2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Robustified Time-optimal Collision-free Motion Planning for Autonomous Mobile Robots under Disturbance Conditions

Shuhao Zhang[1], Mathias Bos[1], Bastiaan Vandewal[1], Wilm Decré[1], Joris Gillis[1], Jan Swevers[1]

Abstract— **This paper presents a robustified time-optimal motion planning approach for navigating an Autonomous Mobile Robot (AMR) from an initial state to a terminal state without colliding with obstacles, even when subjected to disturbances, which are modeled as random process noise and measurement noise. The approach iteratively solves the robustified problem by incorporating updated state-dependent safety margins for collision avoidance, the evolution of which is derived separately from the robustified problem. Additionally, a strategy for selecting an alternative terminal state to reach is introduced, which comes into play when the desired terminal state becomes infeasible considering the disturbances. Both of these contributions are integrated into a robustified motion planning and control pipeline, the efficacy of which is validated through simulation experiments.**

I. Introduction

The prominence of Autonomous Mobile Robots (AMRs) is experiencing a surge, primarily due to their capacity to successfully navigate in complex environments such as industrial facilities, hospitals, warehouses, and homes [1]. This achievement hinges on AMRs' localization and environmental perception through onboard sensors, precise trajectory tracking, flexible maneuvering, and, notably, robust and adaptable motion planning ensuring optimality and safety in response to changing and obstructed surroundings.

Such motion planning is often formulated as an optimal control problem (OCP), and solved in a Nonlinear Model Predictive Control (NMPC) fashion, as it allows to take into account an AMR model, physical limitations, environmental constraints, and desired objective [2]–[4]. Employing the nominal planning and control pipeline in closed-loop applications, i.e., to combine the standard NMPC-based motion planning with an appropriate feedback controller for trajectory tracking, enables the AMR to reject both high-frequency and low-frequency disturbances [5]. Yet, neglecting to explicitly account for these disturbances in the planning process can have severe consequences in terms of satisfying constraints. Time-optimal planning, for instance, often operates on the brink of collision avoidance constraints. In such scenarios, even minor perturbations can result in constraint violations, posing a significant risk to the system's safety. Introducing a fixed heuristically chosen safety margin for each constraint might appear straightforward, but it does not provide guarantees of constraint satisfaction and can lead to an overly conservative result.

This paper focuses on motion planning for an AMR to ensure time-optimality and collision avoidance under disturbances modeled as random process noise and noisy output measurements. We achieve it by formulating motion planning as an OCP, which derives a robustified collision-free trajectory for navigating the AMR from an initial state to a desired terminal state in the shortest time. In terms of robustification, we formulate a state-dependent safety margin for each collision avoidance constraint. This approach showcases significant enhancements in both system safety and closed-loop performance. The simulations, demonstrating point-to-point navigation for an AMR modeled as a differential drive with a specific emphasis on avoiding collisions with circular obstacles, validate the proposed approach.

A. Related Work

To explicitly account for modeling and propagating uncertainty, the OCP involves a linearization-based uncertainty propagation to robustify the nominal constraints through approximated chance constraints as discussed in [6]–[12]. Given that introducing additional states for uncertainty propagation increases the dimension of the state-space from n_s to $O(n_s^2)$, finding methods to reduce computation time for solving the robustified OCP is important. Straightforward application of the standard Multiple Shooting transcription leads to $O(n_s^6)$ computational demand. [7] attained $O(n_s^3)$ by adopting a Single Shooting strategy for the additional states. An alternative method with the same complexity, referred to as a zero-order method, was developed in [9] for stochastic NMPC and in [10] for robust NMPC. This zero-order method solves the OCP and updates the state uncertainty set separately in an iterative manner, and keeps the state uncertainty set fixed in one optimization iteration. Based on this method, [11] presents a trajectory tracking implementation for an AMR under bounded process noise with precomputed feedback gain. [12] proposes a zero-order based algorithm to also optimize the feedback gain in the OCP for counteracting the growth of state uncertainty. These studies do not account for disturbances in output measurements nor do they prioritize time optimality.

An additional challenge arises when the replanning result from the planner, in response to changing environments, is not available in a timely manner. It poses a danger to the AMR while tracking the trajectory. A commonly employed strategy, known as the Real-Time Iterations (RTI) technique [13], aims to mitigate computational complexity by considering only the

[1]MECO Research Team, Department of Mechanical Engineering, KU Leuven, and Flanders Make@KU Leuven, Core lab MPRO, Leuven, Belgium. Email: shuhao.zhang@kuleuven.be

This work has been carried out within the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement ELO-X No.953348, and the framework of Flanders Make SBO ARENA: Agile and Reliable Navigation and FLEXMOSYS: Flexible Multi-Domain Design for Mechatronic Systems.

initial iteration of the OCP. The RTI technique comes at the cost of constraint satisfaction, potentially resulting in infeasible solutions. A most recent technique, referred to as ASAP-MPC [14], proposes an alternative strategy to ensure that the MPC-based planner generates a seamless trajectory despite fluctuations in computation times caused by environmental changes. It selects an initial condition from the previously solved trajectory for the replanning problem. ASAP-MPC has been successfully applied in two different applications: drone navigation [15] and AMR maneuvering [16]. Nonetheless, the replanning process of ASAP-MPC does not take into account state and output feedback due to on-trajectory updates.

B. Contributions:

Two main contributions are presented in this work:

1) A robustified time-optimal motion planner by adapting the method in [9], [10]. This adaptation considers disturbances in both the process model and the measurement model when deriving state uncertainty sets and state-dependent safety margins for avoiding collisions. As a result, it generates a robustified trajectory in a time-optimal and collision-free manner.
2) A strategy to reselect a feasible terminal state to reach in the event that the desired terminal state becomes infeasible under disturbances. This strategy introduces a slack variable into the terminal state constraint, which is minimized to zero when the desired or reselected terminal state is feasible.

We then propose a robustified motion planning and control pipeline, integrating the robustified motion planner, the state estimator, the feedback controller, and the ASAP-MPC update strategy [14]. The cooperation between the robust motion planner and the ASAP-MPC update strategy allows feedback to be taken into account when replanning. We validate the presented robustified pipeline in a ROS-based simulation.

C. Notation

At a specific time step k, a time-dependent vector variable is represented using a bold letter with a subscript, i.e., $\boldsymbol{s}_k := \boldsymbol{s}(t_k)$. A trajectory of a variable is denoted by $\boldsymbol{s}$ or $\{\boldsymbol{s}_n\}_0^N := \boldsymbol{s}_n, \forall n \in \{0, ..., N\}$, interchangeably. The absolute value is denoted by $|\cdot|$.

D. Paper Structure

In Section II, the nominal time-optimal planning and control approach is introduced, while Section III presents the developed robustified time-optimal planning and control approach. The case study setup and validation results are presented in Section IV. Section V concludes the paper.

II. Nominal Time-optimal Planning and Control

We consider a general stochastic nonlinear system,

$$\frac{d\boldsymbol{s}}{dt} = \mathcal{F}(\boldsymbol{s}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)), \tag{1}$$

with the system's state $\boldsymbol{s}(t) \in \mathbb{R}^{n_s}$, control input $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$, and process noise $\boldsymbol{w}(t) \in \mathbb{R}^{n_w}$, to represent the kinematics of an AMR. The state of an AMR contains its position in the 2D plane and its heading angle.

The nominal time-optimal motion planning and control pipeline for the AMR consists of two ingredients: 1) A nominal time-optimal planner discussed in Sec. II-A, which plans a collision-free trajectory for the nominal AMR system, disregarding process noise. 2) A stabilizing feedback controller discussed in Sec. II-B to track the position and orientation reference encompassed in the planned nominal trajectory and reject disturbances. The ASAP-MPC update strategy [14] explained in Sec. II-C coordinates the planner and the controller in the pipeline to handle varying replanning times in changing environments with guaranteed convergence.

A. Nominal Time-optimal Planner

The nominal time-optimal motion planning is posed as an OCP where the objective is to minimize the overall trajectory duration required for the AMR to navigate from the initial state to the desired terminal state. It takes the form

$$\begin{aligned}
\underset{T,\bar{\boldsymbol{s}},\bar{\boldsymbol{u}}}{\text{minimize}} \quad & T \\
\text{subject to} \quad & \bar{\boldsymbol{s}}_0 = \boldsymbol{s}_{\text{t0}} \\
& \bar{\boldsymbol{s}}_{n+1} = \bar{F}(\bar{\boldsymbol{s}}_n, \bar{\boldsymbol{u}}_n, \Delta t_{\text{plan}} := T/N) && \forall n \in \{0,...,N-1\} \\
& \bar{\boldsymbol{s}}_N = \boldsymbol{s}_{\text{tf}} \\
& \boldsymbol{u}_{\min} \le \bar{\boldsymbol{u}}_n \le \boldsymbol{u}_{\max} && \forall n \in \{0,...,N\} \\
& \dot{\boldsymbol{u}}_{\min} \le \dot{\bar{\boldsymbol{u}}}_n := \frac{\bar{\boldsymbol{u}}_{n+1} - \bar{\boldsymbol{u}}_n}{\Delta t_{\text{plan}}} \le \dot{\boldsymbol{u}}_{\max} && \forall n \in \{0,...,N-1\} \\
& h(\bar{\boldsymbol{s}}_n) + \bar{\boldsymbol{\beta}}_n^{\text{safe}} \le 0 && \forall n \in \{0,...,N\},
\end{aligned} \tag{2}$$

where N denotes the planning horizon. The optimization variables $\bar{\boldsymbol{s}}$, $\bar{\boldsymbol{u}}$ represent the nominal state and control input trajectory, respectively. The subscript $n \in \{0, ..., N\}$ is used to denote the time index of the nominal trajectory in the planning. The initial and the desired terminal state condition at the current time step are denoted by $\boldsymbol{s}_{\text{t0}}$ and $\boldsymbol{s}_{\text{tf}}$. The optimization variable T, representing the overall trajectory duration from the initial to the desired terminal state during the horizon, is minimized within the objective function. The nominal system $\dot{\bar{\boldsymbol{s}}}(t) = \mathcal{F}(\bar{\boldsymbol{s}}(t), \bar{\boldsymbol{u}}(t), 0)$, of which the process noise is disregarded by setting $\boldsymbol{w}(t) = 0$, is evaluated with the nominal trajectory and discretized by numerical integration with the sampling time Δt_{plan} in a multiple shooting scheme: $\bar{\boldsymbol{s}}_{n+1} = \bar{\boldsymbol{F}}(\bar{\boldsymbol{s}}_n, \bar{\boldsymbol{u}}_n, \Delta t_{\text{plan}}) : \mathbb{R}^{n_s} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_s}$. We adopt a piecewise linear parameterization for the control input to enhance the smoothness of the control input and state trajectory, and imposing constraints on both the control input and its derivative calculated by finite differences. In terms of the collision avoidance constraints $h(\bar{\boldsymbol{s}}_n) : \mathbb{R}^{n_s} \to \mathbb{R}^{n_h}$, we introduce a fixed heuristically chosen safety margin term $\bar{\boldsymbol{\beta}}_n^{\text{safe}}$ for keeping a safe distance between the AMR and obstacles. It is important to note that the incorporation of a fixed safety margin term to tighten collision avoidance constraints, while not accounting for feedback, does not provide any guarantees of constraint satisfaction for the real-world system. This is our main motivation for developing the robustified motion planning approach in Sec III.

B. Linear Stabilizing Feedback Controller

The feedback controller is designed and implemented to track the nominal state trajectory $\bar{s}$ while mitigating disturbances, working in tandem with the nominal control input $\bar{u}$ as the feedforward control. Given the assumption of full observability for all state variables and the system being controlled with a sampling time of Δt_{ctrl}, the control policy at each time step t_k is expressed as follows:

$$u_k = \bar{u}_k + K_k(s_k - \bar{s}_k), \tag{3}$$

where s_k and K_k denote the true state and the feedback control gain matrix, respectively. We assume a precomputed K_k is available, which will be further discussed in Sec. IV.

C. ASAP-MPC Replanning Strategy

Considering that Δt_{plan} derived by the time-optimal planner is typically longer than the fixed predefined Δt_{ctrl} and variable due to changes in the environment, the pipeline uses the ASAP-MPC strategy [14] to effectively deal with the variability in computation time incurred when replanning. Moreover, it ensures trajectory smoothness and allows solution of the OCP to convergence.

This strategy is initiated by resampling the current nominal state and control trajectory at the sampling time Δt_{ctrl}. The initial state condition of the replanning of the problem (2) is then selected from the resampled nominal state trajectory at a fixed update time interval, which is a tuning parameter. The replanning result is seamlessly integrated into the current nominal trajectory from the initial condition that was taken, ensuring a smooth transition without any discontinuities. Given that problem (2) does not incorporate state feedback from the on-trajectory updates, this strategy relies on the feedback controller to closely track the nominal trajectory.

III. Robustified Time-optimal Planning and Control

We now propose the robustified time-optimal motion planning and control approach, enhancing the nominal one through two aspects. First, we adapt the zero-order method [9], [10] for time-optimal motion planning, resulting in the formulation of a robustified time-optimal OCP in Sec. III-C. This robustified OCP strengthens collision avoidance constraints by introducing state-dependent safety margins (Sec. III-B), which account for both state and output feedback, and are iteratively derived through linearization-based uncertainty propagation (Sec. III-A). Secondly, we introduce a strategy to reselect a terminal state condition during replanning in Sec. III-C. This strategy is employed when the desired terminal state is deemed unsafe under the disturbances.

A. Linearization-based Covariance Propagation

We consider the discrete-time process model of the system (1) and the measurement model as the following system

$$\begin{aligned} s_{k+1} &= f_k(s_k, u_k, w_k), \\ z_{k+1} &= g_{k+1}(s_{k+1}, v_{k+1}), \end{aligned} \tag{4}$$

which is discretized in the same way as the discrete-time system in the problem (2) but by the sampling time Δt_{ctrl} for designing the feedback controller and the state estimator. Therein, $z \in \mathbb{R}^{n_z}$ denotes the measurement variables. The discrete-time process noise $w_k \sim \mathcal{N}(0, Q_k)$, the measurement noise $v_k \sim \mathcal{N}(0, R_k)$, and the initial state $s_0 \sim \mathcal{N}(\bar{s}_0, \Sigma_0)$ are all assumed to be normally distributed. The function $f_k(s_k, u_k, w_k)$ is assumed to be differentiable w.r.t. the state, control input, and process noise, and the function $g_{k+1}(s_{k+1}, v_{k+1})$ is assumed to be differentiable w.r.t. the state and measurement noise.

The uncertainty propagation can be made computationally tractable by linearization around the resampled nominal state and control input trajectory, yielding the linear system

$$s_{k+1} - \bar{s}_{k+1} = \bar{A}_k(s_k - \bar{s}_k) + \bar{B}_k(u_k - \bar{u}_k) + \bar{\Gamma}_k w_k, \tag{5a}$$

$$z_{k+1} - \bar{z}_{k+1} = \bar{C}_{k+1}(s_{k+1} - \bar{s}_{k+1}) + \bar{\Lambda}_{k+1} v_{k+1}, \tag{5b}$$

with the nominal measurement variable $\bar{z}_{k+1} = g_{k+1}(\bar{s}_{k+1}, 0)$, and Jacobian matrices $\bar{A}_k = \frac{\partial f_k(\bar{s}_k, \bar{u}_k, 0)}{\partial s_k}$, $\bar{B}_k = \frac{\partial f_k(\bar{s}_k, \bar{u}_k, 0)}{\partial u_k}$, $\bar{\Gamma}_k = \frac{\partial f_k(\bar{s}_k, \bar{u}_k, 0)}{\partial w_k}$, $\bar{C}_{k+1} = \frac{\partial g_{k+1}(\bar{s}_{k+1}, 0)}{\partial s_{k+1}}$, $\bar{\Lambda}_{k+1} = \frac{\partial g_{k+1}(\bar{s}_{k+1}, 0)}{\partial v_{k+1}}$.
The equation (5a) defines the dynamics of the error state $e_{k+1} := s_{k+1} - \bar{s}_{k+1}$.

The state estimate $\hat{s}_{k+1}$ can be obtained via an extended Kalman filter as

$$\hat{s}_{k+1} = f_k(\hat{s}_k, u_k, 0) + L_{k+1}(z_{k+1} - \bar{z}_{k+1}), \tag{6}$$

where L_{k+1} denotes the Kalman gain. Regarding the linearized system (5), a time-varying Kalman filter is derived by linearizing the estimator (6) around the nominal trajectory. The Kalman gain is computed based on the linearized system matrices and thus are essentially functions of the nominal trajectory. The dynamics of the estimated error state $\hat{e}_{k+1} := \hat{s}_{k+1} - s_{k+1}$ can then be formulated as follows:

$$\begin{aligned} \hat{s}_{k+1} - s_{k+1} = (\mathbb{I} - L_{k+1}\bar{C}_{k+1})(\bar{A}_k(\hat{s}_k - s_k) - \bar{\Gamma}_k w_k) \\ + L_{k+1}\bar{\Lambda}_{k+1} v_{k+1}. \end{aligned} \tag{7}$$

Considering $\hat{e}_k$ and e_k, the control policy is reformulated as

$$\begin{aligned} u_k &= \bar{u}_k + K_k(\hat{s}_k - \bar{s}_k) \\ &= \bar{u}_k + K_k\hat{e}_k + K_k e_k. \end{aligned} \tag{8}$$

Introducing an augmented error state $\tilde{e}_k = [e_k^\top \ \hat{e}_k^\top]^\top$ and input $\tilde{u}_k^e = [w_k^\top \ v_{k+1}^\top]^\top$, we obtain the augmented error system

$$\begin{aligned} \tilde{e}_{k+1} &= \tilde{A}_k\tilde{e}_k + \tilde{B}_k\tilde{u}_k^e, \\ \tilde{A}_k &:= \begin{bmatrix} \bar{A}_k + \bar{B}_k K_k & \bar{B}_k K_k \\ 0 & (\mathbb{I} - L_{k+1}\bar{C}_{k+1})\bar{A}_k \end{bmatrix}, \\ \tilde{B}_k &:= \begin{bmatrix} \bar{\Gamma}_k & 0 \\ -(\mathbb{I} - L_{k+1}\bar{C}_{k+1})\bar{\Gamma}_k & L_{k+1}\bar{\Lambda}_{k+1} \end{bmatrix}, \end{aligned} \tag{9}$$

of which the state has a zero mean $\mathbb{E}\{\tilde{e}_k\} = 0$, and a covariance defined by

$$\tilde{\Sigma}_k = \mathbb{E}\{\tilde{e}_k\tilde{e}_k^\top\} = \begin{bmatrix} \Sigma_k & \check{\Sigma}_k^\top \\ \check{\Sigma}_k & \hat{\Sigma}_k \end{bmatrix}, \tag{10}$$

where $\Sigma_k = \mathbb{E}\{e_k e_k^\top\}$ is the predicted uncertainty for deviations of the true state s_k from the nominal state trajectory $\bar{s}_k$, $\hat{\Sigma}_k = \mathbb{E}\{\hat{e}_k\hat{e}_k^\top\}$ is the predicted uncertainty of the state

estimation error, and $\check{\Sigma}_k = \mathbb{E}\{\hat{\boldsymbol{e}}_k \boldsymbol{e}_k^\top\}$ is the correlation between the estimated error state $\hat{\boldsymbol{e}}_k$ and the error state $\boldsymbol{e}_k$.

The covariance propagation of the error system (9) is approximated as

$$\tilde{\Sigma}_{k+1} = \tilde{A}_k \tilde{\Sigma}_k \tilde{A}_k^\top + \tilde{B}_k \tilde{\Sigma}_k^{\tilde{u}^e} \tilde{B}_k^\top, \tag{11}$$

with the initial augmented error state distribution $\tilde{\boldsymbol{e}}_0 \sim \mathcal{N}(0, \tilde{\Sigma}_0)$, and augmented input distribution $\tilde{\boldsymbol{u}}_k^e \sim \mathcal{N}(0, \tilde{\Sigma}_k^{\tilde{u}^e})$, where

$$\tilde{\Sigma}_0 := \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \otimes \Sigma_0, \; \tilde{\Sigma}_k^{\tilde{u}^e} := \begin{bmatrix} Q_k & 0 \\ 0 & R_{k+1} \end{bmatrix},$$

and $\otimes$ denotes the Kronecker product.

B. Robustified Path Constraints Formulation

We introduce individual chance constraints to ensure that the probability of satisfying each path constraint $h_j(\boldsymbol{s}_k) : \mathbb{R}^{n_s} \to \mathbb{R}$ is higher than a defined probability level ϵ_j as

$$\Pr(h_j(\boldsymbol{s}_k) \leq 0) \geq \epsilon_j, \; j \in \{1, ..., n_h\}. \tag{12}$$

Subsequently, we proceed to demonstrate how the individual probabilistic constraints can be approximated as deterministic constraints, which is the sum of the nominal constraint $h_j(\bar{\boldsymbol{s}}_k)$, and the state-dependent safety margin based on Σ_k, the upper left term of (10), by the method introduced in [8, Sec 3.2.2] as

$$h_j(\bar{\boldsymbol{s}}_k) + \beta_{j,k}^{\text{safe}} := h_j(\bar{\boldsymbol{s}}_k) + \alpha_j \sqrt{H_{j,k} \Sigma_k H_{j,k}^\top} \leq 0, \tag{13}$$

with the Jacobian matrix $H_{j,k} = \frac{\partial h_j(\bar{\boldsymbol{s}}_k)}{\partial \boldsymbol{s}_k}$. The safety margin coefficient α_j is selected to ensure the probability level ϵ_j.

Remark. *One option for general probability distributions is to use the Chebyshev inequality $\alpha_j = \sqrt{\frac{\epsilon_j}{1-\epsilon_j}}$, which may lead to relatively conservative safety margin term [17]. Assuming a normal distribution, we approximate it by using $\alpha_j := \text{cdf}^{-1}(\epsilon_j)$, where $\text{cdf}^{-1}(\cdot)$ is the inverse cumulative density function of a standard normal distribution.*

By vectorizing each individual safety margin, we obtain a vector of the state-dependent safety margins

$$\boldsymbol{\beta}_k^{\text{safe}} = [\beta_{1,k}^{\text{safe}}, \cdots, \beta_{n_h,k}^{\text{safe}}]^\top \tag{14}$$

for robustifying the nominal constraints $h(\bar{\boldsymbol{s}}_k)$ at time step t_k.

C. Robustified Time-optimal Planning and Control

We propose the following robustified OCP, which aims to integrate time-optimal, collision-free motion planning with state-dependent safety margins, as outlined below:

$$\begin{aligned}
\underset{T^i, \boldsymbol{\xi}_N^i, \bar{\boldsymbol{s}}^i, \bar{\boldsymbol{u}}^i}{\text{minimize}} \quad & T^i + \|\boldsymbol{\xi}_N^i\|_{W_{\text{tf}}}^2 \\
\text{subject to} \quad & \bar{\boldsymbol{s}}_0^i = \boldsymbol{s}_{\text{t0}} \\
& \bar{\boldsymbol{s}}_{n+1}^i = \bar{F}(\bar{\boldsymbol{s}}_n^i, \bar{\boldsymbol{u}}_n^i, \Delta t_{\text{plan}}^i) && \forall n \in \{0,...,N-1\} \\
& \bar{\boldsymbol{s}}_N^i + \boldsymbol{\xi}_N^i = \boldsymbol{s}_{\text{tf}}^i \\
& \boldsymbol{u}_{\min} \leq \bar{\boldsymbol{u}}_n^i \leq \boldsymbol{u}_{\max} && \forall n \in \{0,...,N\} \\
& \dot{\boldsymbol{u}}_{\min} \leq \dot{\bar{\boldsymbol{u}}}_n^i \leq \dot{\boldsymbol{u}}_{\max} && \forall n \in \{0,...,N-1\} \\
& h(\bar{\boldsymbol{s}}_n^i) + \boldsymbol{\beta}_n^{\text{safe},i}(\bar{\boldsymbol{s}}^{i-1}, \bar{\boldsymbol{u}}^{i-1}) \leq 0 && \forall n \in \{0,...,N\}.
\end{aligned} \tag{15}$$

The superscript $i \in \{0, ..., i_{\max}\}$ on variables indicates the iteration index used for solving the problem with the same initial state condition and collision avoidance constraints, where $i_{\max}$ represents the maximum number of iterations. The state-dependent safety margin term $\boldsymbol{\beta}_n^{\text{safe},i}(\bar{\boldsymbol{s}}^{i-1}, \bar{\boldsymbol{u}}^{i-1})$ is defined according to equations (13) and (14), yet its computation relies on the nominal trajectory from the preceding iteration, aligning with the idea of the zero-order method [9], [10].

Compared with the nominal problem (2), this robustified problem optimizes an additional slack variable $\boldsymbol{\xi}_N^i \in \mathbb{R}^{n_s}$, representing the difference between the planned and desired terminal state. The quadratic norm of $\boldsymbol{\xi}_N^i$, weighted by a large weighting matrix W_{tf}, is minimized within the objective function. In the normal situation, $\boldsymbol{\xi}_N^i$ is minimized to zero, enabling the AMR to attain the desired terminal state in the shortest time. However, there might arise situations where both time optimality and feasibility under the disturbance conditions cannot be met. In such instances, $\boldsymbol{\xi}_N^i$ assumes a non-zero value, indicating that it is unsafe to reach the desired terminal state, and therefore a new feasible terminal state is required. The terminal state is then adapted by subtracting the value of the slack variable from the desired terminal state.

We then encapsulate the robustified time-optimal motion planning approach in Algorithm 1. Given $\boldsymbol{s}_{\text{t0}}$, Σ_0, $\boldsymbol{s}_{\text{tf}}^0$, the initial guess on the safety margin term $\boldsymbol{\beta}^{\text{safe},0}$, and fixed predefined parameters including N, Δt_{ctrl}, $i_{\max}$, a time variable convergence threshold δT, and a slack variable convergence threshold $\delta \boldsymbol{\xi}$, the robustified approach begins by solving the initial iteration of the problem (15) with $\boldsymbol{\beta}^{\text{safe},0}$. There are four main steps in each subsequent iteration: First, it resamples the preceding normal trajectory using Δt_{ctrl}, resulting $\{\bar{\boldsymbol{s}}_k^{i-1}\}_0^{M^{i-1}}, \{\bar{\boldsymbol{u}}_k^{i-1}\}_0^{M^{i-1}}$ with a length of M^{i-1}. Then, the resampled nominal trajectory is employed to compute the covariance trajectory $\{\Sigma_k^{i-1}\}_0^{M^{i-1}}$ via (9) and (11). Following

Algorithm 1 Robustified time optimal motion planner

Given $\boldsymbol{s}_{\text{t0}}$, Σ_0, $\boldsymbol{s}_{\text{tf}}^0$, $\boldsymbol{\beta}^{\text{safe},0}$, N, Δt_{ctrl}, $i_{\max}$, δT, $\delta \boldsymbol{\xi}$, $i = 0$
Solve the initial iteration of the problem (15)
- Derive the nominal trajectory $\{\bar{\boldsymbol{s}}_n^0\}_0^N, \{\bar{\boldsymbol{u}}_n^0\}_0^N$.
repeat
- $i \leftarrow i + 1$
Resample the nominal trajectory with Δt_{ctrl}
- $\{\bar{\boldsymbol{s}}_k^{i-1}\}_0^{M^{i-1}}, \{\bar{\boldsymbol{u}}_k^{i-1}\}_0^{M^{i-1}} \leftarrow \{\bar{\boldsymbol{s}}_n^{i-1}\}_0^N, \{\bar{\boldsymbol{u}}_n^{i-1}\}_0^N$.
Derive the covariance matrix $\{\Sigma_k^{i-1}\}_0^{M^{i-1}}$
- Propagate the augmented covariance via (9) and (11).
Update the safety margin
- Derive $\{\Sigma_n^{i-1}\}_0^N$ by interpolating $\{\Sigma_k^{i-1}\}_0^{M^{i-1}}$. with $\Delta t_{\text{plan}}^{i-1} := T^{i-1}/N$.
- Derive $\boldsymbol{\beta}_n^{\text{safe},i}(\bar{\boldsymbol{s}}^{i-1}, \bar{\boldsymbol{u}}^{i-1})$ via (13) and (14).
Resolve the problem (15)
- Update $\boldsymbol{s}_{\text{tf}}^i \leftarrow \boldsymbol{s}_{\text{tf}}^{i-1} - \boldsymbol{\xi}_N^{i-1}$ if $|\boldsymbol{\xi}_N^{i-1}| \geq \delta \boldsymbol{\xi}$.
- Derive the nominal trajectory $\{\bar{\boldsymbol{s}}_n^i\}_0^N, \{\bar{\boldsymbol{u}}_n^i\}_0^N$.
until $(|T^i - T^{i-1}| \leq \delta T$ and $|\boldsymbol{\xi}_N^i| \leq \delta \boldsymbol{\xi})$ or $i \geq i_{\max}$
Output $\{\bar{\boldsymbol{s}}_k\}_0^{M^i}, \{\bar{\boldsymbol{u}}_k\}_0^{M^i} \leftarrow \{\bar{\boldsymbol{s}}_n^i\}_0^N, \{\bar{\boldsymbol{u}}_n^i\}_0^N$.

this, the covariance trajectory is interpolated using $\Delta t_{\text{plan}}^{i-1} := T^{i-1}/N$ to match the horizon length N, and the safety margin term is updated by using equation (13) and (14). Finally, we resolve the problem (15) with the updated state-dependent safety margin, and the reselected terminal state if $|\boldsymbol{\xi}_N^{i-1}| \geq \delta\boldsymbol{\xi}$. The entire process continues iteratively until either $|\boldsymbol{\xi}_N^{i}| \leq \delta\boldsymbol{\xi}$ and the absolute difference between the optimization variable T from two consecutive iterations is smaller than δT, or the iteration index i surpasses $i_{\max}$. The robustified planner finally outputs the resampled nominal trajectory $\{\bar{\boldsymbol{s}}_k\}_0^{M^i}, \{\bar{\boldsymbol{u}}_k\}_0^{M^i}$.

The robustified time-optimal motion planning and control pipeline integrates the robustified motion planner, the state estimator, the feedback controller, and the ASAP-MPC update strategy [14]. It is important to underline that feedback can indeed be considered during the replanning process. The initial state condition $\bar{\mathbf{s}}_0 = \mathbf{s}_{\text{t0}}$ for replanning, where $\mathbf{s}_{\text{t0}}$ is selected from the current resampled nominal trajectory, denotes the mean value of the state uncertainty distribution that the system can reach under disturbances. The covariance of this distribution leads to formulate the initial collision avoidance constraint as $h(\bar{\mathbf{s}}_0) + \boldsymbol{\beta}_0^{\text{safe},0} \leq 0$.

IV. Case Study and Discussion

We first validate the presented robustified motion planning approach through a case study that focuses on avoiding collisions with a circular obstacle and a wall represented as a straight line. Subsequently, we demonstrate the replanning capability within the presented robustified planning and control pipeline, showcasing collision avoidance with multiple circular obstacles in a ROS simulation environment.

We assume the AMR is a point mass and represent it using the differential drive model

$$\begin{bmatrix}\dot{x}(t)\\ \dot{y}(t)\\ \dot{\theta}(t)\end{bmatrix} = \begin{bmatrix}v(t)\cos\theta(t)\\ v(t)\sin\theta(t)\\ \omega(t)\end{bmatrix} + \boldsymbol{w}(t), \tag{16}$$

which is discretized using the explicit Runge-Kutta method of order 4. The state variables $\boldsymbol{s} = [x\ y\ \theta]^\top \in \mathbb{R}^3$ includes position x(m) and y(m) in the 2D plane, and the heading angle θ(rad). The control inputs $\boldsymbol{u} = [v\ \omega]^\top \in \mathbb{R}^2$ includes the forward velocity v(m/s) and the angular velocity ω(rad/s). Note that the case study incorporates additive normally distributed process noise, i.e., $\boldsymbol{w}(t)$ in (16), and measurement noise. Consequently, we have the condition $n_w = n_z = n_s = 3$, and the Jacobian matrices $\bar{\Gamma}$, $\bar{\Lambda}$, and $\bar{C}$ are all identity matrices. The value of the noise covariance in the discrete-time are $Q_k = \text{diag}[4\text{cm}^2/\text{s}\ 4\text{cm}^2/\text{s}\ 4\text{deg}^2/\text{s}]\Delta t_{\text{ctrl}}$, and $R_k = \text{diag}[2\text{cm}^2\text{s}\ 2\text{cm}^2\text{s}\ 1\text{deg}^2\text{s}]/\Delta t_{\text{ctrl}}$, respectively. The value of the safety margin coefficient $\alpha = 3$ is selected based on the probability level $\epsilon \approx 0.997$ for all collision avoidance constraints with the assumption of normal distribution.

To design the feedback controller gain matrix K, we adopt the approach presented in [16], [18], which takes into account the deviations of state variables expressed in the trajectory coordinate frame and is piecewise linear w.r.t the nominal forward velocity $\bar{v}$. K_k at the time step t_k is

$$K_k = -\begin{bmatrix}K_x(\bar{v}_k) & 0 & 0\\ 0 & K_y(\bar{v}_k) & K_\theta(\bar{v}_k)\end{bmatrix}\begin{bmatrix}\cos(\bar{\theta}_k) & \sin(\bar{\theta}_k) & 0\\ -\sin(\bar{\theta}_k) & \cos(\bar{\theta}_k) & 0\\ 0 & 0 & 1\end{bmatrix}.$$

TABLE I: Values of common parameters

Δt_{ctrl}	$v_{\min/\max}$	$\dot{v}_{\min/\max}$	$\omega_{\min/\max}$	$\dot{\omega}_{\min/\max}$
0.04s	0/1m/s	$\pm 0.2\text{m/s}^2$	$\pm\frac{\pi}{6}$rad/s	$\pm 0.9\text{rad/s}^2$
$\delta\boldsymbol{\xi}$	δT	N	W_{tf}	$i_{\max}$
$[2\text{mm}\ 2\text{mm}\ 2\text{mrad}]^\top$	2ms	30	5000	5

The robustified OCP is formulated in Python using the Rockit toolbox for rapid OCP prototyping [19], and solved with Ipopt [20] using ma57 as linear solver [21]. Key steps in the implementation including integrator and Jacobian matrix calculation are handled using Casadi [22]. All computations are performed on a laptop with an Intel® Core™ i7-1185G7 processor with eight cores at 3GHz and 31.1GB RAM.

A. Case Study on Robustified Planning

In this case, we study the performance of the presented robustified motion planning approach. We plan a trajectory for the AMR in the presence of disturbances, starting from $\boldsymbol{s}_{\text{t0}}$ and reaching $\boldsymbol{s}_{\text{tf}}$ in the shortest time while avoiding collisions with a circular obstacle and a wall. The large circular obstacle is located at coordinates (2m, 2m), and it has a radius of 2m. This configuration is chosen to ensure that the collision avoidance constraint $(x - 2\text{m})^2 + (y - 2\text{m})^2 \leq (2\text{m})^2$ is frequently triggered. The straight wall is located at $x = 3.8$m, and the feasible region is $x \leq 3.8$m. We choose a desired terminal state $\boldsymbol{s}_{\text{tf}} = [3.8\text{m}\ 3.6\text{m}\ 0\text{rad}]^\top$ for the AMR to reach. The initial condition of AMR is $\boldsymbol{s}_{\text{t0}} = [-0.5\text{m}\ 2\text{m}\ \frac{\pi}{2}\text{rad}]^\top$ without initial uncertainty, i.e., $\Sigma_0 = 0$.

As depicted in Fig. 1, the time-optimal robustified trajectory is generated after three iterations. In the initial iteration, we solve the problem without taking the safety margin terms into account. The resulting nominal trajectory can successfully reach the desired terminal state without violating $\delta\boldsymbol{\xi}$. In the second iteration, we incorporate the safety margin term derived from the results of the initial iteration. However, this results in a trajectory that violates both δT and $\delta\boldsymbol{\xi}$, notably a non-zero $\boldsymbol{\xi}_N^1$ with a 0.1 meter difference in the x position. In the third iteration, we incorporate the safety margin term in the same way. Additionally, we reselect a terminal state to reach by subtracting $\boldsymbol{\xi}_N^1$ from $\boldsymbol{s}_{\text{tf}}$, resulting in the final time-optimal trajectory that satisfies both δT and $\delta\boldsymbol{\xi}$ without violation. The final trajectory has a total duration of 10.315 seconds, which corresponds to a total of 258 tracking points. To assess the advantage of the robustified approach, this trajectory is then subsequently tracked by the control policy (8) for 100 times under different realizations of the process and measurement noises. The outcomes reveals that the actual trajectories of these 100 samples remain essentially within the derived uncertainty sets, satisfying the requirement on ϵ.

Regarding the computation time, we present in Fig. 2 the average results for each step when solving this case 20 times. In

this context, the initial iteration, primarily focused on solving the nominal OCP without warm starting, consumes roughly one-third of the total computation time. In the subsequent iterations, the computation time required to resolve the OCP is significantly reduced due to the warm starting provided by the previous results. Additionally, the covariance propagation step consumes some time, as it involves matrix inversion computations. Both the computation time of the covariance propagation step and the trajectory resampling step depend on the total trajectory length and Δt_{ctrl}, essentially on how finely the trajectory is resampled. To enhance computational efficiency, one can explore strategies to smartly warm start the initial solve, use more efficient solvers such as Fatrop [23], and prototype the robustified approach in a C++ environment.

B. Case Study on Robustified Replanning

In this case, we study the replanning performance of the robustified motion planning and control pipeline when handling new obstacles within a ROS simulation environment. There are three randomly positioned circular obstacles with radii ranging from 0.3m to 0.5m. The AMR is able to detect circular obstacles from a perfect bird's eye view image using the OpenCV Library [24]. The goal for the AMR is to reach the desired terminal state $\boldsymbol{s}_{\text{tf}} = [6\text{m}\ 3\text{m}\ \frac{\pi}{4}\text{rad}]^\top$ from the initial state condition $\boldsymbol{s}_{\text{t0}} = [0\text{m}\ 0\text{m}\ \frac{\pi}{2}\text{rad}]^\top$ with $\Sigma_0 = 0$, in the shortest time and without colliding with any obstacles in the field of view (FOV). To use the ASAP-MPC strategy, we select a 1.8-second update time interval. This interval strikes a balance between allowing both sufficient time for replanning computations to complete, and newly detected information to be incorporated into the replanning process promptly.

As depicted in Fig. 3, initially, only obstacle A is in the FOV, and consequently, the planned trajectory only avoids this one. Upon completion of the current planning, the pipeline promptly initiates a new planning by considering updated initial conditions and obstacle information. As an illustration, the pipeline takes into account both obstacles A and B for the first time in the 5th replanning, seamlessly integrating the replanned trajectory with the previous one to avoid both obstacles. In the 8th replanning, the pipeline takes into account the obstacle C for the first time, which blocks the trajectory to $\boldsymbol{s}_{\text{tf}}$. It results in a trajectory that avoids a collision with obstacle C and reaches a reselected feasible terminal state. A video of this simulation can be found at `https://youtu.be/qPGFleQklSo`.

In summary, the pipeline yields a robustified time-optimal trajectory by combining all replannings that avoids collisions with all three obstacles and reaches a reselected feasible terminal state. The AMR cannot track this trajectory perfectly with the feedback controller under disturbances. However, the tracking safety is guaranteed to a significant extent, as the actual AMR motion trajectory usually remains within the uncertainty sets used to derive the safety margins. The derived safety margins also give a better idea of which routes are accessible in very restrictive environments, e.g., cluttered with obstacles or consisting of very narrow passages. It can even act as a guide to decide upon a lower velocity which decreases the uncertainty and makes passages that would not be safe to pass at high velocities feasible.

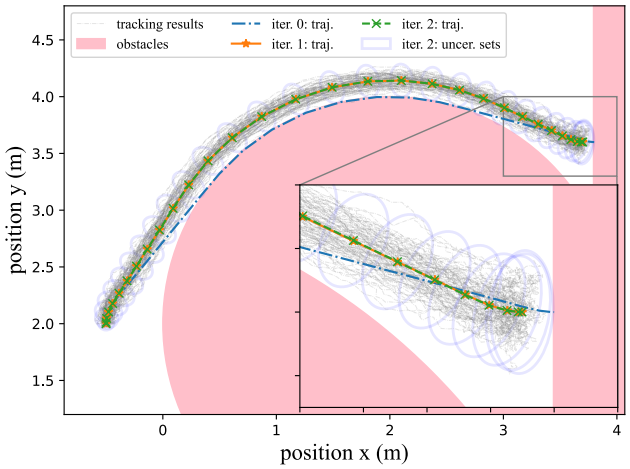

Fig. 1: Single robustified motion planning and closed-loop trajectory tracking.

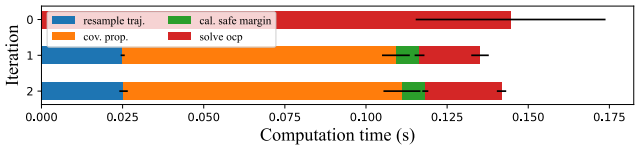

Fig. 2: Average computation time of each step in each iteration, with black bars denote the standard deviation.

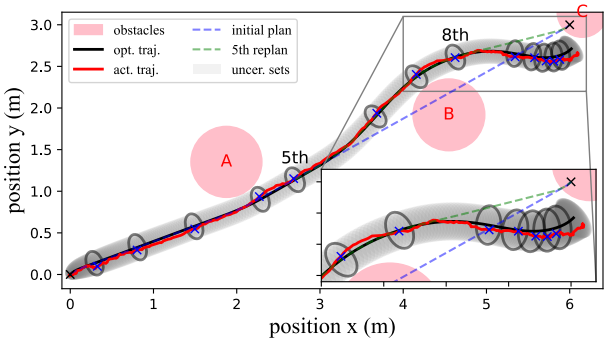

Fig. 3: Closed-loop replanning results. Bold ellipses indicate the initial uncertainty set for each replanning, within which the actual position of the corresponding time step (×) lies.

V. Conclusion

We present a robustified approach to plan time-optimal and collision-free trajectories for the AMR under disturbances that account for both process and measurement noise. This approach iteratively solves the robustified OCP by integrating state-dependent safety margins to ensure collision avoidance, and integrates a strategy to reselect a terminal state when the desired terminal state becomes infeasible. We then introduce and validate a robustified motion planning and control pipeline within a ROS simulation environment. Future work involves improving computational efficiency and implementing the proposed pipeline on an actual AMR.

References

[1] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39 830–39 846, 2020.

[2] S. Gros and M. Diehl, "Numerical optimal control," (draft manuscript). [Online]. Available: https://www.syscop.de/files/2020ss/NOC/book-NOCSE.pdf

[3] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[4] K. Berntorp, "Path planning and integrated collision avoidance for autonomous vehicles," in *2017 American Control Conference (ACC)*, 2017, pp. 4023–4028.

[5] M. Neunert, C. de Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1398–1404.

[6] B. Houska and M. Diehl, "Robustness and stability optimization of power generating kite systems in a periodic pumping mode," in *2010 IEEE International Conference on Control Applications*, 2010, pp. 2172–2177.

[7] J. Gillis and M. Diehl, "A positive definiteness preserving discretization method for lyapunov differential equations," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 7759–7764.

[8] J. Gillis, "Practical methods for approximate robust periodic optimal control ofnonlinear mechanical systems," PhD thesis, 2015.

[9] X. Feng, S. D. Cairano, and R. Quirynen, "Inexact adjoint-based sqp algorithm for real-time stochastic nonlinear mpc," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6529–6535, 2020.

[10] A. Zanelli, J. Frey, F. Messerer, and M. Diehl, "Zero-order robust nonlinear model predictive control with ellipsoidal uncertainty sets," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 50–57, 2020.

[11] Y. Gao, F. Messerer, J. Frey, N. v. Duijkeren, and M. Diehl, "Collision-free motion planning for mobile robots by zero-order robust optimization-based mpc," in *2023 European Control Conference (ECC)*, 2023, pp. 1–6.

[12] F. Messerer and M. Diehl, "An efficient algorithm for tube-based robust nonlinear optimal control with optimal linear feedback," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 6714–6721.

[13] M. Diehl, R. Findeisen, and F. Allgöwer, "A stabilizing real-time implementation of nonlinear model predictive control," in *Real-Time PDE-Constrained Optimization*. SIAM, 2007, pp. 25–52.

[14] D. Dirckx, B. Vandewal, M. Bos, W. Decré, and J. Swevers, "Applying asynchronous mpc to complex, nonlinear robotic systems in uncertain environments (asap-mpc)," in *The 42nd Benelux Meeting on Systems and Control*, 2023.

[15] D. Dirckx, M. Bos, W. Decré, and J. Swevers, "Optimal and reactive control for agile drone flight in cluttered environments," in *Proceedings of the IFAC World Congress 2023*, 2023, pp. 6837 – 6842.

[16] M. Bos, B. Vandewal, W. Decré, and J. Swevers, "Mpc-based motion planning for autonomous truck-trailer maneuvering," in *Proceedings of the IFAC World Congress 2023*, 2023, pp. 5324 – 5329.

[17] D. Telen, M. Vallerio, L. Cabianca, B. Houska, J. Van Impe, and F. Logist, "Approximate robust optimization of nonlinear systems under parametric uncertainty and process noise," *Journal of Process Control*, pp. 140–154, 2015.

[18] L. Jacobs, A. De Preter, J. Anthonis, J. Swevers, and G. Pipeleers, "H∞ controller synthesis for agv trajectory tracking using a linearized kinematic model," *IFAC-PapersOnLine*, vol. 52, pp. 61–66, 01 2019.

[19] J. Gillis, B. Vandewal, G. Pipeleers, and J. Swevers, "Effortless modeling of optimal control problems with rockit." in *The 39nd Benelux Meeting on Systems and Control*, 2020.

[20] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, p. 25–57, 2006.

[21] HSL (Harwell Subroutine Library), "A collection of fortran codes for large scale scientific computation," https://www.hsl.rl.ac.uk/, 2023, accessed on: Sep. 15, 2023.

[22] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[23] L. Vanroye, A. Sathya, J. D. Schutter, and W. Decré, "Fatrop : A fast constrained optimal control problem solver for robot trajectory optimization and control," 2023.

[24] OpenCV, "Open source computer vision library," https://github.com/opencv/opencv, 2015.