

Comparison of Advanced Control Strategies for Automated Docking

Stefan Wirtensohn* Oliver Hamburger* Hannes Homburger*
 Leticia Mayumi Kinjo** Johannes Reuter*

* *University of Applied Sciences Konstanz, Konstanz, 78462 Germany*
 (e-mail: stwirten@htwg-konstanz.de, ohamburger@htwg-konstanz.de, h.homburger@htwg-konstanz.de, jreuter@htwg-konstanz.de).

** *LAC EA7478, ENSICAEN, 6 Bd du Marechal Juin, 14050 Caen*
Cedex, France (e-mail: leticia.kinjo@ensicaen.fr).

Abstract: In this paper, a systematic comparison of three different advanced control strategies for automated docking of a vessel is presented. The controllers are automatically tuned off-line by applying an optimization process using simulations of the whole system including trajectory planner and state and disturbance observer. Then investigations are conducted subject to performance and robustness using Monte Carlos simulation with varying model parameters and disturbances. The control strategies have also been tested in full scale experiments using the solar research vessel Solgenia. The investigated control strategies all have demonstrated very good performance in both, simulation and real world experiments. Videos are available under <https://www.htwg-konstanz.de/forschung-und-transfer/institute-und-labore/isd/regelungstechnik/videos/>

Copyright © 2021 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Automated Docking of Vessels, Nonlinear Control Design, Monte Carlo Simulations, Docking Experiments

1. INTRODUCTION

The request for automation of critical maneuvers is continuously growing not at least in the realm of maritime systems. Benefits of doing so are related to increased safety, time- and energy efficiency and to a great deal in reduction of operating costs. Accordingly, the amount of related research to assist maritime system operators has steadily increased in recent decades. In particular, the automation of ship operations, e.g. for collision avoidance, station keeping and ship berthing has gained significant attention. In this paper, the focus is put solely on the critical docking maneuvers. A variety of control and estimation methods have been proposed also in recent years, with the majority being tested in simulations, only, e.g. Martinsen et al. (2019) Li et al. (2020) Skulstad et al. (2021). Experimental full scale results have been reported e.g. in Mizuno et al. (2015) and Bitar et al. (2020). In the literature there is a lack of investigations where different algorithms are compared systematically under similar circumstances in order to provide guidance towards a selection of proper algorithms for certain tasks. An exception can be found in Mizuno et al. (2012). The main contribution of this paper is the provision of a comparative study wherein three state of the art control methods for ship docking maneuvers are investigated systematically. In order to obtain meaningful results, all control parameters have been automatically optimized by using the same simulative test scenarios and then have been applied on a vessel without further adjustments. The trials have been done on the fully actuated research vessel Solgenia. Since model based control strategies have been employed, the availability of a dynamic model with parameters describing the systems

behaviour sufficiently accurate is important. Likewise, the accuracy of the state estimates has a major impact of the overall performance and cannot being neglected. Therefore, both topics are addressed. The paper is organized as follows. We start with a brief summary how the modelling of the vessel has been performed, followed by a description of how the motion estimation is performed in the presence of various types of disturbances. In section four, we describe the control algorithms and the systematic tuning of the respective parameters. Section five is devoted to the trajectory planning for forward and backward docking maneuvers. The simulation and experimental results are presented in section six and seven. The paper concludes with a summary and directions of future research.

2. MODELLING

A detailed description of the modeling and identification process of our experimental watercraft Solgenia in terms of the equation of motion, actuator forces, and actuator dynamics can be found in Kinjo et al. (2021). It should be noted, however, that the most important aspects are briefly introduced in this section as they will appear in the description of the trajectory controllers and motion estimation algorithm.

As can be seen in Fig. 1 the experimental watercraft Solgenia is equipped with a 360° rotatable azimuth thruster in the rear and a bow thruster. To describe the planar dynamics of the watercraft the widely used model

$$M\dot{\nu} + C_{RB}(\nu)\nu + D\nu = \tau_c + \tau_d \quad (1)$$

$$\dot{\eta} = J(\psi)\nu \quad (2)$$

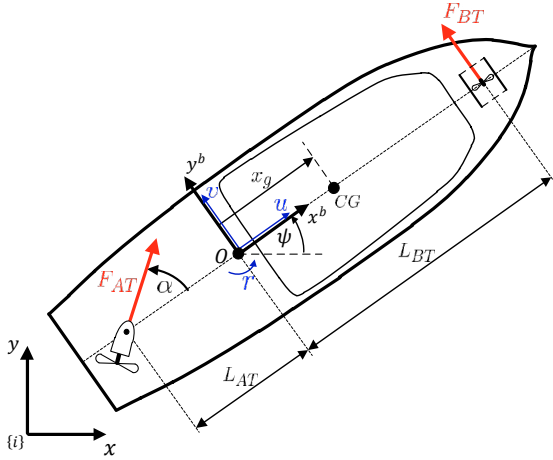


Fig. 1. Earth-fixed and body-fixed frame for the description of the planar ship motion

according to Fossen (2011) is used. Here, $\boldsymbol{\nu} = (u \ v \ r)^T$ combines the surge, sway and yaw velocity components. The influence of the roll, pitch and heave motion on the planar dynamics will be neglected in the following, since it is difficult to determine on the one hand and on the other hand it is assumed that the controller can compensate for these effects. Mass matrix \mathbf{M} , Coriolis matrix \mathbf{C}_{RB} and damping matrix \mathbf{D} are given by

$$\mathbf{M} = \begin{pmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & J_{comb} \end{pmatrix} \quad (3)$$

$$\mathbf{C}_{RB} = \begin{pmatrix} 0 & -mr & 0 \\ mr & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (4)$$

$$\mathbf{D} = \begin{pmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{pmatrix} \quad (5)$$

with m the displacement and J_{comb} the moment of inertia in the yaw axis including added mass effects ($N_{\dot{r}}$). Regarding the damping matrix \mathbf{D} , note that the linear damping was considered sufficient to describe the dynamics of the Solgenia for docking operations. The input vector $\boldsymbol{\tau}_c = (X \ Y \ N)^T$ combines the forces produced by the actuators. In real maritime applications $\boldsymbol{\tau}_c$ cannot be commanded directly. There is a relationship between the real inputs \mathbf{u} (e.g. propeller speed) and $\boldsymbol{\tau}_c$ that can be described by a dynamical model. This relationship is usually handled by a control allocation algorithm. Nevertheless, deviations between the requested and the actual $\boldsymbol{\tau}_c$ are likely. This will not be considered further here. In the following, it is assumed that the overlaying controller can compensates for these deviations. For further details regarding the modelling of the propulsion system of the Solgenia, please refer to Kinjo et al. (2021).

Disturbances due to wind, current, un-modelled effects etc. are combined with $\boldsymbol{\tau}_d = (X_d \ Y_d \ N_d)^T$. The kinematic equation (2) describes the transformation of the body-fixed velocities in the inertial velocities $\dot{\boldsymbol{\eta}} = (\dot{x} \ \dot{y} \ \dot{\psi})^T$ with rotation matrix

$$\mathbf{J} = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (6)$$

3. MOTION ESTIMATION

Wave filtering and disturbance estimation are crucial in maritime control applications. In Wirtensohn et al. (2016) an Unscented Kalman filter (UKF) was applied for this purpose. In the present work, a UKF is also used, but with a modified transition and measurement model, which are described below. Furthermore, outliers in the measurements are removed by means of gating.

3.1 Transition Model

The transition model is only slightly changed compared to Wirtensohn et al., 2016. It could be written as

$$\dot{\mathbf{x}}_a = \begin{pmatrix} \mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\tau}_d) \\ \mathbf{A}_w \boldsymbol{\xi} \\ \mathbf{0} \end{pmatrix} + \boldsymbol{\omega}_a \quad (7)$$

with state vector $\mathbf{x}_a = (\mathbf{x}^T \ \boldsymbol{\xi}^T \ \boldsymbol{\tau}_d^T)^T$, the vector $\mathbf{x} = (\boldsymbol{\eta}^T \ \boldsymbol{\nu}^T)^T$, the input vector $\mathbf{u} = \boldsymbol{\tau}_c$, and the non-linear function

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{\tau}_d) = \begin{pmatrix} \mathbf{J}(\psi) \boldsymbol{\nu} \\ \mathbf{M}^{-1}(\boldsymbol{\tau}_c + \boldsymbol{\tau}_d - \mathbf{C}_{RB}(\boldsymbol{\nu}) \boldsymbol{\nu} - \mathbf{D} \boldsymbol{\nu}) \end{pmatrix}.$$

The uncertainties are collected in the vector $\boldsymbol{\omega}_a = (\mathbf{0}_{1 \times 3} \ \boldsymbol{\omega}_\nu \ \mathbf{0}_{1 \times 2} \ \boldsymbol{\omega}_w \ \boldsymbol{\omega}_d)^T$, modeled as normally distributed white noise with zero-mean and covariance matrix $\mathbf{Q} = E\{\boldsymbol{\omega}_a \boldsymbol{\omega}_a^T\}$.

It is assumed that the wave-induced motion mainly affects roll, pitch and heave. The heave motion will not be considered further here, as we are only interested in the effects on the planar motion. In order to describe the wave induced roll and pitch motion two uncoupled linear second order models driven by white noise are used. They could be written in the following compact way

$$\dot{\boldsymbol{\xi}} = \mathbf{A}_w \boldsymbol{\xi} + \mathbf{E}_w \boldsymbol{\omega}_w \quad (8)$$

with the state vector $\boldsymbol{\xi} = (\phi \ \theta \ \dot{\phi} \ \dot{\theta})^T$ containing the angles and rates induced by wave motion in roll and pitch direction. Matrix

$$\mathbf{A}_w = \begin{pmatrix} \mathbf{0}_{2 \times 2} & \mathbf{E}_{2 \times 2} \\ -\omega_\phi^2 & 0 & -2\zeta_\phi \omega_\phi & 0 \\ 0 & -\omega_\theta^2 & 0 & -2\zeta_\theta \omega_\theta \end{pmatrix} \quad (9)$$

includes the main wave frequencies ω_i and wave damping ζ_i for the already mentioned directions. Noise entry matrix \mathbf{E}_w finally takes the following form:

$$\mathbf{E}_w = \begin{pmatrix} \mathbf{0}_{2 \times 2} \\ \mathbf{E}_{2 \times 2} \end{pmatrix}. \quad (10)$$

Note that for the application the state space model (7) is discretized using Euler's method for the discrete time UKF.

3.2 Discrete Time Measurement Model

The two antenna RTK-GPS system provides the position and velocity of the first antenna in global coordinates. These measurements are transformed into a local ENU-frame and further considered as the actual measurements.

In addition, the GPS system provides the yaw angle measurements due to the second antenna. The built-in 3-axis MEMS gyroscope measures the three angular velocity components in the body-fixed frame $\boldsymbol{\omega}_{BF} = (\omega_x \ \omega_y \ \omega_z)^T$. Therefore, the measurement model results in

$$\mathbf{z}_k = \begin{pmatrix} \boldsymbol{\eta}_k + \mathbf{P}_\eta \mathbf{R}_z(\psi_k) \mathbf{R}_y(\theta_k) \mathbf{R}_x(\phi_k) \mathbf{l}_{BF} \\ \mathbf{R}(\psi_k) \left(\begin{pmatrix} u_k \\ v_k \end{pmatrix} + \mathbf{S}(\boldsymbol{\omega}_{BF}) \mathbf{l}_{BF} \right) \\ \boldsymbol{\omega}_{BF} \end{pmatrix} + \mathbf{v}_k \quad (11)$$

with vector $\mathbf{l}_{BF} = (x_A \ y_A \ z_A)^T$ specifying the leverarm to the first antenna in the body-fixed frame, $\mathbf{R}_i(\alpha) \in \mathbb{R}^{3 \times 3}$ the rotation matrix describing the rotation around the i -th axis with angle α , matrix

$$\mathbf{P}_\eta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (12)$$

and the reduced cross product matrix

$$\mathbf{S}(\boldsymbol{\omega}_{BF}) = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ 0 & \omega_x & 0 \end{pmatrix}. \quad (13)$$

For a rigid body in 3 dimensions the relationship between the angular velocity $\boldsymbol{\omega}_{BF}$ and the temporal change of the roll, pitch and yaw angles are given by

$$\boldsymbol{\omega}_{BF} = \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + \mathbf{R}_y^T(\theta) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + \mathbf{R}_y^T(\theta) \mathbf{R}_z^T(\psi) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix}. \quad (14)$$

The measurement uncertainty $\mathbf{v}_k \in \mathbb{R}^8$ is modelled as white noise sequence with zero mean.

3.3 Outlier Removal

Even rare outliers in the measurements can have a very negative effect on the estimation result. Therefore, an outlier removal strategy is applied. For related groups of measurements (e.g. from one sensor), the Mahalanobis distance is calculated using the corresponding entries of the innovation covariance matrix $\mathbf{S}_k \in \mathbb{R}^{8 \times 8}$. If measured values exceed a previously defined chi-square confidence level, they are accordingly not taken into account in the UKF update.

4. TRAJECTORY CONTROLLER

4.1 Feedback Linearisation (FBL)

This method is a well known approach in robotics. The basic idea is that a nonlinear system is transformed into a linear one by suitable state feedback, allowing for using linear control strategies in the new coordinates. We consider the typical class of mechanical systems of the form

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{n}(\boldsymbol{\nu}, \boldsymbol{\eta}) = \boldsymbol{\tau}_c + \boldsymbol{\tau}_d \quad (15)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu}, \quad (16)$$

with the assumption that all components of the generalized force vector $\boldsymbol{\tau}_c \in \mathbb{R}^n$ can be specified independently. Note that equation (1) and (2) are a special cases of (15) and (16). To obtain the desired linear system we apply the nonlinear state feedback

$$\boldsymbol{\tau}_c = \mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{n}(\boldsymbol{\nu}, \boldsymbol{\eta}) - \boldsymbol{\tau}_d. \quad (17)$$

yielding the linear dynamics

$$\ddot{\boldsymbol{\eta}} = \mathbf{a}^n. \quad (18)$$

as n decoupled double integrators, with \mathbf{a} being the new input. To determine $\dot{\boldsymbol{\nu}}$ we take the time derivative of equation (16)

$$\ddot{\boldsymbol{\eta}} = \mathbf{a}^n = \dot{\mathbf{J}}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} + \mathbf{J}_\Theta(\boldsymbol{\eta})\dot{\boldsymbol{\nu}} \quad (19)$$

Finally, this results in

$$\dot{\boldsymbol{\nu}} = \mathbf{J}_\Theta^{-1}(\boldsymbol{\eta})(\mathbf{a}^n - \dot{\mathbf{J}}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu}). \quad (20)$$

With perfect knowledge of the model parameters and disturbances $\boldsymbol{\tau}_d$, perfect control of the linearized system could be achieved by choosing the new input $\mathbf{a}^n = \ddot{\boldsymbol{\eta}}_d$. For real systems, the control loop is enhanced by a PD controller for stabilization and disturbance rejection, which leads to the control law.

$$\mathbf{a}^n = \ddot{\boldsymbol{\eta}}_d + \mathbf{K}_d \dot{\tilde{\boldsymbol{\eta}}} + \mathbf{K}_p \tilde{\boldsymbol{\eta}} \quad (21)$$

with $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_d - \boldsymbol{\eta}$. For tracking control $\ddot{\boldsymbol{\eta}}_d$, $\dot{\boldsymbol{\eta}}_d$ and $\boldsymbol{\eta}_d$ are directly obtained from the reference trajectory. In order to guarantee stability, standard linear methods like pole placement or LQR design could be applied. Note, that in Fossen (2011) a PID instead of a PD controller is used. In our case it is assumed that integral action is not needed since feed-forward control via $\ddot{\boldsymbol{\eta}}_d$ and the estimated disturbance vector $\hat{\boldsymbol{\tau}}_d$ from the UKF, acting as a disturbance observer, are provided.

4.2 Nonlinear PID Control (NPID)

Another widely known method in fully actuated robotics is nonlinear PD control. In Fossen (2011) the method is extended with integral action and the so-called acceleration feedback. Acceleration feedback is commonly used in Dynamic Positioning (DP) applications to improve disturbance rejection by artificially increasing the inertia of the system. The increase of inertia is not desired in our application and is therefore not used. In Fossen (2011) nonlinear PID Control is applied to systems of the form

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}_c + \boldsymbol{\tau}_d \quad (22)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} \quad (23)$$

and therefore applicable to systems described by equation (1) and (2). The corresponding PID control law is

$$\boldsymbol{\tau}_c = \mathbf{J}_\Theta^T(\boldsymbol{\eta})\boldsymbol{\tau}_{PID} \quad (24)$$

with

$$\boldsymbol{\tau}_{PID} = \mathbf{K}_p \tilde{\boldsymbol{\eta}} + \mathbf{K}_d \dot{\tilde{\boldsymbol{\eta}}} + \mathbf{K}_i \int_0^t \tilde{\boldsymbol{\eta}}(\tau) d\tau. \quad (25)$$

For set point position stabilization $\dot{\boldsymbol{\eta}}_d = 0$ in Fossen (2011), for $\boldsymbol{\tau}_d = \mathbf{0}$, $\mathbf{K}_i = \mathbf{0}$, $\mathbf{K}_p = \mathbf{K}_p^T < \mathbf{0}$, and $\mathbf{K}_d \leq \mathbf{0}$ the global asymptotic stability (GAS) of the closed loop is shown. For $\boldsymbol{\tau}_d \neq \mathbf{0}$, but $\dot{\boldsymbol{\tau}}_d = \mathbf{0}$ boundedness can be proven. It is also pointed out for $\mathbf{K}_i \neq \mathbf{0}$ that a proof for local asymptotic stability (LAS) exists. Usually, the control law could also be applied to stabilization of slow varying trajectories ($\dot{\boldsymbol{\eta}}_d$ small). For the later comparison of the different control strategies $\mathbf{K}_i = 0$ is chosen, since a disturbance feed-forward is also applied here via $\hat{\boldsymbol{\tau}}_d$, which finally leads to the control law:

$$\boldsymbol{\tau}_c = \mathbf{J}_\Theta^T(\mathbf{K}_p \tilde{\boldsymbol{\eta}} + \mathbf{K}_d \dot{\tilde{\boldsymbol{\eta}}}) - \boldsymbol{\tau}_d. \quad (26)$$

4.3 Flatness Based FF with PD-Control (NPDFLAT)

From the literature it is well known that the system described by equation (22) and (23) is differentially flat. This

can be exploited to derive a flatness based feed-forward control law. In brief, a system is called differentially flat if it satisfies the following three conditions:

$$\mathbf{y} = \Phi_1(\mathbf{x}, \mathbf{u}, \dot{\mathbf{u}}, \dots, \mathbf{u}^\alpha)$$

$$\mathbf{x} = \Phi_2(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^\beta)$$

$$\mathbf{u} = \Phi_3(\mathbf{y}, \dot{\mathbf{y}}, \dots, \mathbf{y}^\gamma)$$

With \mathbf{y} called a flat output. With regards to the used notations this translates to:

$$\mathbf{y} = \boldsymbol{\eta}$$

$$\mathbf{x} = (\boldsymbol{\eta} \ \boldsymbol{\nu})^T = \Phi_2(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \dots, \boldsymbol{\eta}^\beta)$$

$$\mathbf{u} = \boldsymbol{\tau}_c = \Phi_3(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \dots, \boldsymbol{\eta}^\gamma)$$

The first condition is trivial. The second condition can be written as follows:

$$\mathbf{x} = \begin{pmatrix} \boldsymbol{\eta} \\ \mathbf{J}_{\Theta}^T(\boldsymbol{\eta})\dot{\boldsymbol{\eta}} \end{pmatrix}$$

The third condition is the most interesting, since it leads directly to the relation

$$\mathbf{u} = \boldsymbol{\tau}_{flat} = \mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{\tau}_d \quad (27)$$

with $\boldsymbol{\nu} = \mathbf{J}_{\Theta}^T(\boldsymbol{\eta})\dot{\boldsymbol{\eta}}$ and $\dot{\boldsymbol{\nu}} = \dot{\mathbf{J}}_{\Theta}^T(\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \mathbf{J}_{\Theta}^T(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}}$.

The feed-forward is obtained by replacing $\boldsymbol{\eta}$, $\dot{\boldsymbol{\eta}}$, and $\ddot{\boldsymbol{\eta}}$ with the desired quantities $\boldsymbol{\eta}_d$, $\dot{\boldsymbol{\eta}}_d$, and $\ddot{\boldsymbol{\eta}}_d$. Combined with a PD controller the final control law is yield by

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_{flat} + \mathbf{J}_{\Theta}^T(\mathbf{K}_p\tilde{\boldsymbol{\eta}} + \mathbf{K}_d\dot{\tilde{\boldsymbol{\eta}}}). \quad (28)$$

4.4 Tuning of the Controllers

For enabling the comparison of the different control strategies in an appropriate manner, $\mathbf{K}_p = (K_x \ K_y \ K_\psi)^T$ and $\mathbf{K}_d = (K_{dx} \ K_{dy} \ K_{d\psi})^T$ are determined using an optimization process for each of the three controllers. The control error $\mathbf{e}_k = (\tilde{\boldsymbol{\eta}}_k \ \dot{\tilde{\boldsymbol{\eta}}}_k)^T$ and the temporal change of the inputs $\dot{\mathbf{u}}_k = (\dot{n}_{Az} \ \dot{\alpha}_{Az} \ \dot{n}_{Bow})^T$ are included into the associated cost function

$$\mathcal{J} = \sum_{n=1}^{N_R} \frac{1}{N_n} \sum_{k=1}^{N_n} \mathbf{e}_k^T \mathbf{Q}_e^{-1} \mathbf{e}_k + \dot{\mathbf{u}}_k^T \mathbf{Q}_u^{-1} \dot{\mathbf{u}}_k \quad (29)$$

with \mathbf{Q}_e and \mathbf{Q}_u the weighting matrices, N_R the number of different simulation runs and N_n the number of samples of a corresponding simulation run. The individual simulation runs are generated by simulating the complete controlled system with UKF, dynamic model of the watercraft (incl. actuators), trajectory planning and the corresponding control algorithm. In order to include the aspect robustness of the control system into the optimization, trajectories, measurement disturbances and deviations in the model parameters are randomly selected within defined limits (e.g. $\pm 20\%$ for the model parameters). Note, that the random selection is made in advance and remains identical during optimization for all controllers. For symmetry reasons $K_x = K_y$ and $K_{dx} = K_{dy}$ were assumed, which leads to 4 optimization variables. The used weighting matrices, number of runs and obtained values for the optimization can be found in Tab. 1. Finally, the optimized \mathbf{K}_p and \mathbf{K}_d values obtained from the optimization are used for the following Monte Carlo Analysis and the controller comparison in section 5.

Table 1. Obtained values from controller parameter optimization including weighting matrices and number of simulation runs

	FBL	NPID	NPIDFLAT
K_x / K_y	0.3336	2019	923
K_ψ	0.7427	18845	13815
K_{dx} / K_{dy}	1.6	5299	4742
$K_{d\psi}$	1.46	18845	11865
\mathbf{Q}_e	$\text{diag}(5 \ 5 \ 28.6 \ 200 \ 200 \ 57.3)$		
\mathbf{Q}_u	$\text{diag}(0.0017 \ 0.0667 \ 0.0002)$		
N_R	50		

Table 2. Performance metrics of the different controllers for the Monte Carlo simulation

	FBL	NPID	NPIDFLAT
RMSE Pos [m]	0.0636	0.0738	0.0734
RMSE Yaw [°]	0.3579	0.592	0.4021
RMSE dPos [m/s]	0.0213	0.0249	0.0221
RMSE dYaw [°/s]	0.1775	0.2913	0.2212
Max. RMSE Pos [m]	0.1528	0.1429	0.17
Max. RMSE Yaw [°]	0.8333	1.9882	1.0068
Max. RMSE dPos [m/s]	0.0483	0.053	0.0485
Max. RMSE dYaw [°/s]	0.4643	2.6938	0.8211
Mean $ \dot{n}_{Az} $ [RPM/s]	202.84	233.15	180.54
Mean $ \dot{\alpha}_{Az} $ [°/s]	31.09	34.13	27.90
Mean $ \dot{n}_{Bow} $ [RPM/s]	827.48	748.47	572.99

4.5 Monte Carlo Simulation

To investigate the behavior of the different control strategies under parameter change, a Monte Carlo analysis is performed. As in the optimization, the trajectories, model parameters and the measurement noise are chosen randomly. In addition, slowly varying environmental disturbances $\boldsymbol{\tau}_d$ are added. For each controller 100 Monte Carlo simulations are performed. The performance achieved is summarized in Table 2. On the one hand, the RMSE errors averaged over all MC simulations and, on the other hand, the maximum RMSE for the worst MC run are listed. For better comparability, the average changes of the actuation variables are also listed.

Looking at the average RMSE values, the three control strategies are relatively close to each other. Only a small advantage can be found for FBL and NPIDFLAT in contrast to NPID. However, this advantage is significantly larger for the maximum RMSE values, but with FBL performing better than NPIDFLAT. With regard to the change of the actuation variables, a clear advantage of NPIDFLAT to FFL and NPID can be identified. This can also be seen in Fig. 2, which shows the error curves over all MC simulations for NPID. It can be clearly seen from the errors in yaw and yaw rate that oscillations occur with NPID in some scenarios. The oscillation do not occur for the other controllers, but the related figures are not shown here due to space reason. It should be noted again that the MC runs for the different controllers are identical.

In summary, despite significant changes in model parameters ($\pm 30\%$ was used), the two methods with model-based feed-forward have still an advantage in terms of performance and change in actuation variables. In the following chapter, it will be investigated whether this can be observed in the same way for the real tests.

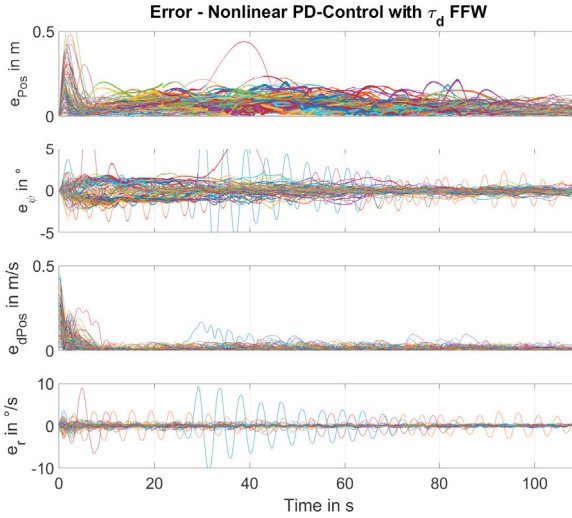


Fig. 2. Errors during Monte Carlo runs. From top to bottom: Euclidean position error, yaw angle error, Euclidean velocity error, yaw rate error.

5. EXPERIMENTAL RESULTS

In this section, the results of the real docking tests are presented. The tests were carried out on the university's own jetty at the Rhine river. The final pose of the dock at the jetty is known. By activating the docking procedure, a trajectory from the current pose to the docking pose is planned. The trajectory generation is not the subject of this paper, but the procedure of the docking process will be briefly explained with reference to Figs. 3 and 4. For better orientation, the jetty is schematically drawn at the top right of Fig. 3. The trajectory can be divided into 4 sections:

- (1) The position remains constant and the boat turns to the initial orientation of the trajectory
- (2) The boat moves tangentially along the trajectory with continuous curvature (euler spiral)
- (3) The boat moves tangentially along the straight line section into the dock
- (4) The boat moves sideways to the jetty with constant orientation

Continuous transitions are provided using velocity ramps. Note that in Fig. 3 the boat moves backwards along the trajectory into the dock, since it is possible to perform the docking process forward and backward. A total of three tests in the forward direction and three tests in the reverse direction were performed. The tests were carried out directly one after the other and care was taken to ensure an almost identical initial pose. For a better overview and space reasons, only the 3 backward docking operations are shown in Fig. 3 to 5, while the control errors of all 6 trials are drawn in Fig. 6.

The related performance metrics are listed in Tab. 3. In contrast to Results of the Monte Carlo Simulation in Table 2, the RMSE Values of the worst run is not listed here, since there are only 2 trails per controller. The RMSE values of the real tests confirm the very good impression during the real tests. The boat was able to reach the docking pose in a very precise and repeatable manner.

Table 3. Performance metrics of the different control strategies for the experimental trails

	FBL	NPID	NPIDFLAT
RMSE Pos [m]	0.0735	0.0764	0.0814
RMSE Yaw [°]	0.7634	0.9066	0.7371
RMSE dPos [m/s]	0.0316	0.0367	0.0306
RMSE dYaw [°/s]	0.6516	0.8277	0.6277
Mean \dot{n}_{Az} [/s]	625.6	822.9	606
Mean $\dot{\alpha}_{Az}$ [°/s]	77	69.2	74.4
Mean \dot{n}_{Bow} [RPM/s]	2589.3	2070.2	1855.8

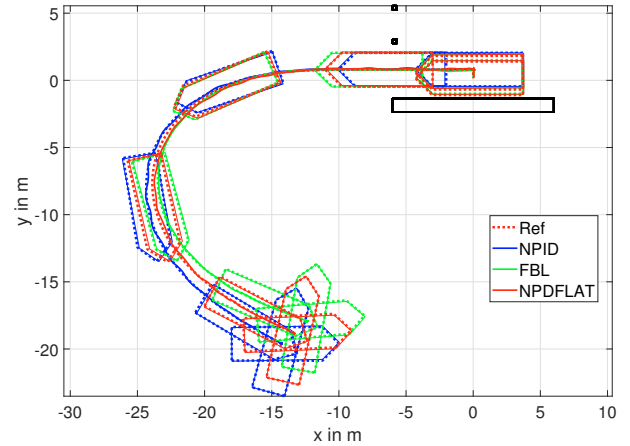


Fig. 3. X-Y-Plot the real docking scenarios (only backward scenarios)

In comparison with the Monte Carlo analyses, the RMSE values are somewhat worse. Nevertheless, it is close to the accuracy of the used RTK GPS system. Since a trajectory with a relatively strong curvature was selected for the real tests, it is difficult to compare changes in the actuator variables, since the starting point of the trajectories was randomly selected in the Monte Carlo analysis and thus many trajectories with a significantly weaker curvature were also selected.

Comparing the controllers with each other, it is difficult to identify a clear winner on the basis of the table. A significant difference are that NPID performs worse in terms of accuracy in yaw and yaw rate. This becomes even clearer when looking at Fig. 4. Oscillations in yaw are most pronounced with NPID and this is independent of whether docking is done in forward or backward direction. This becomes clear from Fig. 6 and was also noticeable during the real test. At this point it should be noted that the oscillations have almost the same frequency as the oscillations already noticed in the Monte Carlo analysis. Furthermore, it is important to mention that NPIDFLAT has a slight advantage with regard to the change of the thruster speeds.

In summary, all three controllers show very good results in the real tests. However, there is a small advantage of NPIDFLAT when considering performance and change of actuator variables. It should be noted that the performance of the overall system also depends to a large extent on the motion estimation which also provides disturbance estimation and wave filtering. However, due to space limitations, no additional results regarding the motion estimation are presented.

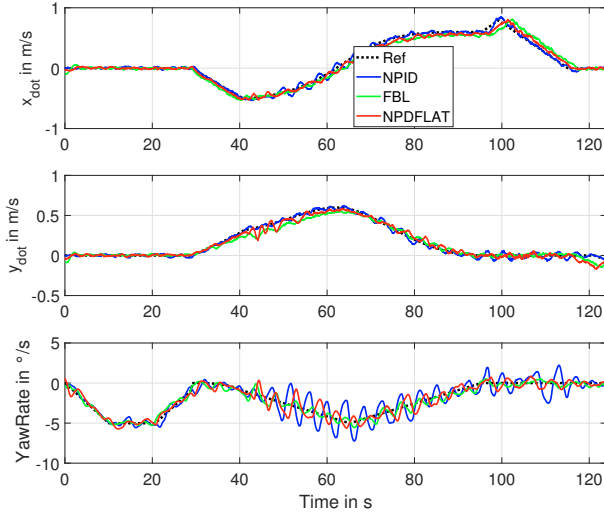


Fig. 4. Velocities in local coordinates during the real docking scenarios (only backward scenarios)

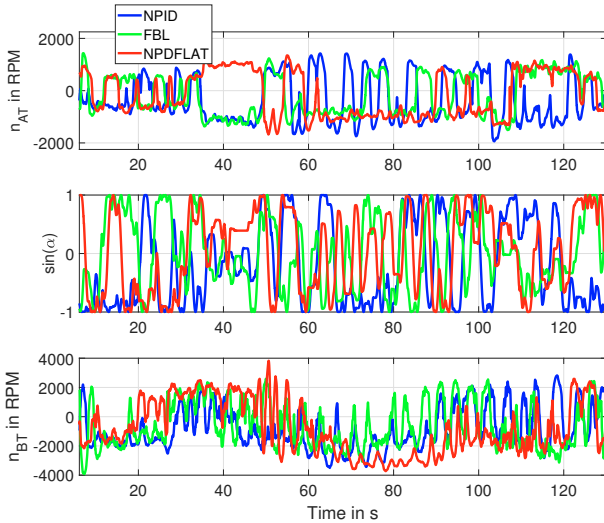


Fig. 5. Actuation variables during the real docking scenarios (only backward scenarios)

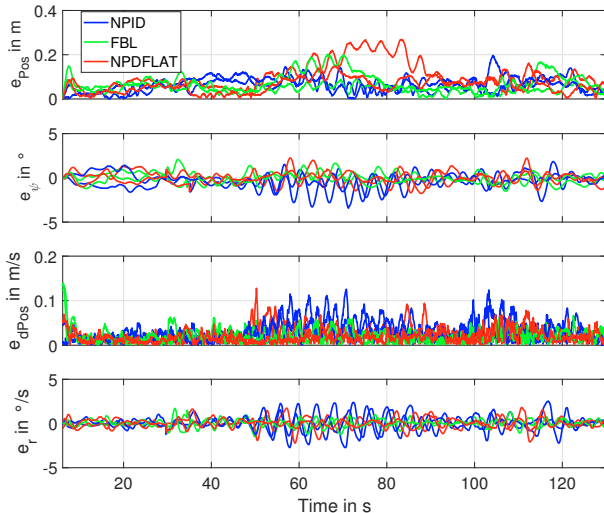


Fig. 6. Errors during the real docking scenarios (forward and backward scenarios)

6. CONCLUSION AND FUTURE WORK

The paper presents a systematic analysis of three state of the art trajectory controllers with application for automated docking. For comparability reasons, the controller parameters are subject to an optimization based on a quality function that penalizes the control error and change of the actuation variables. The optimized controller parameters are used directly for both a Monte Carlo analysis and the real tests trails with the university's experimental watercraft Solgenia. Particular attention was paid to ensuring that the conditions in the real tests were as similar as possible. The investigation showed that all of the three controllers exhibit excellent performance in simulation and in the real tests and are thus suitable for docking applications. A slightly better performance of the NPDFLAT controller was observed, but the relatively small number of tests does not allow a final statement. The present study represents a ground truth for further investigations of modern approaches like model predictive control and reinforcement learning for automated docking applications at the Institute of System Dynamics Konstanz.

REFERENCES

- Bitar, G., Martinsen, A. B., Lekkas, A. M., Breivik, M., 2020. Trajectory planning and control for automatic docking of asvs with full-scale experiments. IFAC-PapersOnLine 53 (2), 14488–14494, 21th IFAC World Congress.
- Fossen, T. I., 2011. Marine Craft Hydrodynamics and Motion Control.
- Kinjo, L. M., Wirtensohn, S., Reuter, J., Menard, T., Gehan, O., 2021. Trajectory tracking of a fully-actuated surface vessel using nonlinear model predictive control. In: Proc. of the 13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles (CAMS21).
- Li, S., Liu, J., Negenborn, R. R., Wu, Q., 2020. Automatic docking for underactuated ships based on multi-objective nonlinear model predictive control. IEEE Access 8, 70044–70057.
- Martinsen, A. B., Lekkas, A. M., Gros, S., 2019. Autonomous docking using direct optimal control. IFAC-PapersOnLine 52 (21), 97–102.
- Mizuno, N., Kakami, H. K., Okazaki, T., 2012. Parallel simulation based predictive control scheme with application to approaching control for automatic berthing. In: IFAC Conference on Manoeuvring and Control of Marine Craft. IFAC, pp. 19–25.
- Mizuno, N., Uchida, Y., Okazaki, T., 2015. Quasi real-time optimal control scheme for automatic berthing. IFAC-PapersOnLine 48 (16), 305–312, 10th IFAC Conference on Manoeuvring and Control of Marine Craft MCMC 2015.
- Skulstad, R., Li, G., Fossen, T. I., Vik, B., Zhang, H., 2021. A hybrid approach to motion prediction for ship docking—integration of a neural network model into the ship dynamic model. IEEE Transactions on Instrumentation and Measurement 70, 1–11.
- Wirtensohn, S., Schuster, M., Reuter, J., 2016. Disturbance estimation and wave filtering using an unscented kalman filter. IFAC-PapersOnLine 49 (23), 518–523.