

EAST: Environment-Aware Safe Tracking for Robot Navigation in Dynamic Environments

Zhichao Li^{1†}, Yinzhuang Yi^{1*†}, Zhuolin Niu¹, Nikolay Atanasov¹

¹Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, 92093, CA, USA.

*Corresponding author(s). E-mail(s): yyi@ucsd.edu;

Contributing authors: zhichaoli@ucsd.edu; zhniu@ucsd.edu; natanasov@ucsd.edu;

[†]These authors contributed equally to this work.

Abstract

This paper considers the problem of autonomous mobile robot navigation in unknown environments with moving obstacles. We propose a new method to achieve environment-aware safe tracking (EAST) of robot motion plans that integrates an obstacle clearance cost for path planning, a convex reachable set for robot motion prediction, and safety constraints for dynamic obstacle avoidance. EAST adapts the motion of the robot according to the locally sensed environment geometry and dynamics, leading to fast motion in wide open areas and cautious behavior in narrow passages or near moving obstacles. Our control design uses a reference governor, a virtual dynamical system that guides the robot's motion and decouples the path tracking and safety objectives. While reference governor methods have been used for safe tracking control in static environments, our key contribution is an extension to dynamic environments using convex optimization with control barrier function (CBF) constraints. Thus, our work establishes a connection between reference governor techniques and CBF techniques for safe control in dynamic environments. We validate our approach in simulated and real-world environments, featuring complex obstacle configurations and natural dynamic obstacle motion.

Keywords: safe robot navigation in dynamic environments, reference governor control, control barrier function

1 Introduction

Autonomous mobile robots are being increasingly integrated into human environments to support services including transportation, infrastructure inspection, cleaning, and medical assistance. Reliable robot navigation is a crucial aspect for enabling these services, leading to a growing focus on developing certifiably safe yet efficient robot navigation techniques.

In this paper, we consider the problem of safe robot navigation in unknown environments

with moving obstacles. The objective is to enable a mobile robot to navigate to a desired goal safely, potentially without prior knowledge of the environment, relying solely on onboard sensing.

The safe control problem can be approached using model predictive control (MPC) (Borrelli, Bemporad, & Morari, 2017; Bravo, Alamo, & Camacho, 2006; Dirckx et al., 2025; Y. Gao, Gray, Tseng, & Borrelli, 2014; Saccani, Cecchin, & Fagiano, 2023; Santillo & Jankovic, 2021; Wen, Dong, & Chen, 2024), quadratic programming with control barrier function (CBF) constraints

(Allibhoy & Cortés, 2024; Ames et al., 2019; Ames, Xu, Grizzle, & Tabuada, 2017; Borrmann, Wang, Ames, & Egerstedt, 2015; Chandra, Zinage, Bakolas, Stone, & Biswas, 2025; Dai et al., 2023; Mestres, Nieto-Granda, & Cortés, 2024; Squires, Pierpaoli, & Egerstedt, 2018), or Hamilton–Jacobi (HJ) reachability analysis (Bajcsy, Bansal, Bronstein, Tolani, & Tomlin, 2019; Ding, Li, Huang, & Tomlin, 2011; Fisac, Chen, Tomlin, & Sastry, 2015; Ganai, Gao, & Herbert, 2024; Margellos & Lygeros, 2011; Sharpless, Shinde, Kim, Chow, & Herbert, 2023). However, many of these methods require prior knowledge of the obstacle configuration to form safety constraints, and their extension to time-varying safety constraints is non-trivial.

In this paper, we consider the reference governor method (Bemporad, 1998; Garone, Cairano, & Kolmanovsky, 2017; Garone & Nicotra, 2016; Li, Arslan, & Atanasov, 2020; Nicotra & Garone, 2018a), which does not impose constraints on the system dynamics directly. Instead, a virtual dynamical system, called *reference governor*, is introduced to decouple the stabilization and safety objectives. The governor state acts as a desired equilibrium point for the robot, while safety constraints are enforced by controlling the evolution of the governor state by comparing the obstacle distance to the robot’s Lyapunov function. With few exceptions (Hosseinzadeh, Sinopoli, & Bobick, 2020; Miguel & Kolmanovsky, 2024), reference governor techniques have been used to achieve safe tracking only in static environments.

Our **contribution** is a new reference governor tracking control design based on optimization with Lyapunov function constraints to account for static obstacles and time-varying CBF constraints to account for dynamic obstacles. The resulting optimization is a convex quadratically constrained quadratic program, which can be solved efficiently. Our formulation establishes a connection between reference governor techniques and CBF techniques for safe control synthesis. Based on this formulation, we develop an environment-aware safe tracking (EAST) method that dynamically adjusts the governor state to avoid both static and dynamic obstacles, while guiding the robot along a reference path. EAST also integrates a motion planner with an obstacle clearance cost to periodically replan the reference path for the governor as new observations from the environment are received by the robot. Thus, EAST can guide a mobile robot

through an unknown environment, sensing obstacles online and relying on low-frequency planning and high-frequency control to make necessary maneuvers, while providing rigorous stability and safety guarantees.

We evaluate the performance of EAST extensively in both simulated and real-world environments, featuring large areas with complex obstacle configurations and natural obstacle motion. An open-source implementation of EAST is available at <https://github.com/ExistentialRobotics/EAST>.

2 Related Work

In safety-critical robotics applications, safety requirements are typically formulated as constraints in an optimization problem, aiming to synthesize control inputs or control policies for the robot. These safety requirements may arise from diverse considerations, including geometric constraints, actuator limits (Fan, Liu, & Belabbas, 2019), or more abstract task-specific and semantic-rich constraints (Nakamura, Peters, & Bajcsy, 2025). MPC (Borrelli et al., 2017) is a widely used approach for safe control synthesis, which approximates an infinite-horizon optimal control problem with a sequence of finite-horizon problems. To ensure recursive feasibility, MPC formulations typically require the constraint sets to be polytopic (Borrelli et al., 2017; Y. Gao et al., 2014) or the feasible space to be approximated with convex regions (F. Gao, Wu, Lin, & Shen, 2018). The EVA-Planner (Quan, Zhang, Zhong, Xu, & Gao, 2021) is an MPC-based method for safe tracking that is closely related to our work. To deal with general constraints, the EVA-Planner incorporates the constraints as (soft) cost terms and uses a two-level MPC formulation proposed in (Y. Gao, Lin, Borrelli, Tseng, & Hrovat, 2010). The high-level MPC uses a reduced-order model to generate a feasible reference trajectory efficiently, while the low-level MPC uses the full-order model to generate control inputs for path tracking. In contrast, our method imposes hard constraints using a reduced-order reference governor system and directly handles general (non-convex) constraints.

In the presence of dynamic obstacles, the constraints become time-varying. Applying MPC

techniques in such settings requires motion prediction of the dynamic obstacles (Brito, Floor, Ferranti, & Alonso-Mora, 2019; Lindqvist, Mansouri, Agha-mohammadi, & Nikolakopoulos, 2020). Our method also requires obstacle motion prediction but can directly handle non-convex constraints encoded as time-varying CBF on the governor system input.

CBF methods (Ames et al., 2019; Ames, Grizzle, & Tabuada, 2014; Ames et al., 2017) are increasingly used in safety-critical control applications, owing to their simplicity for ensuring safety with non-convex and dynamics-aware constraints. As discussed in Ames et al. (2014), for control-affine systems, the CBF conditions can be expressed as linear constraints in the control input. This structure enables the synthesis of safe controllers via quadratic programming. However, constructing valid CBFs remains a challenging task (Ames et al., 2019), often requiring prior knowledge of obstacles, which can limit applicability in dynamic or partially known environments. Long, Qian, Cortés, and Atanasov (2021) construct CBF constraints for safe robot navigation online from onboard range sensing. However, this approach is limited to static obstacles. Furthermore, CBF constraints derived from onboard measurements (Keyumarsi, Atman, & Gusrialdi, 2024; Long et al., 2021) typically require the system to have relative degree one. Extending such methods to higher-relative-degree systems requires non-trivial generalizations (Nguyen & Sreenath, 2016; Xiao & Belta, 2019). In contrast, our formulation imposes the CBF constraints on the governor system, which is of relative degree one by design, thus circumventing the complexity associated with higher-order CBF constructions.

Reachability-based approaches for safe robot navigation depend on accurate reachable set approximations. Various techniques have been developed to approximate reachable sets, including sum-of-squares optimization (Kousik, Vaskov, Bu, Johnson-Roberson, & Vasudevan, 2020), funnels (Burridge, Rizzi, & Koditschek, 1999), and Hamilton-Jacobi (HJ) reachability analysis (Bansal, Chen, Herbert, & Tomlin, 2017; Ding et al., 2011). Notably, HJ reachability analysis has demonstrated its effectiveness for safe robot navigation (Borquez, Chakraborty, Wang, & Bansal, 2024; Herbert et al., 2017; Seo, Lee, Son, Tomlin,

& Kim, 2019). Although time-varying constraints can be handled by the HJ reachability approach (Fisac et al., 2015; Margellos & Lygeros, 2011), the reachability computations require solving HJ partial differential equations, which is challenging for high-dimensional systems. In contrast, our approach enforces safety in dynamic environments by formulating a convex quadratically constrained quadratic program for the reduced-order reference governor system, providing a computationally efficient alternative to HJ reachability methods.

To address the challenge of satisfying safety and stability constraints simultaneously, reference governor techniques (Bemporad, 1998; Garone et al., 2017; Garone & Nicotra, 2016) assume the availability of a pre-defined control law that stabilizes the system to an arbitrary equilibrium. The method introduces a virtual reference governor system to act as an equilibrium for the original system, and safety constraints are imposed on the reference governor, thus decoupling the objectives of stabilization and safety. Reference governor methods have been successfully employed in a variety of applications, notable for ensuring safe aerial robot navigation (Convens, Merckaert, Nicotra, & Vanderborght, 2022; Tartaglione, Nicotra, Naldi, & Garone, 2024) and safe ground robot navigation in unknown environments (Arslan & Koditschek, 2017; Li & Atanasov, 2023). Gautam, Nechyporenko, Lin, Roncone, and Nicotra (2025) also extended the reference governor method to enforce kinematic and force constraints for robot manipulators, while transitioning between free motion and contact modes. However, existing applications of the reference governor are largely restricted to static environments. Hosseinzadeh et al. (2020) extended the explicit reference governor approach of (Nicotra & Garone, 2018b) to accommodate time-varying constraints, under the assumption that the evolution of the constraints is known a priori. To address this limitation, Miguel and Kolmanovsky (2024) recast the control problem with time-varying constraints as an equivalent problem with time-invariant constraints and an unmeasured disturbance, thereby enabling the application of robust reference governor techniques (Castroviejo-Fernandez, Li, Cotoruelo, Garone, & Kolmanovsky, 2024). In contrast, our method offers a simple yet effective extension to the reference governor method by incorporating time-varying CBF constraints to account for

dynamic obstacles. Our approach formulates a convex quadratically constrained quadratic program to adapt the reference governor input, enabling its application in environments with moving obstacles.

3 Problem Formulation

Consider a ground wheeled robot with differential-drive kinematics:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = G(\mathbf{x})\mathbf{u}, \quad (1)$$

where the state $\mathbf{x} = (x, y, \theta)$ consists of position $\mathbf{p} = (x, y)$ and orientation θ , while the input $\mathbf{u} = (v, \omega)$ consists of linear velocity v and angular velocity ω . The robot's body is contained in a Euclidean ball $\mathcal{B}_r(\mathbf{p}) := \{\mathbf{z} \in \mathbb{R}^2 \mid \|\mathbf{z} - \mathbf{p}\| \leq r\}$ centered at \mathbf{p} with radius r .

The robot is operating in an unknown environment $\mathcal{W} \subset \mathbb{R}^2$ containing both static and moving obstacles. The static obstacles are modeled as a closed set $\Omega \subset \mathcal{W}$. Each moving obstacle $i \in \mathcal{I} = \{1, 2, \dots, k\}$ is modeled as a ball $\mathcal{B}_{r_i}(\mathbf{p}_i)$ with radius r_i , position \mathbf{p}_i , and velocity \mathbf{v}_i such that:

$$\dot{\mathbf{p}}_i = \mathbf{v}_i, \quad (2)$$

where the velocity can be time varying.

The static obstacle space can be inflated to capture all positions such that the robot body is not contained in the free space of the static environment:

$$\Omega^+ = \bigcup_{\mathbf{q} \in \Omega} \mathcal{B}_r(\mathbf{q}). \quad (3)$$

Similarly, define an inflated obstacle space capturing both the static and dynamic obstacles:

$$\mathcal{O}^+(t) = \left(\bigcup_{i \in \mathcal{I}} \mathcal{B}_{r_i+r}(\mathbf{p}_i(t)) \right) \bigcup \Omega^+. \quad (4)$$

Definition 1 The robot is *safe* if it remains collision-free with respect to both static and moving obstacles at all times, i.e., $\mathbf{p}(t) \in \text{cl}(\mathcal{W} \setminus \mathcal{O}^+(t))$, $\forall t \geq 0$, where $\text{cl}(\cdot)$ denotes the closure of a set.

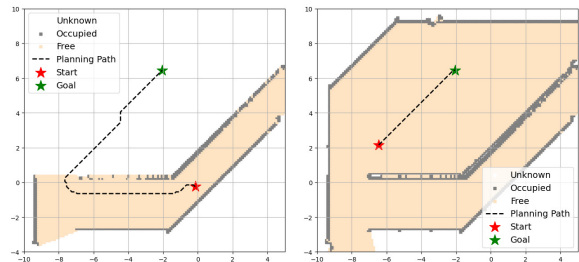


Fig. 1: Map update and path re-plan. The left figure shows the partial map and the planned path at the start. The right figure shows an updated map and a re-planned path as the robot proceeds to the goal.

The robot is equipped with a sensor, such as a LiDAR or depth camera, that provides distance measurements from the robot's position $\mathbf{p}(t)$ to the obstacles $\mathcal{O}^+(t)$. We assume that the robot is able to measure the positions $\mathbf{p}_i(t)$ and velocities $\mathbf{v}_i(t)$ of the moving obstacles (e.g., using an object tracking algorithm (Dendorfer et al., 2020)).

The quadratic distance between a point \mathbf{a} and a set \mathcal{B} is defined as:

$$d_{\mathbf{Q}}(\mathbf{a}, \mathcal{B}) := \inf_{\mathbf{b} \in \mathcal{B}} \|\mathbf{a} - \mathbf{b}\|_{\mathbf{Q}}, \quad (5)$$

where $\|\mathbf{x}\|_{\mathbf{Q}} := \sqrt{\mathbf{x}^{\top} \mathbf{Q} \mathbf{x}}$ for a symmetric positive-definite matrix \mathbf{Q} . Similarly, the quadratic distance between two sets \mathcal{A}, \mathcal{B} is defined as:

$$d_{\mathbf{Q}}(\mathcal{A}, \mathcal{B}) := \inf_{\mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}} \|\mathbf{a} - \mathbf{b}\|_{\mathbf{Q}}. \quad (6)$$

When $\mathbf{Q} = \mathbf{I}$, $d_{\mathbf{Q}}$ reduces to the Euclidean distance, and we drop the subscript to simplify the notation.

The distance measurements provided by the sensor are used to construct an occupancy map of the environment that contains free space, occupied space, and unknown space. Due to the limited sensing range of the robot, the evolving partial map is used to repeatedly re-plan a path to a desired goal location. As the robot moves, it receives new measurements, updates the map, and re-plans the path, e.g., using a motion planning algorithm such as A* (Hart, Nilsson, & Raphael, 1968) or RRT (LaValle, 1998) assuming that the unknown space is free, as illustrated in Fig. 1.

A *path* is a piecewise continuous function $\rho : [0, 1] \rightarrow \mathcal{W}$ that maps a path-length parameter



Fig. 2: Jackal unmanned ground vehicle.

$\sigma \in [0, 1]$ to the interior of the free space considered by the planner. A path starts at $\rho(0)$ and ends at $\rho(1)$.

Our objective is to design a control policy for the robot with kinematics in (1) to track a planned path $\rho(\sigma)$ while remaining safe at all times (Def. 1).

4 Robot System Design

This section overviews the robot system we used to evaluate our safe autonomous navigation method, including the robot’s computation, sensing, and communication hardware and its localization, mapping, motion planning, and control software.

4.1 Hardware Architecture

We used a Clearpath Jackal unmanned ground vehicle, shown in Fig. 2. The Jackal is a differential-drive robot with four wheels and dimensions $508 \times 430 \times 250$ mm. It weighs approximately 17 kg and runs at 2.0 m/sec top speed.

The robot was equipped with an on-board computer with an Intel i7-9700TE CPU with 32GB RAM, an Ouster OS1-32 LiDAR and a 9-axis IMU UM7. It can be controlled manually by a Bluetooth joystick or remotely by Wi-Fi access. The joystick was used for manual data collection or as an emergency stop controller. A local network, configured through a standard router, was used to monitor the status of our algorithms through visualization tools in the Robot Operating System (ROS) (Quigley et al., 2009).

4.2 Software Architecture

The software components fall in three categories: localization and mapping, planning, and control.

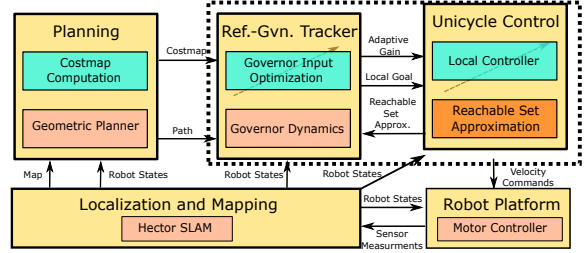


Fig. 3: System architecture overview. This paper focuses on the planning block (top left) and the control block (top right, dashed).

We used ROS to exchange messages among these components and the Gazebo physics simulator (Koenig & Howard, 2004) to test and debug our algorithms before the hardware deployment. Fig. 3 provides an overview of the software architecture.

Localization and Mapping. We used Hector SLAM (Kohlbrecher, Meyer, von Stryk, & Klingauf, 2011), which takes 2D LiDAR scans as input and provides a 2D occupancy grid map and robot poses (positions and orientations) in the map frame as output. The ROS package `pointcloud_to_laserscan` was used to convert point clouds from the Ouster LiDAR to 2D LiDAR scans.

Planning. The planning module consists of two parts: cost map construction and geometric path planning using the A^* algorithm (Hart et al., 1968). A cost map is constructed as a distance field over the occupancy grid map provided by Hector SLAM and is tuned to trade off obstacle clearance with traveled distance. Details about the cost map construction are discussed in Sec. 5.1.

Control. We used a low-level velocity controller provided by Clearpath Robotics to convert linear and angular velocity inputs to motor torque inputs. Our main contribution is the design of a trajectory tracking controller to enforce safety constraints with respect to static and dynamic obstacles. Our controller consists of two parts: (1) a stabilizing control law with associated convex reachable set over-approximation and (2) an adaptive trajectory tracker that balances stabilization and safety constraints. The control law is designed to stabilize the differential-drive robot to a desired equilibrium and to provide a convex set that contains the predicted trajectory from the initial robot state to the equilibrium state. To bridge the gap between the geometric path provided by

the planner and the point stabilization capability of the controller, we design an environment-aware safe tracker. The tracker generates local reference points along the path for the controller adaptively based on comparisons between the local safe region around the robot and the predicted controlled reachable set. On the one hand, when there are no obstacles around the robot, the reachable set can be large, corresponding to a distant reference point along the path and high robot velocity, without endangering safety. On the other hand, when there are obstacles around, the local safe space is small requiring a small reachable set to ensure safety and, hence, a nearby reference point along the path and low robot velocity.

5 Safe Tracking via Reference Governor

This section describes our cost map design for path planning (Sec. 5.1) and our control design for environment-aware safe tracking (Sec. 5.2).

5.1 Cost Function Design

Given the robot position $\mathbf{p}(t)$ at time t and an occupancy map approximating the obstacle space $\mathcal{O}^+(t)$ obtained from SLAM, we periodically re-plan a path ρ to a desired goal position \mathbf{p}^* . We assume that unknown parts of the map are free for the purpose of motion planning. The motion planning problem at time t is formulated as:

$$\begin{aligned} \min_{\rho} C(\rho), \\ \text{s.t. } \rho(0) = \mathbf{p}(t), \rho(1) = \mathbf{p}^*, \\ \rho(\sigma) \in \mathcal{W} \setminus \mathcal{O}^+(t), \forall \sigma \in (0, 1), \end{aligned} \quad (7)$$

where the constraints require that the path ρ starts at the current robot position $\mathbf{p}(t)$, ends at the goal \mathbf{p}^* , and the robot remains collision-free along the way with respect to the occupancy information $\mathcal{O}^+(t)$ at time t . We restrict the path $\rho(\sigma)$ to be a piecewise-linear function characterized by $N + 1$ vertices $\mathcal{P} := \{\rho(\sigma_i)\}_{i=0}^N$ defined as:

$$\rho(\sigma) := \rho(\sigma_i) + \frac{\sigma - \sigma_i}{\sigma_{i+1} - \sigma_i}(\rho(\sigma_{i+1}) - \rho(\sigma_i)), \quad (8)$$

for $\sigma_i \leq \sigma < \sigma_{i+1}$.

The motion planning problem in (7) can be solved by a variety of motion planning algorithms, such as A^* (Hart et al., 1968), PRM (Kavraki, Kolountzakis, & Latombe, 1998) and RRT (LaValle, 1998). We focus on designing a cost function $C(\rho)$ to trade off obstacle clearance and travel distance of the form:

$$C(\rho) := \sum_{i=0}^N \|\rho(\sigma_{i+1}) - \rho(\sigma_i)\| + c(\rho(\sigma_{i+1})), \quad (9)$$

where the first term captures travel distance and the second term captures obstacle clearance. The *obstacle clearance* term $c: \mathbb{R}^2 \rightarrow \mathbb{R}$ is defined as:

$$c(\mathbf{p}) := c_u \exp(-\kappa d(\mathbf{p}, \mathcal{O}^+)), \quad (10)$$

where $c_u > 0$ is a parameter placing an upper bound on the clearance cost term and $\kappa > 0$ is an exponential decay rate parameter. One way to obtain the distance $d(\mathbf{p}, \mathcal{O}^+)$ in practice is via a distance transform (Bradski, 2000) applied to the occupancy map. Since $c(\mathbf{p})$ in (9), (10) is positive, the Euclidean distance $h(\mathbf{p}) = \|\mathbf{p} - \mathbf{p}^*\|$ is an admissible and consistent heuristic (Likhachev, Gordon, & Thrun, 2004) for the motion planning problem in (7).

The safety constraint $\rho(\sigma) \in \mathcal{W} \setminus \mathcal{O}^+(t)$ in (7) can be enforced by setting a minimum acceptable obstacle clearance $c(\rho(\sigma)) \leq c_f$ and considering positions with obstacle clearance cost larger than c_f occupied.

Proposition 1 Consider the safety constraint $\rho(\sigma) \in \mathcal{W} \setminus \mathcal{O}^+(t)$ in (7) and the obstacle clearance cost defined in (10). Let $c_f < c_u$. The safety constraint in (7) can be reformulated as:

$$c(\rho(\sigma)) \leq c_f, \quad \forall \sigma \in (0, 1). \quad (11)$$

Proof By the definition of the obstacle clearance cost in (10), $c(\rho(\sigma)) \leq c_f < c_u$ implies $d(\mathbf{p}, \mathcal{O}^+) > 0$. \square

The obstacle clearance cost with $c_u = 8.3$ and $\kappa = 7$ is visualized in Fig. 4b. Based on Fig. 4b, we set $c_f = 5$ to ensure $\rho(\sigma) \in \mathcal{W} \setminus \mathcal{O}^+(t)$. An occupancy grid map and the corresponding obstacle clearance cost are shown in Fig. 4c and Fig. 4d.

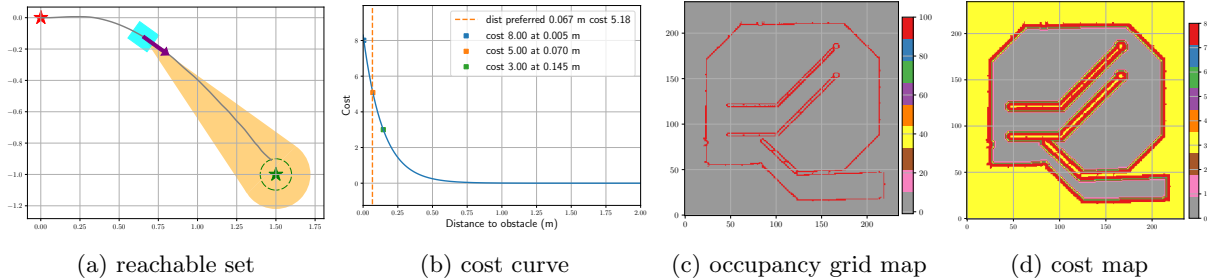


Fig. 4: Ice-cream cone approximation (orange) of the reachable set for a differential-drive system (1) with control law in (12) is shown in (a). The start and goal positions are depicted as red and green stars, respectively. The robot is shown as a cyan rectangle with a purple arrow showing its orientation. The obstacle clearance cost design for motion planning is shown in (b). Note that the obstacle clearance (10) is a function of \mathbf{p} , but we plot the value of $c(\mathbf{p})$ versus $d(\mathbf{p}, \mathcal{O}^+)$. An occupancy grid map of a simulated environment is shown in (c), with cell values equal to the probability (in percent) of being occupied by obstacles. The corresponding clearance cost is shown in (d), with clearance cost set to 3 for unknown regions in the occupancy grid map.

5.2 Reference Governor Safe Tracker

Given a path $\rho(\sigma)$ obtained from the motion planning problem in (7), our next objective is to design a control policy for the differential-drive robot in (1) to track $\rho(\sigma)$ subject to the safety constraint in (11). We first present existing results considering the case where no moving obstacles are present in Sec. 5.2.1. Our main contribution is presented in Sec. 5.2.2, where we extend the existing results to dynamic environments.

5.2.1 Safe Tracking in Static Environments

We focus on designing a control law to track a reference path $\rho(\sigma)$ subject to the safety constraint $\mathbf{p}(t) \in \mathcal{W} \setminus \Omega^+$ that the robot remains in the free space of the static environment for all t . First, we consider stabilization to a point \mathbf{p}^* in the absence of constraints and approximate the reachable set of the robot. We use the control policy proposed by Isleyen, van de Wouw, and Arslan (2023):

$$\mathbf{k}(\mathbf{x}, \mathbf{p}^*) = \begin{bmatrix} v(\mathbf{x}, \mathbf{p}^*) \\ \omega(\mathbf{x}, \mathbf{p}^*) \end{bmatrix} = \begin{bmatrix} k_v e_v \\ k_\omega \arctan(e_v^\perp / e_v) \end{bmatrix}, \quad (12)$$

where $\mathbf{x} = (\mathbf{p}, \theta)$ is the robot state, $k_v > 0$ and $k_\omega > 0$ are control gains for the linear and angular velocities, and the error terms e_v and e_v^\perp are

defined as:

$$e_v = \begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}^\top (\mathbf{p}^* - \mathbf{p}), \quad e_v^\perp = \begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}^\top (\mathbf{p}^* - \mathbf{p}).$$

We set $\omega(\mathbf{x}, \mathbf{p}^*) = 0$ when $\mathbf{p} = \mathbf{p}^*$.

Proposition 2 (Isleyen et al. (2023, Lemma 1)) *The control law in (12) applied to the system in (1) asymptotically steers all initial states (\mathbf{p}_0, θ_0) in $\mathbb{R}^2 \times [-\pi, \pi)$ to a given goal position $\mathbf{p}^* \in \mathbb{R}^2$, i.e., the closed-loop trajectory $(\mathbf{p}(t), \theta(t))$ satisfies $\lim_{t \rightarrow \infty} \mathbf{p}(t) = \mathbf{p}^*$.*

Accurate prediction of the closed-loop robot trajectory is important for fast and safe path following. However, the closed-loop system is nonlinear and predicting its trajectory requires numerical integration. An alternative is to compute a reachable set that provides bounds on the possible trajectories. Isleyen et al. (2023) show that trajectories generated by (1) under control law (12) are contained in a reachable set with ice-cream cone shape, illustrated in Fig. 4a.

Proposition 3 (Isleyen et al. (2023, Proposition 4)) *Consider the set $\mathcal{M}(\mathbf{x}, \mathbf{p}^*) := \mathcal{R}(\mathbf{p}, \mathbf{p}^*, \|e_v^\perp\|)$, where:*

$$\mathcal{R}(\mathbf{a}, \mathbf{b}, r) := \{\mathbf{a} + \alpha(\mathbf{z} - \mathbf{a}) \mid \alpha \in [0, 1], \mathbf{z} \in \mathcal{B}_r(\mathbf{b})\}.$$

For any given goal position \mathbf{p}^ and any initial condition $\mathbf{x}_0 = (\mathbf{p}_0, \theta_0)$, the control law (12) applied to system (1) renders $\mathcal{M}(\mathbf{x}_0, \mathbf{p}^*) \times [-\pi, \pi)$ forward invariant.*

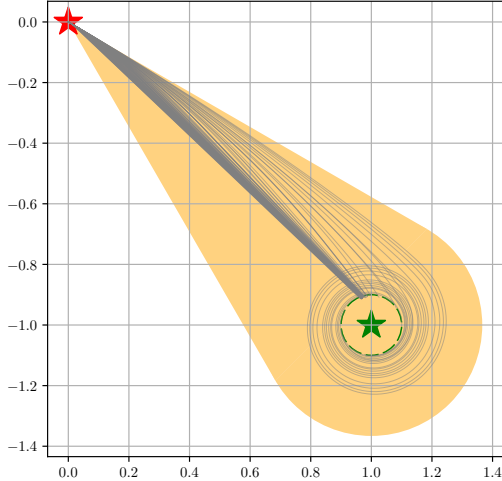


Fig. 5: Simulation of differential-drive trajectories (gray curves) under the control law in (12) with $k_v = 2.0$, $k_\omega = 5.0$, and control bounds $|v| \leq v_{max}$ and $|\omega| \leq \omega_{max}$ such that $v_{max}, \omega_{max} \in \{0.1, 0.2, \dots, 1.0\}$. The robot position and goal position are depicted as red and green stars, respectively. The reachable set derived in Prop. 3 is shown as a yellow region. To ease visualization, the trajectories are terminated when they enter a ball of radius 0.1 centered at the goal (green dashed circle). Despite variations in the control limits, all trajectories remain entirely within the reachable set, illustrating that the reachable set depends solely on the robot’s position and the governor’s state.

The reachable set approximation $\mathcal{M}(\mathbf{x}, \mathbf{p}^*)$ is convex, making distance computations efficient. Additionally, it is more accurate than ellipsoidal reachable set approximations derived from the Lyapunov function associated with (12), resulting in less conservative distance computation.

As stated in Prop. 3, the reachable set prediction depends only on the robot position and the goal position, and is independent of the particular control gains k_v and k_ω . We illustrate this with a simulation using fixed control gains $k_v = 2.0$ and $k_\omega = 5.0$ and varying control bounds $|v| \leq v_{max}$ and $|\omega| \leq \omega_{max}$, with $v_{max}, \omega_{max} \in \{0.1, 0.2, \dots, 1.0\}$. We generated 100 differential-drive trajectories using different combinations of the control bounds. The results are shown in Fig. 5. We observe that, for a fixed linear velocity bound, as the angular velocity bound approaches

zero ($\omega_{max} \rightarrow 0$), the trajectory lies closer to the boundary of the predicted reachable set. In contrast, for larger angular velocity bounds, the trajectory tends to approximate a straight line connecting the starting point to the goal point.

Given the controller in (12) and the reachable set approximation in Prop. 3, we enforce the safety constraint $\mathbf{p}(t) \in \mathcal{W} \setminus \Omega^+$ using the reference governor method (Arslan & Koditschek, 2017; Nicotra & Garone, 2018b). A *reference governor* is a first-order virtual system with dynamics:

$$\dot{\mathbf{g}} = -k_g(\mathbf{g} - \mathbf{u}_g), \quad (13)$$

where $\mathbf{g} \in \mathbb{R}^2$ is the governor state, $\mathbf{u}_g \in \mathbb{R}^2$ is the governor input, and $k_g > 0$ is a parameter. A governor is used to decouple the stabilization of a dynamical system from the enforcement of safety constraints.

We use the governor state \mathbf{g} as the reference point in the stabilizing controller $\mathbf{k}(\mathbf{x}, \mathbf{g})$ in (12). This way, the closed-loop system aims to reach the governor state, while the governor can move along the reference path $\rho(\sigma)$ in a way to ensure that the reachable set $\mathcal{M}(\mathbf{x}, \mathbf{g})$ remains in free space. To satisfy the safety constraint, the input \mathbf{u}_g of the governor system (13) must be chosen by considering the size of the free space Ω^+ in comparison to the size of the reachable set approximation $\mathcal{M}(\mathbf{x}, \mathbf{g})$. We measure the difference in the sizes of these sets by a distance metric $d(\mathcal{M}(\mathbf{x}, \mathbf{g}), \Omega^+)$ and use it to define a set of feasible governor inputs, termed local safe zone.

Definition 2 A *local safe zone* is a set, determined by the joint system-governor state (\mathbf{x}, \mathbf{g}) and the distance $d(\mathcal{M}(\mathbf{x}, \mathbf{g}), \Omega^+)$ between the reachable set $\mathcal{M}(\mathbf{x}, \mathbf{g})$ and the static obstacle space Ω^+ :

$$\mathcal{LS}(\mathbf{x}, \mathbf{g}) := \left\{ \mathbf{q} \in \mathbb{R}^2 \mid \|\mathbf{q} - \mathbf{g}\|^2 \leq d(\mathcal{M}(\mathbf{x}, \mathbf{g}), \Omega^+) \right\}.$$

The definitions of the local safe zone and governor input are illustrated in Fig. 6. The governor input at system-governor state (\mathbf{x}, \mathbf{g}) is chosen as the point \mathbf{u}_g in the local safe zone $\mathcal{LS}(\mathbf{x}, \mathbf{g})$ that is furthest along the reference path ρ :

$$\mathbf{u}_g = \rho(\bar{\sigma}), \quad \bar{\sigma} = \arg \max_{\sigma \in [0,1]} \{\sigma \mid \rho(\sigma) \in \mathcal{LS}(\mathbf{x}, \mathbf{g})\}. \quad (14)$$

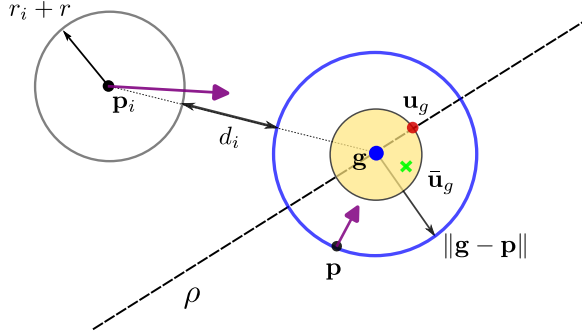


Fig. 6: Geometric relationship between the local safe zone \mathcal{LS} (yellow ball), governor input \mathbf{u}_g (red dot) and modified governor input $\bar{\mathbf{u}}_g$ (green cross). The blue ball $\mathcal{B}_{\|\mathbf{g}-\mathbf{p}\|}(\mathbf{g})$ represents an outer approximation of the reachable set $\mathcal{M}(\mathbf{x}, \mathbf{g})$, the gray ball $\mathcal{B}_{r_i+r}(\mathbf{p}_i)$ represents the inflated obstacle space for moving obstacle i , and d_i is the distance between the two balls.

Combining the system in (1) with controller $\mathbf{k}(\mathbf{x}, \mathbf{g})$ in (12) and the governor dynamics (13) with controller in (14), we have the closed-loop system-governor dynamics:

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} G(\mathbf{x})\mathbf{k}(\mathbf{x}, \mathbf{g}) \\ -k_g\mathbf{g} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ k_g I \end{bmatrix} \mathbf{u}_g = A(\mathbf{x}, \mathbf{g}) + B\mathbf{u}_g. \quad (15)$$

The closed-loop system-governor dynamics satisfy the safety constraint $\mathbf{p}(t) \subset \mathcal{W} \setminus \Omega^+$.

Proposition 4 *Given a reference path ρ , consider the closed-loop system-governor dynamics in (15). Suppose that the initial state $(\mathbf{x}_0, \mathbf{g}_0)$ satisfies:*

$$d(\mathcal{M}(\mathbf{x}_0, \mathbf{g}_0), \Omega^+) > 0, \quad \mathbf{g}_0 = \rho(0) \in \mathcal{W} \setminus \Omega^+. \quad (16)$$

Then, the position $\mathbf{p}(t)$ converges to the end of the path $\rho(1)$ without collision, i.e., $\mathbf{p}(t) \in \mathcal{W} \setminus \Omega^+, \forall t \geq 0$.

Proof The proof follows from the results in Arslan (2022); Isleyen et al. (2023); Li et al. (2020) and the references therein. It consists of two parts. First, we show that the closed-loop dynamics in (15) ensure the safety of the system position, i.e., $\mathbf{p}(t) \in \mathcal{W} \setminus \Omega^+, \forall t \geq 0$. Second, we establish that the system position asymptotically converges to the endpoint of the reference path.

Since the distance between the reachable set and the static obstacle space $d(\mathcal{M}(\mathbf{x}, \mathbf{g}), \Omega^+)$ is Lipschitz continuous (Isleyen et al., 2023, Proposition 9), one can

consider a partition of the time interval $[0, \infty)$ based on distinct time instances $(t_0 = 0, t_1, \dots, t_i, \dots)$ with $t_i < t_{i+1}$ such that $d(\mathcal{M}(\mathbf{x}(t), \mathbf{g}(t)), \Omega^+)$ is either positive or zero over $[t_i, t_{i+1})$, and alternates between these cases across the partition. Given such a time interval $[t_i, t_{i+1})$, we distinguish two cases.

- If $d(\mathcal{M}(\mathbf{x}(t), \mathbf{g}(t)), \Omega^+) > 0$ for any $t \in [t_i, t_{i+1})$, then both the reachable set and the robot position are in the interior of the free space, i.e., $\mathcal{M}(\mathbf{x}(t), \mathbf{g}(t)) \subset \text{int}(\mathcal{W} \setminus \Omega^+)$ and $\mathbf{p}(t) \in \mathcal{W} \setminus \Omega^+$, because, by definition $\mathbf{p}(t) \in \mathcal{M}(\mathbf{x}(t), \mathbf{g}(t))$.
- If $d(\mathcal{M}(\mathbf{x}(t), \mathbf{g}(t)), \Omega^+) = 0$ for any $t \in [t_i, t_{i+1})$, we have $\text{int}(\mathcal{M}(\mathbf{x}(t), \mathbf{g}(t))) \cap \text{int}(\Omega^+) = \emptyset$. To show this, notice that the distance between the reachable set and the static obstacle space $d(\mathcal{M}(\mathbf{x}, \mathbf{g}), \Omega^+)$ is Lipschitz continuous and was strictly positive in the previous time interval, i.e., $d(\mathcal{M}(\mathbf{x}(t), \mathbf{g}(t)), \Omega^+) > 0$ for any $t \in [t_{i-1}, t_i)$. As a result, at time $t = t_i$, the reachable set and the obstacle space are in contact at their boundaries without overlapping interiors: $\text{int}(\mathcal{M}(\mathbf{x}, \mathbf{g})) \cap \text{int}(\Omega^+) = \emptyset$, and $\partial\mathcal{M}(\mathbf{x}, \mathbf{g}) \cap \partial\Omega^+ \neq \emptyset$, where $\partial(\cdot)$ denotes the boundary of a set. Hence, it follows from $\dot{\mathbf{g}} = \mathbf{0}$ over $[t_i, t_{i+1})$ and Prop. 3 that under the proposed control policy (12), the robot position trajectory satisfies $\mathbf{p}(t) \in \mathcal{M}(\mathbf{x}(t), \mathbf{g}(t)) \subset \text{cl}(\mathcal{W} \setminus \Omega^+)$ for all $t \in [t_i, t_{i+1})$.

Thus, the position of the system is collision-free under the closed-loop dynamics (15).

To establish convergence to the goal position, note that $\dot{\mathbf{g}} = \mathbf{0}$ if and only if $\mathbf{g} = \rho(1)$ or $d(\mathcal{M}(\mathbf{x}, \mathbf{g}), \Omega^+) = 0$. To conclude that $\rho(1)$ is the only stable governor state under governor dynamics (13), observe that condition $d(\mathcal{M}(\mathbf{x}, \mathbf{g}), \Omega^+) = 0$ might be zero only for a finite time. Since the distance between the system position and the governor state $\|\mathbf{p} - \mathbf{g}\|$ is Lipschitz continuous and asymptotically decays to zero under control policy (12), as shown in Prop. 2. Moreover, the reference path lies in the interior of the free space, i.e., $d(\rho(\omega), \Omega^+) > 0$ for all $\omega \in [0, 1]$. Hence, the governor dynamics might be zero only for a finite duration away from $\rho(1)$, and always stays strictly nonzero for at least some finite time. Thus, since the control policy (12) is point stabilizing, it follows from LaSalle's invariance principle that both the governor state and the system position asymptotically converge to the end of the reference path, which completes the proof of convergence. \square

To make the robot system converge to the goal efficiently, the gain parameter k_v in (12)

should be well chosen. Based on results from our previous work (Li et al., 2020), we develop an adaptive gain that considers the heading of the robot with respect to the surrounding environment. We define a directional distance metric that penalizes obstacles along the robot’s heading direction $\mathbf{v} = [\cos \theta, \sin \theta]^\top$ as the quadratic distance between the reachable set and the obstacle space $d_{\mathbf{Q}[\mathbf{v}]}(\mathcal{M}(\mathbf{x}, \mathbf{g}), \Omega^+)$ where

$$\mathbf{Q}[\mathbf{v}] = q_2 \mathbf{I} + (q_1 - q_2) \frac{\mathbf{v}\mathbf{v}^\top}{\|\mathbf{v}\|^2}, \quad (17)$$

with scalars $q_2 > q_1 > 0$. We obtain an adaptive control gain by computing the ratio of the directional distance metric to the regular one:

$$k_v := \begin{cases} \frac{d_{\mathbf{Q}[\mathbf{v}]}(\mathcal{M}(\mathbf{x}, \mathbf{g}), \Omega^+)}{d(\mathcal{M}(\mathbf{x}, \mathbf{g}), \Omega^+)}, & d(\mathcal{M}(\mathbf{x}, \mathbf{g}), \Omega^+) > 0, \\ 1, & \text{otherwise.} \end{cases} \quad (18)$$

Remark 1 The gain k_v in (18) is uniformly bounded. Following the proof by Isleyen et al. (2023), one can verify that the reachable set prediction in Prop. 3 does not change if $k_v > 0$. As a result, Thm. 4 still holds, indicating that this choice of gain does not endanger safety.

The reference governor design guides the robot to navigate safely and efficiently in static environments. However, when moving obstacles are present, choosing the governor input \mathbf{u}_g along the reference path according to (14) is not sufficient to guarantee safety with respect to the moving obstacles. In this case, we need to modify the governor input \mathbf{u}_g , considering both static and moving obstacles. This motivates our extension of the reference governor formulation to dynamic environments.

5.2.2 Safe Tracking in Dynamic Environments

When moving obstacles are present, the robot may need to deviate from its planned path to ensure safety. To achieve this, we choose an appropriate input \mathbf{u}_g for the governor in (13), which guides the robot to execute avoidance maneuvers.

Inspired by safe control synthesis techniques using CBFs (Ames et al., 2019), we formulate a

convex optimization problem to modify the governor input \mathbf{u}_g in a minimally invasive way to avoid moving obstacles. We define a CBF that encodes the safety of the system-governor system with respect to the i th moving obstacles. Consider a ball $\mathcal{B}_{\|\mathbf{p}-\mathbf{g}\|}(\mathbf{g})$ centered at the governor that contains the robot position \mathbf{p} and a ball $\mathcal{B}_{r_i+r}(\mathbf{p}_i)$ centered at the i th obstacle position \mathbf{p}_i with radius r_i inflated by the robot radius r . According to Isleyen et al. (2023, Prop. 8), the reachable set $\mathcal{M}(\mathbf{x}, \mathbf{g})$ is a subset of the ball $\mathcal{B}_{\|\mathbf{p}-\mathbf{g}\|}(\mathbf{g})$. Hence, the distance between the reachable set $\mathcal{M}(\mathbf{x}, \mathbf{g})$ and the inflated i th moving obstacle is lower bounded by the distance between the two balls, which can be computed as the distance between their centers $\|\mathbf{g} - \mathbf{p}_i\|$ minus the radii:

$$d_i = \|\mathbf{g} - \mathbf{p}_i\| - \|\mathbf{g} - \mathbf{p}\| - r_i - r.$$

This is illustrated in Fig. 6. Due to non-differentiability of d_i at $\|\mathbf{g} - \mathbf{p}\| = 0$, the distance d_i can not be used as a CBF directly. However, note that for $a \geq 0$ and $b > 0$, $(a - b)(a + b) = a^2 - b^2 \geq 0$ implies $a - b \geq 0$ since $a + b > 0$. Hence, a candidate CBF can be constructed by setting $a = \|\mathbf{g} - \mathbf{p}_i\|$ and $b = \|\mathbf{g} - \mathbf{p}\| + r_i + r$:

$$h_i(\mathbf{x}, \mathbf{g}, \mathbf{p}_i) = \|\mathbf{g} - \mathbf{p}_i\|^2 - (r_i + r + \|\mathbf{g} - \mathbf{p}\|)^2. \quad (19)$$

Thus, a CBF constraint for moving obstacle i can be formulated as:

$$s_i(\mathbf{u}_g) := \dot{h}_i(\mathbf{x}, \mathbf{g}, \mathbf{p}_i, \mathbf{u}_g) + \alpha_i(h_i(\mathbf{x}, \mathbf{g}, \mathbf{p}_i)) \geq 0, \quad (20)$$

where $\alpha_i(\cdot)$ is a class- \mathcal{K} function to be designed and the time derivative of the CBF is:

$$\dot{h}_i(\mathbf{x}, \mathbf{g}, \mathbf{p}_i, \mathbf{u}_g) = \left[\frac{\partial h_i}{\partial \mathbf{x}} \quad \frac{\partial h_i}{\partial \mathbf{g}} \right] (A(\mathbf{x}, \mathbf{g}) + B\mathbf{u}_g) + \frac{\partial h_i}{\partial \mathbf{p}_i} \mathbf{v}_i$$

with $A(\mathbf{x}, \mathbf{g})$ and B defined in (15).

The CBF constructed above does not take static obstacles into account. According to our discussion in the previous subsection, a governor input that satisfies static obstacle constraints has to lie in the local safe zone (Def. 2). Thus, we formulate an optimization problem to find a modified governor input $\bar{\mathbf{u}}_g$ that deviates minimally from the desired governor input \mathbf{u}_g in (14) and *both* lies within the local safe zone, $\bar{\mathbf{u}}_g \in \mathcal{LS}(\mathbf{x}, \mathbf{g})$, and

satisfies the CBF constraints, $s_i(\bar{\mathbf{u}}_g) \geq 0$:

$$\begin{aligned} & \min_{\bar{\mathbf{u}}_g \in \mathbb{R}^2} \|\bar{\mathbf{u}}_g - \mathbf{u}_g\|^2 \\ \text{s.t. } & s_i(\bar{\mathbf{u}}_g) \geq 0, \quad \forall i \in \mathcal{I} \\ & \bar{\mathbf{u}}_g \in \mathcal{LS}(\mathbf{x}, \mathbf{g}). \end{aligned} \quad (21)$$

Note that the functions s_i in (20) are linear in $\bar{\mathbf{u}}_g$, while the last constraint is quadratic in $\bar{\mathbf{u}}_g$. Hence, (21) is a convex quadratically constrained quadratic program (QCQP), which can be solved in polynomial time (Park & Boyd, 2017).

When moving obstacles endanger the robot's motion, the modified governor input $\bar{\mathbf{u}}_g$ guides the governor and subsequently the robot to deviate from the path $\rho(\sigma)$ to reduce collision risk. An illustrative example is shown in Fig. 6, where a modified governor input $\bar{\mathbf{u}}_g$ is found to avoid an incoming obstacle.

According to Thm. 4, the proposed control synthesis method is guaranteed to be safe in a static environment. When moving obstacles are present, ensuring safety depends on both the robot's motion and the obstacles' motion, and the local safe zone should remain non-empty $\mathcal{LS}(\mathbf{x}, \mathbf{g})$. According to the local safe zone definition (Def. 2), the modified governor input can be decomposed as $\bar{\mathbf{u}}_g = \mathbf{g} + \mathbf{e}$, where $\mathbf{e} \in \mathcal{B}_{d(\mathcal{M}, \Omega^+)}(\mathbf{0})$. The governor dynamics can therefore be expressed as $\dot{\mathbf{g}} = k_g \mathbf{e}$ and the CBF constraint for moving obstacle i becomes:

$$\frac{\partial h_i}{\partial \mathbf{g}} \mathbf{e} + \frac{\partial h_i}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial h_i}{\partial \mathbf{p}_i} \mathbf{v}_i + \alpha_i(h_i) \geq 0. \quad (22)$$

Breeden and Panagou (2023) show that if the CBFs are non-interfering, i.e., $\nabla_{\mathbf{g}}^\top h_i \nabla_{\mathbf{g}} h_j \geq 0$ for all $i, j \in \mathcal{I}$, then the optimization problem (21) remains feasible provided that, for each $i \in \mathcal{I}$:

$$-\|\nabla_{\mathbf{g}} h_i\| \leq \sqrt{2} \frac{\frac{\partial h_i}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial h_i}{\partial \mathbf{p}_i} \mathbf{v}_i + \alpha_i(h_i)}{k_g d(\mathcal{M}, \Omega^+)}. \quad (23)$$

The non-interfering condition on the CBFs precludes cases where the CBF gradients with respect to \mathbf{g} point in opposite directions. In particular, the CBF constraints in (21) can be stacked into a linear system of inequalities: $A\bar{\mathbf{u}}_g \leq \mathbf{b}$. According to Farkas' Lemma Bertsimas and Tsitsiklis (1997), infeasibility occurs if there exists a vector $\mathbf{y} \geq \mathbf{0}$ such that $\mathbf{y}^\top A = 0$ and $\mathbf{y}^\top \mathbf{b} < 0$, regardless of the

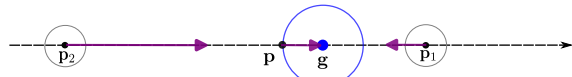


Fig. 7: Illustration of the case where the optimization problem (21) becomes infeasible in the presence of two moving obstacles. The moving obstacles are shown as gray circles centered at \mathbf{p}_1 and \mathbf{p}_2 , respectively. The robot position is denoted by \mathbf{p} , and the governor position by \mathbf{g} . Purple arrows indicate the velocity vectors of the robot and the moving obstacles, reflecting both their directions and magnitudes.

size of the local safe zone, as illustrated in Fig. 7. In the case of two moving obstacles, this condition implies that the CBF gradients with respect to \mathbf{g} must point in opposite directions for infeasibility to occur.

The condition in (23) further characterizes the size of the local safe zone required for the robot to have sufficient leeway to avoid the moving obstacles, given the configuration of the robot and the moving obstacles. If the local safe zone is too small, e.g., when the robot is close to static obstacles, it becomes difficult to avoid moving obstacles. Conversely, in a relatively empty environment, the condition in (23) is satisfied as long as $|\nabla_{\mathbf{g}} h_i| \neq 0$, which corresponds to the case where the robot position coincides with the governor position, reducing the reachable set to a single point. In this case, any movement of the governor enlarges the reachable set and may therefore compromise the robot's safety.

5.3 Generalization to Other Systems

Assuming the existence of a point-stabilizing controller and associated bounded forward-invariant reachable set prediction, our adaptive reference governor tracking method can be generalized to other nonlinear systems. Consider a general nonlinear system:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (24)$$

with state $\mathbf{x} = (\mathbf{p}, \mathbf{s})$, including position $\mathbf{p} \in \mathbb{R}^{n_p}$ and other variables $\mathbf{s} \in \mathbb{R}^{n_s}$, and input $\mathbf{u} \in \mathbb{R}^{n_u}$. We generalize the workspace $\mathcal{W} \subset \mathbb{R}^{n_p}$, the static obstacle space $\Omega^+ \subset \mathcal{W}$, and the obstacle space $\mathcal{O}^+ \subset \mathcal{W}$ to the n_p -dimensional setting, consistent with the definitions in Sec. 3. The reference path $\rho: [0, 1] \rightarrow \mathcal{W}$ is obtained by solving (7).

A crucial component in our adaptive reference governor formulation is a point-stabilizing control policy. Since deriving such a policy requires knowledge of the system dynamics, we assume its existence.

Assumption 1 There exists a control policy $\mathbf{k}(\mathbf{x}, \mathbf{p}^*)$ such that, when applied to (24), the closed-loop trajectory $(\mathbf{p}(t), \mathbf{s}(t))$ converges asymptotically from any initial state $(\mathbf{p}_0, \mathbf{s}_0) \in \mathbb{R}^{n_p+n_s}$ to the desired goal position $\mathbf{p}^* \in \mathbb{R}^{n_p}$. That is, $\lim_{t \rightarrow \infty} \mathbf{p}(t) = \mathbf{p}^*$.

We also assume the existence of a reachable set prediction, which contains all possible closed-loop trajectories under the policy $\mathbf{k}(\mathbf{x}, \mathbf{p}^*)$.

Assumption 2 There exists a reachable set $\mathcal{M}(\mathbf{x}, \mathbf{p}^*)$ such that, for any goal position \mathbf{p}^* and initial condition $\mathbf{x}_0 = (\mathbf{p}_0, \mathbf{s}_0)$, the control policy $\mathbf{k}(\mathbf{x}, \mathbf{p}^*)$ applied to system (24) renders $\mathcal{M}(\mathbf{x}, \mathbf{p}^*) \times \mathbb{R}^{n_s}$ forward invariant, i.e., $\mathbf{p}(t) \in \mathcal{M}(\mathbf{x}(t), \mathbf{p}^*)$ for all $t \geq 0$. Moreover, the reachable set is bounded as $\mathcal{M}(\mathbf{x}(t), \mathbf{p}^*) \subseteq \mathcal{B}_{\|\mathbf{p}(t) - \mathbf{p}^*\|}(\mathbf{p}^*)$ for all $t \geq 0$.

Given Assumption 1 and Assumption 2, the reference governor system (13) can be generalized to guide the nonlinear system in (24). The governor input \mathbf{u}_g is specified by (14), where the local safe zone $\mathcal{LS}(\mathbf{x}, \mathbf{g})$ is defined according to Def. 2. Under these conditions, Prop. 4 remains valid, and the construction of the adaptive governor input $\bar{\mathbf{u}}_g$ follows directly from the exposition in Sec. 5.2.2.

6 Evaluation

This section evaluates our Environment-Aware Safe Tracking (EAST) method for safe autonomous robot navigation. We begin by testing EAST in simulated static environments. This includes specifying the design parameters in Sec. 6.1, analyzing the impact of the obstacle clearance term (10) in Sec. 6.2, evaluating the adaptive control gain (18) in Sec. 6.3, and comparing EAST with the EVA-Planner (Quan et al., 2021) in Sec. 6.4. Next, we evaluate EAST in a large cluttered real-world static environment in Sec. 6.5. Finally, we evaluate EAST in simulated and real-world dynamic environments in Sec. 6.6.

Table 1: Control gain parameters.

Parameter	k_g	k_v	k_ω	γ_i	q_1	q_2
Value	2.0	Eq. (18)	1.5	0.2	1	9

Table 2: Obstacle clearance cost parameters.

Design	κ	c_u	c_f
minimum clearance	15.0	3.2	1
medium clearance	7.0	8.3	5
maximum clearance	1.0	16.9	15

6.1 Experiment Setup and Parameters.

All experiments share the same parameters and control gains unless explicitly stated otherwise. We used Hector SLAM (Kohlbrecher et al., 2011) to obtain robot pose estimates at 20 Hz and an occupancy grid map with 0.1 m resolution. The control gain parameters are summarized in Tab. 1. The class- \mathcal{K} function in (20) is defined as:

$$\alpha_i(h_i) = \gamma_i h_i^2, \quad (25)$$

where γ_i is a known parameter specified in Tab. 1.

6.2 Obstacle Clearance Cost Study

We study the performance of EAST under different parameters for the obstacle clearance cost in (10), presented in Tab. 2. Paths were planned in a simulated environment for each of the clearance cost designs, shown in Fig. 8. EAST can drive the robot to the goal safely in each case. Associated quantitative results are summarized in Tab. 3. The results show that, even though the minimum clearance cost design leads to the shortest path, the tracking time is significantly longer than the medium and maximum clearance cost designs. The maximum clearance cost design achieves the shortest tracking time but the longest path. The medium clearance cost design strikes a balance between them: the path is significantly shorter compared to the maximum clearance cost design while the tracking time is similar. These results suggest that a medium clearance cost design is preferable.

Table 3: Simulation results for three cost map designs.

Costmap Design	Plan Path Length	Robot Traj. Length	Finish Time	Avg. Clearance	Min. Clearance
minimum clearance	15.10 m	15.66 m	32.80 sec	0.45 m	0.13 m
medium clearance	16.19 m	15.63 m	20.62 sec	0.63 m	0.34 m
maximum clearance	22.28 m	20.32 m	18.96 sec	1.31 m	0.38 m

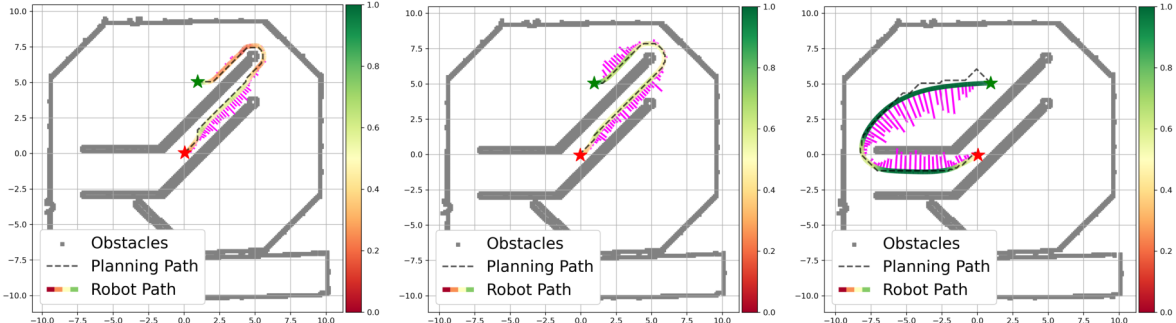


Fig. 8: Gazebo simulation with three different obstacle clearance cost designs. The left, middle, and right figures show the results for the minimum, median, and maximum clearance designs, respectively. The paths are shown as a black dashed lines with start and goal denoted by a red and green star, respectively. The colored curve represents the actual robot trajectory, with the robot to the inflated obstacle space distance $d(\mathbf{p}, \Omega^+)$ shown by gradient color corresponding to the side color bar. The tracking velocity profiles are shown as magenta arrows perpendicular to the robot’s trajectory.

6.3 Adaptive Control Gain Evaluation

To demonstrate the effectiveness of the adaptive control gain (18), we created a C-shaped simulated environment using Gazebo with a pre-specified reference path, shown in Fig. 9. We compared the performance of the safe tracking controller with fixed control gain $k_v = 1$ versus with the adaptive control gain in (18). With the adaptive control gain, the robot speeds up faster in straight lines and keeps a low speed in turns, finishing the task in less time compared to fixed control gain design.

6.4 Baseline Comparison

In this section, we compare EAST with the EVA-Planner (Quan et al., 2021), which solves the safe tracking problem via two-layer hierarchical MPC. In the first layer, the EVA-Planner refines a planned path utilizing a low-order approximation of the robot’s motion model. This refined path is then used to define the cost function in the second layer, which considers tracking error, safety constraints, and dynamics feasibility. All constraints are encoded in the cost function, leading to an

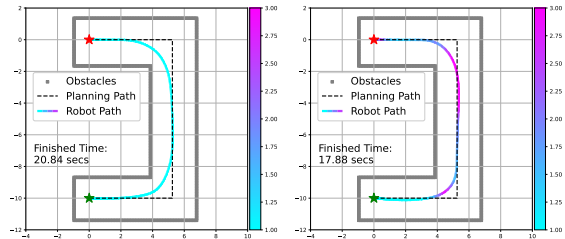


Fig. 9: Simulation of our safe tracking controller in a C-shape environment with reference path shown as a black dashed line and start and goal denoted by a red and green star, respectively. The colored curve represents the actual robot trajectory with control gain shown by gradient color corresponding to the side color bar.

unconstrained, nonlinear, non-convex optimization problem, which is solved by a gradient-based numerical solver. We carried out two experiments to compare the methods with emphasis on safety and stability, respectively.

Ten Point Test. This test evaluates the safety of EAST and the EVA-Planner in a static environment. Ten feasible goal points were selected in a simulated Gazebo environment, as

shown in Fig. 10. In each trial, the robot was tasked to reach one of these predetermined goals without any prior knowledge of the environment. The experiment was repeated ten times, so all the goal points were tested.

The results in Fig. 10 show that EAST can reach all ten goals safely, while the EVA-Planner fails to reach one. Since the EVA-Planner encodes the safety constraint in the cost function, solutions to its optimization problem do not guarantee safety, and the output trajectory goes through obstacles during our test. In contrast, EAST guarantees safety via Prop. 4 as long as the planned path ρ is in the interior of the free space and the robot is sufficiently close to the start of the path initially.

Maze Test. This test evaluates the stability of EAST and the EVA-Planner in a static environment. A maze-like simulated environment was created, shown in Fig. 11, with the objective of going from a start position at the origin to a goal position at (6, 6) in the maze center. To accomplish this task, the robot must go through corridors that become narrower from the outer (about 3 m wide) to the inner (less than 1 m wide) part of the environment.

In Fig. 11, the left plot shows the trajectory generated by the EVA-Planner and the right plot shows the trajectory generated by our EAST method. Our method accomplishes this task successfully while adapting its speed according to the local environment geometry, i.e., moving faster when the corridor is wide and slower when it becomes narrower. In contrast, the EVA-Planner cannot handle this task. The robot fails to adjust its speed as the corridor becomes narrower, which leads to oscillatory motion and eventually a safety violation. This oscillatory behavior arises from the gradient-based safety metric employed by the EVA-Planner, which continuously attempts to push the robot away from nearby obstacles. Consequently, within a narrow corridor, when the robot is repelled from one side, it rapidly approaches the opposite side, causing a repeated cycle of oscillations. This back-and-forth behavior persists until a safety violation occurs.

6.5 Hardware Evaluation in Cluttered Static Environment

This experiment tests the safety and stability of EAST in a real environment of size 40×20 m, containing static obstacles of different sizes, such as robots, boxes, desks, etc., as shown in Fig. 12a-12c.

During this experiment, a few goals in unknown regions are specified, and the robot is required to navigate autonomously. The robot has no prior knowledge of the environment and is expected to reach each goal without collisions. A first-person video captured by the onboard camera of the robot can be found in the supplementary material.

The experiment is visualized in Fig. 12, where the final obstacle clearance cost map is depicted in Fig. 12d. Three local scenes (marked in green boxes) are shown in the top row, with associated quantitative results below each of them. The robot navigates adaptively, slowing down when entering obstacle-dense areas and speeding up when in wide open spaces, as shown in Fig. 12a-12c. From Fig. 12b, we can see that the speed of the robot (orange curve) is higher when the distance to the inflated obstacle space (green line) is larger, and the adaptive control gain (purple curve) increases when the robot heading is aligned with the local environment.

6.6 Evaluation in Dynamic Environments

This section evaluates the safety and stability of EAST in dynamically changing environments. The robot is assumed to know the position and velocity of the moving obstacles. In the simulated experiments, the trajectories of the moving obstacles were pre-specified with position and velocity known to the robot. In the hardware experiments, a Vicon motion capture system was used to track the position and velocity of human actors and transmit the motion information to the robot using a wireless network. In practice, this information can be obtained from object detection and tracking using visual or LiDAR sensors. The control parameters used in this section are summarized in Tab. 4.

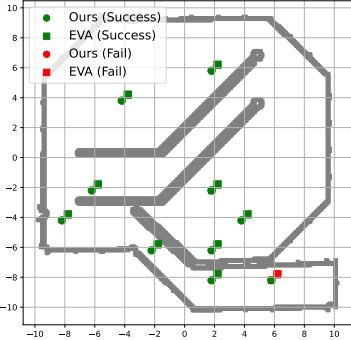


Fig. 10: EAST and EVA-Planner (Quan et al., 2021) navigating to randomly selected goals in simulation. The successful and unsuccessful goal executions are shown in green and red, respectively.

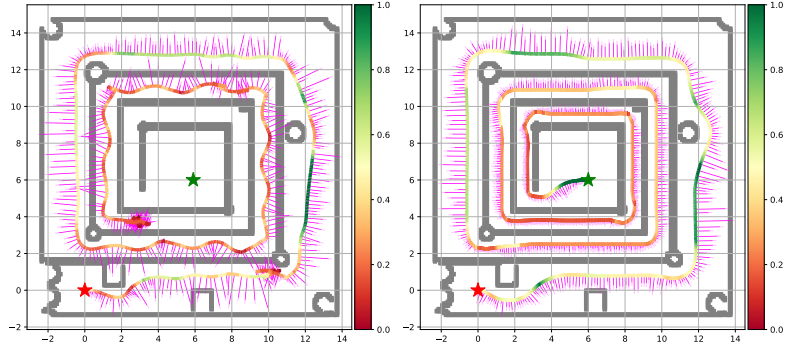


Fig. 11: Adaptivity test in the maze environment. The start and the goal are denoted by a red and green star, respectively. The left figure presents the results produced by the EVA-Planner method, while the right figure displays the results obtained using our EAST method. The colored curve represents the actual robot trajectory with the robot to the inflated obstacle space $d(\mathbf{p}, \Omega^+)$ shown by gradient color corresponding to the side color bar. The tracking velocity profiles are shown as magenta arrows perpendicular to the robot trajectories.

Table 4: Control gain parameters used in the dynamic environment simulations.

Parameter	k_g	k_v	k_ω	γ_i	q_1	q_2
Value	2.0	2.0	5.0	0.15	1	1

Simulation experiment. We evaluated EAST in a simulated dynamically changing environment, containing four static obstacles and six moving obstacles (light gray circles) of different sizes and velocities, shown in Fig. 13a-13c. Each moving obstacle follows a pre-specified path (gray dashed lines). The robot is required to reach a goal location without collisions with either static or moving obstacles.

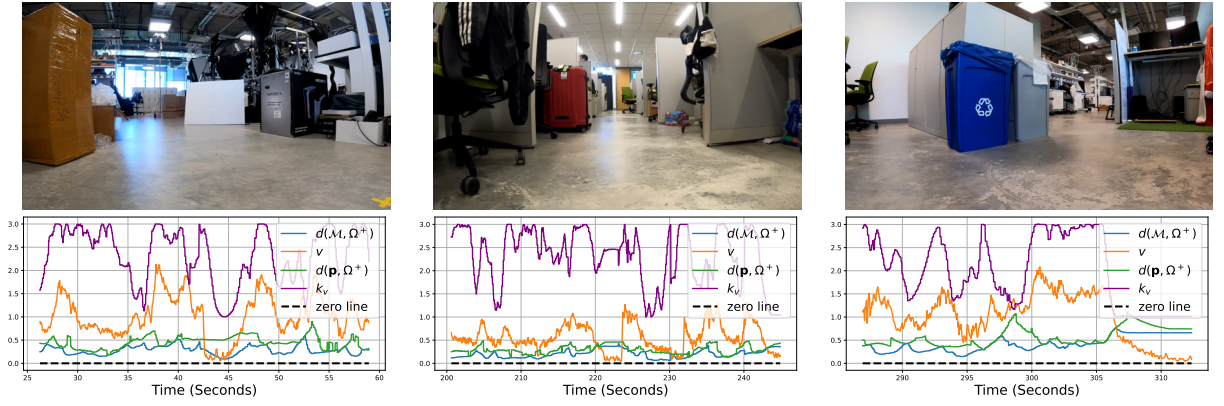
The results of this experiment can be found in Fig. 13. The robot is able to reach the goal successfully without collisions. The three snapshots in Fig. 13 show interactions between the robot and the moving obstacles. The robot deviates from its planned trajectory to avoid collisions with the moving obstacles effectively and subsequently returns to its original path once the obstructions are no longer present.

Hardware experiment. Finally, we evaluated the safety and stability of EAST in a real, dynamically changing environment. The experiment setup is shown in Fig. 14 with static

obstacles (boxes, walls, pillars, etc.) and moving obstacles (two human pedestrians wearing helmets). The robot does not have prior information about this environment, except for receiving moving obstacle positions and velocities from the Vicon system. The robot is required to visit three pre-specified goal states and return to the start state without any collision.

The quantitative results from this experiment are shown in Fig. 14c. We observe that our method successfully executes the experiment. During the test, the planner failed to adapt to the pedestrians due to its low re-planning frequency, and the robot made multiple deviations from the planned path to avoid moving obstacles. This stems from the planner's dependence on an occupancy map, which is not updated sufficiently fast to capture changes in a timely manner. Consequently, safe navigation in dynamic settings cannot rely solely on the planned path. The control algorithm plays a critical role in departing from the nominal path whenever strict adherence would place the robot in unsafe states.

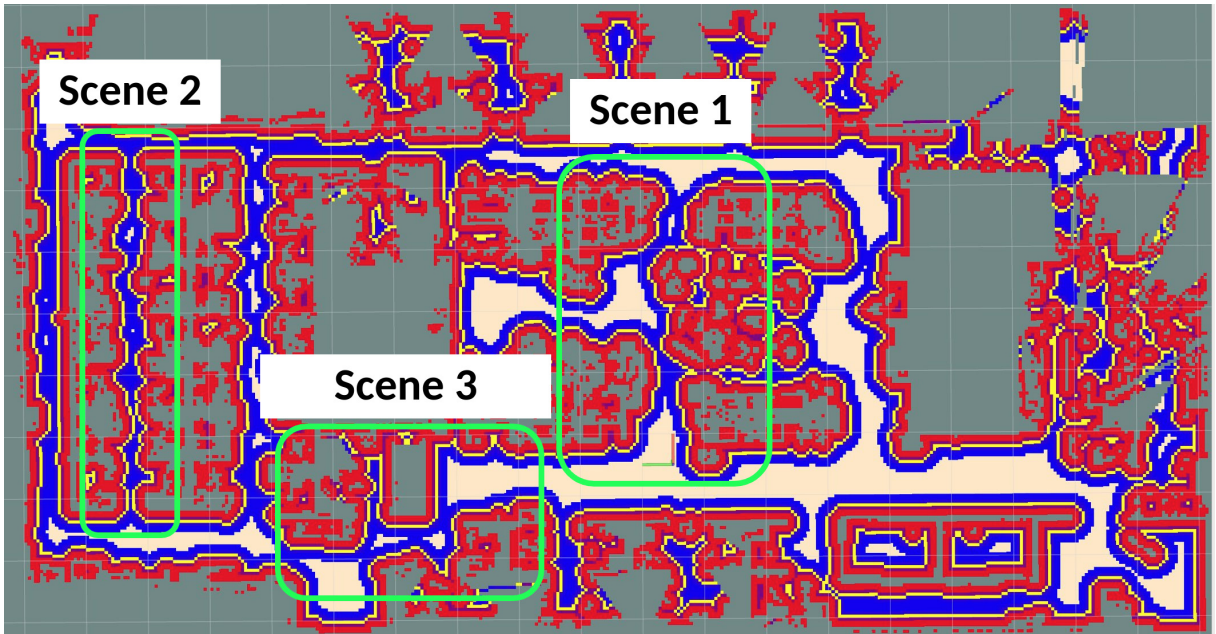
Both the simulation and hardware experiments demonstrate that our EAST method is able to successfully handle static and dynamic obstacles by optimizing the reference governor behavior in real time. Our QCQP optimization of the reference governor input with CBF constraints capturing the dynamic obstacles demonstrates several



(a) Scene 1: Going from wide-open area to cluttered region.

(b) Scene 2: Going through narrow aisle with obstacles.

(c) Scene 3: Passing from one region to another through narrow gap.

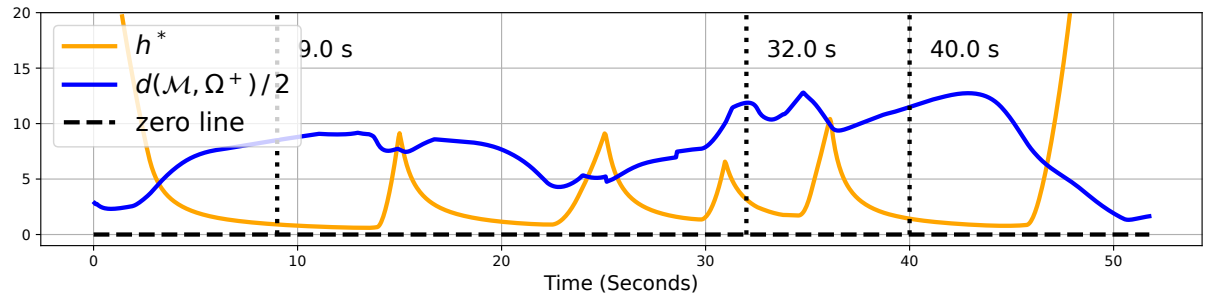
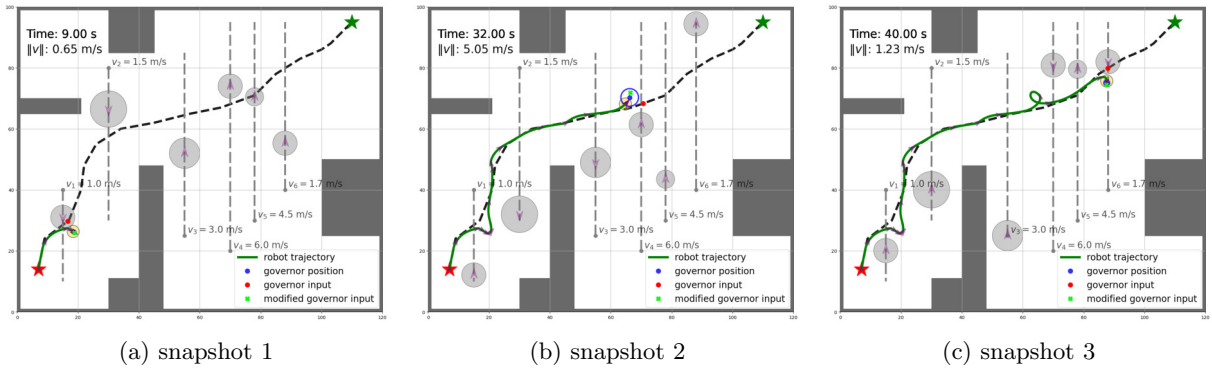


(d) Obstacle clearance cost map with cost values from high to low shown as red, yellow, blue, and white.

Fig. 12: Hardware experiment in large lab environment (about $40 \text{ m} \times 20 \text{ m}$). Three representative scenes, shown in (a)-(c), are marked on the final obstacle clearance cost map (d) using green regions. The distance from the motion prediction set to the inflated obstacle space $d(\mathcal{M}, \Omega^+)$, the robot linear velocity v , the distance from the robot to the inflated obstacle space $d(\mathbf{p}, \Omega^+)$, and the adaptive control gain k_v are plotted beneath each scene.

advantages. First, imposing the CBF constraint on the simpler reference-governor dynamics substantially lowers the computational load of the online optimization, enabling real-time adaptation to moving obstacles. Second, the reference signal is altered only when its direct execution would

compromise safety, thereby preserving the nominal stability of the original reference governor formulation with minimal degradation. Moreover, by decoupling path-tracking control from safety enforcement and confining the complex safety constraints to the reference-governor dynamics,



(d) Distance from motion prediction set to inflated obstacles $d(\mathcal{M}, \Omega^+)$ and minimum CBF value $h^* = \min_i h_i$.

Fig. 13: Snapshots (a)-(c) from simulation with six dynamic obstacles with different sizes and speeds, shown as gray circles. The pre-specified paths for the moving obstacles are depicted using gray dashed lines. The start and goal locations for the robot are shown by a red and green star, respectively. The robot is represented as a yellow circle. The robot's reference path is shown as a black dashed line. The actual robot trajectory is shown as a green solid line with purple arrows indicating its heading. The governor position \mathbf{g} , governor input \mathbf{u}_g , and modified governor input $\bar{\mathbf{u}}_g$ are shown by blue, red, and green dots, respectively.

our formulation is readily transferable to robot platforms with different system dynamics.

7 Conclusion

This paper presented an environment-aware safe tracking controller using a reference governor design to enforce safety in dynamic environments. Our control design separates path tracking from safety constraint enforcement using a reduced-order reference governor system. Using prior results, we defined a local safe zone for the robot-governor system as the set of governor inputs where the reachable set of the robot remains outside the obstacle set. While choosing governor inputs in the local safe zone guarantees static obstacle safety, our key contribution

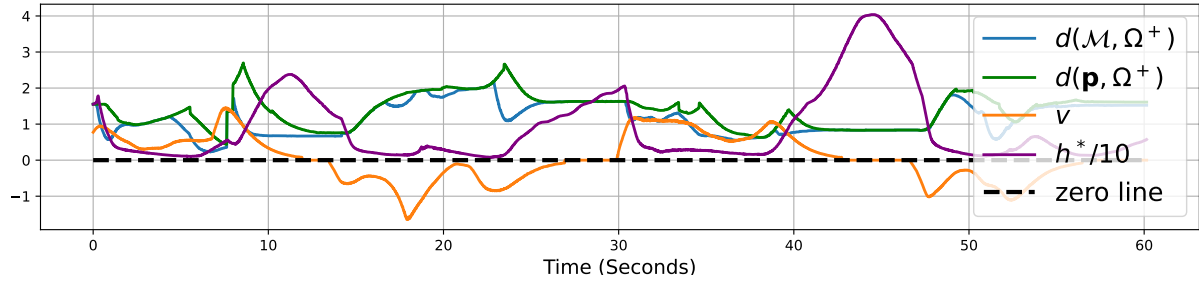
is to optimize the governor input to also satisfy CBF constraints capturing dynamic obstacles. The resulting convex QCQP optimization can be solved in real-time, enabling fast adaptation in dynamic environments. The method guarantees safety in static environments, while in dynamic environments safety is maintained only when the QCQP is feasible. We analyzed the feasibility of the optimization problem in Sec. 5.2.2 and discussed how our method can be applied to a broader class of systems in Sec. 5.3. Our controller outperformed a baseline method in simulation and demonstrated safety, efficiency, and fast adaptation to moving obstacles in several real environments. Future work will focus on extending our adaptive reference governor design to robots with more complex dynamics, such as mobile manipulators.



(a) Experiment site with Vicon motion capture system



(b) Robot and helmet with markers



(c) Distance from motion prediction set to inflated obstacles $d(\mathcal{M}, \Omega^+)$, distance from robot to inflated obstacles $d(\mathbf{p}, \Omega^+)$, robot linear velocity v , and minimum CBF value $h^* = \min_i h_i$.

Fig. 14: Hardware experiment with Jackal robot and two pedestrians, serving as moving obstacles. The actors' positions are captured by a Vicon motion capture system and passed to the robot over a WiFi network. The robot re-plans its path at 10 Hz online to reach different goals while avoiding the pedestrians crossing its path. The three goals are marked as green circles, and the origin is marked as a red circle in (a). The robot and helmets worn by the pedestrians are shown in (b). Quantitative results from the experiment are shown in (c). Please refer to the supplementary material for a video from the experiment.

Funding

We gratefully acknowledge support from ONR N00014-23-1-2353 and NSF RI IIS-2007141.

Statements and Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Author contribution Conceptualization: ZL, NA; Methodology: ZL, YY, ZN, NA; Theoretical

analysis: ZL, YY, ZN, NA; Algorithm development: ZL, YY, ZN, NA; Simulation validation and hardware evaluation: ZL, YY, ZN; Writing - initial draft: ZL, YY; Writing - review and editing: YY, NA; Supervision: NA.

References

Allibhoy, A., & Cortés, J. (2024). Control-barrier-function-based design of gradient flows for constrained nonlinear programming. *IEEE Transactions on Automatic Control*, 69(6), 3499-3514, <https://doi.org/10.1109/TAC>

.2023.3306492

- Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., Tabuada, P. (2019). Control barrier functions: Theory and applications. *IEEE European Control Conference (ECC)* (pp. 3420–3431).
- Ames, A.D., Grizzle, J.W., Tabuada, P. (2014). Control barrier function based quadratic programs with application to adaptive cruise control. *IEEE Conference on Decision and Control (CDC)* (pp. 6271–6278).
- Ames, A.D., Xu, X., Grizzle, J.W., Tabuada, P. (2017). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control (TAC)*, 62(8), 3861–3876, <https://doi.org/10.1109/TAC.2016.2638961>
- Arslan, O. (2022). *Time governors for safe path-following control*. (Preprint at <https://arxiv.org/abs/2212.01444>)
- Arslan, O., & Koditschek, D.E. (2017). Smooth extensions of feedback motion planners via reference governors. *IEEE International Conference on Robotics and Automation (ICRA)* (p. 4414–4421).
- Bajcsy, A., Bansal, S., Bronstein, E., Tolani, V., Tomlin, C.J. (2019). An efficient reachability-based framework for provably safe autonomous navigation in unknown environments. *IEEE Conference on Decision and Control (CDC)* (p. 1758–1765).
- Bansal, S., Chen, M., Herbert, S., Tomlin, C.J. (2017). Hamilton-jacobi reachability: A brief overview and recent advances. *IEEE Conference on Decision and Control (CDC)* (pp. 2242–2253).
- Bemporad, A. (1998). Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control (TAC)*, 43(3), 415–419, <https://doi.org/10.1109/9.661611>
- Bertsimas, D., & Tsitsiklis, J. (1997). *Introduction to linear optimization*. Athena Scientific.
- Borquez, J., Chakraborty, K., Wang, H., Bansal, S. (2024). On safety and liveness filtering using hamilton–jacobi reachability analysis. *IEEE Transactions on Robotics*, 40, 4235–4251, <https://doi.org/10.1109/TRO.2024.3454470>
- Borrelli, F., Bemporad, A., Morari, M. (2017). *Predictive control for linear and hybrid systems*. Cambridge University Press.
- Borrmann, U., Wang, L., Ames, A.D., Egerstedt, M. (2015). Control barrier certificates for safe swarm behavior. *IFAC-PapersOnLine*, 68–73, <https://doi.org/10.1016/j.ifacol.2015.11.154>
- Bradski, G. (2000). The OpenCV library. *Dr. Dobbs's Journal of Software Tools*, ,
- Bravo, J.M., Alamo, T., Camacho, E.F. (2006). Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets. *Automatica*, 42(10), 1745–1751, <https://doi.org/10.1016/j.automatica.2006.05.003>
- Breedon, J., & Panagou, D. (2023). Compositions of multiple control barrier functions under input constraints. *American Control Conference (ACC)* (p. 3688–3695).
- Brito, B., Floor, B., Ferranti, L., Alonso-Mora, J. (2019). Model predictive contouring control for collision avoidance in unstructured dynamic environments. *IEEE Robotics and Automation Letters*, 4(4), 4459–4466, <https://doi.org/10.1109/LRA.2019.2929976>
- Burridge, R., Rizzi, A., Koditschek, D. (1999). Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research (IJRR)*,

- Castroviejo-Fernandez, M., Li, H., Cotorruelo, A., Garone, E., Kolmanovsky, I. (2024). Less conservative robust reference governors and their applications. *European Journal of Control*, 79, 101084, <https://doi.org/https://doi.org/10.1016/j.ejcon.2024.101084>
- Chandra, R., Zinage, V., Bakolas, E., Stone, P., Biswas, J. (2025). Deadlock-free, safe, and decentralized multi-robot navigation in social mini-games via discrete-time control barrier functions. *Autonomous Robots*, 49(2), 12, <https://doi.org/10.1007/s10514-025-10194-8>
- Convens, B., Merckaert, K., Nicotra, M.M., Vandenberght, B. (2022). Safe, fast, and efficient distributed receding horizon constrained control of aerial robot swarms. *IEEE Robotics and Automation Letters*, 7(2), 4173-4180, <https://doi.org/10.1109/LRA.2022.3148455>
- Dai, B., Khorrambakht, R., Krishnamurthy, P., Gonçalves, V., Tzes, A., Khorrami, F. (2023). Safe navigation and obstacle avoidance using differentiable optimization based control barrier functions. *IEEE Robotics and Automation Letters*, 8(9), 5376-5383, <https://doi.org/10.1109/LRA.2023.3295654>
- Dendorfer, P., Rezatofghi, H., Milan, A., Shi, J., Cremers, D., Reid, I., ... Leal-Taixé, L. (2020). *MOT20: A benchmark for multi object tracking in crowded scenes*. (Preprint at <https://arxiv.org/abs/2003.09003>)
- Ding, J., Li, E., Huang, H., Tomlin, C.J. (2011). Reachability-based synthesis of feedback policies for motion planning under bounded disturbances. *IEEE International Conference on Robotics and Automation (ICRA)*
- (p. 2160-2165).
- Dirckx, D., Bos, M., Vandewal, B., Vanroye, L., Swevers, J., Decré, W. (2025). Asap-mpc: an asynchronous update scheme for online motion planning with nonlinear model predictive control. *Autonomous Robots*, 49(1), 8, <https://doi.org/10.1007/s10514-025-10192-w>
- Fan, Y., Liu, S., Belabbas, M.-A. (2019). Mid-air motion planning of floating robot using heat flow method. *IFAC-PapersOnLine*, 52(22), 19-24, <https://doi.org/https://doi.org/10.1016/j.ifacol.2019.11.041>
- Fisac, J.F., Chen, M., Tomlin, C.J., Sastry, S.S. (2015). Reach-avoid problems with time-varying dynamics, targets and constraints. *International Conference on Hybrid Systems: Computation and Control (HSCC)* (pp. 11–20).
- Ganai, M., Gao, S., Herbert, S.L. (2024). Hamilton-jacobi reachability in reinforcement learning: A survey. *IEEE Open Journal of Control Systems*, 3, 310-324, <https://doi.org/10.1109/OJCSYS.2024.3449138>
- Gao, F., Wu, W., Lin, Y., Shen, S. (2018). Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial. *IEEE International Conference on Robotics and Automation (ICRA)* (p. 344-351).
- Gao, Y., Gray, A., Tseng, H.E., Borrelli, F. (2014). A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6), 802–823, <https://doi.org/10.1080/00423114.2014.902537>
- Gao, Y., Lin, T., Borrelli, F., Tseng, E., Hrovat, D. (2010). Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. *ASME Dynamic Systems and Control Conference* (p. 265-272).

- Garone, E., Cairano, S.D., Kolmanovsky, I. (2017). Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, 75, 306–328, <https://doi.org/10.1016/j.automatica.2016.08.013>
- Garone, E., & Nicotra, M.M. (2016). Explicit reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control (TAC)*, 61(5), 1379–1384, <https://doi.org/10.1109/TAC.2015.2476195>
- Gautam, Y., Nechyporenko, N., Lin, C.-H., Roncone, A., Nicotra, M.M. (2025). *Compliant explicit reference governor for contact friendly robotic manipulators*. (Preprint at <https://arxiv.org/abs/2504.09188>)
- Hart, P.E., Nilsson, N.J., Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100–107, <https://doi.org/10.1109/TSSC.1968.300136>
- Herbert, S.L., Chen, M., Han, S., Bansal, S., Fisac, J.F., Tomlin, C.J. (2017). FaSTrack: A modular framework for fast and guaranteed safe motion planning. *IEEE Conference on Decision and Control (CDC)* (pp. 1517–1522).
- Hosseinzadeh, M., Sinopoli, B., Bobick, A.F. (2020). An explicit reference governor for time-varying linear constraints. *IEEE Conference on Decision and Control (CDC)* (p. 3323-3328).
- Isleyen, A., van de Wouw, N., Arslan, O. (2023). Feedback motion prediction for safe unicycle robot navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (p. 10511-10518).
- Kavraki, L.E., Kolountzakis, M.N., Latombe, J.-C. (1998). Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and Automation (TRA)*, 14(1), 166–171, <https://doi.org/10.1109/70.660866>
- Keyumarsi, S., Atman, M.W.S., Gusrialdi, A. (2024). LiDAR-based online control barrier function synthesis for safe navigation in unknown environments. *IEEE Robotics and Automation Letters*, 9(2), 1043-1050, <https://doi.org/10.1109/LRA.2023.3339059>
- Koenig, N., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (p. 2149-2154).
- Kohlbrecher, S., Meyer, J., von Stryk, O., Klingauf, U. (2011). A flexible and scalable SLAM system with full 3D motion estimation. *IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)* (p. 155-160).
- Kousik, S., Vaskov, S., Bu, F., Johnson-Roberson, M., Vasudevan, R. (2020). Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots. *The International Journal of Robotics Research (IJRR)*, 39(12), 1419–1469, <https://doi.org/10.1177/0278364920943266>
- LaValle, S. (1998). *Rapidly-exploring random trees: A new tool for path planning* (TR 98-11). Comp. Sci. Dept., Iowa State University.
- Li, Z., Arslan, O., Atanasov, N. (2020). Fast and safe path-following control using a state-dependent directional metric. *IEEE International Conference on Robotics and Automation (ICRA)* (p. 6176-6182).
- Li, Z., & Atanasov, N. (2023). Governor-parameterized barrier function for safe output tracking with locally sensed constraints. *Automatica*, 152, 110996, <https://doi.org/10.1016/j.automatica.2023.110996>

- Likhachev, M., Gordon, G.J., Thrun, S. (2004). ARA*: Anytime A* with provable bounds on sub-optimality. *Advances in Neural Information Processing Systems (NIPS)* (pp. 767–774).
- Lindqvist, B., Mansouri, S.S., Agha-mohammadi, A.-a., Nikolakopoulos, G. (2020). Nonlinear mpc for collision avoidance and control of uavs with dynamic obstacles. *IEEE Robotics and Automation Letters*, 5(4), 6001-6008, <https://doi.org/10.1109/LRA.2020.3010730>
- Long, K., Qian, C., Cortés, J., Atanasov, N. (2021). Learning barrier functions with memory for robust safe navigation. *IEEE Robotics and Automation Letters*, 6(3), 4931-4938, <https://doi.org/10.1109/LRA.2021.3070250>
- Margellos, K., & Lygeros, J. (2011). Hamilton–jacobi formulation for reach–avoid differential games. *IEEE Transactions on Automatic Control*, 56(8), 1849-1861, <https://doi.org/10.1109/TAC.2011.2105730>
- Mestres, P., Nieto-Granda, C., Cortés, J. (2024). Distributed safe navigation of multi-agent systems using control barrier function-based controllers. *IEEE Robotics and Automation Letters*, 9(7), 6760-6767, <https://doi.org/10.1109/LRA.2024.3414268>
- Miguel, C.-F., & Kolmanovsky, I. (2024). Reference governor for linear uncertain systems with time-varying constraints. *IFAC-PapersOnLine*, 58(18), 250-256, <https://doi.org/10.1016/j.ifacol.2024.09.039>
- Nakamura, K., Peters, L., Bajcsy, A. (2025). *Generalizing safety beyond collision-avoidance via latent-space reachability analysis*. (Preprint at <https://arxiv.org/abs/2502.00935>)
- Nguyen, Q., & Sreenath, K. (2016). Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. *IEEE American Control Conference (ACC)* (pp. 322–328).
- Nicotra, M.M., & Garone, E. (2018a). The explicit reference governor: A general framework for the closed-form control of constrained nonlinear systems. *IEEE Control Systems Magazine*, 38(4), 89-107, <https://doi.org/10.1109/MCS.2018.2830081>
- Nicotra, M.M., & Garone, E. (2018b). The explicit reference governor: A general framework for the closed-form control of constrained nonlinear systems. *IEEE Control Systems Magazine*, 38(4), 89–107, <https://doi.org/10.1109/MCS.2018.2830081>
- Park, J., & Boyd, S. (2017). *General heuristics for nonconvex quadratically constrained quadratic programming*. (Preprint at <https://arxiv.org/abs/1703.07870>)
- Quan, L., Zhang, Z., Zhong, X., Xu, C., Gao, F. (2021). EVA-planner: Environmental adaptive quadrotor planning. *IEEE International Conference on Robotics and Automation (ICRA)* (pp. 398–404).
- Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., ... Ng, A.Y. (2009). ROS: An open-source robot operating system. *IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open-Source Software* (p. 5).
- Saccani, D., Cecchin, L., Fagiano, L. (2023). Multitrajectory model predictive control for safe uav navigation in an unknown environment. *IEEE Transactions on Control Systems Technology*, 31(5), 1982-1997, <https://doi.org/10.1109/TCST.2022.3216989>
- Santillo, M., & Jankovic, M. (2021). Collision free navigation with interacting, non-communicating obstacles. *American Control Conference (ACC)* (p. 1637-1643).

- Seo, H., Lee, D., Son, C.Y., Tomlin, C.J., Kim, H.J. (2019). Robust trajectory planning for a multicopter against disturbance based on hamilton-jacobi reachability analysis. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (p. 3150-3157).
- Sharpless, W., Shinde, N., Kim, M., Chow, Y.T., Herbert, S. (2023). *Koopman-hopf hamilton-jacobi reachability and control*. (Preprint at <https://arxiv.org/abs/2303.11590>)
- Squires, E., Pierpaoli, P., Egerstedt, M. (2018). Constructive barrier certificates with applications to fixed-wing aircraft collision avoidance. *IEEE Conference on Control Technology and Applications (CCTA)* (pp. 1656–1661).
- Tartaglione, G., Nicotra, M.M., Naldi, R., Garone, E. (2024). A constrained control framework for unmanned aerial vehicles based on explicit reference governor. *Automatica*, 166, 111696, <https://doi.org/10.1016/j.automatica.2024.111696>
- Wen, Z., Dong, M., Chen, X. (2024). Collision-free robot navigation in crowded environments using learning based convex model predictive control. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (p. 5452-5459).
- Xiao, W., & Belta, C. (2019). Control barrier functions for systems with high relative degree. *IEEE Conference on Decision and Control (CDC)* (pp. 474–479).