Feature-Based Proposal Density Optimization for Nonlinear Model Predictive Path Integral Control

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

***Abstract*— This paper presents a novel feature-based sampling strategy for nonlinear Model Predictive Path Integral (MPPI) control. In MPPI control, the optimal control is calculated by solving a stochastic optimal control problem online using the weighted inference of stochastic trajectories. While the algorithm can be excellently parallelized the closed-loop performance is dependent on the information quality of the drawn samples. Because these samples are drawn using a proposal density, its quality is crucial for the solver and thus the controller performance. In classical MPPI control, the explored state-space is strongly constrained by assumptions that refer to the control value variance, which are necessary for transforming the 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 used to determine the proposal density and thus, the 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. Further, the developed algorithm is applied on an autonomous vessel that follows a track and concurrently avoids collisions using an emergency braking feature.**

I. INTRODUCTION

Recent scientific and technical advances in the context of control engineering and available computational power allow nonlinear stochastic optimal control problems (OCP) to be solved in real time by lately explored numerical methods. For a restricted class of stochastic nonlinear control problems the HJB equation can be transformed into a linear partial second-order differential equation [1]. This class is characterized by arbitrary but input affine dynamics, containing input noise, and the cost function is only restricted to be quadratic in the controls. Further, in [1] the solution derived for both the optimal control and the optimal value function can be expressed as a Feynman-Kac path integral. In the path integral framework [2], the solution of the transformed HJB is formulated as a conditional expectation value with respect to the stochastic dynamical system. As a result, the optimal control can be estimated using Monte Carlo sampling [3]. While in general the resulting optimal control function has an unknown structure, there are different approaches for its representation. Beside offline learning parametrized policy approaches [4],[5] nonlinear model predictive control (NMPC) has become the de-facto technological standard [6]-[8]. Based on path integrals, a new type of NMPC algorithm called MPPI is presented in [6]. Using a free energy definition [9], the input affine requirement is completely removed by [7]. Because MPPI is based on Monte Carlo simulation, the information content of the thereby drawn samples is highly dependent on the proposal density [10]. While in [11] a robust MPPI version and in [12] a covariance steering approach are introduced, in this paper, a novel feature-based MPPI extension is presented. In section II, 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 III. In section IV, an application scenario is defined containing the equations of motions of a vessel and cost functions, which are combined as an optimal control problem (OCP). Further, an emergency braking feature is defined. In section V, a comparison of usual MPPI control and its feature-based extension is given. Finally, in section VI the conclusions of this paper and ideas for future work are presented.

[1] Institute of System Dynamics, University of Applied Sciences Konstanz, 78462 Konstanz, Germany {hhomburg, stwirten, jreuter}@htwg-konstanz.de

[2] Department of Microsystems Engineering (IMTEK) and Department of Mathematics, University of Freiburg, 79110 Freiburg, Germany moritz.diehl@imtek.uni-freiburg.de

II. MPPI CONTROL

Recently, an NMPC algorithm based on path integrals was presented by [7], calculating the optimal control inputs by solving the time-discrete nonlinear stochastic OCP

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

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

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

in real time, where $\boldsymbol{F} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ denotes the 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 additive white Gaussian noise (AWGN) $\boldsymbol{\varepsilon}_t \in \mathbb{R}^m$ with covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{m \times m}$. The costs are given by the terminal costs denoted by $\phi(\boldsymbol{X}_T)$, the instantaneous state costs denoted by $C(\boldsymbol{X}_t)$ and a quadratic input term with weighting matrix $\boldsymbol{R} \in \mathbb{R}^{m \times m}$. The cost function minimized in (1a) evaluates the expected costs subject to the commanded system inputs $\boldsymbol{U} = \{\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{T-1}\} \in \mathbb{R}^{m \times T}$ with regard to an initial state $\boldsymbol{x}_0$. In this section, the basics of this novel algorithm are described before it will be extended in the next section. According to [7], the distribution function of the uncontrolled system denoted by $\mathbb{P}$ with zero input $\boldsymbol{u}_t = \boldsymbol{0}$, $t = 0, 1, ..., T-1$ leads to the probability density

function (PDF)

$$p(\boldsymbol{V}) = \prod_{t=0}^{T-1} \frac{1}{((2\pi)^m |\boldsymbol{\Sigma}|)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\boldsymbol{v}_t^T \boldsymbol{\Sigma}^{-1} \boldsymbol{v}_t\right), \quad (2)$$

where $\boldsymbol{V} = \{\boldsymbol{v}_0, \boldsymbol{v}_1, ..., \boldsymbol{v}_{T-1}\}$ denotes the sequence of actual input values. The distribution of the controlled system denoted by $\mathbb{Q}$ with an open-loop sequence of manipulated variables $\boldsymbol{u}_t = \boldsymbol{u}_t^P, t = 0, 1, ..., T-1$ leads to the PDF

$$q(\boldsymbol{V}) = \prod_{t=0}^{T-1} \frac{1}{((2\pi)^m |\boldsymbol{\Sigma}|)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{v}_t - \boldsymbol{u}_t^P)^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{v}_t - \boldsymbol{u}_t^P)\right). \quad (3)$$

An initial state $\boldsymbol{x}_0$ and a realized sequence of input values $\boldsymbol{V}$ can be uniquely assigned to a trajectory without stochastic influence using (1b). Introducing the cumulated state-dependent path costs $S(\boldsymbol{V}) = \phi(\boldsymbol{X}_T) + \sum_{t=0}^{T-1} C(\boldsymbol{X}_t)$ the value function for the OCP (1) is given by

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

with respect to the uncontrolled dynamics [7]. To express the value function with respect to $\mathbb{Q}$ the likelihood ratio $\frac{p(V)}{q(V)}$ must be added, which yields

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

Applying Jensen's inequality according to [7] yields

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

where the bound is tight with

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

where $\eta \in \mathbb{R}$ denotes a normalization factor. In [7] it is shown that the associated optimal control values are given by

$$\boldsymbol{U}^* = \underset{\boldsymbol{U}}{\operatorname{argmin}}\ \mathbb{D}_{KL}(\mathbb{Q}^* || \mathbb{Q}) \quad (8a)$$

$$\boldsymbol{u}_t^* = \mathbb{E}_{\mathbb{Q}^*}[\boldsymbol{v}_t] \quad (8b)$$

where $\mathbb{D}_{KL}$ denotes the Kullback-Leibler divergence and $\mathbb{Q}^*$ denotes the abstract optimal distribution. According to [7], the optimal input is given by

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

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

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

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

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

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

where N samples are drawn from the system dynamics (1b) with the commanded control input sequence $\boldsymbol{U} = \{\boldsymbol{u}_0, \boldsymbol{u}_1, ..., \boldsymbol{u}_{T-1}\}$. The iterative procedure 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 [7]

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;
Σ, ϕ, C, λ: Control hyper-parameters;
Output: $\boldsymbol{U}^O$: Optimized control sequence
$\overline{S}$: Average costs;

for $k \in \{0, 1, ..., K-1\}$ **do**
  $\boldsymbol{x}_{k,0} \leftarrow \boldsymbol{x}_0$;
  Sample $\{\boldsymbol{\varepsilon}_0^k, \boldsymbol{\varepsilon}_1^k, ..., \boldsymbol{\varepsilon}_{T-1}^k\}$;
  $S(k) \leftarrow 0$;
  for $t \in 1, 2, ..., T$ **do**
    $\boldsymbol{x}_{k,t} \leftarrow \boldsymbol{F}(\boldsymbol{x}_{k,t-1}, \boldsymbol{u}_{t-1}^I + \boldsymbol{\varepsilon}_{t-1}^k)$;
    $S(k) += C(\boldsymbol{x}_{k,t}) + \lambda \boldsymbol{u}_{t-1}^{I\top} \Sigma^{-1} \boldsymbol{\varepsilon}_{t-1}^k$;
  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 $k \leftarrow 0$ to $K-1$ **do**
  $\omega(k) \leftarrow \frac{1}{\eta} \exp(-\frac{1}{\lambda} S(k) - \beta)$;
end for
for $t \in \{0, 1, ..., T-1\}$ **do**
  $\boldsymbol{u}_t^O \leftarrow \boldsymbol{u}_t^I + \sum_{k=0}^{K-1} \omega(k) \boldsymbol{\varepsilon}_t^k$;
end for
$\overline{S} \leftarrow \frac{1}{K} \sum_{k=0}^{K-1} S(k)$;
return $\boldsymbol{U}^O = \{\boldsymbol{u}_0^O, \boldsymbol{u}_1^O, ..., \boldsymbol{u}_{T-1}^O\}$ and $\overline{S}$

III. Extension to Feature Based Proposal Density

A. 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 (3). Due to the restrictive assumption $\boldsymbol{R} = \lambda \boldsymbol{\Sigma}^{-1}$, the proposal PDF is only parametrized by a sequence of inputs $\boldsymbol{U}^P = \{\boldsymbol{u}_0^P, \boldsymbol{u}_1^P, ..., \boldsymbol{u}_{T-1}^P\}$. The sampling efficiency is dependent on the quality of the proposal density [10]. In [7] 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

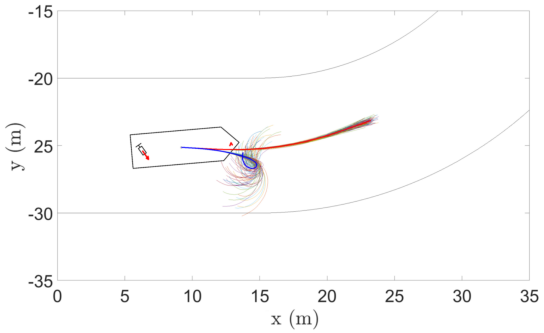

Fig. 1. Explored state-space using stochastic trajectories with mean $\boldsymbol{U}^0$ (red) and mean $\boldsymbol{U}^F$ (blue) for emergency braking feature

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.

B. 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 artificial OCP

$$\boldsymbol{U}^F = \underset{\boldsymbol{U}}{\operatorname{argmin}}\ \mathbb{E}_{\mathbb{Q}}\left\{\phi_F(\boldsymbol{X}_T) + \sum_{t=0}^{T-1}\left[C_F(\boldsymbol{X}_t) + \boldsymbol{u}_t^T \boldsymbol{R}\boldsymbol{u}_t\right]\right\} \tag{12a}$$

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

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

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

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

C. Resulting Feature-Based MPPI Algorithm

The resulting feature-based MPPI algorithm is presented in algorithm 2. First, the optimal 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

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;
Σ, ϕ, C, λ: Control hyper-parameters;
I: Number of features with feature index $i = \{1, 2, ..., I\}$;
$\boldsymbol{U}^i$: Initial control sequence of i_{th} feature;
ϕ_i, C_i: State dependent costs of i_{th} feature;
K_i: Number of i_{th} features samples;

1: **while** *Controller is active* **do**
2: $[\boldsymbol{U}^0, \overline{S}_0] \leftarrow \text{OCS}(K, T, \boldsymbol{U}^0, \boldsymbol{x}_0, \Sigma, \phi, C, \lambda)$
3: **for** $i \in \{1, 2, ..., I\}$ **do**
4: $[\boldsymbol{U}^i, -] \leftarrow \text{OCS}(K_i, T, \boldsymbol{U}^i, \boldsymbol{x}_0, \Sigma, \phi_i, C_i, \lambda)$
5: $[-, \overline{S}_i] \leftarrow \text{OCS}(K_i, T, \boldsymbol{U}^i, \boldsymbol{x}_0, \Sigma, \phi, C, \lambda)$
6: **end for**
7: $i^* \leftarrow \underset{i}{\operatorname{argmin}}\ \overline{S}_i,\ \text{with } i = 0, 1, .., I$
8: $\boldsymbol{U}^0 \leftarrow \boldsymbol{U}^{i^*}$
9: SendToActuator$\left(\boldsymbol{u}_0^0\right)$;
10: **for** $t \in \{1, 2, ..., T-1\}$ **do**
11: $\boldsymbol{u}_{t-1}^0 \leftarrow \boldsymbol{u}_t^0$
12: **end for**
13: $\boldsymbol{u}_{T-1}^0 \leftarrow \text{Initialize}\left(\boldsymbol{u}_{T-1}^0\right)$
14: **end while**

IV. Application Scenario: Emergency Brake

In this section, the feature-based MPPI control algorithm is applied in an emergency braking scenario for a vessel. In the scenario being discussed, the vessel is traveling at full speed along a 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.

A. Dynamics of the Vessel

In [13] 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 given. Nevertheless, a short description of the modeling is given in the following since it is an elementary component of the scenario. According to [14], the state vector is 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 Fig. 2. This model was extended in [15] to also consider the dynamics of the actuators. Thus, the actual actuator states $\boldsymbol{a} = (n_{\text{AT}}\ \ \alpha\ \ n_{\text{BT}})^\top$, 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

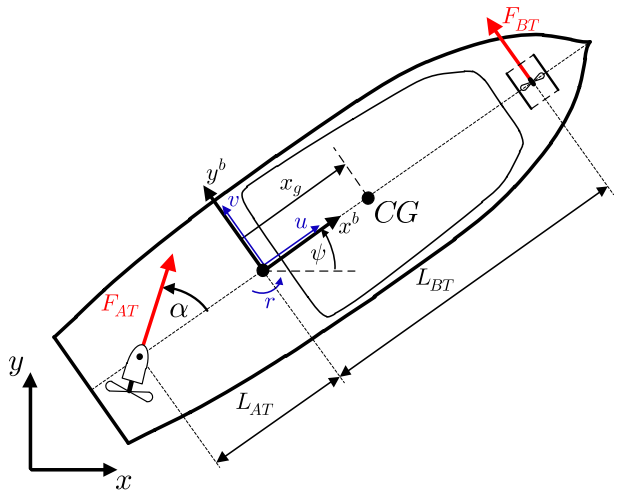

Fig. 2. Local and body-fixed coordinate systems, geometrical parameters and thruster forces

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 part in the system state

$$\boldsymbol{x} = (\boldsymbol{\eta}^\top \quad \boldsymbol{\nu}^\top \quad \boldsymbol{a}^\top \quad \boldsymbol{w}^\top)^\top \in \mathbb{R}^{12}. \tag{14}$$

The whole dynamic model of the vessel is given by

$$\dot{\boldsymbol{w}} = \boldsymbol{u} \tag{15a}$$
$$\dot{\boldsymbol{a}} = \boldsymbol{d}(\boldsymbol{a}, \boldsymbol{w}) \tag{15b}$$
$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{N}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}_c(\boldsymbol{a}, \boldsymbol{\nu}) + \boldsymbol{\tau}_d \tag{15c}$$
$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\psi)\boldsymbol{\nu}, \tag{15d}$$

where in (15a) the desired variable vector $\boldsymbol{w}$ includes the integral action $\boldsymbol{u}$. In (15b) the actuators' dynamics are taken into account by $\boldsymbol{d} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$. The time derivative of the body-fixed velocity $\dot{\boldsymbol{\nu}}$ is implicitly described in (15c) as a function of the body-fixed velocity $\boldsymbol{\nu}$, the matrices $\boldsymbol{M}$, $\boldsymbol{C}_{RB}(\boldsymbol{\nu})$ and $\boldsymbol{N}(\boldsymbol{\nu})$ including the system parameters, the input vector $\boldsymbol{\tau}_c$ and disturbance vector $\boldsymbol{\tau}_d$. The kinematic equation (15d) 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

$$\boldsymbol{\tau}_c(\boldsymbol{a}, \boldsymbol{\nu}) = \begin{pmatrix} F_{\mathrm{AT}}(n_{\mathrm{AT}}, \boldsymbol{\nu}) \cos\alpha \\ F_{\mathrm{AT}}(n_{\mathrm{AT}}, \boldsymbol{\nu}) \sin\alpha + F_{\mathrm{BT}}(n_{\mathrm{BT}}, \boldsymbol{\nu}) \\ F_{\mathrm{BT}}(n_{\mathrm{BT}}, \boldsymbol{\nu}) L_{\mathrm{BT}} - L_{\mathrm{AT}} F_{\mathrm{AT}}(n_{\mathrm{AT}}, \boldsymbol{\nu}) \sin\alpha \end{pmatrix} \tag{16}$$

depends on the geometric parameters L_{BT}, L_{AT} and α and the thrusts F_{AT} and F_{BT}, shown in Fig. 2. The thrust F_{AT} is generated by the AT, F_{BT} denotes the thrust generated by the BT. In [13] it is shown how the thrusts F_{AT} and F_{BT} can be calculated using nonlinear dependencies of various constants, the body-fixed velocity vector $\boldsymbol{\nu}$ and the actuator state vector $\boldsymbol{a}$. For more detailed information about the dynamics, the reader is referred to [13] and [15].

B. Inequality Constraints

Due to the maximum speed of the actuators two inequality constraints

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}) = \begin{pmatrix} |n_{\mathrm{AT}}| - |n_{\mathrm{AT,max}}| \\ |n_{\mathrm{BT}}| - |n_{\mathrm{BT,max}}| \end{pmatrix} \leq \boldsymbol{0}, \quad \forall t \in [t_0, t_0 + T], \tag{17}$$

are defined. By considering the actuator dynamics (15b) in the model, no further constraints are required. In the next part, the cost function is presented for the given scenario.

C. Cost Function

The instantaneous state-dependent cost function $C(\boldsymbol{x})$ is significantly responsible for the behavior of the controlled system, since it is used to evaluate the predicted stochastic trajectories. In the present application example, the quality of a trajectory is evaluated based on three parts

$$C(\boldsymbol{x}) = C_{\mathrm{pos}}(\boldsymbol{\eta}) + C_{\mathrm{vel}}(\boldsymbol{\nu}) + C_{\mathrm{col}}(\boldsymbol{\eta}), \tag{18}$$

where $C_{\mathrm{pos}}(\boldsymbol{\eta})$ denotes the position dependent costs used to stay on the parcours shown in Fig. 3, $C_{\mathrm{vel}}(\boldsymbol{\nu})$ denotes the part of the cost function that is dependent on the body-fixed velocity $\boldsymbol{\nu}$ and a collision is penalized by $C_{\mathrm{col}}(\boldsymbol{\eta})$. The vessel should move on a predefined track shown in Fig. 3. Moreover, the vessel's distance to the boundaries of the track should be maximized. To create a function with these described properties, first the indicator function

$$\zeta(x, y) = \begin{cases} 0 & \text{if } \{x, y\} \in \boldsymbol{\eta}_{\mathrm{track}} \\ 1 & \text{if } \{x, y\} \in \boldsymbol{\eta}_{\mathrm{outside}} \end{cases} \tag{19}$$

is defined, where $\boldsymbol{\eta}_{\mathrm{track}}$ is the subset of the state-space where the vessel is on the track and $\boldsymbol{\eta}_{\mathrm{outside}}$ is a disjoint subset. Then, this function is smoothed using a moving average filter with a square filter with side length d_{filt}. Finally the smoothed function is normalized so that $\max[C_{\mathrm{pos}}(\boldsymbol{\eta})] = c_{\mathrm{pos,max}}$. This yields:

$$C_{\mathrm{pos}}(\boldsymbol{\eta}) = c \sum_{k_x=-N/2}^{N/2} \sum_{k_y=-N/2}^{N/2} \zeta(x + k_x\Delta, y + k_y\Delta), \tag{20}$$

where c denotes a normalization factor, Δ denotes the sample widths and $N = \lfloor d_{\mathrm{filt}}/\Delta \rfloor$. As a further quality criterion, the difference between the body-fixed velocity component u and a reference velocity u_{ref} with

$$C_{\mathrm{vel}}(\boldsymbol{\nu}) = c_u(u_{\mathrm{ref}} - u)^2 \tag{21}$$

is penalized quadratically. By this choice of the velocity dependent part of the cost function, the velocity components v and r have no influence on the cost function. Thus, a drifting behavior of the vessel is neither penalized nor made desirable. To avoid a collision, the indicator function

$$C_{\mathrm{col}}(\boldsymbol{\eta}) = \begin{cases} c_{\mathrm{col}} & \text{if } \boldsymbol{\eta} \in \boldsymbol{\eta}_C \\ 0 & \text{else} \end{cases} \tag{22}$$

is defined, where $\boldsymbol{\eta}_C$ denotes the region of the state-space where a collision of the vessel would occur and $c_{\mathrm{col}} \in \mathbb{R}^+$ denotes a penalizing parameter.

TABLE I

CONTROLLER PARAMETERS

Parameter	MPPI	feature-based MPPI
K	750	500
T	41	41
$\boldsymbol{\Sigma}$	diag(114.56, 2.39, 301.7)	diag(114.56, 2.39, 301.7)
λ	1	1
$C(\boldsymbol{x})$	Eq. (23)	Eq. (23)
$\phi(\boldsymbol{x})$	0	0
K_1	-	250
$C_1(\boldsymbol{x})$	-	Eq. (28)
$\phi_1(\boldsymbol{x})$	-	Eq. (28)

D. Resulting Problem Formulation

To minimize the cumulated costs (18) subject to the given dynamics (15a)-(15d) and the inequality constraints (17) using MPPI control the OCP has to be formulated in the assumed structure (1a)-(1c). Consequently, the inequality constraints must be considered in the cost function with

$$C_{\text{MPPI}}(\boldsymbol{x}) = C(\boldsymbol{x}) + c_{\text{ineq}} \max[\boldsymbol{0}, \boldsymbol{h}(\boldsymbol{a})], \tag{23}$$

where $c_{ineq} \in \mathbb{R}$ denotes a coefficient of the penalty term. In addition, the end cost function $\phi(\boldsymbol{x})$ must be determined. Because this function is not needed in the given scenario, it is defined as

$$\phi(\boldsymbol{x}) = 0. \tag{24}$$

In MPPI control, a discrete-time system dynamics with input noise is assumed in (1b). Therefore, the time continuous system dynamics given in (15a)-(15d) are discretized using the explicit fourth order Runge-Kutta method with a step size h. Consequently, the time discrete vessel dynamics is given by

$$\boldsymbol{X}_{t+1} = \boldsymbol{F}_v(\boldsymbol{X}_t, \boldsymbol{u}_t). \tag{25}$$

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

$$\boldsymbol{u}_t \sim \mathcal{N}(\boldsymbol{u}_{\text{des},t}, \boldsymbol{\Sigma}), \tag{26}$$

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 OCP is given by

$$\min_{\boldsymbol{U}_{des}} \mathbb{E}\left\{ \sum_{t=0}^{T-1} [C_{\text{MPPI}}(\boldsymbol{X}_t) + \boldsymbol{u}_{\text{des},t}^T \boldsymbol{R} \boldsymbol{u}_{\text{des},t}] | \boldsymbol{X}_0 = \boldsymbol{x}_0 \right\} \tag{27a}$$

$$\text{with } \boldsymbol{X}_{t+1} = \boldsymbol{F}_v(\boldsymbol{X}_t, \boldsymbol{u}_t), \quad \forall t \in \{0, 1, ..., T-1\}, \tag{27b}$$

$$\text{and } \boldsymbol{u}_t \sim \mathcal{N}(\boldsymbol{u}_{\text{des},t}, \boldsymbol{\Sigma}), \quad \boldsymbol{R} = \lambda \boldsymbol{\Sigma}^{-1}, \tag{27c}$$

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

E. Feature definition

The selection of the feature can be done by inverse reinforcement learning according to [16] 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. 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 brake feature is defined by choosing

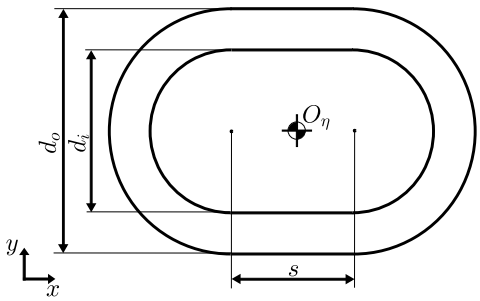

Fig. 3. Track with inside diameter d_i, outside diameter d_o, length of the straight s and origin of the local frame O_η

$$C_1(\boldsymbol{x}) = \phi_1(\boldsymbol{x}) = \boldsymbol{v}^\top \boldsymbol{Q} \boldsymbol{v}, \quad \boldsymbol{Q} = \text{diag}(c_u, c_v, c_r), \tag{28}$$

where $c_u, c_v, c_r \in \mathbb{R}^+$ denote the coefficients of the quadratic components of the body-fixed velocity. Consequently, this feature leads to the exploration of the region of the state-space, that leads to a decreasing velocity.

V. SIMULATION RESULTS

The MPPI controller (Sec. II) and the feature based MPPI controller (Sec. III) are embedded in a simulation environment to compare their performances controlling the vessel model (Sec. IV). The used controller parameters are listed in Tab. I. Note, for an objective comparison, the parameters are chosen to be equal. The covariance Σ is determined for minimal cumulated costs results using Monte-Carlo simulations. In the feature-based MPPI, a part of the trajectories drawn is used to evaluate the feature. Thus, both controllers have almost the same computational effort. The step size for discretizing the system dynamics is chosen as $h = 360$ ms. The parameters for the position dependent costs (20) are given by $d_i = 40$ m, $d_a = 60$ m, $s = 30$ m, $d_{\text{filt}} = 9$ m,

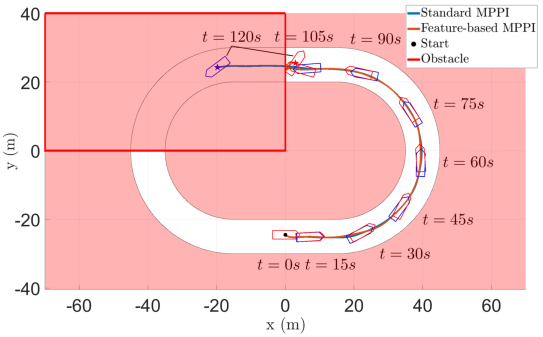

Fig. 4. Resulting trajectories of simulation scenarios and 2D pose of the vessel at equidistant time instances. The red shaded area corresponds to $\boldsymbol{\eta}_C$.

$\Delta = 0.5\,\mathrm{m}$ and $c_{\mathrm{pos,max}} = 100\,\mathrm{m}$. The reference surge velocity, used in (21) is given by $u_{\mathrm{ref}} = 1\,\mathrm{m/s}$ and the collision space is given by $\boldsymbol{\eta}_C \in \{\mathbb{R}^3 | (x < 0 \wedge y > 0) \vee \boldsymbol{\eta}_{\mathrm{outside}}\}$ with costs $c_{\mathrm{col}} = 10000$. A violation of the inequality constraint (23) is penalized by $c_{\mathrm{ineq}} = 10000$. The velocity cost coefficient is given by $c_u = 250$. The feature cost parameters (28) are determined as $c_u = 5000$, $c_v = 100$, $c_r = 100$. The total time of the simulation is set to 135 s. Further, the initial 2D pose of the vessel is set to $\boldsymbol{\eta}_0 = (0\,\mathrm{m} \quad -25\,\mathrm{m} \quad 0\,\mathrm{rad})^\top$ while the other states with $\boldsymbol{\nu}_0 = \boldsymbol{a}_0 = \boldsymbol{w}_0 = \mathbf{0}$ are set to zero. The resulting trajectories and the vessel's orientation at several time instances are shown in Fig. 4. The time courses of the minimal path costs β are plotted over time in Fig. 5. Both, the conventional MPPI controlled system and the feature-based MPPI controlled system show almost the same behavior for $t \leq 103\,\mathrm{s}$ in the absence of an obstacle. At $t = 0\,\mathrm{s}$ the costs were caused by the initial body-fixed velocity $\boldsymbol{\nu}_0 = \mathbf{0}$. Due to fewer predicted stochastic trajectories of the feature-based MPPI with regard to the main costs (23), the surge velocity is slightly overshooting at the end of the acceleration phase. After that, both systems behave identically, and the position distance doesn't increase any further. Starting at $t = 98\,\mathrm{s}$ some of the predicted trajectories would collide with the obstacle within the prediction horizon. The standard MPPI controller only decelerates briefly and then accelerates for $t > 103\,\mathrm{s}$. 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 $t > 103\,\mathrm{s}$. Consequently, predicted trajectories with surge velocity close to u_{ref} cause lower costs. In contrast, the feature-based MPPI controlled system explores more important areas of the state-space shown in Fig. 1. Thus, it can also react excellently for $t > 103\,\mathrm{s}$ and brake in time to avoid a collision.

VI. CONCLUSION AND FUTURE WORK

In this paper, an extension of the MPPI algorithm [7] is presented improving the sample efficiency by adapting the proposal density to defined features. This is the first possibility to incorporate information about a scenario additive to the usual cost function into the controller. The conventional MPPI and the presented feature-based MPPI are compared in a collision avoidance scenario. Therefore, a stochastic nonlinear OCP for enableling 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. In the simulation, 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. Concerning other NMPC approaches, the MPPI control algorithm is characterized by a very good parallelizability drawing the samples. This property is not affected by the presented feature-based extension. In future work, it would be interesting to evaluate the scenario in a real-world experiment. In addition, an evaluation of how the presented feature-based approach can be used in other NMPC algorithms could be scientifically valuable.

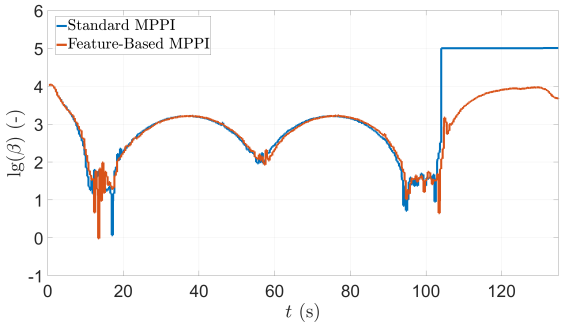

Fig. 5. Comparison of minimal path costs plotted over time. Note the logarithmic representation of the minimal pathcost.

REFERENCES

[1] H. J. Kappen, *Path Integrals and Symmetry Breaking for Optimal Control Theory*. In Journal of Statistical Mechanics Theory and Experiment, 2005.

[2] H. Kleinert, *Path Integrals in Quantum Mechanics, Statistics, Polymer Physics, and Financial Markets*. World Scientific Publishing Ltd, 5th Edition, Singapore, Republic of Singapore, 2009.

[3] S. Schaal and C. Atkeson, *Learning control in robotics*. In Robotics and Automation Magazine, IEEE, Vol.17, pp. 20 – 29, 2010.

[4] E. Theodorou, J. Buchli und S. Schaal, *A Generalized Path Integral Control Approach to Reinforcement Learning*. In Journal of Machine Learning Research, Vol. 11, S. 3137-3181, 2010.

[5] H. J. Kappen and H. Ruiz, *Adaptive importance sampling for control and inference*. In Journal of Statistical Physics, Vol. 162, 2016, pp. 1244–1266.

[6] V. Gómez, S. Thijssen, A. Symington, S. Hailes and H. J. Kappen, *Real-Time Stochastic Optimal Control for Multi-agent Quadrotor Systems*. In 26th International Conference on Automated Planning and Scheduling (ICAPS 16), London, 2015.

[7] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. Rehg, B. Boots, and E. Theodorou, *Information Theoretic MPC for Model-Based Reinforcement Learning*. In IEEE International Conference on Robotics and Automation (ICRA), Singapore, Republic of Singapore, 2017.

[8] H. Homburger, S. Wirtensohn and J. Reuter, *Swinging Up and Stabilization Control of the Furuta Pendulum using Model Predictive Path Integral Control*. In 30th Mediterranean Conference on Control and Automation (MED), Athens, 2022.

[9] E. Theodorou and E. Todorov, *Relative entropy and free energy dualities: Connections to path integral and KL control*. In IEEE 51st Annual Conference on Decision and Control (CDC), 2012, pp. 1466–1473.

[10] S. Thijssen and H. J. Kappen, *Path Integral Control and State Dependent Feedback*. In Phys. Rev. E 91, 032104, 2015.

[11] M. S. Gandhi, B. Vlahov, J. Gibson, G. Williams and E. Theodorou. *Robust Model Predictive Path Integral Control: Analysis and Performance Guarantees*. In IEEE Robotics and Automation Letters, Preprint Version, 2021.

[12] J. Yin, Z. Zhang, E. Theodorou and P. Tsiotras, *Improving Model Predictive Path Integral using Covariance Steering*. In arXiv preprints, arXiv:2109.12147, 2021.

[13] L. M. Kinjo, S. Wirtensohn, J. Reuter, T. Menard, O. Gehan, *Trajectory tracking of a fully-actuated surface vessel using nonlinear model predictive control*. In 13th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles (CAMS), 2021.

[14] T. I. Fossen, *Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics AS, Trondheim, 2002.

[15] H. Homburger, S. Wirtensohn and J. Reuter, *Docking Control of a Fully-Actuated Autonomous Vessel using Model Predicitve Path Integral Control*. In 20th European Control Confernce (ECC), London, 2022.

[16] A. Ng and S. Russel, *Algorithms for Inverse Reinforcement Learning*. In Proceedings of the Seventeenth International Conference on Machine Learning (IMCL), Berkeley, 2000, pp. 663-670.