2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

# Real-Time-Feasible Collision-Free Motion Planning For Ellipsoidal Objects

Yunfan Gao[1,2], Florian Messerer[2], Niels van Duijkeren[1], Boris Houska[3], Moritz Diehl[2,4]

*Abstract*— **Online planning of collision-free trajectories is a fundamental task for robotics and self-driving car applications. This paper revisits collision avoidance between ellipsoidal objects using differentiable constraints. Two ellipsoids do not overlap if and only if the endpoint of the vector between the center points of the ellipsoids does not lie in the interior of the Minkowski sum of the ellipsoids. This condition is formulated using a parametric over-approximation of the Minkowski sum, which can be made tight in any given direction. The resulting collision avoidance constraint is included in an optimal control problem (OCP) and evaluated in comparison to the separating-hyperplane approach. Not only do we observe that the Minkowski-sum formulation is computationally more efficient in our experiments, but also that using pre-determined over-approximation parameters based on warm-start trajectories leads to a very limited increase in suboptimality. This gives rise to a novel real-time scheme for collision-free motion planning with model predictive control (MPC). Both the real-time feasibility and the effectiveness of the constraint formulation are demonstrated in challenging real-world experiments.**

## I. INTRODUCTION

Collision-free motion planning for robotic systems receives significant attention in real-world applications. Solving optimal control problems (OCPs) sufficiently fast enables a model predictive control (MPC) scheme to promptly replan online and react to the environment in a timely manner. The constraint formulation in OCPs, in particular collision avoidance constraints, greatly impacts computational efficiency.

A substantial body of work has investigated differentiable collision avoidance constraints for non-circular objects. The authors of [1] address the collision avoidance between a circle and an ellipsoid by offline predetermining an ellipsoid that covers the collision region with minimally enlarged semi-axes. The enlarged ellipsoid, however, does not provide a tight bound of the collision region in every direction. With respect to polytopic objects, the authors of [2] use duality theory to formulate collision-avoidance constraints. The impact of shape representation on computational efficiency is investigated in [3]. The authors conclude that a vertex-based representation of polyhedra results in shorter solution times than a half-space-based representation. Additionally, in [4], the authors smoothen rectangle obstacles progressively by $p$-norms over the OCP prediction horizon.

In this paper, we focus on collision avoidance between ellipsoidal objects. Ellipsoids are a helpful geometric object in collision-free motion planning, not only to approximate robot shapes, but also to model uncertainties for point objects [5]. Ellipsoidal calculus has been investigated in many different areas, e.g., ellipsoid packaging [6] and collision detection in computer graphics [7]. The distance computation between non-overlapping ellipsoids is a convex optimization problem. In the 2D case, the distance can be computed by solving a polynomial equation [8]. The optimization problem of signed-distance computation for overlapping ellipsoids is non-convex. The authors of [9] propose to compute the signed distance by identifying the points that satisfy relaxed Karush-Kuhn-Tucker (KKT) conditions and subsequently evaluating the signed distance values at these points.

To detect the overlapping of ellipsoids, the authors of [9], [10], and [7, Section 11.9.2] shrink or grow the two ellipsoids until they share exactly one point. The scale factor being greater than one indicates that the intersection is an empty set. Although this condition is useful in overlapping detection, it is not differentiable. Differentiable collision avoidance constraints are formulated in [6], [11], [12] by ensuring the existence of a separating hyperplane [13, Theorem 11.3]. To simplify the constraint formulation, the authors of [6] transform one of the two ellipsoids into a unit circle (or sphere) via an affine transformation. The collision-free condition can be ensured by imposing that the distance from the circle center to the scaled ellipsoid is no smaller than one.

In the present paper, we ensure collision avoidance by using a different necessary and sufficient condition for two ellipsoids not to overlap. The condition requires the endpoint of the vector between the center of the two ellipsoids to be located outside or on the boundary of their Minkowski sum centered at the origin. We numerically formulate this condition via a parametric over-approximation of the Minkowski sum, which can be made tight in any direction. By leaving the parameter of the over-approximation as an optimization variable, we achieve collision avoidance without introducing extra separation distance. For the resulting formulation, we observe a better computational performance compared to the separating-hyperplane approach.

Furthermore, we facilitate real-time capability by computing the over-approximation parameters based on the latest OCP solution and keeping their values fixed while solving the OCP. This gives rise to suboptimality, but values for the fixed parameters resulting in minor suboptimality can be obtained

[1] Robert Bosch GmbH, Corporate Research, Stuttgart, Germany {yunfan.gao, niels.vanduijkeren}@de.bosch.com

[2] Department of Microsystems Engineering (IMTEK), University of Freiburg, Germany {florian.messerer, moritz.diehl}@imtek.uni-freiburg.de

[3] School of Information Science and Technology, ShanghaiTech University, Shanghai, China borish@shanghaitech.edu.cn

[4] Department of Mathematics, University of Freiburg, Germany

The research that led to this paper was funded by Robert Bosch GmbH. This work was also supported by DFG via Research Unit FOR 2401, project 424107692 and 525018088, by BMWK via 03EI4057A and 03EN3054B, and by the EU via ELO-X 953348.

with little computational effort. Thereby, the computational complexity is reduced without compromising the capability of the robot navigating through cluttered environments.

The OCP formulation is evaluated in a model predictive control (MPC) scheme both in simulation and on a physical differential-drive robot. We show that planning collision-free trajectories using the proposed constraint formulation is real-time feasible. The main contributions of this work are:

1) A computationally efficient formulation of differentiable collision-avoidance constraints for ellipsoidal objects. Non-conservative collision avoidance can be achieved despite the over-approximation of the Minkowski sum of ellipsoids.
2) Improvement of real-time feasibility at an insignificant cost of robot motion inefficiency by updating the over-approximation parameters outside the OCP.
3) Experimental validation in an environment cluttered with virtual ellipsoidal obstacles.

*Notation:* In this paper, when referring to collision avoidance, we allow that two sets *touch*, but the interior of the two sets must not intersect. The interior of a set $\mathcal{B}$ is denoted by $\operatorname{int}\mathcal{B}$. The sequence of natural numbers for an interval $[a,b]$ is denoted by $\mathbb{N}_{[a,b]}$. An ellipsoid in $\mathbb{R}^n$ is a set of the form

$$\mathcal{E}(t,M) := \left\{\tau \in \mathbb{R}^n \middle| (\tau - t)M^{-1}(\tau - t) \leq 1\right\}, \tag{1}$$

where $t \in \mathbb{R}^n$ is the center of the ellipsoid and $M \in \mathbb{S}^n_{++}$ is a positive-definite matrix.

## II. MOTION PLANNING PROBLEM STATEMENT

In the following, we describe the problem set-up and define the OCP we want to solve.

### A. Robot System Dynamics

Let $x \in \mathbb{R}^{n_x}$ and $u \in \mathbb{R}^{n_u}$ be the robot system state and the control input respectively. The discrete (or discretized) system dynamics take the form

$$x_{k+1} = \psi_k(x_k, u_k),\ k \in \mathbb{N}_{[0,N-1]}. \tag{2}$$

Due to the physical limitations of the system and control design objectives such as recursive feasibility, the trajectories are subject to state-input constraints $g_k(x_k,u_k) \leq 0$ and terminal constraints $g_N(x_N) \leq 0$. The functions $\psi_k$, $g_k$, and $g_N$ are twice continuously differentiable in all arguments.

### B. Robot Shape Modeling

The robot shape is modeled by an ellipsoid $\mathcal{E}(0,G)$. The system state $x_k$ contains information on the center position $p_c : \mathbb{R}^{n_x} \to \mathbb{R}^{n_w}$, where $n_w \in \{2,3\}$ is the dimension of the physical world. The mapping from the robot state to the rotation matrix is denoted by $R : \mathbb{R}^{n_x} \to \mathbb{R}^{n_w \times n_w}$. Given the robot state $x_k$, the space occupied by the robot is a rotation and translation of the ellipsoid $\mathcal{E}(0,G)$, which is given by $\mathcal{E}(p_c(x_k), R(x_k)GR(x_k)^\top)$. To simplify notation, let $\tilde{G}(x_k) := R(x_k)GR(x_k)^\top$.

### C. Obstacle Avoidance

Consider a set of ellipsoidal obstacles: $\{\mathcal{E}(t_1,M_1), \mathcal{E}(t_2,M_2), \cdots, \mathcal{E}(t_{n_m}, M_{n_m})\}$. We aim to ensure that for each time index $k \in \mathbb{N}_{[0,N]}$ and each obstacle $m \in \mathbb{N}_{[1,n_m]}$, the interior of the robot and the interior of the obstacle do not intersect. Let $l_k(x_k,u_k)$ and $l_N(x_k)$ be the stage cost and terminal cost functions, which are twice continuously differentiable. Let $\bar{x}_0$ be the current robot state. The discrete-time optimal control problem (OCP) is formulated as follows:

$$\min_{\substack{x_0,\cdots,x_N,\\ u_0,\cdots,u_{N-1}}} \quad \sum_{k=0}^{N-1} l_k(x_k,u_k) + l_N(x_N) \tag{3a}$$

$$\text{s.t.} \quad x_0 = \bar{x}_0, \tag{3b}$$

$$x_{k+1} = \psi_k(x_k,u_k),\ k \in \mathbb{N}_{[0,N-1]}, \tag{3c}$$

$$0 \geq g_k(x_k,u_k),\ k \in \mathbb{N}_{[0,N-1]}, \tag{3d}$$

$$0 \geq g_N(x_N), \tag{3e}$$

$$\varnothing = \operatorname{int}\left(\mathcal{E}\left(p_c(x_k), \tilde{G}(x_k)\right)\right) \cap \operatorname{int}\left(\mathcal{E}(t_m, M_m)\right),$$
$$k \in \mathbb{N}_{[0,N]}, m \in \mathbb{N}_{[1,n_m]}. \tag{3f}$$

## III. COLLISION AVOIDANCE CONSTRAINT FORMULATION

In this section, we first present the preliminary knowledge related to the proposed constraint formulation and then derive the Minkowski-sum-based constraint formulation.

### A. Preliminaries

Supporting functions are an important tool for the analysis of convex sets. Minkowski sums of ellipsoids have been analyzed in many contexts, in particular reachable sets for dynamic systems and robust optimization [14]. Here we present the preliminary knowledge of the supporting function and the Minkowski sum related to our constraint formulation.

**Definition 1** (Supporting function and supporting halfspace)**.** Given a closed convex set $\mathcal{B} \subset \mathbb{R}^n$, the supporting function $V(\eta;\mathcal{B})$ and the supporting halfspace $\mathcal{H}(\eta;\mathcal{B})$ for a direction $\eta \in \mathbb{R}^n \setminus \{0\}$ are given by

$$V(\eta;\mathcal{B}) := \max_{b \in \mathcal{B}} \eta^\top b, \tag{4a}$$

$$\mathcal{H}(\eta;\mathcal{B}) := \left\{b \in \mathbb{R}^n \middle| \eta^\top b \leq V(\eta;\mathcal{B})\right\}. \tag{4b}$$

For an ellipsoid $\mathcal{E}(0,M)$, the supporting function can be expressed in closed form [14, Example 2.10]:

$$V(\eta;\mathcal{E}(0,M)) = \sqrt{\eta^\top M \eta}. \tag{5}$$

**Definition 2** (Minkowski sum)**.** The Minkowski sum of two sets $\mathcal{B} \subset \mathbb{R}^n$ and $\mathcal{D} \subset \mathbb{R}^n$ is defined as

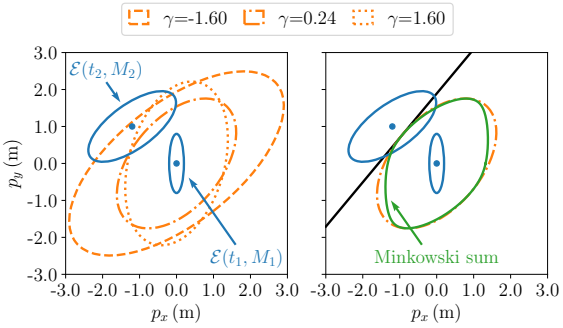
$$\mathcal{B} \oplus \mathcal{D} := \{b + d \mid b \in \mathcal{B} \text{ and } d \in \mathcal{D}\}. \tag{6}$$

**Lemma 1.** The Minkowski sum of the two ellipsoids can be over-approximated by a third ellipsoid:

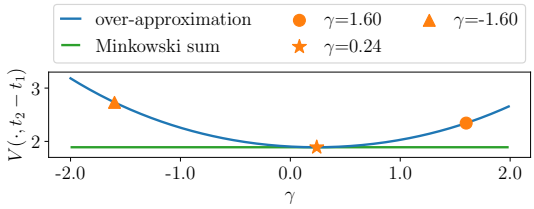
$$\mathcal{E}(0,M_1) \oplus \mathcal{E}(0,M_2) \subseteq \mathcal{E}\left(0, \frac{M_1}{\beta_1} + \frac{M_2}{\beta_2}\right), \tag{7}$$

for any $\beta_1, \beta_2 > 0$ satisfying $\beta_1 + \beta_2 = 1$.

*Proof.* The main idea of the proof is that for any direction $\eta \in \mathbb{R}^n \setminus \{0\}$, the value of the supporting function associated with the right-hand side of (7) is greater than or equal to its counterpart of the left-hand side. Therefore, the supporting halfspace of the left-hand side is a subset of its counterpart of the right-hand side. Since a closed convex set can be represented by the intersection of its supporting halfspaces,




(a) Minkowski sum (colored in green) of two ellipsoids (colored in blue) and different over-approximations (colored in orange). The solid black line depicts the boundary of the supporting halfspace of the Minkowski-sum, which coincides with the over-approximation counterpart for $\gamma = 0.24$. The intersections between the ellipsoid $\mathcal{E}(t_2, M_2)$ and the Minkowski sum (and its over-approximations) are irrelevant.




(b) The over-approximation for $\gamma = 0.24$ achieves the minimum value of the supporting function (the same value as the Minkowski sum counterpart).

Fig. 1: Illustrative example

Lemma 1 can be derived (see the detailed proof in [14, Theorem 2.4]). □

**Lemma 2.** Given one direction $\eta \in \mathbb{R}^n \setminus \{0\}$, there exists $\beta_1^*, \beta_2^* > 0$ satisfying $\beta_1^* + \beta_2^* = 1$ such that the over-approximation is tight in the direction of $\eta$:

$$V\left(\eta; \mathcal{E}\left(0, \frac{M_1}{\beta_1^*} + \frac{M_2}{\beta_2^*}\right)\right) = V\left(\eta; \mathcal{E}(0, M_1) \oplus \mathcal{E}(0, M_2)\right). \quad (8)$$

*Proof.* We refer to [15, Theorem 4.1] for the proof. □

An illustrative example demonstrating the tightness of over-approximations is shown in Fig. 1.

### B. Minkowski-sum-based Constraint Formulation

**Lemma 3.** Consider two ellipsoidal sets $\mathcal{E}(t_1, M_1)$ and $\mathcal{E}(t_2, M_2)$. The interior of the two sets does not intersect if and only if the point $t_1 - t_2$ is not in the interior of the Minkowski sum of two ellipsoids centered at the origin:

$$\begin{aligned} &\varnothing = \operatorname{int}\left(\mathcal{E}(t_1, M_1)\right) \cap \operatorname{int}\left(\mathcal{E}(t_2, M_2)\right) \Leftrightarrow \\ &t_1 - t_2 \notin \operatorname{int}\left(\mathcal{E}(0, M_1) \oplus \mathcal{E}(0, M_2)\right). \end{aligned} \quad (9)$$

*Proof.* Lemma 3 can be derived from the observation that an ellipsoid is symmetric with respect to its center:

$$t_1 - t_2 \in \operatorname{int}\left(\mathcal{E}(0, M_1) \oplus \mathcal{E}(0, M_2)\right), \quad (10a)$$

$$\Leftrightarrow \text{there exists } \tau_1, \tau_2 \in \mathbb{R}^n \text{ such that } \tau_1 + \tau_2 = t_1 - t_2,$$
$$\tau_1^\top M_1^{-1} \tau_1 < 1, \text{ and } \tau_2^\top M_2^{-1} \tau_2 < 1, \quad (10b)$$

$$\Leftrightarrow \text{there exists } \tau_1, \tau_2' \in \mathbb{R}^n \text{ such that } \tau_1 - \tau_2' = t_1 - t_2,$$
$$\tau_1^\top M_1^{-1} \tau_1 < 1, \text{ and } \left(-\tau_2'\right)^\top M_2^{-1} \left(-\tau_2'\right) < 1, \quad (10c)$$

$$\Leftrightarrow \text{there exists } \tau_1 \in \mathbb{R}^n \text{ such that } \tau_1^\top M_1^{-1} \tau_1 < 1 \text{ and}$$
$$\left(t_1 - t_2 - \tau_1\right)^\top M_2^{-1} \left(t_1 - t_2 - \tau_1\right) < 1, \quad (10d)$$

$$\Leftrightarrow \varnothing = \operatorname{int}\left(\mathcal{E}(0, M_1)\right) \cap \operatorname{int}\left(\mathcal{E}(t_1 - t_2, M_2)\right), \quad (10e)$$

$$\Leftrightarrow \varnothing = \operatorname{int}\left(\mathcal{E}(t_1, M_1)\right) \cap \operatorname{int}\left(\mathcal{E}(t_2, M_2)\right). \quad (10f)$$

□

**Corollary 4.** Given two ellipsoidal sets $\mathcal{E}(t_1, M_1)$ and $\mathcal{E}(t_2, M_2)$, the interior of the two ellipsoidal sets does not intersect if and only if there exists at least one over-approximation such that the point $t_1 - t_2$ is not in the interior of the over-approximation.

$$\begin{aligned} &\varnothing = \operatorname{int}\left(\mathcal{E}(t_1, M_1)\right) \cap \operatorname{int}\left(\mathcal{E}(t_2, M_2)\right) \Leftrightarrow \exists\, \beta_1^*, \beta_2^* > 0: \\ &\beta_1^* + \beta_2^* = 1,\ t_1 - t_2 \notin \operatorname{int}\left(\mathcal{E}\left(0, \frac{M_1}{\beta_1^*} + \frac{M_2}{\beta_2^*}\right)\right). \end{aligned} \quad (11)$$

*Proof.* Corollary 4 follows from Lemma 2 and Lemma 3. □

For computational considerations, we substitute the variables $\beta_1$ and $\beta_2$ by introducing a new variable $\gamma \in \mathbb{R}$:

$$\beta_1 = \frac{1}{1 + \exp(\gamma)} \text{ and } \beta_2 = \frac{1}{1 + \exp(-\gamma)}. \quad (12)$$

The constraints on $\beta_1$ and $\beta_2$ are satisfied for any $\gamma \in \mathbb{R}$:

$$\frac{1}{1 + \exp(\gamma)} > 0, \ \frac{1}{1 + \exp(-\gamma)} > 0,$$
$$\frac{1}{1 + \exp(\gamma)} + \frac{1}{1 + \exp(-\gamma)} = \frac{1}{1 + \exp(\gamma)} + \frac{\exp(\gamma)}{1 + \exp(\gamma)} = 1.$$

Constraint (3f) is thereby reformulated using an additional optimization variable $\gamma_{m,k}$ for each obstacle $m$ and every time index $k$ as follows:

$$\begin{aligned} 1 \leq\ &(p_c(x_k) - t_m)^\top \left(\left(1 + \exp(\gamma_{m,k})\right) \tilde{G}(x_k) \right. \\ &\left. + \left(1 + \exp(-\gamma_{m,k})\right) M_m\right)^{-1} (p_c(x_k) - t_m). \end{aligned} \quad (13)$$

**Remark 5.** This formulation can be extended to zonotopes. A zonotope can be viewed as the Minkowski sum of a set of line segments, i.e., degenerate ellipsoids whose corresponding $M$ matrices are positive *semi*-definite. The constraints for zonotope objects can thereby be formulated by taking nested over-approximations, one on the object shape and the other on the Minkowski sum of the two objects.

### C. Bounds on Optimization Variable $\gamma$

Numerical solvers may encounter difficulties when the values of $\exp(\gamma)$ and $\exp(-\gamma)$ are close to zero or very large. A bound on the variable $\gamma$ can be imposed to improve numerical robustness. In this context, we derive an appropriate bound for $\gamma$ such that the optimality is unaffected.

Recall the closed-form expression for the value of the supporting function with respect to an ellipsoid (5). For an over-approximation given by $(1 + \exp(\gamma)) M_1 + (1 + \exp(-\gamma)) M_2$, the value of $\gamma^*$ that obtains the minimum value of the supporting function is given by:

$$\gamma^*(\eta) := \arg\min \sqrt{(1 + \exp(\gamma))\, \eta^\top M_1 \eta + (1 + \exp(-\gamma))\, \eta^\top M_2 \eta}.$$

The supporting function is a convex function of $\gamma$ given a direction $\eta$. The optimal $\gamma^*(\eta)$ can be expressed in closed form:

$$\gamma^*(\eta) = \frac{1}{2} \log\left(\frac{\eta^\top M_2 \eta}{\eta^\top M_1 \eta}\right). \quad (14)$$

We have $\frac{\lambda_{\min}(M_2)}{\lambda_{\max}(M_1)} \leq \frac{\eta^\top M_2 \eta}{\eta^\top M_1 \eta} \leq \frac{\lambda_{\max}(M_2)}{\lambda_{\min}(M_1)}$, where $\lambda_{\min}$ and $\lambda_{\max}$ denote the least and the largest eigenvalues. Therefore, we

can impose a bound on $\gamma$ for numerical robustness without compromising the optimality of the over-approximation. The bound on $\gamma$ is given by

$$\left[\frac{1}{2}\log\left(\frac{\lambda_{\min}(M_2)}{\lambda_{\max}(M_1)}\right), \frac{1}{2}\log\left(\frac{\lambda_{\max}(M_2)}{\lambda_{\min}(M_1)}\right)\right]. \tag{15}$$

## IV. REAL-TIME NUMERICAL OPTIMIZATION

In this section, we present several approaches to improve the numerical properties and facilitate real-time applications.

### A. Fixed Parameterization of Over-approximation

It can be seen from Corollary 4 that for any over-approximation of the Minkowski-sum of two ellipsoids $\mathcal{E}(t_1, M_1)$ and $\mathcal{E}(t_2, M_2)$, the point $t_1 - t_2$ not being in the interior of the over-approximation is a sufficient condition for the interior of the two ellipsoidal sets not to intersect. Therefore, solving the OCP where the over-approximation, i.e., the value of $\gamma$, is fixed still yields a collision-free trajectory. Meanwhile, fixing $\gamma$ reduces the nonlinearity of the collision-avoidance constraint (13). This comes at the cost of obtaining a suboptimal solution. The degree of suboptimality depends on the proximity of the fixed over-approximation to the optimal over-approximation.

One option to determine the fixed over-approximation is to take the one that minimizes the corresponding supporting function in the direction of the robot center (given the solution of the last time step) to the obstacle center. Recall that for any given non-zero direction, a closed-form expression exists for the value of $\gamma$ achieving the minimum projection length (14). For each obstacle $m$ and each time index $k$, we compute the parameter $\hat{\gamma}_{m,k}$ for the fixed over-approximation as follows:

$$\hat{\gamma}_{m,k} = \frac{1}{2}\log\left(\frac{\eta^\top M_m \eta}{\eta^\top \tilde{G}(x_k)\eta}\right)\Bigg|_{\eta = p_c(x_k) - t_m}. \tag{16}$$

Note that the computation merely consists of several matrix-vector multiplications and one logarithm operation. The computation effort is therefore negligible compared to solving the OCPs.

### B. Regularization

The Gauss-Newton Hessian approximation is widely used in solving quadratic programming (QP) subproblems in the sequential quadratic programming (SQP) method [12, Section 10.3]. It is computationally efficient as it depends only on the first-order derivatives of the objective function. The constraint-related Hessian information is disregarded. Given that the objective function (3a) does not depend on the optimization variables $\gamma$, the Hessian blocks associated with these variables are zero. Overly optimistic Newton steps are avoided through regularization of these Hessian blocks.

## V. SIMULATION AND REAL-WORLD EXPERIMENTS

We solve the OCPs with an SQP-type solver in `acados` [16] and use HPIPM [17] as the QP solver. The simulation experiments are conducted on a laptop with an Intel i7-11850H processor and 32GB of RAM. The real-world experiments are carried out on a Neobotix MP-500 differential-drive robot. Its onboard computer is equipped with an Intel i7-7820EQ processor and 16GB of RAM.

TABLE I: OCP parameters

| Name | Unit | Symbol | Value |
|---|---|---|---|
| Prediction horizon | s | $T$ | 2.0 |
| Discretization intervals | - | $N$ | 20 |
| Robot axis length | m | | $(0.4, 0.7)$ |
| Number of obstacles | - | $n_m$ | 4 |

### A. System Dynamics, Cost Function, and Constraints

Consider a differential-drive robot modeled in a two-dimensional physical space. The robot is centered at $(p_x, p_y)$ with heading $\theta$. Forward and angular velocities are denoted by $v$ and $\omega$, respectively:

$$x := \begin{bmatrix} p_x & p_y & \theta & v & \omega \end{bmatrix}^\top \in \mathbb{R}^5.$$

The control input $u$ consists of forward acceleration $a$ and angular acceleration $\alpha$, i.e., $u := \begin{bmatrix} a & \alpha \end{bmatrix}^\top \in \mathbb{R}^2$. The continuous-time equations of motion are

$$\dot{x} = \begin{bmatrix} v\cos(\theta) & v\sin(\theta) & \omega & a & \alpha \end{bmatrix}^\top. \tag{17}$$

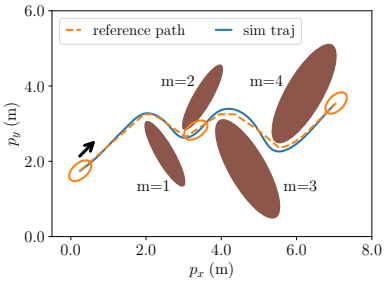
In each discretization interval, the control input is modeled as zero-order hold. The system is discretized by numerical integration (explicit Runge-Kutta integrator of order four). The objective of the OCP is to track given state and input reference trajectories $\left(x_0^{\text{ref}}, \ldots, x_N^{\text{ref}}\right)$ and $\left(u_0^{\text{ref}}, \ldots, u_{N-1}^{\text{ref}}\right)$. The stage costs and terminal costs are the weighted squared reference-tracking errors. The robot is subject to affine constraints on $v$, $\omega$, $a$, and $\alpha$ due to actuator limits. The terminal constraint requires the robot to be stationary at the terminal state, i.e., $-\epsilon_v \le v \le \epsilon_v$ and $-\epsilon_\omega \le \omega \le \epsilon_\omega$, where $\epsilon_v$ and $\epsilon_\omega$ are small positive values.

### B. Simulation Experiments

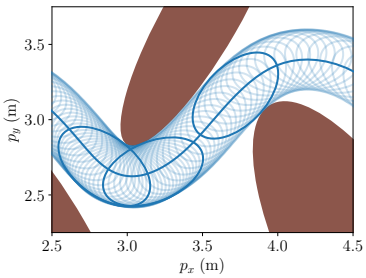
*Experiment Setting:* The Theta* algorithm [18] is employed to determine a path from a given initial position to the goal position. The generated reference path is collision-free for point objects, but not for an ellipsoidal robot (see Fig. 2a). At every time instant of the MPC simulation, we segment one part of the path based on the current robot position. The segmented path is then fitted by a spline. A time-optimal reference trajectory, which takes into account the system limitations of the robot, is subsequently generated [19]. The OCP parameters are collected in Table I.

At the first time step of the MPC simulation, the state variables are initialized by the reference trajectories. The initial guess of the control variables, as well as the additional optimization variables introduced by the collision avoidance constraints, namely, $\gamma_{m,k}$, are set to zeros. In subsequent time steps, the optimization process is warm-started using the solution from the previous time step.

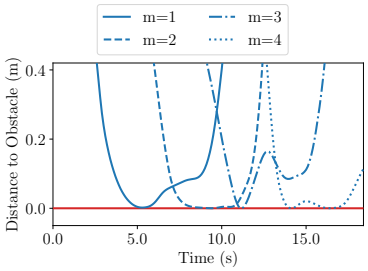
*MPC Simulation Results:* Figure 2 displays the simulated robot trajectory when we optimize the over-approximations within the OCP and iteratively solve QP subproblems until convergence. The robot safely navigates through narrow passages (see Fig. 2b) and reaches the goal position (see Fig. 2a). It is noteworthy that certain sections of the path feature exceptionally narrow passages. At around eleven seconds in the simulation, the available free space on either side of the robot is only a few centimeters.




(a) The robot would crash into the obstacles if it stayed on the reference path.




(b) The robot safely travels through the narrow passages.




(c) Distance to the obstacles over time.

Fig. 2: MPC simulation results. The over-approximations are optimized within the OCPs. The solid-line ellipsoids depict the differential-drive robot at different time steps, and the ellipsoids colored in brown depict the obstacles to avoid.

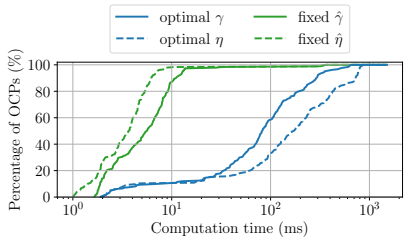



Fig. 3: Computation time for solving OCPs in MPC simulation. The time for computing the value of $\hat{\gamma}$ and $\hat{\eta}$ is excluded. The maximum number of iterations is sufficiently big for the SQP to converge.

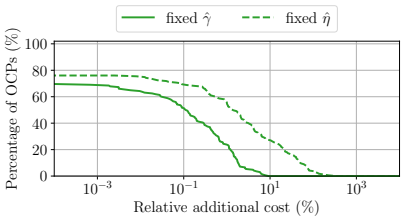



Fig. 4: Relative additional cost due to fixing the over-approximation parameters $\hat{\gamma}$ and fixing the separating hyperplane parameters $\hat{\eta}$. The lines show the percentage of OCPs whose relative additional cost exceeds certain values.

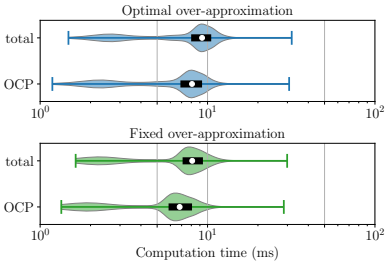



Fig. 5: Computation time in real-world experiments (maximum two QP iterations). The white circle is the median. The black bar goes from the lower to the upper quartile.

*Suboptimality:* We evaluate the suboptimality resulting from fixing the parameterization of the over-approximations in OCPs as discussed in Sec. IV-A. While running the MPC simulation with the over-approximations being optimized, we solve another OCP with the fixed over-approximations. The relative additional cost is the increase in the optimal objective due to the fixed over-approximations divided by the optimal objective value for the optimal over-approximations.

We also evaluate the suboptimality induced by the fixed separating hyperplanes, which also retain collision-freeness. To this end, let $\eta \in \mathbb{R}^{n_w}$ parameterize the separating hyperplane. The collision-avoidance constraint is formulated by $\eta^\top (p_c(x_k) - t_m) - \sqrt{\eta^\top M_m \eta} - \sqrt{\eta^\top \tilde{G}(x_k)\eta} > 0$ and $\eta^\top \eta \leq 1$. We determine the fixed hyperplane $\hat{\eta}$ by finding the two points, one in each ellipsoid, the vector between which describes the shortest vector between the two ellipsoids. The hyperplane perpendicular to this vector is chosen as the fixed hyperplane. The suboptimality induced by fixing the separating hyperplanes is notably greater than the suboptimality due to fixing the over-approximation parameters, as can be seen in Fig. 4. For fixed separating hyperplanes, the median and the worst-case increase in the cost is 1.36% and 335% respectively. In contrast, for fixed over-approximations, where the $\hat{\gamma}$ values are computed using (16), the resulting median and worst-case increase is 0.11% and 9.2% respectively.

Besides the cost comparison, we evaluate the closed-loop MPC in simulation. Determining the value of the over-approximation parameter $\hat{\gamma}$ using (16) enables the robot to safely travel through the narrow passages and reach the target position. The minimum distances to the four obstacles are summarized in Table II.

TABLE II: Minimum distance to obstacles in MPC simulation. The parameters for the over-approximations are updated outside the OCPs using (16) and fixed in the OCPs.

| Obstacle Index | $m=1$ | $m=2$ | $m=3$ | $m=4$ |
|---|---|---|---|---|
| Distance ($\mu$m) | 2007.5 | 171.2 | 67.3 | 1574.9 |

TABLE III: Computation time for solving OCPs in MPC simulation. The time for computing the value of $\hat{\gamma}$ and $\hat{\eta}$ is excluded. The maximum number of SQP iterations is two.

| Time (ms) | optimal $\gamma$ | fixed $\hat{\gamma}$ | optimal $\eta$ | fixed $\hat{\eta}$ |
|---|---|---|---|---|
| median | 1.53 | 1.24 | 1.74 | 1.16 |
| 90% | 1.72 | 1.44 | 2.22 | 1.32 |
| worst | 2.01 | 1.73 | 2.57 | 1.50 |

*Computation Time:* We assess the solution time for the OCPs including the Minkowski-sum-based constraint formulation together with a comparison to the separating-hyperplane approach. When the maximum number of iterations is sufficiently large to allow the SQP to fully converge, the Minkowski-sum-based formulation overall performs better than the separating-hyperplane approach (see Fig. 3). Fixing over-approximation parameters $\hat{\gamma}$ and fixing the separating hyperplanes $\hat{\eta}$ allow a significant decrease in the computation time. The median computation times of the optimal over-approximations and the fixed over-approximations are 81.9 ms and 5.6 ms respectively.

It is not uncommon that the optimization process is terminated in early QP iterations for real-time capability [20]. Here we evaluate the computation time when solving the OCPs for maximum two iterations. The results are reported in Table III. Fixing the over-approximations results in a decrease in the

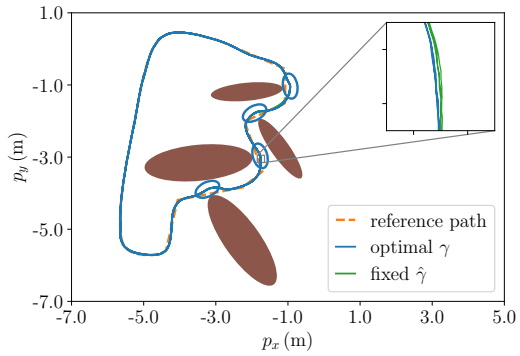



Fig. 6: Robot trajectory in real-world experiments. The blue solid-line ellipsoids depict the differential-drive robot at different time instants. The brown ellipsoids are the obstacles.

computation time by approximately 19%. While early termination greatly enhances real-time feasibility, the solutions are often not optimal and occasionally violate the constraints. In consequence, safety margins need to be incorporated to ensure collision-free motions in practice.

### C. Real-World Experiments

*Experiment Setting:* The MPC is implemented as a Nav2 controller plugin written in C++ [21]. The controller operates at a frequency of 20 Hz. The maximum number of SQP iterations is set to two. As in simulation, the reference trajectory is provided by a time-optimal path tracking module. Four virtual ellipsoidal obstacles are positioned in proximity to the path (see Fig. 6). The obstacles are slightly more separated than in simulation. The MPC has the knowledge of the ground-truth location of the obstacles while the path-tracking module that generates the reference trajectory is unaware of the obstacles. The robot dynamics, in particular the robot motor controller, is not modeled in the (17), resulting in plant-model mismatch. A safety margin of 0.01 is added to the left-hand side of the collision-avoidance constraint (13).

*Results:* The robot trajectory in the real-world experiments is plotted in Fig. 6. For both the optimal and the fixed over-approximation approaches, the robot safely follows the reference path, adjusting its trajectory when potential collisions are imminent. The two trajectories are barely distinguishable. Only in some small sections along the path, the fixed over-approximation approach yields a slightly larger distance to the obstacle compared to the optimal case (see the zoomed-in region). In the absence of collision risks, the robot adheres to the reference path. The robot completes three cycles along the reference path. The trajectories of different cycles overlay.

The computation time is reported in Fig. 5. The total computation time includes the durations for solving the OCP, generating the reference trajectory, and computing the value of $\hat{\gamma}$ for fixed over-approximation parameterization (16). The median of the total computation time is 9.27 ms for the optimal over-approximations and 8.10 ms for the fixed case. The computation manages to complete within the controller frequency, i.e., 50 ms, for our test cases.

## VI. CONCLUSIONS

This paper presented a constraint formulation for collision-free motion planning for ellipsoidal objects, achieving non-conservative collision avoidance through a parametric over-approximation of the Minkowski sum of ellipsoids. Updating the over-approximation parameters online and fixing their values in the OCPs leads to significant computation time savings while retaining the capability of navigation through narrow passages. The effectiveness of the constraint formulation is demonstrated in simulation and on an actual differentiable-drive robot. Future work might aim at analyzing the degree of suboptimality caused by the fixed over-approximations and evaluating the method for dynamic obstacles.

## REFERENCES


[1] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, pp. 4459–4466, Oct. 2019.

[2] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Trans. on Control Systems Technology*, vol. 29, pp. 972–983, May 2021.

[3] C. Dietz, S. Albrecht, A. Nurkanović, and M. Diehl, "Efficient collision modelling for numerical optimal control," in *Eur. Control Conf. (ECC)*, IEEE, June 2023.

[4] R. Reiter, K. Baumgärtner, R. Quirynen, and M. Diehl, "Progressive smoothing for motion planning in real-time NMPC," in *Eur. Control Conf. (ECC)*, IEEE, June 2024.

[5] Y. Gao, F. Messerer, J. Frey, N. van Duijkeren, and M. Diehl, "Collision-free motion planning for mobile robots by zero-order robust optimization-based MPC," in *Eur. Control Conf. (ECC)*, IEEE, June 2023.

[6] E. G. Birgin, R. D. Lobato, and J. M. Martínez, "Packing ellipsoids by nonlinear optimization," *J. of Global Optim.*, vol. 65, pp. 709–743, Dec. 2015.

[7] P. J. Schneider and D. Eberly, *Geometric Tools for Computer Graphics*. USA: Elsevier Science Inc., Sept. 2002.

[8] D. Eberly, "Distance between ellipses in 2d," Mar. 2008.

[9] S. Iwata, Y. Nakatsukasa, and A. Takeda, "Computing the signed distance between overlapping ellipsoids," *SIAM J. on Optim.*, vol. 25, pp. 2359–2384, Jan. 2015.

[10] S. Alfano and M. L. Greer, "Determining if two solid ellipsoids intersect," *J. of Guidance, Control, and Dynamics*, vol. 26, pp. 106–110, Jan. 2003.

[11] J. Kallrath and S. Rebennack, "Cutting ellipses from area-minimizing rectangles," *J. of Global Optim.*, vol. 59, pp. 405–437, Dec. 2013.

[12] J. Nocedal and S. J. Wright, *Numerical optimization*. Springer series in operations research and financial engineering, New York, NY: Springer, second ed., 2006.

[13] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, Dec. 1970.

[14] B. Houska, *Robust Optimization of Dynamic Systems*. PhD thesis, KU Leuven, 2011.

[15] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal techniques for reachability analysis of discrete-time linear systems," *IEEE Trans. on Automatic Control*, vol. 52, pp. 26–38, Jan. 2007.

[16] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Math. Program. Computation*, vol. 14, pp. 147–183, Oct. 2021.

[17] G. Frison and M. Diehl, "HPIPM: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.

[18] K. Daniel, A. Nash, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," *J. of Artificial Intelligence Res.*, vol. 39, pp. 533–579, Oct. 2010.

[19] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The Int. J. of Robotics Res.*, vol. 4, no. 3, pp. 3–17, 1985.

[20] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM J. on Control and Optim.*, vol. 43, pp. 1714–1736, Jan. 2005.

[21] S. Macenski, F. Martin, R. White, and J. Ginés Clavero, "The marathon 2: A navigation system," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2020.