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.

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 \( \eta = [x, y, \psi]^\top \in \mathbb{R}^2 \times S^1 \), and velocity vector \( \nu = [u, v, r]^\top \in \mathbb{R}^3 \). Here, \((x, y)\) describe the Cartesian position in the earth-fixed reference frame, \(\psi\) 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.
\[
\hat{\eta} = J(\psi)\nu, \\
M\hat{\nu} + D(\nu)\nu = \tau,
\]
where \(M \in \mathbb{R}^{3 \times 3}\), \(D(\nu) \in \mathbb{R}^{3 \times 3}\), \(\tau\) and \(J(\psi) \in \text{SO}(3)\) are the inertia matrix, dampening matrix, control input vector, and rotation matrix respectively. The rotational matrix \(J(\psi) \in \text{SO}(3)\) is given by
\[
J(\psi) = \begin{bmatrix}
\cos(\psi) & -\sin(\psi) & 0 \\
\sin(\psi) & \cos(\psi) & 0 \\
0 & 0 & 1
\end{bmatrix}
\]
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 \(T(\alpha) \in \mathbb{R}^{3,n_{\text{thrusters}}}\) which maps the thrust \(f\) from each thruster into the surge, sway and yaw forces and moments in the body frame of the vessel given the thruster angles \(\alpha\).

\[
\tau = T(\alpha)f
\]
Each column \(T_i(\alpha_i)\) in \(T(\alpha)\) gives the configuration of the forces and moments of a thruster \(i\) as follows:
\[
T_i(\alpha_i) = \begin{bmatrix}
F_x \\
F_y \\
F_z \sin(\alpha_i) - l_y \cos(\alpha_i)
\end{bmatrix} = \begin{bmatrix}
f_i \cos(\alpha_i) \\
f_i \sin(\alpha_i) \\
0
\end{bmatrix}
\]
where \(\alpha_i\) is the orientation of the thruster in the body frame, and \(f_i\) is the force it produces. Selecting the orientation \(\alpha\) and force \(f\) of the thrusters in order to generate the desired force \(\tau\) 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.

\[
\alpha_{i,\text{min}} \leq \alpha_i \leq \alpha_{i,\text{max}} \\
f_{i,\text{min}} \leq f_i \leq f_{i,\text{max}}
\]

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.
\[
\epsilon + \det (T(\alpha)W^{-1}T^\top(\alpha))
\]
where \(\epsilon > 0\) is a small constant in order to avoid division by 0, \(\rho > 0\) is the weighting of the maneuverability, and \(W\) 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.
In order to ensure safe operating conditions, we define a
operating region in terms of spatial constraints. To ensure safe operations are ensured when
the spatial constraints are a convex polyhedron:
\[ S_a = \{ x | A_a x \leq b_i \} \]

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.
\[ S_b \subseteq S_a \iff A_a x_i^{NED} \leq b_i \quad \forall x_i^{NED} \in \text{Vertex}(S_b) \] (8)

Since the Vertices 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.
\[ A_a \left( R(\psi) x_i^b + \begin{bmatrix} x \\ y \end{bmatrix} \right) \leq b_i \quad \forall x_i^b \in \text{Vertex}(S_b) \] (9)

Where \( R \) is the rotation from the body frame to NED.
\[ R(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \] (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 \( \eta_d = [x_d, y_d, \psi_d]^T \), we can formulate the following nonlinear continuous time optimal control problem.

\[
J^* = \min_{\eta, \nu, f, \alpha} \int_0^T \left\{ \| \eta - \eta_d \|_Q^2 + \| \nu \|_Q^2 + \| f \|_R^2 + \frac{\rho}{\epsilon + \det(T(\alpha)W^{-1}T^T(\alpha))} \right\} dt \quad (11a)
\]

subject to:
\[
\dot{\eta} = J(\psi) \nu \quad (11b)
\]
\[
M \dot{\nu} + D \nu = T(\alpha) f \quad (11c)
\]
\[
\begin{bmatrix} A_a & \left( R(\psi) x_i^b + \begin{bmatrix} x \\ y \end{bmatrix} \right) \end{bmatrix} \leq b_i \quad \forall x_i^b \in \text{Vertex}(S_b) \quad (11d)
\]
\[
f_{\min} \leq f \leq f_{\max} \quad (11e)
\]
\[
\alpha_{\min} \leq \alpha \leq \alpha_{\max} \quad (11f)
\]
\[
| \dot{\alpha} | \leq \alpha_{\max} \quad (11g)
\]

Initial conditions on \( \eta, \nu, f, \alpha \) (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 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 multipoint 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 an 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 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 found 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 practice may be good enough, and will
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 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 underwater docking procedure for unmanned surface vehicles. IFAC Proceedings Volumes, 44(1), 13630–13635.

Deuffhard, 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 simulation. John Wiley & Sons.

Johansen, T.A. and Fossen, T.I. (2013). Control allocation with singular value avoidance using sequential quadratic programming. IEEE Transactions on Control Systems Technology, 21(4), 1340–1353.

Vukov, M., Gros, S., Horn, G., Frison, G., Geeelen, K., Jorgensen, 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.

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 from the Marine System Simulator (MSS) Toolbox Fossen and Perez (2004). The model used has the following vessel dynamics

\[ \dot{\psi} = J(\psi) \nu, \]

\[ MV + Du = T(\alpha) f \]

With the diagonal normalization matrix \( N = \text{diag}([1, 1, 1]) \) and the non-diagonal (bis-system) given by \( M_{\text{bis}} \) and \( D_{\text{bis}} \), the mass and dampening matrix are given by the following.

\[ M = mNN_{\text{bis}}N, \quad D = m \sqrt{\frac{2}{L}} ND_{\text{bis}}N \]

\[ M_{\text{bis}} = \begin{bmatrix}
1.274 & 0 & 0 \\
0 & 1.8902 & -0.0744 \\
0 & -0.0744 & 0.1278
\end{bmatrix}, \quad D_{\text{bis}} = \begin{bmatrix}
0.0558 & 0 & 0 \\
0 & 0.1183 & -0.0124 \\
0 & -0.0041 & 0.0308
\end{bmatrix} \]

Where the normalization parameters of length gravity and mass are given as \( L = 76.2(m) \), \( g = 9.8(m/s^2) \) and \( m = 6000c(3)(kg) \) 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 \( T(\alpha) \) is as follows.

\[ \begin{bmatrix}
\cos(\alpha_1) & \cos(\alpha_2) & 0 \\
\sin(\alpha_1) & \sin(\alpha_2) & 1
\end{bmatrix} \begin{bmatrix}
l_{x1} & l_{y1} & l_{z1} \\
l_{x2} & l_{y2} & l_{z2} \\
l_{x3} & l_{y3} & l_{z3}
\end{bmatrix} \]

| Thruster | x-position | y-position | angle |
|----------|------------|------------|-------|
| Azimuth 1 | \( l_{x1} = -35m \) | \( l_{y1} = 7m \) | \( \alpha_1 \) |
| Azimuth 2 | \( l_{x2} = -35m \) | \( l_{y2} = -7m \) | \( \alpha_2 \) |
| Tunnel 3 | \( l_{x3} = 35m \) | \( l_{y3} = 0m \) | \( \alpha_3 = \frac{\pi}{2} \) |