Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-21 (2019) 97–102

Autonomous docking using direct optimal control

Andreas B. Martinsen * **Anastasios M. Lekkas** *
Sebastien Gros *

* *Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, NO-7491 Norway (e-mail: andreas.b.martinsen@ntnu.no, anastasios.lekkas@ntnu.no, sebastien.gros@ntnu.no)*

Abstract: We propose a method for performing autonomous docking of marine vessels using numerical optimal control. The task is framed as a dynamic positioning problem, with the addition of spatial constraints that ensure collision avoidance. The proposed method is an all-encompassing procedure for performing both docking, maneuvering, dynamic positioning and control allocation. In addition, we show that the method can be implemented as a real-time MPC-based algorithm on simulation results of a supply vessel.

Keywords: Docking, Optimal Control, Autonomous vehicles, Numerical Optimization, Path planning

1. INTRODUCTION

For most larger vessels, docking has historically been performed by utilizing external help from support vessels such as tug boats. The main reasons for this has been limits in terms of maneuverability as well as limits in the accuracy of the human operators when dealing with relatively slow dynamical systems. With the increasing usage of azimuth thrusters, marine vessels have become increasingly maneuverable. In addition to this, interest in autonomous ferries, and cargo vessels has increased in recent years. Despite this, and contrary to topics such as path following/tracking and control allocation, research on autonomous docking for surface vessels has seen little attention. While there are some methods such as Rae and Smith (1992); Teo et al. (2015); Hong et al. (2003) developed for Autonomous Underwater Vehicles (AUVs), which use fuzzy control schemes for different stages of the docking process. While Breivik and Loberg (2011) and Woo et al. (2016) have developed methods for Unmanned Surface Vehicles (USVs) based on target tracking and artificial potential fields respectively. These existing approaches are usually quite limited, do not take into account the underlying vessel model, and make few guarantees in terms of safety.

In this paper, we present a method for framing the problem of autonomous docking as a optimal control problem. Our proposed method is similar to methods used for dynamic positioning Veksler et al. (2016); Sotnikova and Veremey (2013), with the addition of control allocation optimization Johansen et al. (2004), and spatial constraint, which ensure the vessel operates safely without colliding.

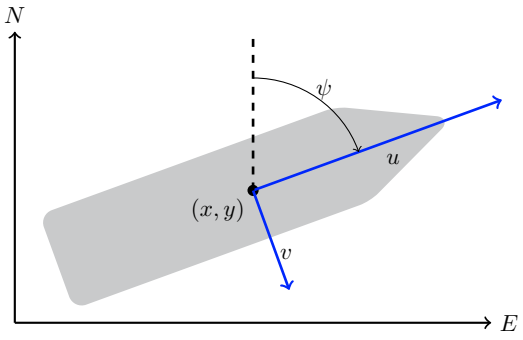

Fig. 1. 3-DOF vessel centered at (x, y), with surge velocity u, sway velocity v, heading ψ in a North-East-Down (NED) reference frame.

2. VESSEL MODEL

2.1 Kinematics

When modeling vessels for the purpose of autonomous docking, we assume the vessel moves on the ocean surface at relatively low velocities. In addition to this we assume that effects of the roll and pitch motions of the vessel are negligible, and hence have little impact on the surge, sway and yaw of the vessel. The mathematical model used to describe the system can then be kept reasonably simple by limiting it to the planar position and orientation of the vessel. The motion of a surface vessel can be represented by the pose vector $\boldsymbol{\eta} = [x, y, \psi]^\top \in \mathbb{R}^2 \times \mathbb{S}$, and velocity vector $\boldsymbol{\nu} = [u, v, r]^\top \in \mathbb{R}^3$. Here, (x, y) describe the Cartesian position in the earth-fixed reference frame, ψ is yaw angle, (u, v) is the body fixed linear velocities, and r is the yaw rate, an illustration is given in Figure 1. Using the notation in Fossen (2011) we can describe a 3-DOF vessel model as follows

2405-8963
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.12.290

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\psi)\boldsymbol{\nu}, \tag{1}$$
$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}, \tag{2}$$

where $\boldsymbol{M} \in \mathbb{R}^{3\times3}$, $\boldsymbol{D}(\boldsymbol{\nu}) \in \mathbb{R}^{3\times3}$, $\boldsymbol{\tau}$ and $\boldsymbol{J}(\psi) \in SO(3)$ are the inertia matrix, dampening matrix, control input vector, and rotation matrix respectively. The rotational matrix $\boldsymbol{J}(\psi) \in SO(3)$ is given by

$$\boldsymbol{J}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

and is the rotation from the body frame to the earth-fixed reference frame.

2.2 Thrust configuration

The control surfaces of the vessel are specified by the thrust configuration matrix $\boldsymbol{T}(\boldsymbol{\alpha}) \in \mathbb{R}^{3,n_{thrusters}}$ which maps the thrust $\boldsymbol{f}$ from each thruster into the surge, sway and yaw forces and moments in the body frame of the vessel given the thruster angles $\boldsymbol{\alpha}$.

$$\boldsymbol{\tau} = \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{f} \tag{4}$$

Each column $\boldsymbol{T}_i(\alpha_i)$ in $\boldsymbol{T}(\boldsymbol{\alpha})$ gives the configuration of the forces and moments of a thruster i as follows:

$$\boldsymbol{T_i}(\boldsymbol{\alpha})f_i = \begin{bmatrix} F_x \\ F_y \\ F_y l_x - F_x l_y \end{bmatrix} = \begin{bmatrix} f_i\cos(\alpha_i) \\ f_i\sin(\alpha_i) \\ f_i(l_x\sin(\alpha_i) - l_y\cos(\alpha_i)) \end{bmatrix} \tag{5}$$

where α_i is the orientation of the thruster in the body frame, and f_i is the force it produces. Selecting the orientation $\boldsymbol{\alpha}$ and force $\boldsymbol{f}$ of the thrusters in order to generate the desired force $\boldsymbol{\tau}$ is called the thrust allocation problem. While there are numerous ways of solving the thrust allocation problem Johansen and Fossen (2013), for our purpose we want to include the thrust allocation as part of the optimization for performing the docking operations. This allows us to take into account physical thruster constraints such as force saturation and feasible azimuth sectors.

$$\alpha_{i,min} \leq \alpha_i \leq \alpha_{i,max}$$
$$f_{i,min} \leq f_i \leq f_{i,max}$$

In order to avoid singular thruster configurations, we add a penalty on the rank deficiency of the thrust configuration matrix, as proposed by Johansen et al. (2004). The singular configuration cost is given as the following.

$$\frac{\rho}{\epsilon + \det\left(\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{W}^{-1}\boldsymbol{T}^{\top}(\boldsymbol{\alpha})\right)} \tag{6}$$

Here $\epsilon > 0$ is a small constant in order to avoid division by 0, $\rho > 0$ is the weighting of the maneuverability, and $\boldsymbol{W}$ is typically diagonal matrix, weighting each individual thruster. A constraint on the singular configuration may alternatively be added, however in our implementation this is added as a cost, which means that avoiding singular thrust configurations become more important when close to the desired docking position.

It should be noted that both the singular configuration cost in (6) and the thrust configuration matrix in (4) are both highly nonlinear due to the trigonometric functions, adding them as costs and constraints in an optimization problem will therefor in general cause the problem to become non-convex.

Fig. 2. Thruster configuration for vessel, where 1 and 2 are azimuth thrusters, and 3 is a tunnel thruster.

2.3 Summary of model

The model used for the simulations is based on the SV Northern Clipper from Fossen et al. (1996). The model is a 3 Degree of Freedom (3-DOF) linear model on the form:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\psi)\boldsymbol{\nu},$$
$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}\boldsymbol{\nu} = \boldsymbol{T}(\alpha)\boldsymbol{f}$$

For thruster configuration, we used two azimuth thrusters in the stern and one tunnel thruster in the bow, giving configuration seen in Figure 2. Additionally saturations were added to the force generatetd by the thrusters, where the maximum thrust for the azimuth thrusters and tunnel thrusers respectively were 1/30 and ±1/60 of the dry ships weight. For the azimuth thrusters additional constraints were added, this included a maximum turnaround time of $30s$ per revolution, and a maximum angle of ±170° giving a 20° forbidden sector illustrated in figure 2, which ensures the thrusters do not produce thrust that directly work against eachother, which may cause damage, this additionally reflects the movement of real world azimuth thrusters which have a finite turning radius. Additional details on the vessel model, and specific parameters are given in Appendix A.

3. AUTONOMOUS DOCKING

3.1 Obstacle avoidance

Docking of autonomous vessels is a complex problem, which includes planning and performing maneuvers to control a vessel to a desired orientation and position, while adhering to spatial constraint in order to avoid collisions. Given a desired position x_d, y_d and a desired heading ψ_d, we define the docking problem as maneuvering a vessel as close to the desired pose as possible, with out the vessel going aground, or running into obstacles, i.e. adhering to spatial constraints.

In order to ensure the vessel does not collide we define a safety margin around the vessel which obstacles should not enter. Given a set $\mathbb{S}_v$ representing the vessel, the convex hull

$$\text{Conv}(\mathbb{S}_v)$$

gives the boundary points of the vessel which form a polyhedron around the vessel. By dilating the set representing the vessel by a desired safety margin $\mathbb{M}$, we get the following polyhedron representing the safety boundary surrounding the vessel.

$$\mathbb{S}_b = \text{Conv}(\mathbb{S}_v \oplus \mathbb{M}) \tag{7}$$

For our simulations we used a safety margin of 10% giving the safety boundary $\mathbb{S}_b$ seen in Figure 3, which is a polyhedron in the body frame of the vessel consisting of five vertices.

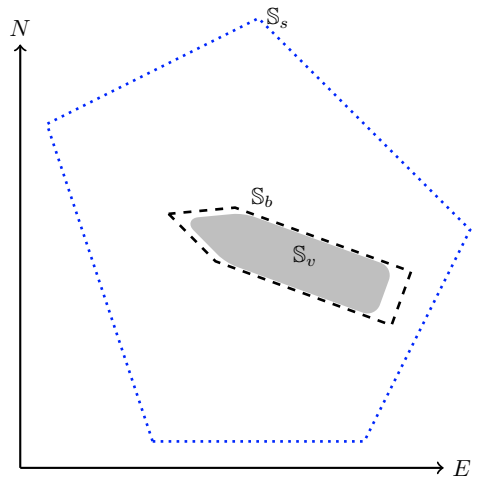

Fig. 3. Vessel $\mathbb{S}_v$ with safety boundary $\mathbb{S}_b$ with black dashed line, and spatial constraints $\mathbb{S}_s$ as blue dotted line, in the NED frame. The vessel will always lie within the spatial constraints $\mathbb{S}_s$ as long as all the vertices of $\mathbb{S}_b$ lie within the spatial constraints.

In order to ensure safe operating conditions, we define a operating region in terms of spatial constraints $\mathbb{S}_s$ for the vessel. The operating region is chosen as the largest convex region that encompasses the desired docking position, while not intersecting with obstacles or land. Choosing the spatial constraints and vessel boundary in this way, safe operations are ensured when $\mathbb{S}_b \subseteq \mathbb{S}_s$, i.e. the vessel with the safety margin is contained within the spatial constraints, this is illustrated in Figure 3. Using the fact that the spatial constraints are a convex polyhedron:

$$\mathbb{S}_s = \{x | \boldsymbol{A}_s x \leq \boldsymbol{b}_s\}$$

we have that the vessel is within the spatial constraints so long as all the vertices of the vessel boundary follow the linear inequality representing the spatial constraints.

$$\mathbb{S}_b \subseteq \mathbb{S}_s \iff \boldsymbol{A}_s \boldsymbol{x}_i^{NED} \leq \boldsymbol{b}_s \quad \forall \boldsymbol{x}_i^{NED} \in \text{Vertex}(\mathbb{S}_b) \tag{8}$$

Since the Vertexes of the vessel boundary are given in the body frame of the vessel we need to transform them from the body frame to the NED frame, giving the following nonlinear constraints.

$$\boldsymbol{A}_s \left(\boldsymbol{R}(\psi) \boldsymbol{x}_i^b + \begin{bmatrix} x \\ y \end{bmatrix} \right) \leq \boldsymbol{b}_s \quad \forall \boldsymbol{x}_i^b \in \text{Vertex}(\mathbb{S}_b) \tag{9}$$

Where $\boldsymbol{R}$ is the rotation from the body frame to NED.

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \tag{10}$$

This can directly be implemented as inequality constraints in an optimization problem, and ensures the vessel is contained within a predefined safe region.

While this constraint is easily implemented in a nonlinear programming (NLP) problem, the constraint is not convex. This means the constraint will enforce safety requirements, however the NLP may not converge to a global optimum.

3.2 Optimal control problem (OCP)

Using the model, and constraints discussed in the previous sections, with the desired docking pose $\boldsymbol{\eta}_d = [x_d, y_d, \psi_d]^\top$, we can formulate the following nonlinear continuous time optimal control problem.

$$J^* = \min_{\boldsymbol{\eta}, \boldsymbol{\nu}, \boldsymbol{f}, \boldsymbol{\alpha}} \int_0^T \left\{ ||\boldsymbol{\eta} - \boldsymbol{\eta}_d||^2_{\boldsymbol{Q}_\eta} + ||\boldsymbol{\nu}||^2_{\boldsymbol{Q}_\nu} + ||\boldsymbol{f}||^2_{\boldsymbol{R}_f} + \frac{\rho}{\epsilon + \det\left(\boldsymbol{T}(\boldsymbol{\alpha}) \boldsymbol{W}^{-1} \boldsymbol{T}^\top(\boldsymbol{\alpha})\right)} \right\} dt \tag{11a}$$

subject to:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\psi)\boldsymbol{\nu} \tag{11b}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}\boldsymbol{\nu} = \boldsymbol{T}(\alpha)\boldsymbol{f} \tag{11c}$$

$$\boldsymbol{A}_s \left(\boldsymbol{R}(\psi) \boldsymbol{x}_i^b + \begin{bmatrix} x \\ y \end{bmatrix} \right) \leq \boldsymbol{b}_s \; \forall \boldsymbol{x}_i^b \in \text{Vertex}(\mathbb{S}_b) \tag{11d}$$

$$\boldsymbol{f}_{min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{max} \tag{11e}$$

$$\boldsymbol{\alpha}_{min} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{max} \tag{11f}$$

$$|\dot{\boldsymbol{\alpha}}| \leq \dot{\boldsymbol{\alpha}}_{max} \tag{11g}$$

$$\text{Initial conditions on } \boldsymbol{\eta}, \boldsymbol{\nu}, \boldsymbol{f}, \boldsymbol{\alpha} \tag{11h}$$

Where we minimize cost (11a), subject to the dynamic model constraints (11b) and (11c), the spatial constraints (11d), the saturation constraint (11e), (11f) and (11g), and the initial conditions (11h) over the time horizon T. For this problem we have opted to use a simple quadratic penalty in order to ensure the vessel converges to the desired pose, however Huber penalty functions as discussed in Gros and Diehl (2013); Gros and Zanon (2017) may give better performance for large pose deviations.

3.3 Implementation

In order to implement the proposed docking system we need to solve the OCP in the previous section. This can be done in multiple ways, however the two main classes of methods are sequential methods, such as direct single shooting Hicks and Ray (1971), and simultaneous methods such as direct multiple shooting Deuflhard (1974), and direct collocation Tsang et al. (1975). For this approach we chose to use direct collocation, in where implicit numerical integration of the ODE constraints (11b) and (11c), as well as the objective function (11a), is performed as part of the nonlinear optimization. In the collocation method, the numerical integration is performed by fitting the derivatives to a degree d Legendre polynomial, with known integral, within N set time intervals called shooting intervals. The shooting intervals are then connected to create the full time horizon, by enforcing constraints on the shooting gaps between intervals.

For this problem we opted to use direct collocation for several reasons. Comparing direct collocation with multiple shooting, they both offer the same stability in terms of the optimization, however direct collocation offers a speedup, as the numerical integration is performed as part of the optimization, and not offloaded to a separate integration routine, giving the optimization problem a nice sparsity structure. While multiple shooting offers more flexibility in terms of the integrator used, the implicit integrator of the direct collocation is sufficient for our purpose. Comparing single shooting to direct collocation the single shooting problem has much fewer decision variables, however the problem often becomes very dense, and hence increases the

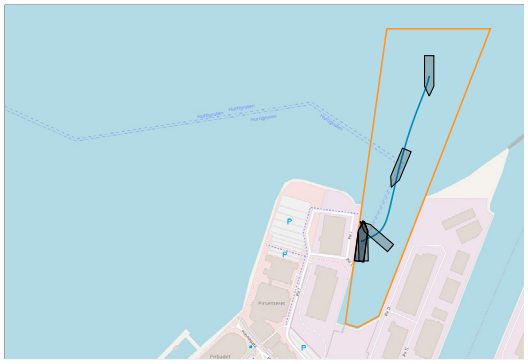

Fig. 4. Vessel docking performed at Hurtigruten terminal in Trondheim Norway.

computation time, single shooting is also more unstable, as propagating the gradients through a long time horizon often cause them to become very small (vanish) or very large (explode), and hence the optimization steps may be oscillatory and unstable.

For the implementation we used CasADi Andersson et al. (In Press, 2018) a software framework for easy implementation of nonlinear optimization and optimal control problems, with IPOPT Wächter and Biegler (2006) an interior point optimizer, for solving the resulting NLP.

Solving the OCP once, gives a open loop trajectory over a time horizon T, which can be used to perform open loop control, or trajectory tracking. We however wish to use the OCP as the basis for a Nonlinear Model Predictive Control (NLMPC). Where at each time step the OCP is solved with the vessel state as initial conditions, and then only the first predicted control action is performed. This gives a closed loop control scheme, which makes the method more robust to modeling errors, and external disturbances due to the feedback.

4. SIMULATION

As a proof of concept, simulations were performed, where the OCP was run as a closed loop Nonlinear Model Predictive Control (NLMPC). For the OCP we used a time horizon of $T = 300$ seconds, with $N = 30$ time steps, making each time step $T/N = 10$ seconds. Using this we performed docking simulations at two different locations, namely Trondheim harbour and Lundevågen harbour, as seen in Figure 4 and 5 respectively. For the docking at Lundevågen harbour, the vessel state and control inputs are shown in Figure 6, 7 and 8, and for the docking at Trondheim harbour, the vessel state and control inputs are shown in Figure 9, 10 and 11. From the simulations we see an expected behaviour, where the vessel will turn and face the bow in the direction of travel, as this is the most efficient way of traveling. As the vessel closes in on the target position, it will start initiating the turn such that it faces in the desired heading, while simultaneously adhering to the defined spatial constraints in order to avoid colliding.

5. CONCLUSION

Based on the results of the simulation, the proposed method works very well, with the vessel approaching the

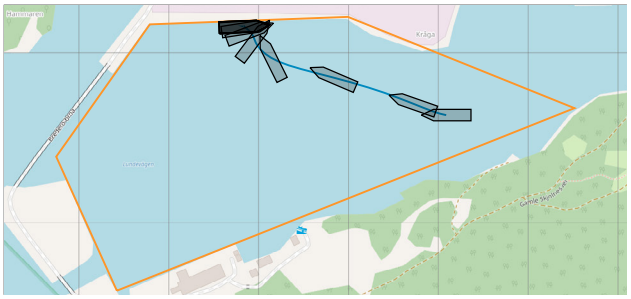

Fig. 5. Vessel docking performed at Lundevågen harbour in Farsund Norway.

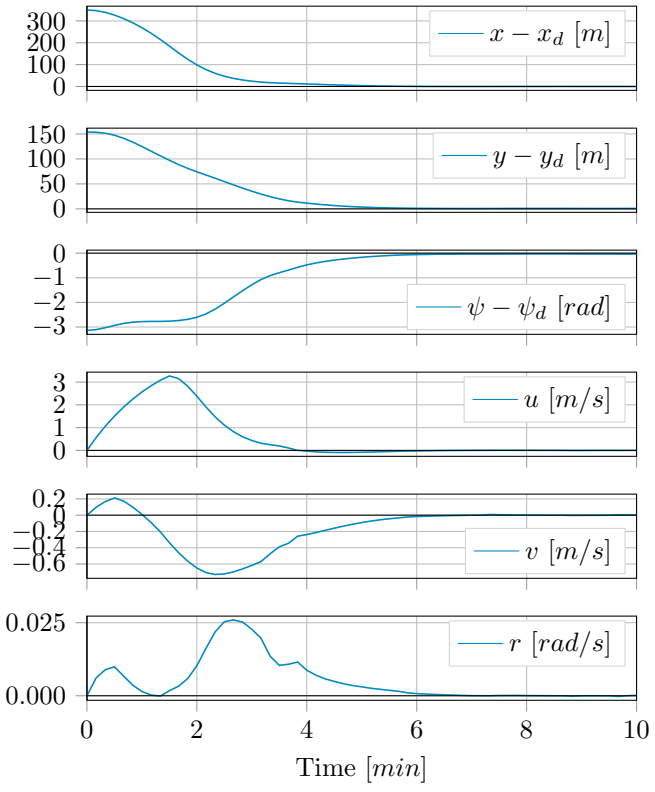

Fig. 6. Vessel pose error $\boldsymbol{\eta} - \boldsymbol{\eta}_d$ and velocity $\boldsymbol{\nu}$ when docking at Lundevågen harbour.

target poses without violating the spatial constraints. Solving the open loop optimization problem with zeros as a trivial initial guess takes $2 - 4$ seconds, while solving the problem using a warm start, a solution is found in about 0.5 seconds. With a purpose build solver this should take even less time, and ensures real time feasibility, as demonstrated by Vukov et al. (2015). NLP solver for the problem should be chosen carefully. We fond that IPOPT worked the best, as it was able to consistently solve the problem from a number of tested initial points, within a reasonable amount of time. With other solvers outright failing, or using excessive amounts of time.

The method does however have some drawbacks, since the proposed problem is non-convex due to the rotation of the azimuth thrusters and the vessel rotation, this means convergence to a global optimum can not be guaranteed. The method will however converge to a locally optimal solution, which in practise may be good enough, and will

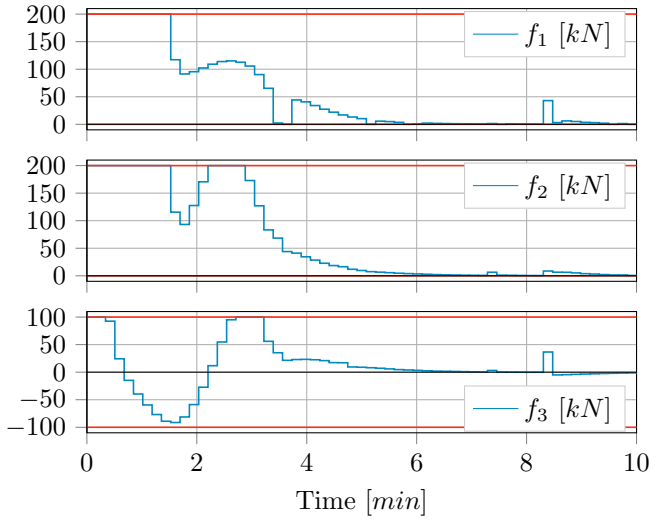

Fig. 7. Thruster force for docking at Lundevågen harbour, with saturation constraints indicated in red.

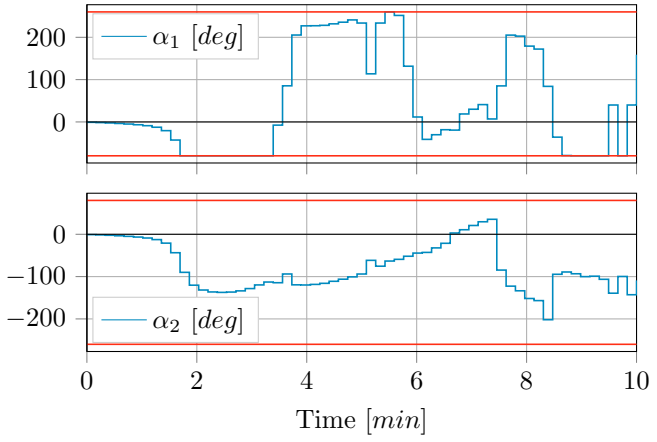

Fig. 8. Azimuth angles when docking at Lundevågen harbour, with saturation constraints indicated in red.

most importantly ensure safe operations. It is also worth noting that the problem is a finite horizon optimization problem, meaning we are only optimizing over a horizon T. This means that maneuvers that are optimal over a time horizon longer than T, may no longer be optimal over T, meaning the horizon must also be carefully chosen to get the desired behaviour.

The proposed method seems very promising, however many improvements can be made. Future research can be done on using more complex nonlinear vessel models, which may include thrust and azimuth dynamics. Different objective functions may be implemented, such as minimizing time, until the vessel reaches a terminal set, or energy expended. The method may also be further generalized by having dynamic spatial constraints, that use the largest convex set that does not intersect obstacles centered about the vessel as constraints. This may make the method not only suitable for docking, but also for general obstacle avoidance while in transit. While the proposed docking method has some measures ensuring robustness and safety while performing docking, future research can be done into

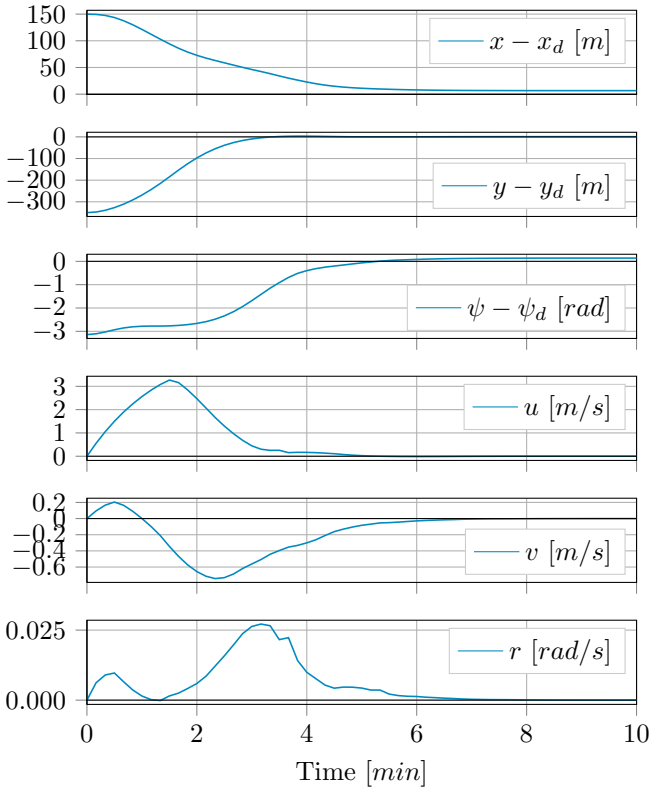

Fig. 9. Vessel pose error $\boldsymbol{\eta} - \boldsymbol{\eta}_d$ and velocity $\boldsymbol{\nu}$ when docking at Trondheim harbour.

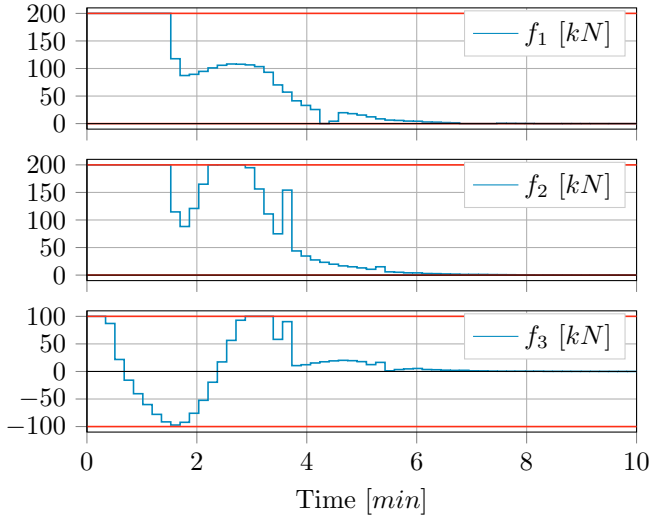

Fig. 10. Thruster force for docking at Trondheim harbour, with saturation constraints indicated in red.

making the method able to handle external environmental forces such as wind waves and currents using for example a scenario based MPC.

REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (In Press, 2018). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*.

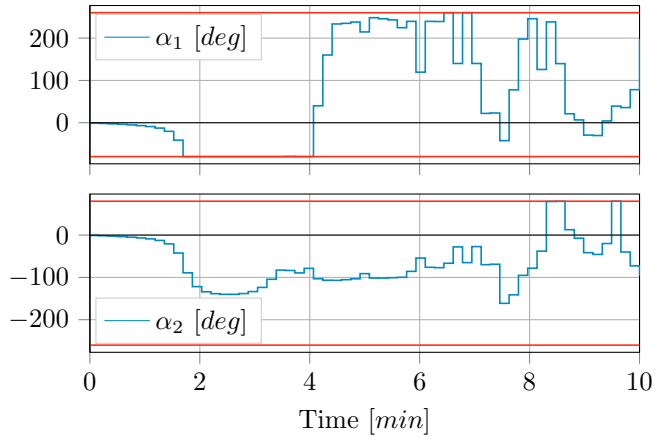

Fig. 11. Azimuth angles when docking at Trondheim harbour, with saturation constraints indicated in red.

Breivik, M. and Loberg, J.E. (2011). A virtual target-based underway docking procedure for unmanned surface vehicles. *IFAC Proceedings Volumes*, 44(1), 13630–13635.

Deuflhard, P. (1974). A modified newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting. *Numerische Mathematik*, 22(4), 289–315.

Fossen, T.I. and Perez, T. (2004). Marine systems simulator (mss). URL `http://www.marinecontrol.org`.

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

Fossen, T.I., Sagatun, S.I., and Sørensen, A.J. (1996). Identification of dynamically positioned ships.

Gros, S. and Diehl, M. (2013). Nmpc based on huber penalty functions to handle large deviations of quadrature states. In *2013 American Control Conference*, 3159–3164. doi:10.1109/ACC.2013.6580317.

Gros, S. and Zanon, M. (2017). Penalty functions for handling large deviation of quadrature states in nmpc. *IEEE Transactions on Automatic Control*, 62(8), 3848–3860. doi:10.1109/TAC.2017.2649043.

Hicks, G. and Ray, W. (1971). Approximation methods for optimal control synthesis. *The Canadian Journal of Chemical Engineering*, 49(4), 522–528.

Hong, Y.H., Kim, J.Y., Oh, J.h., Lee, P.M., Jeon, B.H., Oh, K.H., et al. (2003). Development of the homing and docking algorithm for auv. In *The Thirteenth International Offshore and Polar Engineering Conference*. International Society of Offshore and Polar Engineers.

Johansen, T.A. and Fossen, T.I. (2013). Control allocation—a survey. *Automatica*, 49(5), 1087–1103.

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

Rae, G. and Smith, S. (1992). A fuzzy rule based docking procedure for autonomous underwater vehicles. In *OCEANS 92 Proceedings@ m_Mastering the Oceans Through Technology*, volume 2, 539–546. IEEE.

Sotnikova, M.V. and Veremey, E.I. (2013). Dynamic positioning based on nonlinear mpc. *IFAC Proceedings Volumes*, 46(33), 37–42.

Teo, K., Goh, B., and Chai, O.K. (2015). Fuzzy docking guidance using augmented navigation system on an auv. *IEEE journal of oceanic engineering*, 40(2), 349–361.

Tsang, T., Himmelblau, D., and Edgar, T. (1975). Optimal control via collocation and non-linear programming. *International Journal of Control*, 21(5), 763–768.

Veksler, A., Johansen, T.A., Borrelli, F., and Realfsen, B. (2016). Dynamic positioning with model predictive control. *IEEE Transactions on Control Systems Technology*, 24(4), 1340–1353.

Vukov, M., Gros, S., Horn, G., Frison, G., Geebelen, K., Jørgensen, J., Swevers, J., and Diehl, M. (2015). Real-time nonlinear mpc and mhe for a large-scale mechatronic application. *Control Engineering Practice*, 45, 64 – 78. doi:https://doi.org/10.1016/j.conengprac.2015.08.012. URL `http://www.sciencedirect.com/science/article/pii/S0967066115300095`.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25–57.

Woo, J., Kim, N., et al. (2016). Vector field based guidance method for docking of an unmanned surface vehicle. In *The Twelfth ISOPE Pacific/Asia Offshore Mechanics Symposium*. International Society of Offshore and Polar Engineers.

Appendix A. VESSEL MODEL

The vessel model used in the simulations was based on the SV Northern Clipper Fossen et al. (1996), where the model parameters were taken form the Marine System Simulator (MSS) Toolbox Fossen and Perez (2004). The model used has the following vessel dynamics

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\psi)\boldsymbol{\nu},$$
$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}\boldsymbol{\nu} = \boldsymbol{T}(\alpha)\boldsymbol{f}$$

With the diagonal normalization matrix $\boldsymbol{N} = \text{diag}([1, 1, \text{L}])$, and the non-dimensional (bis-system) given by $\boldsymbol{M}_{bis}$ and $\boldsymbol{D}_{bis}$, the mass and dampening matrix are given by the following.

$$\boldsymbol{M} = m\boldsymbol{N}\boldsymbol{M}_{bis}\boldsymbol{N}, \quad \boldsymbol{D} = m\sqrt{\frac{g}{L}}\boldsymbol{N}\boldsymbol{D}_{bis}\boldsymbol{N}$$

$$\boldsymbol{M}_{bis} = \begin{bmatrix} 1.1274 & 0 & 0 \\ 0 & 1.8902 & -0.0744 \\ 0 & -0.0744 & 0.1278 \end{bmatrix}, \quad \boldsymbol{D}_{bis} = \begin{bmatrix} 0.0358 & 0 & 0 \\ 0 & 0.1183 & -0.0124 \\ 0 & -0.0041 & 0.0308 \end{bmatrix}$$

Where the normalization parameters of length gravity and mass are given as $L = 76.2(m)$, $g = 9.8(m/s^2)$ and $m = 6000e3(kg)$ respectively.

For the vessel, we assume two azimuth thrusters in the aft, with one tunnel thruster in the front giving the thruster position and angle given in Table A.1, and the thrust configuration matrix $\boldsymbol{T}(\boldsymbol{\alpha})$ is as follows.

$$\begin{bmatrix} \cos(\alpha_1) & \cos(\alpha_2) & 0 \\ \sin(\alpha_1) & \sin(\alpha_2) & 1 \\ l_{x_1}\sin(\alpha_1) - l_{y_1}\cos(\alpha_1) & l_{x_2}\sin(\alpha_2) - l_{y_2}\cos(\alpha_2) & l_{x_3} \end{bmatrix}$$

Table A.1. Thruster position and angle

Truster	x-position	y-position	angle
Azimuth 1	$l_{x_1} = -35m$	$l_{y_1} = 7m$	α_1
Azimuth 2	$l_{x_2} = -35m$	$l_{y_2} = -7m$	α_2
Tunnel 3	$l_{x_3} = 35m$	$l_{y_3} = 0m$	$\alpha_3 = \frac{\pi}{2}$