machines

MDPI

Article

Feature-Based MPPI Control with Applications to Maritime Systems †

Hannes Homburger [1,*], Stefan Wirtensohn [1], Moritz Diehl [2,3] and Johannes Reuter [1]

[1] Institute of System Dynamics, HTWG Konstanz—University of Applied Sciences, 78462 Konstanz, Germany
[2] Department of Microsystems Engineering (IMTEK), University of Freiburg, 79110 Freiburg, Germany
[3] Department of Mathematics, University of Freiburg, 79110 Freiburg, Germany
* Correspondence: hhomburg@htwg-konstanz.de

† This paper is an extended version of our paper published in Proceedings of the 6th IEEE Conference on Control Technology and Applications (CCTA), Trieste, Italy, 22–25 August 2022, with the title "Feature-Based Proposal Density Optimization for Nonlinear Model Predictive Path Integral Control".

Abstract: In this paper, a novel feature-based sampling strategy for nonlinear Model Predictive Path Integral (MPPI) control is presented. Using the MPPI approach, the optimal feedback control is calculated by solving a stochastic optimal control (OCP) problem online by evaluating the weighted inference of sampled stochastic trajectories. While the MPPI algorithm can be excellently parallelized, the closed-loop performance strongly depends on the information quality of the sampled trajectories. To draw samples, a proposal density is used. The solver's and thus, the controller's performance is of high quality if the sampled trajectories drawn from this proposal density are located in low-cost regions of state-space. In classical MPPI control, the explored state-space is strongly constrained by assumptions that refer to the control value's covariance matrix, which are necessary for transforming the stochastic Hamilton–Jacobi–Bellman (HJB) equation into a linear second-order partial differential equation. To achieve excellent performance even with discontinuous cost functions, in this novel approach, knowledge-based features are introduced to constitute the proposal density and thus the low-cost region of state-space for exploration. This paper addresses the question of how the performance of the MPPI algorithm can be improved using a feature-based mixture of base densities. Furthermore, the developed algorithm is applied to an autonomous vessel that follows a track and concurrently avoids collisions using an emergency braking feature. Therefore, the presented feature-based MPPI algorithm is applied and analyzed in both simulation and full-scale experiments.

Keywords: sample-based nonlinear model predictive control; stochastic system dynamics; nonlinear model predictive control; maritime systems; collision avoidance

check for updates

Citation: Homburger, H.; Wirtensohn, S.; Diehl, M.; Reuter, J. Feature-Based MPPI Control with Applications to Maritime Systems. *Machines* **2022**, *10*, 900. https://doi.org/10.3390/machines10100900

Academic Editor: Stefan Palis

Received: 13 September 2022
Accepted: 29 September 2022
Published: 6 October 2022

1. Introduction

Recently, empowered by the progresses in efficiency of optimization algorithms and the available computing power, sample-based nonlinear model predictive control (NMPC) approaches can be used to solve nonlinear stochastic OCP in real time [1]. Using the path integral control framework [2], the stochastic HJB equation that corresponds to a restricted class of stochastic OCPs can be transformed into a linear partial second-order differential equation [3]. This class is characterized by arbitrary but input affine dynamics, containing additive white Gaussian noise (AWGN) at the inputs, and the cost function is only restricted to be quadratic in the controls [4]. According to [3], a Feynman–Kac path integral can be used to express the solution for both, the optimal controls and the optimal value function. Thus, the solution of the transformed stochastic HJB is formulated as a conditional expectation value with respect to the system dynamics. As a result, the optimal control can be estimated using Monte Carlo methods drawing samples of stochastic trajectories [5]. While in general the resulting optimal feedback control function has an unknown structure, there are different approaches for its representation. Besides reinforcement learning (RL) approaches based on offline learning a parametrized policy [6,7], NMPC has become the de-facto technological standard [8,9]. Based on path integrals, a new type of sample-based NMPC algorithm was presented in [8]. Using a free energy definition described in [10] and [11], the input affine requirement is completely removed by [4]. Because MPPI is based on Monte Carlo simulation, the information content of the drawn samples is highly dependent on the proposal density [12,13]. Due to this fact, the system's performance can be significantly improved by provisioning a good proposal density. While recently in [14] a robust MPPI version, in [15] a covariance steering approach and in [16] a learning-based algorithm are introduced, in this paper, a novel feature-based MPPI extension is presented. In Section 2, the basics of the MPPI approach are described. This is followed by the derivation of a feature-based extension of the MPPI algorithm in Section 3. In Section 4, an application scenario is defined containing the equations of motions of a vessel and cost functions, which are combined as an OCP. Furthermore, an emergency braking feature is defined. In Section 5, the controller design is presented including the architecture, the controller parameters and the cost function parameters. A comparison of usual MPPI control and its feature-based extension in a simulation environment is given in Section 6. In Section 7, full-scale experiments with the research vessel *Solgenia* are used to compare the MPPI algorithm and its feature-based extension. This is followed by the conclusions of this paper including ideas for future work in Section 8.

2. Model Predictive Path Integral Control

Recently, a sample-based NMPC algorithm was derived by [4] using the path integral framework. This so-called MPPI approach calculates the optimal control inputs by numerically solving the time-discrete nonlinear stochastic OCP

$$\min_{\boldsymbol{U}} \mathbb{E}_{\mathbb{Q}}\left\{\phi(\boldsymbol{X}_T) + \sum_{t=0}^{T-1}[C(\boldsymbol{X}_t) + \boldsymbol{u}_t^\top \boldsymbol{R}\boldsymbol{u}_t] | \boldsymbol{X}_0 = \boldsymbol{x}_0\right\} \tag{1a}$$

$$\text{with } \boldsymbol{X}_{t+1} = \boldsymbol{F}(\boldsymbol{X}_t, \boldsymbol{v}_t), \quad \forall t \in \{0, 1, \dots, T-1\}, \tag{1b}$$

$$\text{and } \boldsymbol{v}_t = \boldsymbol{u}_t + \boldsymbol{\epsilon}_t,\ \boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}),\ \boldsymbol{R} = \lambda\boldsymbol{\Sigma}^{-1},\ \lambda \in \mathbb{R}^+$$

in real time, where $\boldsymbol{F} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denotes the time-discrete nonlinear system dynamics of the stochastic system state $\boldsymbol{X}_t \in \mathbb{R}^n$, and the actual system input denoted by $\boldsymbol{v}_t \in \mathbb{R}^m$ that is given by the commanded system's input $\boldsymbol{u}_t \in \mathbb{R}^m$ with time discrete AWGN $\boldsymbol{\epsilon}_t \in \mathbb{R}^m$ with covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times m}$. The so-called temperature of the stochastic system is denoted by λ. The objective function is given by the expected value of the terminal costs denoted by $\phi(\boldsymbol{X}_T)$, the instantaneous stage costs denoted by $C(\boldsymbol{X}_t)$ and a quadratic input term with weighting matrix $\boldsymbol{R} \in \mathbb{R}^{m \times m}$. The objective function to be minimized in (1a) evaluates the expected costs subject to the commanded system inputs $\boldsymbol{U} = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \dots, \boldsymbol{u}_{T-1}\} \in \mathbb{R}^{m \times T}$ under the measurement distribution $\mathbb{Q}$ corresponding to the probability density function (PDF)

$$q(\boldsymbol{V}) = \prod_{t=0}^{T-1}[(2\pi)^m|\boldsymbol{\Sigma}|]^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{v}_t - \boldsymbol{u}_t^P)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{v}_t - \boldsymbol{u}_t^P)\right), \tag{2}$$

where $\boldsymbol{V} = \{\boldsymbol{v}_0, \boldsymbol{v}_1, \dots, \boldsymbol{v}_{T-1}\}$ denotes the sequence of actual input values. The distribution of the controlled system denoted by $\mathbb{Q}$ corresponds to an open-loop sequence of manipulated variables $\boldsymbol{u}_t = \boldsymbol{u}_t^P$ for $t = 0, 1, \dots, T-1$. According to [17], the expectation value under a measurement is defined as $\mathbb{E}_{\mathbb{Q}}(\rho(\boldsymbol{V})) := \int \rho(\boldsymbol{V})q(\boldsymbol{V})\,d\boldsymbol{V}$, where ρ denotes a scalar function. In this section, the basics of this novel algorithm are described before it will be extended in the next section. According to [4], the density function of the uncontrolled system denoted by $\mathbb{P}$ with zero input $\boldsymbol{u}_t = \mathbf{0},\ t = 0, 1, \ldots, T-1$ leads to the PDF

$$p(\boldsymbol{V}) = \prod_{t=0}^{T-1} [(2\pi)^m |\boldsymbol{\Sigma}|]^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{v}_t^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{v}_t\right). \tag{3}$$

An initial state $\boldsymbol{x}_0$ and a realized sequence of input values $\boldsymbol{V}_k$ can be uniquely assigned to a trajectory without stochastic influence by recursively applying (1b). Introducing the cumulated state-dependent path costs $S(\boldsymbol{V}_k) = \phi(\boldsymbol{x}_{k,T}) + \sum_{t=0}^{T-1} C(\boldsymbol{x}_{k,t})$ according to [4], the value function of the OCP (1a) is given by

$$V(t, \boldsymbol{x}_t) = -\lambda \log \mathbb{E}_{\mathbb{P}}\left\{e^{-\frac{1}{\lambda}S} | \boldsymbol{X}_t = \boldsymbol{x}_t\right\}, \tag{4}$$

with respect to the uncontrolled dynamics [4]. To express the value function with respect to $\mathbb{Q}$, the likelihood ratio $\frac{p(V)}{q(V)}$ must be introduced, which yields

$$V(t, \boldsymbol{x}_t) = -\lambda \log \mathbb{E}_{\mathbb{Q}}\left\{\frac{p(\boldsymbol{V})}{q(\boldsymbol{V})} e^{-\frac{1}{\lambda}S} | \boldsymbol{X}_t = \boldsymbol{x}_t\right\}. \tag{5}$$

Applying Jensen's inequality according to [4] yields

$$V(t, \boldsymbol{x}_t) \leq -\lambda \mathbb{E}_{\mathbb{Q}}\left\{\log \frac{p(\boldsymbol{V})}{q(\boldsymbol{V})} e^{-\frac{1}{\lambda}S} | \boldsymbol{X}_t = \boldsymbol{x}_t\right\}, \tag{6}$$

where the bound is tight with

$$q^*(\boldsymbol{V}) = \frac{1}{\eta} \exp\left(-\frac{1}{\lambda}S(\boldsymbol{V})\right) p(\boldsymbol{V}), \tag{7}$$

where $\eta \in \mathbb{R}$ denotes a normalization constant. In [4], it is shown that the associated optimal control values are given by

$$\begin{aligned} \boldsymbol{u}_t^* &= \arg\min_{\boldsymbol{u}_t} \mathbb{D}_{KL}(\mathbb{Q}^* || \mathbb{Q}) \\ &= \mathbb{E}_{\mathbb{Q}^*}[\boldsymbol{v}_t] \end{aligned} \tag{8}$$

where $\mathbb{D}_{KL}$ denotes the Kullback–Leibler divergence, and $\mathbb{Q}^*$ denotes the abstract optimal distribution. According to [4], the optimal input is given by

$$\boldsymbol{u}_t^* = \int_{\Omega_V} q(\boldsymbol{V}) \underbrace{\frac{q^*(\boldsymbol{V})}{p(\boldsymbol{V})} \frac{p(\boldsymbol{V})}{q(\boldsymbol{V})}}_{\omega(\boldsymbol{V})} \boldsymbol{v}_t d\boldsymbol{V} \tag{9a}$$

$$= \mathbb{E}_{\mathbb{Q}}\{\omega(\boldsymbol{V})\boldsymbol{v}_t\} \tag{9b}$$

minimizing (1a), where Ω_V denotes the image of the sample space, and the importance weighting

$$\omega(\boldsymbol{V}) = \frac{1}{\eta} \exp\left(-\frac{1}{\lambda}S(\boldsymbol{V}) + \sum_{t=0}^{T-1} \frac{1}{2}\boldsymbol{u}_t^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_t - \boldsymbol{v}_t^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{u}_t\right) \tag{10}$$

can be calculated using the PDFs (2), (3) and (7). Using Monte Carlo simulation, (9a) can be estimated via the iterative update law

$$\boldsymbol{u}_t^{i+1} = \boldsymbol{u}_t^i + \sum_{n=1}^{N} \omega(\boldsymbol{V}_n)(\boldsymbol{v}_t^n - \boldsymbol{u}_t^i), \tag{11}$$

where N samples are drawn from the system dynamics (1b) with the commanded control input sequence $\boldsymbol{U} = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{T-1}\}$. The iterative procedure described in Algorithm 1 is used to estimate the optimal commanded control input, and to improve the required importance sample distribution $\mathbb{Q}$ simultaneously.

Algorithm 1 Optimize Control Sequence (OCS) acc. to [4]

```
Input: $\boldsymbol{F}$: Transition model;
    $K$: Number of samples;
    $T$: Number of timesteps;
    $\boldsymbol{U}^I$: Initial control sequence;
    $\boldsymbol{x}_0$: Recent state estimate;
    $\Sigma, \phi, C, \lambda$: Control hyper-parameters;
Output: $\boldsymbol{U}^O$: Optimized control sequence
    $\overline{S}$: Average costs;
for $k \in \{0, 1, \ldots, K-1\}$ do
    $\boldsymbol{x}_{k,0} \leftarrow \boldsymbol{x}_0$;
    Sample $\{\boldsymbol{\epsilon}_0^k, \boldsymbol{\epsilon}_1^k, \ldots, \boldsymbol{\epsilon}_{T-1}^k\}$;
    $S(k) \leftarrow 0$;
    for $t \in 1, 2, \ldots, T$ do
        $\boldsymbol{x}_{k,t} \leftarrow \boldsymbol{F}(\boldsymbol{x}_{k,t-1}, \boldsymbol{u}_{t-1}^I + \boldsymbol{\epsilon}_{t-1}^k)$;
        $S(k) += C(\boldsymbol{x}_{k,t}) + \lambda \boldsymbol{u}_{t-1}^{I\top} \Sigma^{-1} \boldsymbol{\epsilon}_{t-1}^k$;
    end for
    $S(k) += \phi(\boldsymbol{x}_{k,T})$;
end for
$\beta \leftarrow \min_k [S(k)]$;
$\eta \leftarrow \sum_{k=0}^{K-1} \exp(-\frac{1}{\lambda} S(k) - \beta)$;
for $k \in \{0, 1, \ldots, K-1\}$ do
    $\omega(k) \leftarrow \frac{1}{\eta} \exp(-\frac{1}{\lambda} S(k) - \beta)$;
end for
for $t \in \{0, 1, \ldots, T-1\}$ do
    $\boldsymbol{u}_t^O \leftarrow \boldsymbol{u}_t^I + \sum_{k=0}^{K-1} \omega(k) \boldsymbol{\epsilon}_t^k$;
end for
$\overline{S} \leftarrow \frac{1}{K} \sum_{k=0}^{K-1} S(k)$;
return $\boldsymbol{U}^O = \{\boldsymbol{u}_0^O, \boldsymbol{u}_1^O, \ldots, \boldsymbol{u}_{T-1}^O\}$ and $\overline{S}$
```

3. Extension to Feature-Based Proposal Density

In this section, the exploration problem of standard MPPI control is described, the idea of feature-based extension is presented and the resulting feature-based MPPI algorithm is presented.

3.1. Exploration Problem of Classical MPPI Control

To improve the sampling efficiency, both the value function (5) and the optimal control sequence (9b) can be calculated by sampling trajectories under the probability measure $\mathbb{Q}$ with the proposal PDF (2). Due to the restrictive assumption $\boldsymbol{R} = \lambda \boldsymbol{\Sigma}^{-1}$, the proposal PDF is only parametrized by a sequence of inputs $\boldsymbol{U}^P = \{\boldsymbol{u}_0^P, \boldsymbol{u}_1^P, \ldots, \boldsymbol{u}_{T-1}^P\}$. The sampling efficiency is dependent on the quality of the proposal density [13]. In [4], the recent estimate $\boldsymbol{U}_t$ of the optimal control sequence $\boldsymbol{U}^*$ is used to determine the proposal PDF for the next sampling iteration. The objective function (1a) is not necessarily convex. Using an infinite number of samples, the MPPI algorithm is a global minimizer. However, due to a finite number of drawable samples, the explored state-space is concentrated around the last approximation. In order not to remain in a local minimum, the explored state-space must be enlarged. An exemplary possibility to enlarge the explored state–space is shown in Figure 1.

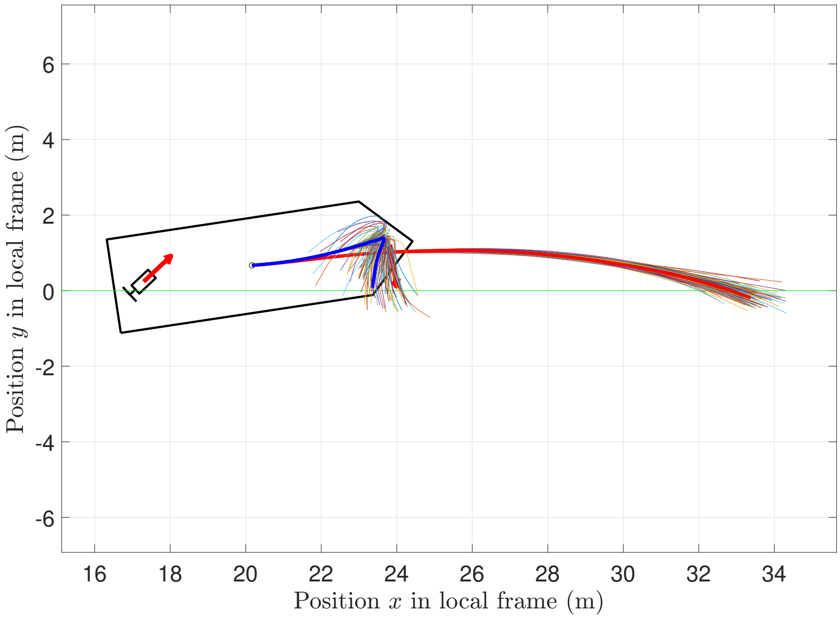


Figure 1. Explored state–space in a full–scale experiment visualized by sampled trajectories using proposal densities with mean $\boldsymbol{U}^0$ (red) and mean $\boldsymbol{U}^F$ (blue) representing the emergency braking feature. The actuators' force vectors are drawn in red. The desired trajectory is visualized in green.

3.2. Feature-Based Extension of the Search Space

An enlargement of the explored state-space is possible by introducing an additional feature-based proposal density to draw a part of the samples from. Therefore, a feature is implicitly defined by solving the artificially designed OCP

$$\boldsymbol{U}^F = \arg\min_{\boldsymbol{U}} \mathbb{E}_{\mathbb{Q}_F}\left\{\phi_F(\boldsymbol{X}_T) + \sum_{t=0}^{T-1}[C_F(\boldsymbol{X}_t) + \boldsymbol{u}_t^\top \boldsymbol{R}\boldsymbol{u}_t | \boldsymbol{X}_0 = \boldsymbol{x}_0]\right\} \tag{12a}$$

$$\text{with } \boldsymbol{X}_{t+1} = \boldsymbol{F}(\boldsymbol{X}_t, \boldsymbol{v}_t) \text{ and } \boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{u}_t, \boldsymbol{\Sigma}),\ \boldsymbol{R} = \lambda\boldsymbol{\Sigma}^{-1}, \tag{12b}$$

where $\phi_F : \mathbb{R}^n \to \mathbb{R}$ denotes the terminal costs of a feature and $C_F : \mathbb{R}^n \to \mathbb{R}$ denotes the instantaneous costs of a feature. Thus, the feature-based proposal density is given by

$$q_F(\boldsymbol{V}) = \prod_{t=0}^{T-1}[(2\pi)^m|\boldsymbol{\Sigma}|]^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{v}_t - \boldsymbol{u}_t^F)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{v}_t - \boldsymbol{u}_t^F)\right), \tag{13}$$

parametrized by a sequence of inputs $\boldsymbol{U}^F = \{\boldsymbol{u}_0^F, \boldsymbol{u}_1^F, \ldots, \boldsymbol{u}_{T-1}^F\}$. Analogous to the previous section, the stochastic OCP (12a) can be solved by using the MPPI algorithm.

3.3. Resulting Feature-Based MPPI Algorithm

The resulting feature-based MPPI algorithm is presented in Algorithm 2. First, the control sequence is optimized using Algorithm 1, and the predicted costs are evaluated. Then, all feature control sequences are improved, and their performances regarding the main cost function are evaluated. The best control sequence is chosen to be the main control sequence. Then, the first element of the control sequence is applied. Subsequently, the sequence is shifted, and the last element is initialized. Note: In the implementation, lines 4 and 5 can be combined to reduce the computational effort.

Algorithm 2 Feature-Based MPPI Control

```
Input: $\boldsymbol{F}$: Transition model;
    $K$: Number of samples;
    $T$: Number of timesteps;
    $\boldsymbol{U}^0$: Initial control sequence;
    $\boldsymbol{x}_0$: Recent state estimate;
    $\Sigma, \phi, C, \lambda$: Control hyper-parameters;
    $I$: Number of features with feature index $i = \{1, 2, \ldots, I\}$;
    $\boldsymbol{U}^i$: Initial control sequence of $i_{\text{th}}$ feature;
    $\phi_i, C_i$: State dependent costs of $i_{\text{th}}$ feature;
    $K_i$: Number of $i_{\text{th}}$ features samples;
while Controller is active do
    $[\boldsymbol{U}^0, \overline{S}_0] \leftarrow \text{OCS}(K, T, \boldsymbol{U}^0, \boldsymbol{x}_0, \Sigma, \phi, C, \lambda)$
    for $i \in \{1, 2, \ldots, I\}$ do
        $[\boldsymbol{U}^i, -] \leftarrow \text{OCS}(K_i, T, \boldsymbol{U}^i, \boldsymbol{x}_0, \Sigma, \phi_i, C_i, \lambda)$
        $[-, \overline{S}_i] \leftarrow \text{OCS}(K_i, T, \boldsymbol{U}^i, \boldsymbol{x}_0, \Sigma, \phi, C, \lambda)$
    end for
    $i^* \leftarrow \underset{i}{\text{argmin}}\ \overline{S}_i, \quad \text{with } i = 0, 1, .., I$
    $\boldsymbol{U}^0 \leftarrow \boldsymbol{U}^{i^*}$
    SendToActuator$(\boldsymbol{u}_0^0)$;
    for $t \in \{1, 2, \ldots, T-1\}$ do
        $\boldsymbol{u}_{t-1}^0 \leftarrow \boldsymbol{u}_t^0$
    end for
    $\boldsymbol{u}_{T-1}^0 \leftarrow \text{Initialize}(\boldsymbol{u}_{T-1}^0)$
end while
```

4. Maritime Application Scenario

Autonomous mobility of maritime systems is a central problem of our time. Due to advances in science and technology, the requirements for the performance are constantly increasing. For example, recent publications such as [18–20] show how NMPC algorithms are used in the maritime environment to solve complex control engineering problems. Moreover, algorithms in the context of RL such as Deep Q-learning [21], Deep Deterministic Policy Gradients [22] or Actor–Critic [23] are increasingly used, which achieve good results for various applications, but the majority of them are only validated in simulation. While MPPI control was used to dock a fully actuated vessel in full-scale autonomously [24], a gap in this research area is the feature-based MPPI control of an autonomous vessel for collision avoidance. In order to also being able to perform autonomous maneuvers, such as heading for a defined target outside the port, it must be ensured that the hazards during these maneuvers are minimized. As regulated by the *International Regulations for Preventing Collisions at Sea* (COLREG) [25], in the event of a predicted collision, it is mandatory to take evasive action. In poor visibility conditions such as fog, heavy rain or due to obstacles appearing through the water surface, there are scenarios where the vessel's sensors will not detect potential collision partners until a short distance away. In such a scenario, using the standard MPPI approach, all sampled trajectories would result in a collision. This overloads the controller and makes it impossible to avoid a collision. Thus, the question arises how feature-based MPPI control can be applied to avoid a collision even in such a scenario. To answer the question, the feature-based MPPI control algorithm is applied in an emergency braking scenario for an autonomous vessel. In the scenario being discussed, the vessel is traveling at full speed along a predefined path,where an obstacle appears at a certain point. Therefore, the vessel has to autonomously perform a so-called last-minute maneuver to avoid a collision. To provide an overview, first the state and dynamics of the fully actuated research vessel *Solgenia* are presented. Then, a standard MPPI controller is parameterized, which causes the vessel to follow a predefined path. This is then extended to a feature-based MPPI controller by introducing an emergency brake feature.

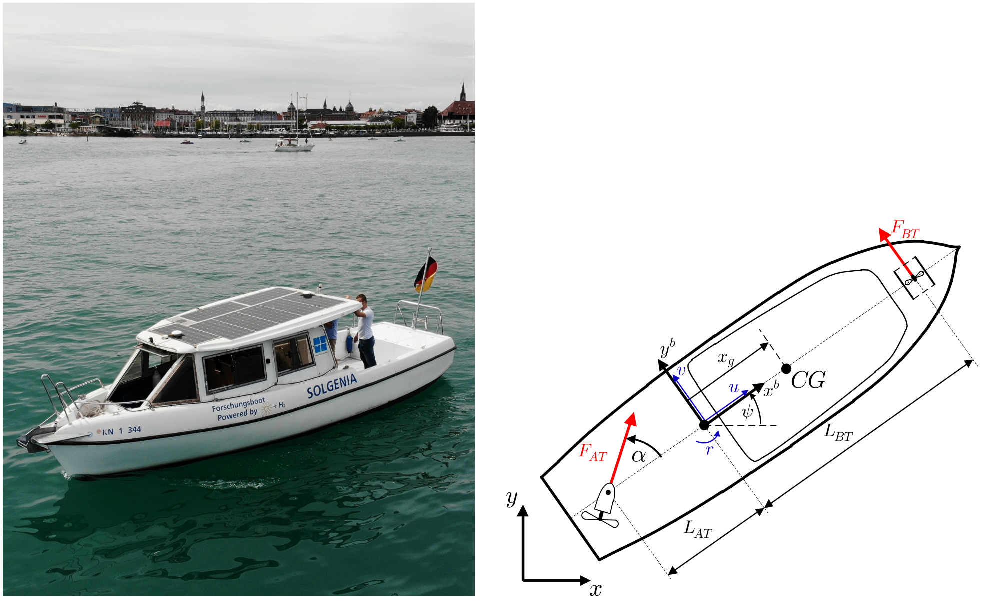
4.1. Dynamics of the Vessel

In [26], a detailed description of the dynamics of the research vessel *Solgenia* including identification of the model parameters and the calculation of the actuator thrusts is outlined. Nevertheless, a short description of the modeling is given in the following since the model is an elementary component of the MPPI approach used in the scenario. A photography of the research vessel *Solgenia* is shown in Figure 2a. According to [27], the vessel's state vector is given by a combination of the 2D pose $\boldsymbol{\eta} = (x \ y \ \psi)^\top$ and the velocity vector $\boldsymbol{\nu} = (u \ v \ r)^\top$ in body-fixed coordinates, shown in Figure 2b. This model was extended in [24] to also consider the dynamics of the actuators. Thus, the transfer properties of the $i = 1, 2, 3$ controlled actuators can be modeled in each case by a first-order low-pass filter with an equivalent time constant T_E, limited slope $\tilde{a}$, and dead time T_T represented by

$$\dot{a}_i(t) = \begin{cases} \tilde{a}_i & \text{if } \frac{a_i(t) - w_i(t - T_{Ti})}{T_{Ei}} > \tilde{a}_i \\ -\tilde{a}_i & \text{if } \frac{a_i(t) - w_i(t - T_{Ti})}{T_{Ei}} < -\tilde{a}_i \\ \frac{a_i(t) - w_i(t - T_{Ti})}{T_{Ei}} & \text{else} \end{cases}, \tag{14}$$

where w_i denotes the i-th actuator's desired state and a_i denotes the i-th actuator's actual state. Thus, the actual actuator states $\boldsymbol{a} = (a_1 \quad a_2 \quad a_3)^\top = (n_{\mathrm{AT}} \quad \alpha \quad n_{\mathrm{BT}})^\top$, where the speed of the azimuth thruster (AT) is denoted by n_{AT}, the orientation of the AT is denoted by α and n_{BT} denotes the speed of the bow thruster (BT), and the desired actuator states $\boldsymbol{w} = (n_{d,\mathrm{AT}} \quad \alpha_d \quad n_{d,\mathrm{BT}})^\top$ denoting the desired variables of $\boldsymbol{a}$ get a part in the system state

$$\boldsymbol{x} = (\boldsymbol{\eta}^\top \quad \boldsymbol{\nu}^\top \quad \boldsymbol{a}^\top \quad \boldsymbol{w}^\top)^\top \in \mathbb{R}^{12}. \tag{15}$$



(**a**) (**b**)

Figure 2. Photography and technical drawing of the research vessel *Solgenia*. (**a**) photography of the research vessel *Solgenia* on Lake Constance in front of the Constance harbor; (**b**) technical drawing of the research vessel *Solgenia* with local and body-fixed coordinate systems, geometrical parameters and thruster forces.

The whole dynamic model of the vessel is given by

$$\begin{aligned} \dot{\boldsymbol{w}} &= \boldsymbol{u} && \text{(16a)}\\ \dot{\boldsymbol{a}} &= \boldsymbol{d}(\boldsymbol{a},\boldsymbol{w}) && \text{(16b)}\\ \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{\mathrm{RB}}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{N}(\boldsymbol{\nu})\boldsymbol{\nu} &= \boldsymbol{\tau}_c(\boldsymbol{a},\boldsymbol{\nu}) + \boldsymbol{\tau}_d && \text{(16c)}\\ \dot{\boldsymbol{\eta}} &= \boldsymbol{J}(\psi)\boldsymbol{\nu}, && \text{(16d)} \end{aligned}$$

where in (16a) the desired variable vector $\boldsymbol{w}$ includes the integral action $\boldsymbol{u}$. In (16b), the actuators' dynamics are considered by $\boldsymbol{d} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$ whose components are given in (14). According to [27], the time derivative of the body-fixed velocity $\dot{\boldsymbol{\nu}}$ is implicitly described in (16c) as a function of the body-fixed velocity $\boldsymbol{\nu}$, the mass matrix

$$\boldsymbol{M} = \begin{pmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - N_{\dot{v}} & J_{comp} \end{pmatrix}, \tag{17}$$

the Coriolis matrix

$$\boldsymbol{C}_{\mathrm{RB}}(\boldsymbol{\nu}) = \begin{pmatrix} 0 & -mr & -mx_g r \\ mr & 0 & 0 \\ mx_g r & 0 & 0 \end{pmatrix} \tag{18}$$

and the damping matrix

$$\boldsymbol{N}(\boldsymbol{\nu}) = \begin{pmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{pmatrix} \tag{19}$$

including the system parameters listed in Table 1, the input vector $\boldsymbol{\tau}_c$ and disturbance vector $\boldsymbol{\tau}_d$. The kinematics equation (16d) describes the transformation of the body-fixed velocity into local coordinates as a function of the rotation matrix $\boldsymbol{J}(\psi)$. The influence of unmodeled effects and environmental disturbances are represented by the disturbance vector $\boldsymbol{\tau}_d$. The controlled force vector

$$\boldsymbol{\tau}_c(\boldsymbol{a},\boldsymbol{\nu}) = \begin{pmatrix} F_{\mathrm{AT}}(n_{\mathrm{AT}},\boldsymbol{\nu})\cos\alpha \\ F_{\mathrm{AT}}(n_{\mathrm{AT}},\boldsymbol{\nu})\sin\alpha + F_{\mathrm{BT}}(n_{\mathrm{BT}},\boldsymbol{\nu}) \\ F_{\mathrm{BT}}(n_{\mathrm{BT}},\boldsymbol{\nu})L_{\mathrm{BT}} - L_{\mathrm{AT}}F_{\mathrm{AT}}(n_{\mathrm{AT}},\boldsymbol{\nu})\sin\alpha \end{pmatrix} \tag{20}$$

depends on the geometric parameters L_{BT}, L_{AT} and α and the thrusts F_{AT} and F_{BT}, shown in Figure 2b. The thrust F_{AT} is generated by the AT, and F_{BT} denotes the thrust generated by the BT. These thrust forces can be modeled according to [28] dependent on various physical constants, the body fixed velocity vector $\boldsymbol{\nu}$ and the actual states of the actuators $\boldsymbol{a}$. Thus, the force generated by the BT is given by

$$F_{\mathrm{BT}} = K_T\, \rho\, d_{\mathrm{BT}}^4\, n_{\mathrm{BT}}\, |n_{\mathrm{BT}}| e^{-c_b u^2}, \tag{21}$$

where K_T denotes a unitless constant with

$$K_T = \begin{cases} d_1 & \text{if} \quad n_{\mathrm{BT}} \geq 0 \\ d_2 & \text{if} \quad n_{\mathrm{BT}} < 0 \end{cases}, \tag{22}$$

ρ denotes the density of the water, the diameter of the propeller is denoted by d_{BT}. Furthermore, the quadratic damping of the thrust dependent on the surge velocity component is scaled by a coefficient denoted by c_b. Using this modeling structure, the influence of the relative speed in the axial direction of the BT is assumed to be small and hence neglected. This assumption cannot be used when modeling the force F_{AT}, since the relative axial velocity component of the AT is large in the given application scenario and given by

$$u_a = u\cos(\alpha) + (v - rL_{\text{AT}})\sin(\alpha), \tag{23}$$

where L_{AT} denotes the distance between the AT and the vessel's center of gravity (CG). According to [26], the force generated by the AT is modeled by

$$F_{\text{AT}} = c_1\,\rho\,d_{\text{AT}}^4\,n_{\text{AT}}\,|n_{\text{AT}}| - c_2\,\rho\,d_{\text{AT}}^3\,u_a\,|n_{\text{AT}}|, \tag{24}$$

where d_{AT} denotes the AT's diameter, and the coefficients c_1 and c_2 are given by

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{cases} (a_1 \;\; b_1)^\top & \text{if} \quad n_{\text{AT}} \geq 0 \wedge u_a \geq 0 \\ (a_1 \;\; 0)^\top & \text{if} \quad n_{\text{AT}} \geq 0 \wedge u_a < 0 \\ (a_2 \;\; b_2)^\top & \text{if} \quad n_{\text{AT}} < 0 \wedge u_a < 0 \\ (a_2 \;\; 0)^\top & \text{if} \quad n_{\text{AT}} < 0 \wedge u_a \geq 0 \end{cases}, \tag{25}$$

where $a_1, a_2, b_1, b_2 \in \mathbb{R}$ denote the coefficients of the AT. The parameters identified by [26] are listed in Table 2. Note that, according to [26], the parameters b_1, b_2 and c_b are eliminated during the evaluation phase of the identification process. For more detailed information about the dynamics, the reader is referred to [24,26,27].

Table 1. Identified vessel parameters of the dynamic model [26] and the parameters of the actuators' models [24].

Vessel's Dynamics Parameter	Value	Actuators' Dynamics Parameter	Value
m	3100 kg	$\lvert n_{\text{AT,max}}\rvert$	2300 rpm
x_g	0 m	$T_{E,1}$	105 ms
J_{comb}	21179 kgm^2	$T_{T,1}$	240 ms
$X_{\dot{u}}$	−155 kg	$\tilde{a}_1$	71.66 1/s^2
$Y_{\dot{v}}$	−1070 kg	$\lvert\alpha_{\max}\rvert$	-
$Y_{\dot{r}}$	−1008 kgm	$T_{E,2}$	95 ms
$N_{\dot{v}}$	−3328 kgm	$T_{T,2}$	160 ms
X_u	−86.5 Ns/m	$\tilde{a}_2$	1.496 rad/s
Y_v	−796 Ns/m	$\lvert n_{\text{BT,max}}\rvert$	3800 rpm
N_r	−5230 Ns/m	$T_{E,3}$	270 ms
Y_r	−896 Ns/m	$T_{T,3}$	80 ms
N_v	−958 Ns/m	$\tilde{a}_3$	10,000 1/s^2

Table 2. Identified parameters of the actuators' thrust models according to [26].

Parameter	Value	Parameter	Value
a_1	0.9047	d_1	0.21
a_2	0.6545	d_2	0.24
d_{AT}	0.36 m	ρ	1000 kg/m^3
d_{BT}	0.23 m	L_{AT}	2.9 m

4.2. Inequality Constraints

Due to the maximum speed of the actuators, two inequality constraints

$$\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u}) = \begin{pmatrix} |n_{\text{AT}}| - |n_{\text{AT,max}}| \\ |n_{\text{BT}}| - |n_{\text{BT,max}}| \end{pmatrix} \leq \boldsymbol{0}, \quad \forall t \in [t_0, t_0 + T], \tag{26}$$

are defined. By considering the actuator dynamics (14) in the model, no further inequality constraints are required.

4.3. Equality Constraints

In the treated scenario, the most important aim is to avoid collisions. In order to achieve this, first the indicator function

$$\zeta(\boldsymbol{\eta}) = \begin{cases} 0 \text{ if } \boldsymbol{\eta} \in \boldsymbol{\eta}_{\text{fine}} \\ 1 \text{ if } \boldsymbol{\eta} \in \boldsymbol{\eta}_{\text{col}} \end{cases} \tag{27}$$

is defined, where $\boldsymbol{\eta}_{\text{col}}$ is the subset of the state-space where the vessel causes a collision and $\boldsymbol{\eta}_{\text{fine}}$ is a disjoint subset. Consequently, the equality constraint to be fulfilled is given by

$$g(\boldsymbol{x}) = \zeta(\boldsymbol{\eta}), \tag{28}$$

where only the indicator function (27) is used. In the next part, the derivation of the cost function is presented for the given scenario.

4.4. Cost Function

The instantaneous state-dependent cost function $C(\boldsymbol{x})$ significantly governs the behavior of the controlled system, since it is used to evaluate the sampled stochastic trajectories. However, for a complex task, the stage cost function can be learned from expert behavior; using inverse reinforcement learning [29–31] for the given scenario, we can define the linguistic criteria for the cost function and then express them as simple equations and subsequently add them up. The vessel should move on a predefined trajectory shown in Figure 3 and should avoid collisions. This task is complicated by significant current, wind and further disturbance effects that occur when moving in the Rhine River. For a good system behavior, the vessel should meet the following criteria, which are sorted in descending order of importance:

1. A collision should be prevented.
2. The actuators must not be overloaded.
3. The position should follow a predefined the trajectory.
4. The orientation should be in line with the trajectory.
5. While the surge velocity component should match a reference, the absolute value of the sway component and yaw-rate component should be minimized.

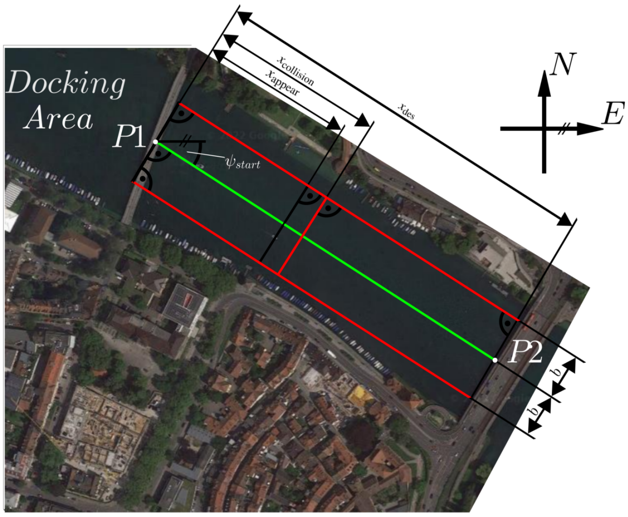


Figure 3. Bird's eye view on the Rhine river between the docking area and the open lake [32]. The locations of $P1$ and $P2$ are specified in Table 3. The desired path is drawn in green. The borders to the inadmissible area next to the river bank are visualized in red. Due to the application scenario, the width and orientation of the river are approximated by $2b = 120$ m and $\psi_{start} = -31.96$ °.

While criteria 1 and 2 are already treated introducing the inequality and equality constraints, to meet criteria 3–5, the quality of sampled trajectories is evaluated based on two parts

$$C(\boldsymbol{x}) = C_{\text{pos}}(\boldsymbol{\eta}) + C_{\text{vel}}(\boldsymbol{\nu}), \tag{29}$$

where $C_{\text{pos}}(\boldsymbol{\eta})$ denotes the position and orientation dependent costs used to meet the criteria 3 and 4, $C_{\text{vel}}(\boldsymbol{\nu})$ denotes the part of the cost function that is dependent on the body-fixed velocity $\boldsymbol{\nu}$ and thus is introduced w.r.t. meeting criterion 5. In the following, the parts of the cost function (29) are presented.

4.4.1. Costs Dependent on the Position

For a clear presentation of the costs dependent on the position, these are defined in transformed local coordinates, which are generated by the linear transformation

$$\overline{\boldsymbol{\eta}} = \begin{pmatrix} \overline{x} & \overline{y} & \overline{\psi} \end{pmatrix}^{\top} = \boldsymbol{J}^{-1}(\psi_{\text{start}})(\boldsymbol{\eta} - \boldsymbol{\eta}_{\text{start}}), \tag{30}$$

where the start orientation of the trajectory is denoted by ψ_{start} and $\boldsymbol{\eta}_{\text{start}}$ denotes the 2D pose located at P1 given in Table 3. Subsequently, the costs dependent on the transformed position are given by

$$C_{\text{pos}}(\overline{\boldsymbol{\eta}}) = c_x(|x_{\text{des}} - \overline{x}|) + c_y|\overline{y}| + c_\psi|\overline{\psi}|, \tag{31}$$

where $c_x, c_y, c_\psi \in \mathbb{R}^+$ denote weighting coefficients, and $x_{\text{des}} \in \mathbb{R}^+$ denotes the distance between $P1$ and $P2$ shown in Figure 3.

Table 3. Significant points specifying the application scenario.

Point	Latitude	Longitude	Description
P1	47.668279218023123	9.174018424704204	Start position
P2	47.666282640256334	9.178596676673507	End position

4.4.2. Costs Dependent on the Velocity

The difference between the body-fixed velocity components and the desired reference velocity is penalized using

$$C_{\text{vel}}(\boldsymbol{\nu}) = c_u|u_{\text{des}} - u| + c_v|v| + c_r|r|, \tag{32}$$

where $c_u, c_v, c_r \in \mathbb{R}^+$ denote the weighting coefficients. Note that this velocity dependent part of the cost function is minimized by the optimal velocity $\boldsymbol{\nu}^* = \begin{pmatrix} u_{\text{des}} & 0 & 0 \end{pmatrix}^{\top}$. Thus, a drifting behavior of the vessel is penalized. In the next section, the resulting problem formulation is given.

4.5. Resulting Problem Formulation

To minimize the cumulated costs (29) subject to the given dynamics (16a)–(16d), for the inequality constraints (26) and the equality constraints (28) using MPPI control, the OCP has to be formulated in the assumed structure (1a)–(1b). Consequently, the equality and inequality constraints must be considered in the cost function with

$$C_{\text{MPPI}}(\boldsymbol{x}) = C(\boldsymbol{x}) + c_{\text{pen}}(\max[\boldsymbol{0}, \boldsymbol{h}(\boldsymbol{x})] + g(\boldsymbol{x})) \tag{33}$$

being defined, where $c_{\text{pen}} \in \mathbb{R}^+$ denotes coefficients of the penalty terms. In addition, the terminal cost function $\phi(\boldsymbol{x})$ must be determined. Because this function is not needed in the given scenario, it is defined as

$$\phi(\boldsymbol{x}) = C_{\text{MPPI}}(\boldsymbol{x}). \tag{34}$$

In MPPI control according to [4], a discrete-time system dynamics with additive input noise is assumed in (1b). Therefore, the time continuous system dynamics given in (16a)–(16d) is discretized using the explicit fourth order Runge–Kutta method with a step size h. Therefore, the input signal is chosen to be piecewise constant within this step size. Consequently, the discrete-time vessel dynamics that maps a system state is given by

$$\boldsymbol{X}_{t+1} = \boldsymbol{F}_v(\boldsymbol{X}_t, \boldsymbol{u}_t). \tag{35}$$

Finally, the assumption that the input of the system is disturbed with time discrete AWGN yields

$$\boldsymbol{u}_t \sim \mathcal{N}(\boldsymbol{u}_{\text{des},t}, \boldsymbol{\Sigma}), \tag{36}$$

where $\boldsymbol{u_t}$ denotes the actual and $\boldsymbol{u}_{\text{des},t}$ the desired system input at time instance t. The covariance matrix of the AWGN is denoted by $\boldsymbol{\Sigma}$. Using these assumptions, the resulting stochastic OCP is given by

$$\min_{\boldsymbol{U}_{des}} \mathbb{E}_{\mathbb{Q}} \left\{ \sum_{t=0}^{T-1} [C_{\text{MPPI}}(\boldsymbol{X}_t) + \boldsymbol{u}_{\text{des},t}^{\top} \boldsymbol{R} \boldsymbol{u}_{\text{des},t}] | \boldsymbol{X}_0 = \boldsymbol{x}_0 \right\} \tag{37a}$$

$$\text{with } \boldsymbol{X}_{t+1} = \boldsymbol{F}_v(\boldsymbol{X}_t, \boldsymbol{u}_t), \quad \forall t \in \{0, 1, \ldots, T-1\}, \tag{37b}$$

$$\text{and } \boldsymbol{u}_t \sim \mathcal{N}(\boldsymbol{u}_{\text{des},t}, \boldsymbol{\Sigma}) \;, \boldsymbol{R} = \lambda \boldsymbol{\Sigma}^{-1},$$

where the sequence of desired inputs is denoted by $\boldsymbol{U}_{\text{des}} = \{\boldsymbol{u}_{\text{des},0}, \boldsymbol{u}_{\text{des},1}, \ldots, \boldsymbol{u}_{\text{des},T-1}\}$. In the next part, the usual MPPI approach is extended to the feature-based MPPI to improve the system's behavior.

4.6. Feature Definition

To use the presented feature-based MPPI algorithm, at least one feature has to be chosen. As already discussed in Section 4.4, the selection of the features can be done by inverse reinforcement learning [30,31] or by the definition of a linguistic quality criterion, which is subsequently formulated mathematically. It is important to note that only well-chosen features lead to an improvement of the system's behavior. In this context, well-chosen means that, in some cases, low-cost regions of the original cost function (33) of state-space are explored by sampling trajectories with controls, which minimizes the feature costs. Note, for the given scenario, the following linguistic quality criterion is defined: a reduction in speed can mitigate or even prevent a collision with an obstacle. For this purpose, an emergency break feature is defined by choosing

$$C_1(\boldsymbol{x}) = \phi_1(\boldsymbol{x}) = \boldsymbol{\nu}^{\top} \boldsymbol{Q} \boldsymbol{\nu}, \quad \boldsymbol{Q} = \text{diag}(c_{\text{F}u}, c_{\text{F}v}, c_{\text{F}r}), \tag{38}$$

where $c_{\text{F}u}, c_{\text{F}v}, c_{\text{F}r} \in \mathbb{R}^+$ denote the coefficients of the velocity dependent costs. Consequently, a control sequence that minimizes this feature leads to the exploration of the region of the state-space decreasing the vessel's velocity. In the following section, the presented scenario is used to compare the performance of classical MPPI and feature-based MPPI control.

5. Controller Design

In this section, the design for both the standard MPPI controller presented in Section 2 and the feature-based MPPI controller presented in Section 3 is shown with application to maritime scenario presented in Section 4. For this purpose, the modularized and distributed control structure used is described. Note that an Unscented Kalman Filter (UKF) according to [33] is used to estimate the system state based on the sensor signals and the actuators' states. These are controlled in subordinate control loops. This is followed by a description of how the control parameters are determined. Finally, the selection of the cost function's coefficients is presented.

5.1. Architecture

An overview of all parts of the used control architecture is given in Figure 4. The state estimation and the communication are performed by the rapid prototyping system *MicroAutoBox2* (MABX2). While the supervisory MPPI algorithm is running on the MABX3, three subordinate 1232SE control units by *Curtis* are used to control the drives. An RTK-GPS and an IMU sensor send their measurement data to the MABX2 using a UDP protocol. Using a UKF, this information is fused with the measured actuator state $\boldsymbol{a}$ to estimate the system's 2D pose denoted by $\hat{\boldsymbol{\eta}}$, the body-fixed velocity vector $\hat{\boldsymbol{\nu}}$ and the disturbance forces $\hat{\boldsymbol{\tau}}_{\mathrm{d}}$, which are sent to the MABX3 via UDP protocol. The MPPI algorithm uses the provided state estimation to calculate the desired system input $\boldsymbol{u}$. Because the system input is defined as the zero-order-hold time derivative of the desired values of the actuator control loops, an explicit Euler integrator can be used to calculate $\boldsymbol{w}$ implemented on the MABX3. Subsequently, $\boldsymbol{w}$ is sent via UDP protocol to the MABX2, which then provides $\boldsymbol{w}$ to the subordinated actuator controllers via CAN protocol. The encoders' measurements of $\boldsymbol{a}$ are denoted by $\boldsymbol{a}_{\mathrm{m}}$ and collected with high quality; therefore, no additional signal processing is needed. The measured actuator states are sent from the encoders to the 1232SE drive controllers, which then send them via UDP to the MABX2 for state estimation.

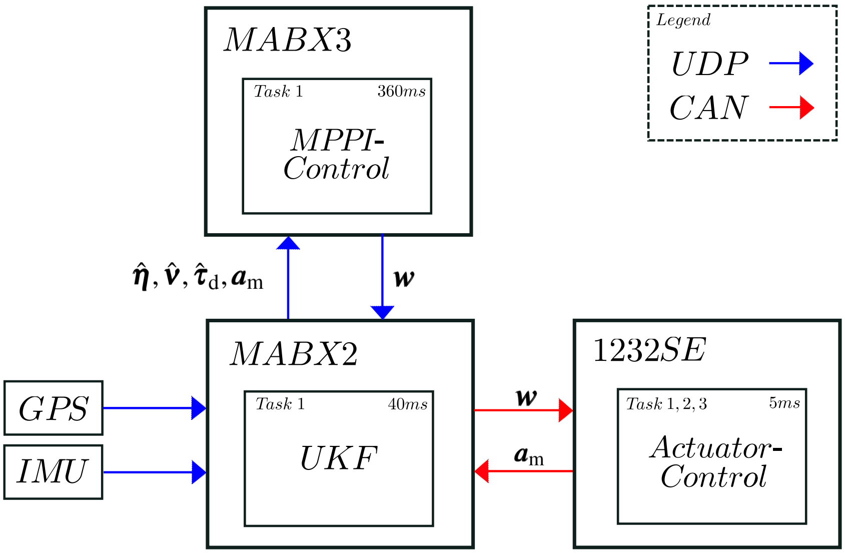

MABX3
Task 1
360ms
MPPI-Control
Legend
UDP
CAN
$\hat{\boldsymbol{\eta}}, \hat{\boldsymbol{\nu}}, \hat{\boldsymbol{\tau}}_{\mathrm{d}}, \boldsymbol{a}_{\mathrm{m}}$
$\boldsymbol{w}$
MABX2
Task 1
40ms
UKF
GPS
IMU
$\boldsymbol{w}$
$\boldsymbol{a}_{\mathrm{m}}$
1232SE
Task 1, 2, 3
5ms
Actuator-Control

Figure 4. Distributed and modularized control architecture for full-scale experiments on research vessel *Solgenia*. The connections are realized via UDP and CAN protocol.

5.2. Controller Parameters

The choice of controller parameters determines the properties of the sampled trajectories, which are realizations over $\mathbb{Q}$. They have a significant influence on the quality of the estimation of the optimal control sequence.

5.2.1. Numerical Solution of the Initial Value Problem

As already discussed, the corresponding initial value problem can be solved numerically using the explicit fourth order Runge–Kutta method with a step size of $h = 40$ ms to sample the trajectories. It should be noted that, in principle, there is no information about the future course of the disturbance variables. Due to the river current, the disturbance vector $\boldsymbol{\tau}_d$ is assumed to be constant within the predicted horizon.

5.2.2. Choice of Controller Step Size

The controller step size denoted by T_{CTRL} should be selected as large as possible to reduce the dimension of the control sequence in the optimization problem, but also small enough to be able to induce a desired system behavior. Furthermore, it has to be considered that T_{CTRL} is a multiple of $h = 40$ ms. Simulations have shown that good performance can be achieved with $T_{CTRL} = 360$ ms.

5.2.3. Prediction Horizon

The dimension of the control sequence, as well as the influence of the inaccuracies of the model, increases with rising prediction horizon denoted by T. However, a sufficiently large prediction horizon is essential for evaluating the result of the choice of control sequence. Based on the vessel's dynamics, the prediction horizon is determined based on the time until the maximum surge velocity is achieved. To reach $u_{max} = 3$ m/s with a maximal acceleration of 0.2 m/s^2, a duration of 15 s is needed and thus a prediction horizon of $T = 41$ time steps is chosen.

5.2.4. Covariance Matrix of Additive Noise

In path integral control framework [12], the choice of the covariance matrix $\mathbf{\Sigma}$ of the AWGN, the quadratic costs of the input is determined by $\mathbf{\Sigma} = \lambda \boldsymbol{R}$. The quadratic penalty term $\boldsymbol{u}^\top \boldsymbol{R} \boldsymbol{u}$ relates to the accelerations of the propellers and the rotational velocity of the AT. We assume uncorrelated noise processes to excite the system's behavior and thus the covariance matrix

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_{\mathrm{AT}}^2 & 0 & 0 \\ 0 & \sigma_\alpha^2 & 0 \\ 0 & 0 & \sigma_{\mathrm{BT}}^2 \end{pmatrix} \tag{39}$$

is chosen. The elements of $\mathbf{\Sigma}$ are chosen proportionately to the corresponding actuator dynamics listed in Table 1. While the maximum dynamics of the AT is given directly, the maximum acceleration of the BT is calculated using a first order Taylor approximation of the delay time, this yields an equivalent time constant of $T_T + T_E = 350$ ms. Thus, the variances of the AWGN are given by

$$(\sigma_{\mathrm{AT}}^2 \;\; \sigma_\alpha^2 \;\; \sigma_{\mathrm{BT}}^2) = c\Big(71.66\ \mathrm{s}^{-4} \;\; 1.4966\ \mathrm{rad}^2/\mathrm{s}^2 \;\; 188.57\ \mathrm{s}^{-4}\Big), \tag{40}$$

where parameter c is introduced to scale the variances. Regarding the explore-exploit dilemma [12], the optimal value $c^* = 0.09$ is approximated by numerically solving the minimization problem

$$c^* = \arg\min_c \sum_{k=0}^{T_{\mathrm{sim}}} [C_{\mathrm{MPPI}}(\tilde{\boldsymbol{x}}_k) + \tilde{\boldsymbol{u}}_k^\top \boldsymbol{R} \tilde{\boldsymbol{u}}_k], \tag{41}$$

where T_{sim} denotes the number of simulation steps, $\tilde{\boldsymbol{u}}_k$ denotes the approximated optimal system input using MPPI control and $\tilde{\mathbf{x}}_k$ denotes the resulting state at the time instance k with $\tilde{\mathbf{x}}_{k+1} = \boldsymbol{F}(\tilde{\mathbf{x}}_k, \tilde{\boldsymbol{u}}_k)$. Using the initial state $\overline{\boldsymbol{\eta}}_0 = \boldsymbol{\nu}_0 = \boldsymbol{a}_0 = \boldsymbol{w}_0 = \mathbf{0}$ and $T_{\mathrm{sim}} = 3T = 123$ simulation steps c^* is found, which parameterizes the MPPI algorithm leading to minimal cost of the scenario. Note that, using the MPPI update law (11), the noise of the AWGN and thus its scaling factor c has a huge influence of the approximated optimal controls $\tilde{\boldsymbol{u}}$.

5.2.5. Temperature

According to [12], the temperature λ with

$$\mathbf{R} = \lambda \mathbf{\Sigma}^{-1} \tag{42}$$

scales the quadratic input costs matrix $\boldsymbol{R}$. The relation between $\mathbf{R}$ and C_{MPPI} has a significant impact on the resulting system behavior. However, C_{MPPI} has already been scaled freely. Thus, $\lambda = 1$ has been chosen.

5.2.6. Number of Predicted Trajectories

Using the MPPI update law (11), the expected value is approximated by the mean over the sampled trajectories. Thus, the quality of the approximation increases with the number of drawn samples used. However, the required computing time also increases with the number of realizations. A buffer $t_{\text{buf}} > 15$ ms is targeted to make the real-time capability of the controller robust. Empirically determined, one core of the TI AM5K2E04 processor with four ARM Cortex-A15 cores 1.4 GHz on the rapid prototyping system MABX3 requires 343 ms to simulate and evaluate 9000 trajectories. Thus, the real-time capability of the controller with a buffer $t_{\text{buf}} = 17$ ms > 15 ms is ensured. The choice of all controller parameters determined in this subsection. To provide a structured overview, the parameters are listed in Table 4. Note that, for an objective comparison of the standard MPPI and the feature-based MPPI approaches, the parameters are chosen to be equal.

Table 4. Controller parameters of the standard MPPI and the feature-based MPPI approaches.

Parameter	MPPI	Feature-Based MPPI
K	9000	8500
T	41	41
c^*	0.0167	0.0167
λ	1	1
$C(\boldsymbol{X_t})$	Equation (33)	Equation (33)
$\phi(\boldsymbol{X_T})$	0	0
K_1	-	500
$C_1(\boldsymbol{X_t})$	-	Equation (38)
$\phi_1(\boldsymbol{X_T})$	-	Equation (38)

5.2.7. Cost Function Parameters

The structure of the cost function (33) and the structure of the feature costs (38) are already derived in previous sections. Since their coefficients have a huge influence on the behavior of the system, they must be determined with a similar diligence as the control parameters. The units of the coefficients are chosen in such a way that the corresponding product is unitless. With respect to the parameters c_x, c_y and c_ψ, which are associated with the positions, reaching the desired position $x_{\text{des}} = 409.9$ m and keeping the desired orientation is weighted more than leaving the path orthogonally. Thus, $c_x = 500$ 1/s > $c_y = 200$ 1/s and $c_\psi = 2500$ 1/rad are chosen. With reference to $c_u = 5000$ s/m, c_v =10 s/m and $c_r = 7500$ s/rad, which are assigned to the velocity components, errors of the surge speed w.r.t. the desired surge speed $u_{\text{des}} = 2$ m/s and yaw rate is weighted more heavily than the occurrence of a sway speed component. The feature cost coefficients with $c_{\text{F}u} = 300$ s/m, $c_{\text{F}v} = 30$ s/m and $c_{\text{F}r} = 10$ s/rad are chosen to weight the surge speed component by far the most, since it can be assumed that this component dominates in the application scenario. Subsequently, it is assumed that there is an obstacle which in the case of $\overline{x} > x_{\text{collision}} = 80$ m would cause a collision. Furthermore, it is assumed that the vessel detects the obstacle if its position is $\overline{x} > x_{\text{appear}} = 60$ m. An overview of the coefficients is given in Table 5.

Table 5. Coefficients of the costs (33) and the feature costs (38).

Parameter	Value	Parameter	Value
c_x	500 1/m	x_{des}	409.9 m
c_y	200 1/m	u_{des}	2 m/s
c_ψ	2500 1/rad	$c_{\text{F}u}$	300 s/m
c_u	5000 s/m	$c_{\text{F}v}$	30 s/m
c_v	10 s/m	$c_{\text{F}r}$	10 s/rad
c_r	7500 s/rad	x_{appear}	60 m
c_{pen}	1×10^8	$x_{\text{collision}}$	80 m

6. Simulation Results

The MPPI controller presented in Section 2 and its feature-based extension introduced in Section 3 are embedded in a simulation environment to compare their performances when controlling the vessel model in the maritime application scenario described in Section 4. Therefore, the parameters derived in Section 5 are used. The vessel model including the actuator dynamics, the UKF according to [33], the MPPI controller and the feature-based MPPI controller are implemented in a simulation environment. According to [33], a constant river current is assumed with a velocity of 0.45 m/s and an orientation of 200° to the $\overline{x}$-axis. The simulation time is determined to 60 s. To compare the vessel's behavior, both the standard MPPI approach and subsequently the feature-based MPPI approach are used to control the vessel. The resulting trajectories are compared in Figure 5. Furthermore, to visualize the orientation of the vessel along the trajectory, the vessel's contour is plotted at the time instances $t_i = i(\Delta T)$ with $i = \{0, 1, 2, 3, 4, 5, 6\}$ and $\Delta T = 10$ s.

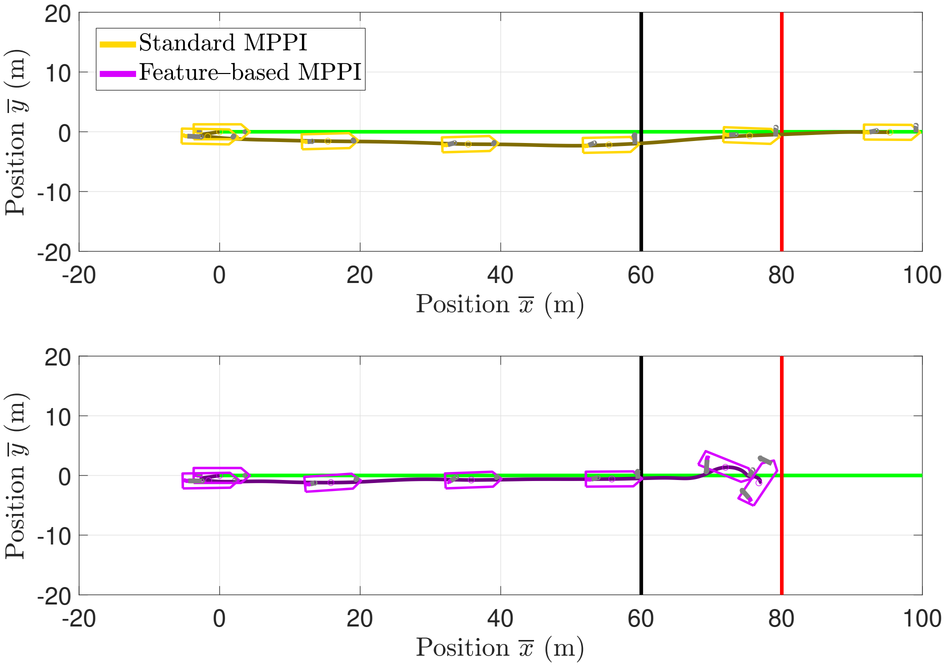


Figure 5. Comparison of the resulting trajectories of the vessel in the simulation environment. While the upper plot shows the performance of standard MPPI control, the lower plot shows the performance of the feature–based MPPI control approach. Both plots use transformed coordinates calculated by (30). The green lines visualize the reference trajectories in both plots. After passing the black line, the vessel detects the obstacle visualized as a red line.

In the absence of an obstacle, both trajectories are quite similar in orientation and position. Note the position difference in $\overline{y}$ occurs since it is only penalized slightly in (1a). After the obstacle appears, the different control approaches cause different behavior. While the system controlled by the standard MPPI controller causes a collision with the obstacle, the feature-based MPPI control approach can prevent this collision by performing a braking maneuver. The time courses of the body-fixed surge velocity and the vessel's velocity in the direction of the $\overline{x}$-axis are shown in Figure 6. In the absence of the obstacle for $t \leq 44$ s, the plotted velocity components show the same behavior. After a learning phase of about 5 s immediately after activating the controllers, the vessel accelerates till it reaches the desired surge velocity. After the acceleration phase, the surge velocity is kept exact at 2 m/s. For $t > 44$ s using the feature-based approach, the vessel decelerates and reaches $\dot{x} = 0$ m/s at $t = 59$ s. A comparison of the desired and the actual actuator values is given in Figure 7. Furthermore, in Figure 8, the caused costs using the different approaches are plotted over time. In contrast, using the standard MPPI controller, the vessel's velocity does not change after the vessel appears. This effect results from the fact that, due to the small explored state-space using conventional MPPI control, all predicted trajectories cause a collision for $t > 44$ s.

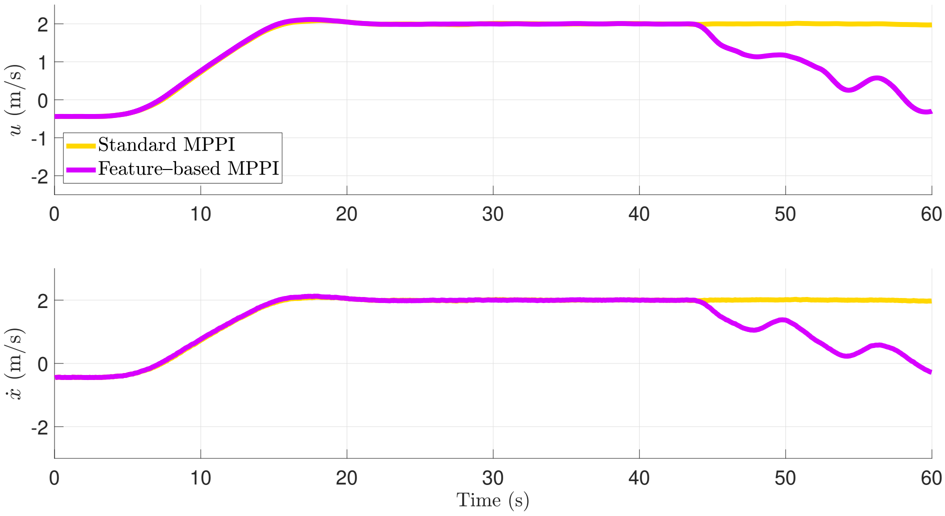


Figure 6. Comparison of the body–fixed surge velocity u and the velocity $\dot{x}$ in simulation.

Consequently, the vessel controlled by the standard MPPI approach collides with the obstacle at full speed at $t = 52.5$ s. Regarding the input trajectories shown in Figure 7, the desired and actual actuator values are well fitting. Furthermore, in absence of an obstacle for $t \leq 44$ s, the input trajectories show nearly the same behavior. To keep the surge velocity at 2 m/s, the AT is driven with about 950 rpm after it reaches its maximal value during the acceleration phase. While the AT is used to reach and keep the desired velocity, the BT is used to compensate the disturbances due to the river current. Using the feature-based MPPI approach for $t > 44$ s, the actuators' trajectories lead to the braking behavior of the system.

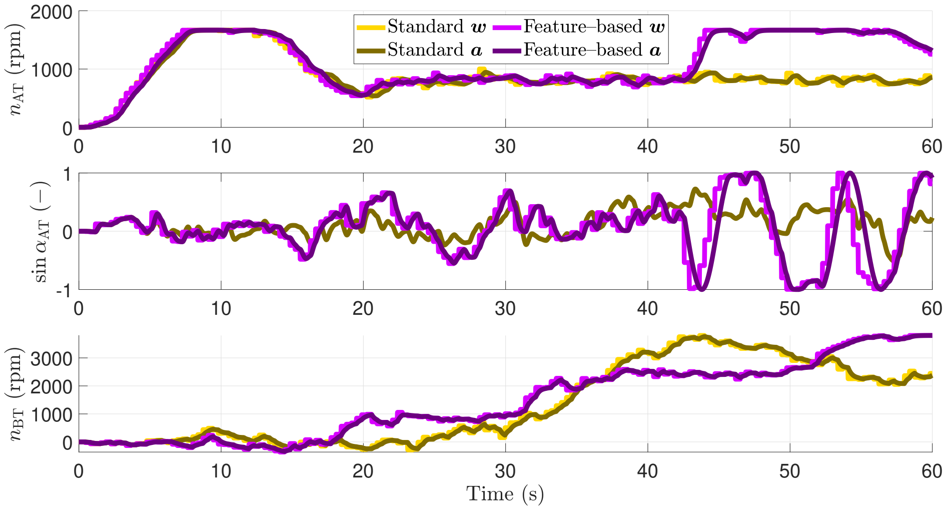


Figure 7. Comparison of the resulting input trajectories of the vessel in the simulation environment.

To reach the maximal deceleration of the vessel, the AT's orientation is inverted while its speed is maximized. Regarding the time series of the stage costs shown in Figure 8 in the absence of an obstacle for $t \leq 44$ s, both approaches cause the same costs. Both controllers find inputs leading to decreasing costs. After the obstacle is detected, the feature-based control approach applies the learned inputs corresponding to the defined feature. These inputs lead to increasing cost due to the decelerating surge velocity; however, in contrast to the standard MPPI approach, a collision is prevented and thus being in the high cost region of state-space is avoided. In this section, it was shown that, using the feature-based MPPI control approach, a significant improvement of the vessel behavior in a maritime application scenario is reached.

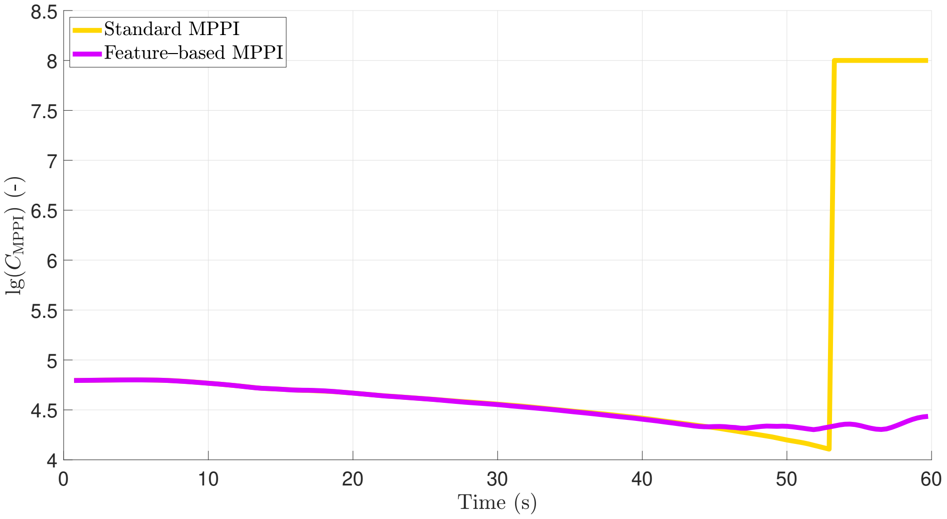


Figure 8. Comparison of costs in the simulation scenarios plotted over time. Note the logarithmic representation of the plotted costs.

7. Full-Scale Results

To validate the simulation results, full-scale scale experiments are used. For this purpose, the algorithms used in the simulation will be transferred to the rapid prototyping systems of the research vessel *Solgenia*. A distributed structure of the components is used as already described in Section 5. In addition, the same controller parameters and cost functions are used as in the previous sections. The data are recorded directly during the experiments on the vessel. Regarding the disturbances immediately before the experiments, a river current of 0.46 m/s was determined in the vicinity of the trajectory. This current has an orientation of about $\psi_{curr} = 165\,^{\circ}$ and is thus almost opposite to the movement of the vessel on the trajectory. However, it should be noted that, although the current values were measured by allowing the research vessel to drift uncontrolled in the vicinity of the trajectory, they vary along the river and can therefore only be regarded as rough estimates, which, in reality, are overlaid by a stochastic component. In the experimental phase, the task described in Section 4 was executed once with a standard MPPI controller and once with a feature-based MPPI controller. For this purpose, the vessel's position is stabilized at the beginning of the trajectory using feedback linearization according to [33] until the MPPI controller is activated. Then, the MPPI controller is activated for 60 s because both a collision or a collision avoidance maneuver could occur within this time span. In Figure 9, a comparison of the vessel's behavior is shown for both experiments. In this figure, the vessel's trajectories are plotted in local coordinates. Furthermore, the vessel's contour and thus its orientation are shown at the time instances $t_i = i(\Delta T)$, with $i = \{0, 1, 2, 3, 4, 5, 6\}$ and $\Delta T = 10$ s. While using the standard MPPI control visualized in the upper plot of Figure 9, the vessel collides with the obstacle; using feature-based MPPI control, a collision can be prevented. Compared to the simulation results presented in the previous section, in the full-scale experiments, the vessel tracks the reference trajectory with higher errors. This behavior is to be expected due to model inaccuracies and the disturbance vector assumed to be constant within the prediction horizon.

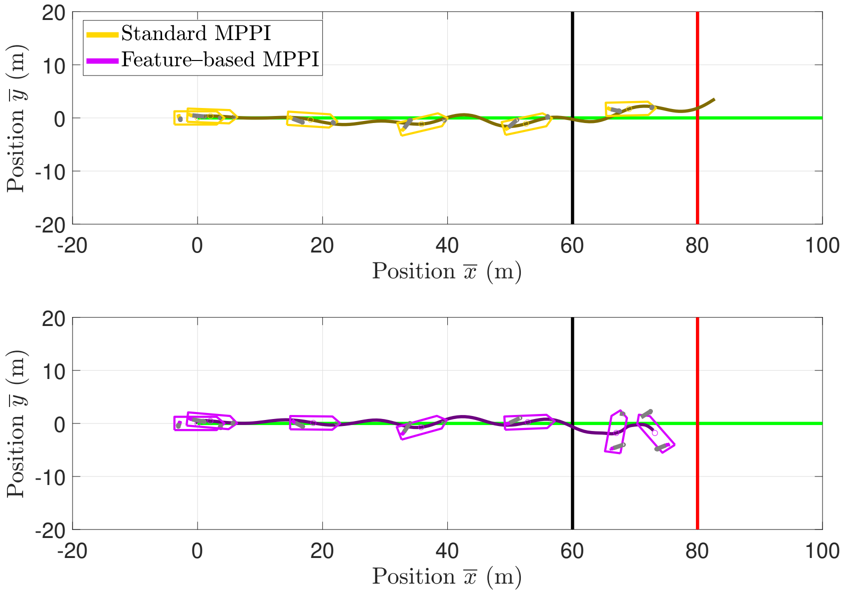


Figure 9. Comparison of the resulting trajectories of the vessel in full–scale experiments at Rhine river in Constance. Both plots use transformed coordinates calculated by (30). The green lines visualize the reference trajectories in both plots. After passing the black line, the vessel detects the obstacle visualized as a red line.

In the application example described, special focus is placed on the surge velocity component, since it is the most weighted in the selection of the coefficients of the cost function and thus the most important property in the case of the absence of an obstacle. However, if an obstacle is detected in this scenario, it is interesting to see how the speed develops in the direction of this obstacle. Thus, the surge velocity in body-fixed frame denoted by u and the velocity in direction of the obstacle denoted by $\dot{x}$ are shown in Figure 10. The corresponding actuators' signals are shown in Figure 11. The minimal path costs β are plotted over time in Figure 12 in logarithmic scale. Using the described subordinated actuator control loops, the actual values of the actuators follow the desired values pretty good. Both the conventional MPPI controlled system and the feature-based MPPI controlled system show almost the same behavior for $t \leq 44.6$ s in the absence of an obstacle.

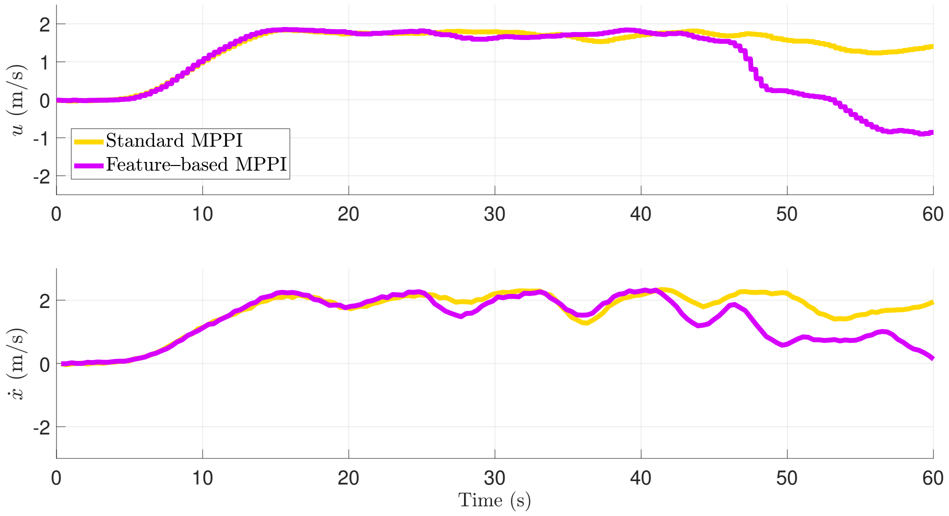


Figure 10. Comparison of the body–fixed surge velocity u and the velocity $\dot{x}$ in $\overline{x}$ direction in local frame.

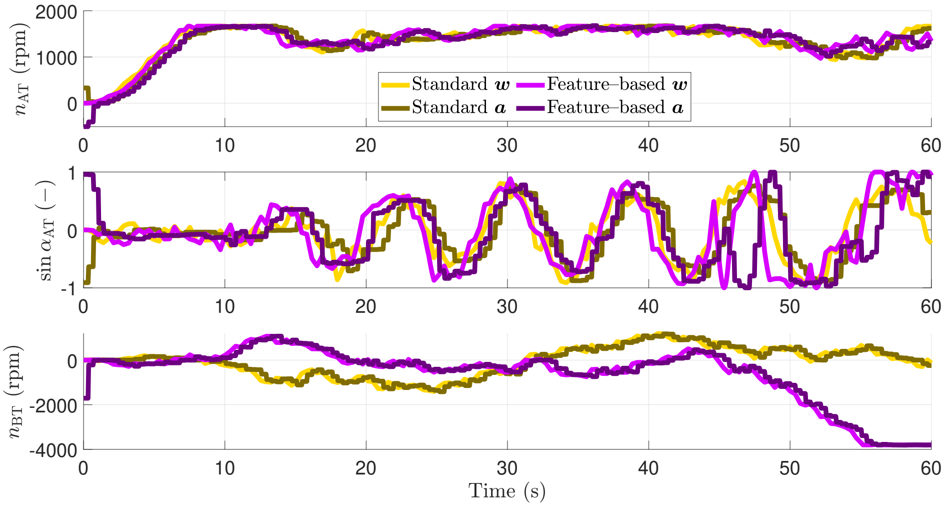


Figure 11. Comparison of the resulting input trajectories of the vessel in the full–scale experiments.

Standalone paragraphs require initial indentation, i change to indentation, please confirm At $t = 5.1$ s, the surge velocity increases until a maximum is reached with $u = 1.93$ m/s at $t = 15.5$ s. After that, the surge velocity is kept at about 1.9 m/s by both controllers. It should be noted that both controllers achieve a good response to the disturbances. The difference between the desired and actual surge velocity is assumed to be due to the limitation of the propeller speed of the AT and model inaccuracies in this state region of high speed. At $t = 44.6$ s, the vessel collides with the obstacle within the prediction horizon. The standard MPPI controller does not decelerate after the detection of the obstacle. This effect results from the fact that, due to the small explored state-space using conventional MPPI control, all predicted trajectories cause a collision for $t > 51.2$ s. Consequently, predicted trajectories with surge velocity close to u_{des} cause lower costs. In contrast, the feature-based MPPI controlled system explores more important areas of the state-space shown in Figure 1.

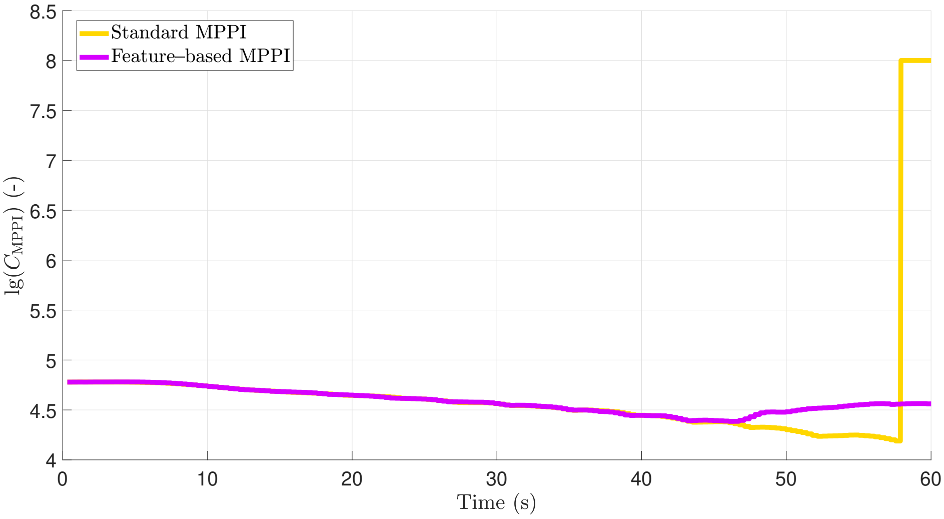


Figure 12. Comparison of costs plotted over time in full-scale experiments. Note the logarithmic representation of the cost.

This allows the feature-based MPPI controller to apply the learned braking trajectory, which is optimal regarding the emergency braking feature (38), as soon as the obstacle is detected. Thus, it can also react excellently for $t > 44.6$ s and brake in time to avoid a collision. As can be seen in Figure 9, braking is achieved by counterclockwise rotation of the vessel, which allows the hydrodynamic effects of the vessel to be utilized in addition to the thruster forces of the propellers. As a consequence, while using the standard MPPI controller, the vessel's velocity in the direction of the obstacle is 1.85 m/s at the moment hitting the obstacle; using the feature-based MPPI approach, the feature-based MPPI controller decelerates the velocity in the direction of the obstacle down to 0.05 m/s and thus avoids a collision. The minimum path costs, which are shown in Figure 12, are equal due to the equivalent system behavior of the two control approaches up to the detection of the obstacle. Using the standard MPPI approach after the detection, a low-cost region of state-space next to η_{col} is explored. However, this leads to the fact that, already for $t > 51$ s, all predicted trajectories cause a collision. In contrast, using feature-based MPPI control, directly after the obstacle detection, the explored state-space is located at a region around the braking trajectory. This yields to higher costs, but even prevent the real high state-dependent costs caused by a collision. Using an MPC approach, the consideration of costs directly provides a quantitative quality criterion. Thus, if the cost curves in Figures 8 and 12 are compared, it can be seen that the quality with regard to the cost function to be minimized is almost identical. In this section, it was shown that feature-based MPPI control in full-scale conditions can greatly expand and improve the behavior of a vessel and prevent collision without increasing the computational cost.

8. Conclusions

In this paper, an extension of the MPPI algorithm [4] is presented improving the sample efficiency by a knowledge-based feature definition. This is the first possibility to incorporate information subject to a scenario additive to the usual cost function into the control algorithm. The conventional MPPI and the presented feature-based MPPI control approaches are compared in a collision avoidance scenario. Therefore, a stochastic nonlinear OCP for enabling a vessel to follow a track and avoid collisions is presented and subsequently solved with both approaches. The performances of the algorithms are evaluated using data out of a simulation environment and full-scale experiments. In both simulation and full-scale experiments, while the standard MPPI would collide with an obstacle, the feature-based MPPI can prevent the collision although the same number of trajectories were drawn in both cases. In the absence of an obstacle, the feature-based MPPI controller learns an optimal braking trajectory that minimizes the feature costs subject to the modeled system dynamics. When the obstacle is detected, this is then successfully applied to avoid a collision. Thus, by inserting an emergency braking feature, the performance and usability of the algorithm could be increased significantly. Concerning other NMPC approaches, the MPPI control algorithm is characterized by a very good parallelization capability, since the samples can be drawn in different kernels. This property is not affected by the presented feature-based extension. Thus, the number of features that could be explored and learned can be increased by adding further parallel computing power. In future work, an evaluation of how the presented feature-based approach can be used in other sample-based NMPC algorithms could be scientifically valuable.

Author Contributions: Conceptualization, H.H., S.W. and J.R.; methodology, H.H.; software, H.H. and S.W.; validation, H.H. and S.W.; formal analysis, H.H.; investigation, H.H.; resources, J.R. and S.W.; data curation, H.H. and S.W.; writing—original draft preparation, H.H.; writing—review and editing, H.H., S.W., M.D. and J.R.; visualization, H.H.; supervision, S.W., M.D. and J.R.; project administration, J.R. All authors have read and agreed to the published version of the manuscript.

Funding: The article processing charge was funded by the Baden-Württemberg Ministry of Science, Research and Culture and the HTWG Konstanz—University of Applied Sciences in the funding programme Open Access Publishing.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The recorded data from simulations and real experiments, which are presented in this paper, can be downloaded using the link https://www.htwg-konstanz.de/fileadmin/pub/ou/isd/Regelungstechnik/Regelungstechnik_Data/MDPI_Machines22_Feature_Based_MPPI_Data_Simulation_and_Experiment.zip. Accessed on 23 May 2022.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AWGN	Additive white Gaussian noise
CG	Center of gravity
HJB	Hamilton–Jacobi–Bellman
MABX	MicroAutoBox
MPPI	Model predictive path integral
NMPC	Nonlinear model predictive control
OCP	Optimal control problem
RL	Reinforcement learning
UKF	Unscented Kalman filter

References

1. Homburger, H.; Wirtensohn, S.; Reuter, J. Feature-Based Proposal Density Optimization for Nonlinear Model Predictive Path Integral Control. In Proceedings of the 6th IEEE Conference on Control Technology and Applications (CCTA), Trieste, Italy, 22–25 August 2022.
2. Kleinert, H. *Path Integrals in Quantum Mechanics, Statistics, Polymer Physics, and Financial Markets*, 5th ed.; World Scientific Publishing Ltd.: Singapore, 2009.
3. Kappen, H.J. Path Integrals and Symmetry Breaking for Optimal Control Theory. *J. Stat. Mech. Theory Exp.* **2005**, *2005*, P11011. [CrossRef]
4. Williams, G.; Wagener, N.; Goldfain, B.; Drews, P.; Rehg, J.; Boots, B; Theodorou, E. Information Theoretic MPC for Model-Based Reinforcement Learning. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May 2017–3 June 2017.
5. Schaal, S.; Atkeson, C. Learning Control in Robotics. *IEEE Robot. Autom. Mag.* **2010**, *10*, 20–29. [CrossRef]
6. Theodorou, E.A.; Buchli, J.; Schaal, S. A Generalized Path Integral Control Approach to Reinforcement Learning. *J. Mach. Learn. Res.* **2010**, *11*, 3137–3181.
7. Kappen, H.J.; Ruiz, H. Adaptive Importance Sampling for Control and Inference. *J. Stat. Phys.* **2016**, *162*, 1244–1266. [CrossRef]
8. Gómez, V.; Thijssen, S.; Symington, A.; Hailes S.; Kappen, H.J. Real-Time Stochastic Optimal Control for Multi-agent Quadrotor Systems. In Proceedings of the 26th International Conference on Automated Planning and Scheduling (ICAPS 16), 12–17 June 2016, London, UK, 2015.
9. Homburger, H.; Wirtensohn, S.; Reuter, J. Swinging Up and Stabilization Control of the Furuta Pendulum using Model Predictive Path Integral Control. In Proceedings of the 30th Mediterranean Conference on Control and Automation (MED), Athens, Greece, 28 June–1 July 2022.
10. Theodorou, E.A.; Todorov, E. Relative entropy and free energy dualities: Connections to path integral and KL control. In Proceedings of the IEEE 51st Annual Conference on Decision and Control (CDC), Grand Wailea Maui, HI, USA, 10–13 December 2012; pp. 1466–1473.
11. Theodorou, E.A. Nonlinear Stochastic Control and Information Theoretic Dualities: Connections, Interdependencies and Thermodynamic Interpretations. *Entropy* **2015**, *17*, 3352–3375. [CrossRef]
12. Kappen, H.J. An Introduction to Stochastic Control Theory, Path Integrals and Reinforcement Learning. *Coop. Behav. Neural Syst. Ninth Granada Lect.* **2007**, *887*, 149–181.
13. Thijssen S.; Kappen, H. J. Path Integral Control and State Dependent Feedback. *Phys. Rev. E* **2015**, *91*, 032104. [CrossRef] [PubMed]
14. Gandhi, M.S.; Vlahov, B.; Gibson, J.; Williams, G.; Theodorou, E. Robust Model Predictive Path Integral Control: Analysis and Performance Guarantees. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1423–1430. [CrossRef]
15. Yin, J.; Zhang, Z.; Theodorou, E.; Tsiotras, P. Improving Model Predictive Path Integral using Covariance Steering. *arXiv* **2021**, arXiv:2109.12147.
16. Kusumoto, R.; Palmieri, L.; Spies, M.; Csiszar A.; Arras, K.O.; Informed Information Theoretic Model Predictive Control. In Proceedings of the International Conference on Robotics and Automation (ICRA), Montreal, Canada, 20–24 May 2019; pp. 2047–2053.
17. Williams, G.; Drews, P.; Goldfain, B.; Rehg, J.M.; Theodorou, E. Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving. *IEEE Trans. Robot.* **2018**, *34*, 1603–1622. [CrossRef]
18. Abdelaal, M.; Hahn, A. NMPC-based trajectory tracking and collision avoidance of unmanned surface vessels with rule-based colregs confinement. In Proceedings of the IEEE Conference on Systems, Process and Control (ICSPC), Hammamet, Tunisia, 16–18 December 2016; pp. 23–28.
19. Lutz, M.; Meurer, T. Optimal trajectory planning and model predictive control of underactuated marine surface vessels using a flatness-based approach. *arXiv* **2021**, arXiv:2101.12730.

20. Bärlund, A.; Linder, J.; Feyzmahdavian, H.R.; Lundh, M.; Tervo, K. Nonlinear MPC for combined motion control and thrust allocation of ships. In Proceedings of the 21st IFAC Wolrd Congress, Berlin, Germany, 11–17 July 2020.
21. Zare, N.; Brandoli, B.; Sarvmaili, M.; Soares, A.; Matwin, S. Continuous Control with Deep Reinforcement Learning for Autonomous Vessels. *arXiv* **2021**, arXiv:2106.14130.
22. Martinsen, A.; Lekkas, A. Curved Path Following with Deep Reinforcement Learning: Results from Three Vessel Models. In Proceedings of the IEEE Oceans MTS, Charleston, SC, USA, 22–25 October 2018; pp. 1–8.
23. Yin, Z.; He, W.; Yang, C.; Sun, C. Control Design of a Marine Vessel System Using Reinforcement Learning. *Neurocomputing* **2018**, *311*, 353–362. [CrossRef]
24. Homburger, H.; Wirtensohn, S.; Reuter, J. Docking Control of a Fully-Actuated Autonomous Vessel using Model Predicitve Path Integral Control. In Proceedings of the 20th European Control Conference (ECC), London, UK, 12–15 July 2022.
25. Lloyd. *Articles of the Convention on the International Regulations for Preventing Collisions at Sea, 1972*; Lloyd's Register of International Maritime Organization: London, UK, 2005.
26. Kinjo, L.M.; Wirtensohn, S.; Reuter, J.; Menard, T.; Gehan, O. Trajectory tracking of a fully-actuated surface vessel using nonlinear model predictive control. In Proceedings of the 13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles (CAMS), Oldenburg, Germany, 22–24 September 2021; pp. 51–56.
27. Fossen, T.I. *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*, 1st ed.; Marine Cybernetics: Trondheim, Norway, 2002.
28. Blanke, M. *Ship Propulsion Losses Related to Automatic Steering and Prime Mover Control*, 1st ed.; Technical University of Denmark: Lyngby, Denmark, 1981.
29. Ramachandran, D.; Amir, E. Bayesian Inverse Reinforcement Learning. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, 6–12 January 2007.
30. Ng, A.Y.; Russell, S. Algorithms for Inverse Reinforcement Learning. In Proceedings of the 17th International Conference on Machine Learning (ICML), San Francisco, CA, USA, 29 June–2 July 2000.
31. Arora, S.; Doshi, P. A Survey of Inverse Reinforcement Learning: Challenges, Methods and Progress. *Artif. Intell.* **2021**, *297*, 103500. [CrossRef]
32. Google Maps. Rhine River. Available online: https://www.google.com/maps/@47.6679863,9.1762762,16.96z (accessed on 5 May 2022).
33. Wirtensohn, S.; Hamburger, O.; Homburger, H.; Kinjo, L.M.; Reuter, J. Comparison of Advanced Control Strategies for Automated Docking. In Proceedings of the 13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles (CAMS), Oldenburg, Germany, 22–24 September 2021; pp. 295–300.