Delta_{\mathbb{F}_x} - \lambda_2 \Delta_{\mathbb{F}_y} - \lambda_3 \Delta_{\mathbb{M}_z}. \tag{3.12}$$

For a given thruster i, collecting terms linear in $f_{x,i}$ and $f_{y,i}$ gives $u_i f_{x,i} + v_i f_{y,i}$ where $u_i = \lambda_1 - y_i\lambda_3$ and $v_i = \lambda_2 + x_i\lambda_3$. Using the relations $f_{x,i} = T_i\cos(\alpha_i)$ and $f_{y,i} = T_i\sin(\alpha_i)$, the collected terms are written as $T_i(u_i\cos(\alpha_i) + v_i\sin(\alpha_i))$ and further summarized to $T_i h_i$ using $h_i = u_i\cos(\alpha_i) + v_i\sin(\alpha_i)$. This gives

$$\Lambda_{\nu=\frac{3}{4}}(\mathbf{T},\mathbf{A},\mathbf{L}) = \sum_{i=1}^{N} \left({T_i}^{\frac{3}{2}} + T_i h_i \right) - \lambda_1 \mathbb{F}_x - \lambda_2 \mathbb{F}_y - \lambda_3 \mathbb{M}_z \ . \tag{3.13}$$

In stationary points, the partial derivatives of the Lagrange function $\Lambda_{\nu=\frac{3}{4}}$, with respect to T_i and α_i for each thruster i, must be zero and can be solved:

$$0 = \frac{\partial \Lambda_{\nu=\frac{3}{4}}}{\partial T_i} = \frac{3}{2}\sqrt{T_i} + h_i \quad \Rightarrow \quad T_i = \frac{4}{9}{h_i}^2 \ , \tag{3.14}$$

$$0 = \frac{\partial \Lambda_{\nu=\frac{3}{4}}}{\partial \alpha_i} = -T_i u_i \sin(\alpha_i) + T_i v_i \cos(\alpha_i) \quad \overset{T_i \neq 0}{\Longrightarrow} \quad \tan(\alpha_i) = \frac{v_i}{u_i} \ . \tag{3.15}$$

If $T_i = 0$, the thruster i is switched off and any value of α_i is a solution.

With the above, thrust T_i can be rewritten as a function of u_i and v_i:

$$h_i = \pm u_i \underbrace{\left(\frac{u_i}{\sqrt{{u_i}^2 + {v_i}^2}} \right)}_{\cos(\alpha_i)} \pm v_i \underbrace{\left(\frac{v_i}{\sqrt{{u_i}^2 + {v_i}^2}} \right)}_{\sin(\alpha_i)} = \pm\sqrt{{u_i}^2 + {v_i}^2}. \tag{3.16}$$

Both T_i and α_i depend only on the Lagrange multipliers λ_1, λ_2 and λ_3. The stationary points are thus characterized by the multipliers, that can be found by substitution T_i and α_i into the equality constraints (3.2),(3.3) and (3.4):

$$\begin{cases} \mathbb{F}_x = \sum_{i=1}^{N} T_i\cos(\alpha_i) & = \frac{4}{9}\sum_{i=1}^{N} h_i u_i, \\ \mathbb{F}_y = \sum_{i=1}^{N} T_i\sin(\alpha_i) & = \frac{4}{9}\sum_{i=1}^{N} h_i v_i, \\ \mathbb{M}_z = \sum_{i=1}^{N} T_i(x_i\sin(\alpha_i) - y_i\cos(\alpha_i)) & = \frac{4}{9}\sum_{i=1}^{N} h_i(x_i v_i - y_i u_i). \end{cases} \tag{3.17}$$

These are thus three equations for the three unknown multipliers λ_1, λ_2 and λ_3. When such a stationary point exists inside the domain, the optimal value of the power P is then equal to $\sum_{i=1}^{N} {T_i}^{\frac{3}{2}} = \frac{8}{27}\sum_{i=1}^{N} {h_i}^3$.

Solutions on the boundary: saturation and forbidden angles

With the Lagrange multipliers technique, it is possible to take inequalities into account, introducing a new multiplier μ_j for each inequality. Derivations lead to the following conclusion: either inequality j becomes an equality (border of the domain) or $\mu_j = 0$. Hence, inside the domain all μ_j vanish and the extended Lagrange function is equal to the previous $\Lambda = \Lambda_{\nu=\frac{3}{4}}$.

On a border, optimizing the total power can also be done by the multiplier technique using the restriction of Lagrange function Λ to this border of the domain. As the domain is expressed as boundary values for T_i and α_i, the restriction of Λ to a border amounts to fixing the value of T_i (resp. α_i) in Λ. Hence all previous derivations remain valid excepted $\frac{\partial \Lambda}{\partial T_i}$ (resp. $\frac{\partial \Lambda}{\partial \alpha_i}$) that is no longer needed to determine T_i (resp. α_i) as a function of the λ's. Observe that when T_i is saturated, equation $\frac{\partial \Lambda}{\partial \alpha_i} = 0$ still allows to determine $\alpha_i = \arctan\left(\frac{v_i}{u_i}\right)$ and conversely, if α_i is a forbidden angle, equation $\frac{\partial \Lambda}{\partial T_i} = 0$ still allows to determine $T_i(\lambda_1, \lambda_2, \lambda_3)$. Finally, if for some i, both T_i and α_i reach boundary values, then the allocation problem is reduced to a problem with $N-1$ thrusters and new known constraints ($\mathbb{F}_x := \mathbb{F}_x - f_{x,i}$, and similar for $\mathbb{F}_y$ and $\mathbb{M}_z$) and it reduces to the same system of 3 equations for a subset of indices in the summations.

Idea III: Iteration on a linearized Lagrange problem

For this strategy, we distinguish a simplified version of the problem and the full problem. By first solving a simplified version, an initial estimate of the solution is obtained. This can be used in an iterative algorithm which adapts the simple version to obtain improved versions. A few of these defect correction type iterations yield the final solution.

The simplified problem ($\nu = 1$)

We simplify the optimization problem (3.8) by taking $\nu = 1$ and the angular dependent maximal thrust constraint (3.7) instead of constraints (3.5) and (3.6):

$$\begin{array}{c} \text{minimize (3.1) for } \nu = 1 \\ \text{subject to (3.2), (3.3), (3.4) and} \\ \text{(3.7) for } i = 1, \ldots, N \ . \end{array} \tag{3.18}$$

Denoting $\mathcal{J}$ as the set of indices j for which thruster j is saturated, e.g. inequality (3.7) becomes an equality, using vectors $\mathbf{F} = [f_{x,1}, f_{y,1}, \ldots, f_{x,N}, f_{y,N}]^\mathsf{T}$ $\mathbf{M} = \{\mu_j\}_{j \in \mathcal{J}}$ and $\mathbf{L} = [\lambda_1, \lambda_2, \lambda_3]^\mathsf{T}$, the Lagrange function becomes

$$\begin{aligned} \Lambda_{\nu=1}(\mathbf{F}, \mathbf{L}, \mathbf{M}) = P_{\nu=1} - \lambda_1 \Delta_{\mathbb{F}_x} - \lambda_2 \Delta_{\mathbb{F}_y} - \lambda_3 \Delta_{\mathbb{M}_z} \\ - \textstyle\sum_{j \in \mathcal{J}} \mu_j \left(\overline{T}_j(\alpha_j)^2 - {f_{x,j}}^2 - {f_{y,j}}^2 \right). \end{aligned} \tag{3.19}$$

Now we can find $\mathbf{F}$, $\mathbf{L}$, and $\mathbf{M}$ such that $(\mathbf{F}, \mathbf{L}, \mathbf{M})$ is a stationary point for the Lagrange function (3.19), by solving the following system (for notational convenience,

we write the system only for $N=4$ thrusters):

$$
\left[\begin{array}{cccccccc|ccc|cc}
2 & & & & & & & & 2f_{x,1} & & & 1 & -y_1 \\
 & 2 & & & & & & & 2f_{y,1} & & & 1 & x_1 \\
 & & 2 & & & & & & & 2f_{x,2} & & 1 & -y_2 \\
 & & & 2 & & & & & & 2f_{y,2} & & 1 & x_2 \\
 & & & & 2 & & & & & & 2f_{x,3} & 1 & -y_3 \\
 & & & & & 2 & & & & & 2f_{y,3} & 1 & x_3 \\
 & & & & & & 2 & & & & 2f_{x,4} & 1 & -y_4 \\
 & & & & & & & 2 & & & 2f_{y,4} & 1 & x_4 \\
\hline
2f_{x,1} & 2f_{y,1} & & & & & & & & & & & \\
 & & 2f_{x,2} & 2f_{y,2} & & & & & & & & & \\
 & & & & 2f_{x,3} & 2f_{y,3} & & & & & & & \\
 & & & & & & 2f_{x,4} & 2f_{y,4} & & & & & \\
\hline
1 & & 1 & & 1 & & 1 & & & & & & \\
 & 1 & & 1 & & 1 & & 1 & & & & & \\
-y_1 & x_1 & -y_2 & x_2 & -y_3 & x_3 & -y_4 & x_4 & & & & &
\end{array}\right]
\left[\begin{array}{c}
f_{x,1} \\ f_{y,1} \\ f_{x,2} \\ f_{y,2} \\ f_{x,3} \\ f_{y,3} \\ f_{x,4} \\ f_{y,4} \\
\hline
\mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \\
\hline
\lambda_1 \\ \lambda_2 \\ \lambda_3
\end{array}\right]
$$

$$
= \left[\begin{array}{c}
0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\
\hline
\overline{T}_1(\alpha_1)^2 + {f_{x,1}}^2 + {f_{y,1}}^2 \\
\overline{T}_2(\alpha_2)^2 + {f_{x,2}}^2 + {f_{y,2}}^2 \\
\overline{T}_3(\alpha_3)^2 + {f_{x,3}}^2 + {f_{y,3}}^2 \\
\overline{T}_4(\alpha_4)^2 + {f_{x,4}}^2 + {f_{y,4}}^2 \\
\hline
\mathbb{F}_x \\ \mathbb{F}_y \\ \mathbb{M}_z
\end{array}\right]. \tag{3.20}
$$

For each non-saturated thruster j, the corresponding μ_j, row and column should be omitted from the system. Notice that the system is symmetric in any case.

Using $\mathbf{R} = [\mathbb{F}_x, \mathbb{F}_y, \mathbb{M}_z]^\mathsf{T}$ for the requirements and $\hat{\mathbf{T}}^2$ for part of the right hand side, i.e. $\hat{\mathbf{T}}^2 = \left[\overline{T}_1{}^2 + {f_{x,1}}^2 + {f_{y,1}}^2, \ldots, \overline{T}_N{}^2 + {f_{x,N}}^2 + {f_{y,N}}^2\right]^\mathsf{T}$, we can write this system as

$$
\begin{bmatrix} \mathbf{C} & \mathbf{B}^\mathsf{T} & \mathbf{E}^\mathsf{T} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} \\ \mathbf{E} & \mathbf{0} & \mathbf{0} \end{bmatrix}
\begin{bmatrix} \mathbf{F} \\ \mathbf{M} \\ \mathbf{L} \end{bmatrix} =
\begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{T}}^2 \\ \mathbf{R} \end{bmatrix}, \tag{3.21}
$$

where $\mathbf{C}$ is an diagonal matrix of dimension $2N \times 2N$; the matrix $\mathbf{E}$ is $3 \times 2N$ and the matrix $\mathbf{B}$ is $k \times 2N$, where k is the number of indices in $\mathcal{J}$, that is the number of inequalities (3.7) that is reduced to equalities (i.e. the number of saturated thrusters).

In the initial step we assume none of the constraints (3.7) is active, i.e. $\mathcal{J} = \emptyset$. This implies that in the initial step the rows and columns with $\mathbf{B}$, $\mathbf{L}$ and $\hat{\mathbf{T}}$ are absent from (3.21). So we solve the system (3.21) for $\mathbf{F}$ and $\mathbf{L}$ to find the estimate $\mathbf{F}^{[0]}$.

As soon as an estimate $\mathbf{F}^{[m]}$ is available we can compute the thrusts T_i

$$T_i = \overline{T}_i \left(\arctan \left(\frac{(f_{y,i})^{[m]}}{(f_{x,i})^{[m]}} \right) \right) \quad \text{for } i = 1, \cdots, N \tag{3.22}$$

and find $\mathcal{J}$, the set of indices j for which the constraints (3.7) is violated. For those indices, we scale that vector so that the trust is equal to the maximum:

$$f_{x,j} := \frac{\overline{T}_j(\alpha_i)}{T_j} f_{x,j} \quad \text{and} \quad f_{y,j} := \frac{\overline{T}_j(\alpha_i)}{T_j} f_{y,j} \ . \tag{3.23}$$

If $\mathcal{J} \neq \emptyset$ we compute $\mathbf{B}$, $\overline{T}_i$ and $\hat{\mathbf{T}}$, using $\mathbf{F}^{[m]}$, and solve the system (3.21) again. The iterative process ends when the set $\mathcal{J}$ does not change any more[10].

After the first iteration, instead of system (3.21) with the artificial $\nu = 1$, it is more realistic to solve the system that will be described in the following.

The true, non-simplified problem ($\nu = \frac{3}{4}$)

In order to remove the simplification made in Section 3.3 by setting $\nu = 1$ instead of $\nu = \frac{3}{4}$ in the expression P_ν, we have to correct for it. The function P_ν was the original object function for optimization, and the value of ν is reflected in the system (3.21) only in the $\mathbf{C}$. In fact $\mathbf{C}$ is the Hessian of P, which is constructed to make the gradient of P vanish.

Namely, for $\nu = 1$ the gradient of $P_{\nu=1}$ reads

$$\nabla P_{\nu=1}(\mathbf{F}) = \frac{\partial P_{\nu=1}}{\partial \mathbf{F}} = 2(f_{x,1}, f_{y,1}, f_{x,2}, f_{y,2}, \cdots, f_{x,N}, f_{y,N})^{\mathsf{T}} \tag{3.24}$$

and the equations $\mathbf{CF} = 0$ make the gradient vanish.

If $\nu = \frac{3}{4}$ the gradient can be written with '$\div$', an element-wise division

$$\nabla P_{\nu=3/4}(\mathbf{F}) = \frac{\partial P_{\nu=3/4}}{\partial \mathbf{F}} = \frac{3}{2}\mathbf{F} \div \left[\sqrt{T_1}, \sqrt{T_1}, \sqrt{T_2}, \sqrt{T_2}, \ldots, \sqrt{T_N}, \sqrt{T_N} \right]^{\mathsf{T}} .$$

Applying the defect correction principle [1], we compute the true problem by replacing the system (3.21) by

$$\begin{bmatrix} \mathbf{C} & \mathbf{B}^{\mathsf{T}} & \mathbf{E}^{\mathsf{T}} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} \\ \mathbf{E} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \\ \mathbf{L} \end{bmatrix} = \begin{bmatrix} \nabla P_{\nu=1}(\mathbf{F}) - \nabla P_{\nu=3/4}(\mathbf{F}) \\ \hat{\mathbf{T}}^2 \\ \mathbf{R} \end{bmatrix}, \tag{3.25}$$

[10] Because of the weak non-linearity of the system, a nice convergence behavior of the iteration is expected.

so that the iterative improvement of the solution of the problem we are interested in, is achieved by solving for $\mathbf{F}^{[m+1]}$ from the system

$$\begin{bmatrix} \mathbf{C} & \mathbf{B}^{[m]^\mathsf{T}} & \mathbf{E}^\mathsf{T} \\ \mathbf{B}^{[m]} & \mathbf{0} & \mathbf{0} \\ \mathbf{E} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{F}^{[m+1]} \\ \mathbf{M} \\ \mathbf{L} \end{bmatrix} = \begin{bmatrix} \nabla P_{\nu=1}(\mathbf{F}^{[m]}) - \nabla P_{\nu=3/4}(\mathbf{F}^{[m]}) \\ \hat{\mathbf{T}}^{[m]^2} \\ \mathbf{R} \end{bmatrix}, \tag{3.26}$$

for $m = 0, 1, 2, \ldots$. Upon convergence, this leads to the stationary point of (3.19) with $\nu = \frac{3}{4}$, hence to the solution of the full optimization problem.

3.4 Results

The optimization using fmincon was used to solve the allocation problem for a sequence of $\mathbb{F}_x$, $\mathbb{F}_y$ and $\mathbb{M}_z$, provided by MARIN. The resulting thrusts T_i and orientations α_i are plotted for the first 250 seconds in Figure 3.4-3.5, where the second figure uses the modified algorithm with the robust handling of the overloaded situation (see Section 3.3).

The baseline implementation took on average 0.15s per iteration for the given sequence. This was reduced to about 0.03s by optimizing the code and by providing some of the derivatives analytically (profiling showed that the finite difference approximations of this derivatives took most of the time).

We tried to mimic the approach that MARIN uses nowadays (with $\nu = 1$). It is unclear if this is exactly identical to their program (there is no way of verifying due to confidentiality of their actual results). Our implementation, in which we use the realistic value $\nu = \frac{3}{4}$, proved to require about 2% less power on average than when considering the previously used quadratic objective function ($\nu = 1$). It is clearly worth considering the realistic case with $\nu = \frac{3}{4}$: in some time steps, the excess power for $\nu = 1$ reached up to 5% of our optima.

In order to facilitate the interpretation, a MATLAB visualization was written to show the thrust together with their orientation (see Figure 3.6). By joining static plots for the given sequence of required values, a movie was made. It allows for a very natural way of inspecting the results because the human eye is able to see trends, even for several thrusters simultaneously. Waves for example, can easily be spotted in this manner because all thrusters re-orient in a similar fashion. This video also shows the required and achieved forces and moments. Violations of the constraints are indicated by the use of color, as explained in Figure 3.6. For the time being, the visualization is specialized for the azimuthal thrusters, but it can be extended to other type of actuators, as can the analysis in Section 3.3.

3.5 Recommendations

If the speed and quality of the straight forward MATLAB implementation is sufficient, this optimization routine can be used in the software with minimal effort. If not, the other approaches should be investigated further.

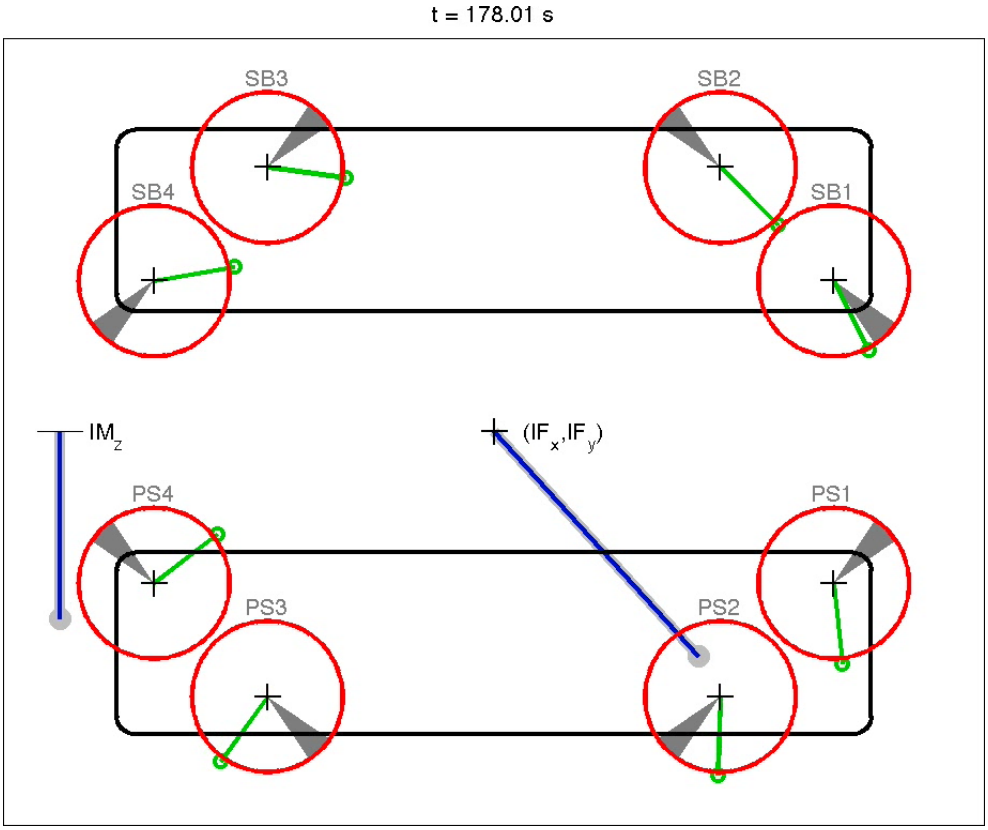


Figure 3.4: Resulting thrust and orientation for the direct nonlinear optimization (Section 3.3) for a given sequence of requested force and momentum. Each plot represents one of the eight thrusters on starboard (SB) and port side (PS). The left axis shows the thrust T_i with an indication of the maximum $\overline{T}_i$ (at some points in time, the thrusters are indeed overloaded), the right axis shows the azimuth angle α_i and the forbidden angle range $[\alpha_{F,i} - \alpha_{C,i}, \alpha_{F,i} + \alpha_{C,i}]$ (only within visible range for thruster 'PS3').

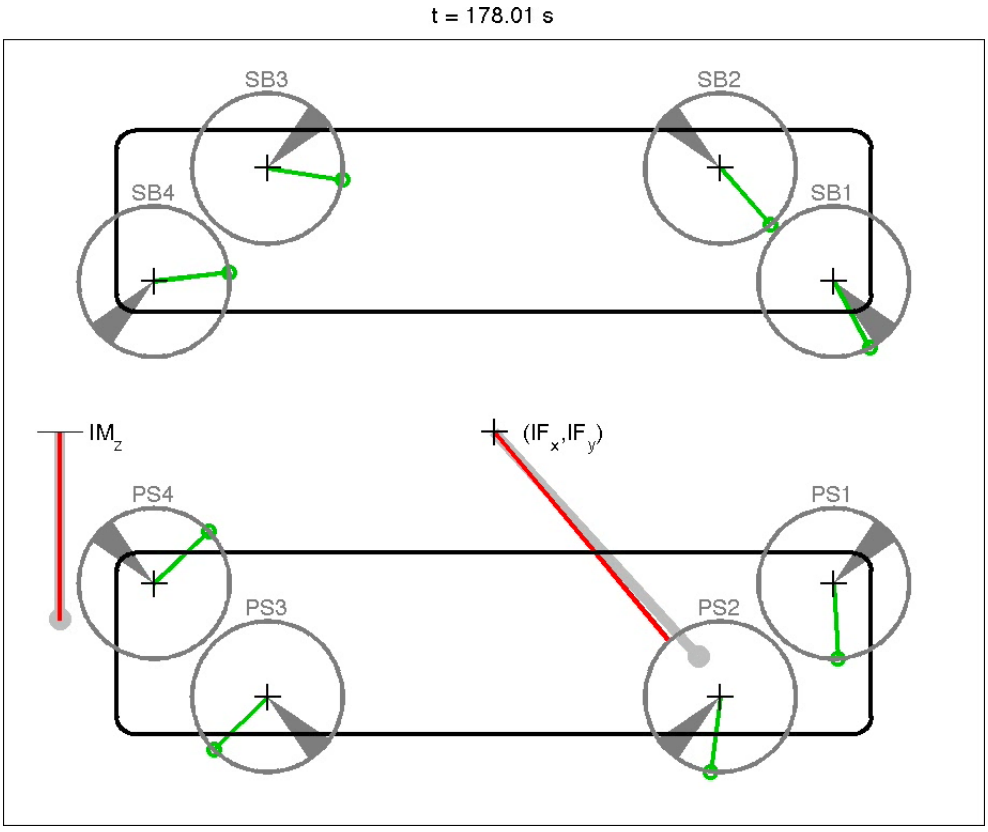
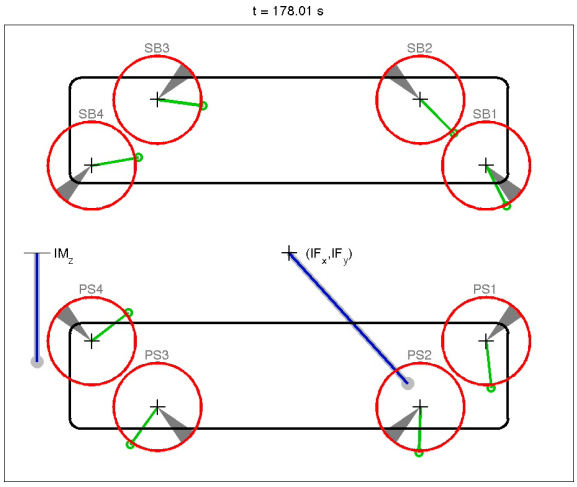


Figure 3.5: Identical setup as in Figure 3.4, but with results for the approach in which the penalty function formulation is used in case the thrusters are overloaded. This avoids overloading, as can be seen in the graphs, by loosening the constraint on the required force and momentum (see Section 3.3).

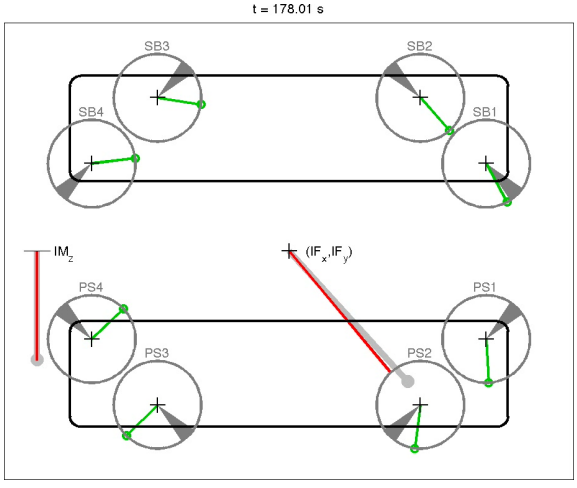
The use of the thrust/angle notation proved to reduce the complexity of the problem quite dramatically to only 3 equations with as many unknowns. Using this in the software could result in another significant speedup.

The iterative method with linearized Lagrange problem seems a promising algorithm. It should only require about 4 or 5 iterations to find a sufficiently accurate solution. However, during the workshop, we had no time to make an implementation and thus have no results at the time of writing.

It should be noted that optimizing only the allocation block in Figure 3.1 might not be ideal. A model-predictive approach, that combines the EKF, PID and allocation units might lead to even better results. Another important aspect of this approach would be the concept of time horizon: the power can be minimized over a given period, the next two hours for example. However, this would require a full model of the vessel, together with models for the wind and the waves. This is not that



(a) Still image from the visualization of the direct nonlinear optimization from Section 3.3. At this moment, the thrusters are overloaded as indicated by the red circles that show the maximal thrust and the thrust vectors that are longer than that radius.



(b) Still image form the visualization for the nonlinear optimization with special handling of the overloaded thruster situation as explained in Section 3.3. At this time, the thrusters would be overloaded, as shown in 3.6(a). So the alternative approach with penalty function is used: instead of overloading the thrusters, the required force and momentum constraint is loosened and matched as closely as possible given the available capacity.

Figure 3.6: MATLAB visualization of the resulting thrusters settings. The required (gray) and achieved (blue, red if not produced) total force ($\mathbb{F}_x, \mathbb{F}_y$) and moment $\mathbb{M}_z$ are shown, together with all the thruster forces of this semi-submersible. The constraints are also shown: filled sections show the forbidden angles and the circle's radius is the maximum thrust capacity. If constraints are violated, the corresponding section or circle is shown in red.

straightforward.

3.6 References

[1] K Böhmer, P Hemker, and H J Stetter. The defect correction approach. In *Defect Correction Methods. Theory and Applications*, pages 1–32. Springer-Verlag, 1984.

[2] Jorrit-Jan Serraris. Time domain analysis for dp simulations. *ASME Conference Proceedings*, 2009(43413):595–605, 2009.