Fast and Safe Path-Following Control using a State-Dependent Directional Metric

Zhichao Li[1]   Ömür Arslan[2]   Nikolay Atanasov[1]

Abstract—This paper considers the problem of fast and safe autonomous navigation in partially known environments. Our main contribution is a control policy design based on ellipsoidal trajectory bounds obtained from a quadratic state-dependent distance metric. The ellipsoidal bounds are used to embed directional preference in the control design, leading to system behavior that is adapted to local environment geometry, carefully considering medial obstacles while paying less attention to lateral ones. We use a virtual reference governor system to adaptively follow a desired navigation path, slowing down when system safety may be violated and speeding up otherwise. The resulting controller is able to navigate complex environments faster than common Euclidean-norm and Lyapunov-function-based designs, while retaining stability and collision avoidance guarantees.

I. INTRODUCTION

Advances in embedded sensing and computation have enabled robot applications in unstructured environments and in close interaction with humans, including autonomous transportation, inspection and cleaning services, and medical robotics. Safe, yet efficient robot navigation is important for these applications but is challenging due to partially known or rapidly changing operational conditions.

In motion planning, optimality guarantees have been achieved for geometric path planning [1], [2] but incorporating robot dynamics without violating these guarantees remains an active area of research. To achieve efficient behavior for a dynamical system, optimal control theory is used together with sampling-based or search-based motion planners. Sampling-based methods connect neighboring states using locally optimal control such as linear-quadratic-regulation (LQR) [3] or fixed-final-state-free-final-time optimal control [4]. Locally optimal control, however, does not necessarily lead to global optimality [5]. Search-based methods construct a safe corridor [6], [7] (a connected safe region in free space) and seek a composition of short motion primitives within. Depending on the primitive design, the resulting trajectory may already be dynamically feasible [8] or may be optimized locally using model predictive control (MPC) [6]. These techniques do not provide formal guarantees for joint collision avoidance and stability.

To guarantee safety formally, most existing works rely on Lyapunov theory and reachable set computations. A
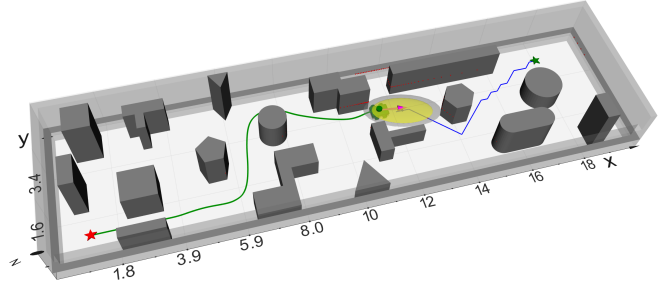
Fig. 1: A robot equipped with a lidar scanner navigates in an unknown cluttered environment. A state-dependent metric that considers the robot's direction of motion is designed to approximate the robot's future trajectory (yellow ellipse) and quantify its safety (gray ellipse) with respect to surrounding obstacles. An adaptive controller guarantees safety and stability based on these measures.

funnel is an outer approximation of the reachable set of a dynamical system in the presence of disturbances [9]. Building on the seminal work of [10], sequential composition of funnels offers effective means of guaranteeing safe navigation [11]–[14]. Using sum-of-squares optimization [15], these techniques can deal with nonlinear systems, non-holonomic constraints, and bounded disturbances. Recently, control barrier functions methods [16]–[19] have received significant attention. While optimizing performance without sacrificing stability using control Lyapunov functions (CLF), safety constraints are handled by a control barrier function (CBF). A virtual reference governor system [20], [21] may also be used to enforce safety constraints as an add-on control scheme to pre-stabilized dynamical systems. Using a reference governor design, [22] enables safe navigation for an acceleration-controlled robot among spherical obstacles by adaptive tracking a first-order vector field.

The importance of considering configuration space geometry and system dynamics jointly when designing a steering function for sampling-based kinodynamic planning is discussed in [5]. Metrics based on Mahalanobis distance, linear-quadratic-regulator cost, and a Gram matrix derived from system linearization are considered. Inspired by this work, we observe that using a static distance metric to quantify the safety of a robot with respect to surrounding obstacles can significantly impact its performance in real applications. For example, an autonomous golf cart running on campus has to simultaneously maintain safe distance from pedestrians and, yet, be able to squeeze through narrow passages such as doors or road block pillars. Static safety measures do not take the system's velocity direction into account, leading to overly cautious behavior even if the direction of travel is completely orthogonal to nearby obstacle surfaces. We refer

to this limitation as the *corridor effect* and aim to design an adaptive path-following controller, mitigating this effect via a new metric that takes the system's state into account when quantifying safety. The two main **contributions** of this work are highlighted as follows. First, we propose a new *state-dependent directional metric* and develop accurate system trajectory bounds for linearized robot dynamics that take direction of motion into account. Second, we develop an adaptive feedback controller, based on the directional trajectory bounds, and prove that it ensures stable and collision-free navigation. The controller relies only on local obstacle information, easily obtainable from onboard sensors, and provides fast tracking performance in complex unknown environments. See Fig. 1 for an illustration.

II. NOTATION

Let $\mathbb{S}_{>0}^n$ and $\mathbb{S}_{\geq 0}^n$ denote the set of $n \times n$ symmetric positive and semi-definite matrices. Let $\succ$ and $\succeq$ denote the generalized inequalities associated with $\mathbb{S}_{>0}^n$ and $\mathbb{S}_{\geq 0}^n$. Denote the Euclidean (ℓ^2) norm by $\|\mathbf{x}\|$ and the quadratic norm induced by $\mathbf{Q} \in \mathbb{S}_{>0}^n$ as $\|\mathbf{x}\|_{\mathbf{Q}} := \sqrt{\mathbf{x}^T \mathbf{Q} \mathbf{x}}$. Let $\lambda_{\max}(\mathbf{Q})$ and $\lambda_{\min}(\mathbf{Q})$ be the maximum and minimum eigenvalues of $\mathbf{Q}$. Let $d_{\mathbf{Q}}(\mathbf{x}, \mathcal{A}) := \inf_{\mathbf{a} \in \mathcal{A}} \|\mathbf{x} - \mathbf{a}\|_{\mathbf{Q}}$ denote the quadratic norm distance from a point $\mathbf{x}$ to a set $\mathcal{A}$. Given $\mathbf{Q} \in \mathbb{S}_{>0}^n$ and scaling $\eta \geq 0$, denote the associated ellipsoid centered at $\mathbf{q} \in \mathbb{R}^n$ by $\mathcal{E}_{\mathbf{Q}}(\mathbf{q}, \eta) := \{\mathbf{x} \in \mathbb{R}^n \mid (\mathbf{x} - \mathbf{q})^T \mathbf{Q}(\mathbf{x} - \mathbf{q}) \leq \eta\}$.

III. PROBLEM FORMULATION

Consider a robot operating in an unknown environment $\mathcal{W} \subset \mathbb{R}^n$. Denote the obstacle space by a closed set $\mathcal{O}$ and the free space by an open set $\mathcal{F} := \mathcal{W} \setminus \mathcal{O}$. Suppose that the robot dynamics are controllable, linear, and time-invariant:

$$\dot{\mathbf{s}} = \mathbf{A}\mathbf{s} + \mathbf{B}\mathbf{u} \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ is the control input and $\mathbf{s} := (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n_s}$ is the robot state[1], decomposed into constrained variables $\mathbf{x}$ and free variables $\mathbf{y}$. Throughout this paper, $\mathbf{x}(t) \in \mathbb{R}^n$ represents the robot position, required to remain within $\mathcal{F}$ for all $t \geq 0$, while $\mathbf{y}(t) \in \mathbb{R}^{n_s - n}$ denotes higher-order (velocity, acceleration, jerk, ...) terms. Our objective is to design a closed-loop control policy $\mathbf{u}(t)$, ensuring that the robot follows a given navigation path in the free space.

Definition 1. A *path* is a piecewise-continuous function $\mathcal{P} : [0, 1] \mapsto \mathring{\mathcal{F}}$ that maps a path-length parameter $\alpha \in [0, 1]$ to the interior of free space. The start $\mathcal{P}(0)$ and the end $\mathcal{P}(1)$ of a navigation path $\mathcal{P}(\alpha)$ are in the interior of free space, i.e., $\mathcal{P}(0), \mathcal{P}(1) \in \mathring{\mathcal{F}}$.

A path $\mathcal{P}$ may be provided by a geometric planning algorithm [1], [2], [23]. We consider the following problem.

Problem 1. Given a path $\mathcal{P}$, design a control policy $\mathbf{u}(t)$ so that the robot (1) is asymptotically steered from the start to the end of $\mathcal{P}$ while remaining collision-free, i.e., $\mathbf{x}(t) \in \mathcal{F}$ for all $t \geq 0$.

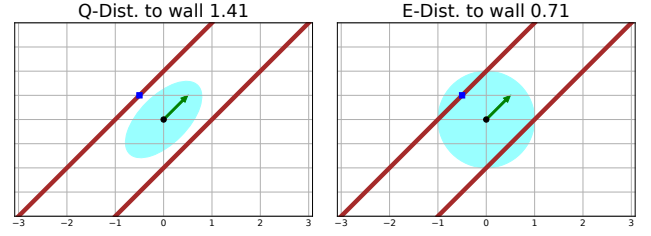[1]Transpose operations are omitted when grouping vectors for conciseness.

Fig. 2: Robot (black dot) moving in direction $\mathbf{v} := \left[\sqrt{2}/2, \sqrt{2}/2\right]$ (green arrow) along a corridor. The distances, measured by a quadratic norm $\|\cdot\|_{\mathbf{Q}}$ (left) and Euclidean norm $\|\cdot\|$ (right), from the robot to the closest point (small blue square) on the wall (red line) are 1.41 and 0.71. The matrix $\mathbf{Q} = [[2.5 - 1.5], [-1.5, 2]]$ is defined as a *directional matrix* $\mathbf{Q}[\mathbf{v}]$.

IV. TECHNICAL APPROACH

In this section, we propose a novel state-dependent directional metric (SDDM) and show that closed-loop trajectories of (1) can be bounded in the SDDM by solving a convex optimization problem. We develop a feedback control law that exploits the trajectory bounds to stabilize the robot and follow the path $\mathbf{r}$ adaptively, slowing down when safety may be endangered and speeding up otherwise.

A. State-Dependent Directional Metric

As mentioned in the introduction, measuring safety using a static Euclidean norm may lead to system performance suffering from the *corridor effect*. We propose a quadratic distance measure $\|\cdot\|_{\mathbf{Q}}$ that assigns priority to obstacles depending on the robot's moving direction. The level sets of $\|\cdot\|_{\mathbf{Q}}$ are ellipsoids $\mathcal{E}_{\mathbf{Q}}(\mathbf{0}, \eta)$ whose shape and orientation are determined by the matrix $\mathbf{Q}$. Our idea is to encode a desired directional preference in the distance metric via an appropriate choice of $\mathbf{Q}$. Consider the example in Fig. 2. A quadratic norm, well-aligned with the local environment geometry, may provide a more accurate evaluation of safety than a static Euclidean norm. Based on this observation, we propose a general construction of a directional matrix $\mathbf{Q}[\mathbf{v}]$, in the direction of vector $\mathbf{v}$, that defines a state-dependent directional metric.

Definition 2. A *directional matrix* associated with vector $\mathbf{v}$ and scalars $c_2 > c_1 > 0$ is defined as

$$\mathbf{Q}[\mathbf{v}] = \begin{cases} c_2\mathbf{I} + (c_1 - c_2)\frac{\mathbf{v}\mathbf{v}^T}{\|\mathbf{v}\|^2}, & \text{if } \mathbf{v} \neq 0, \\ c_1\mathbf{I}, & \text{otherwise.} \end{cases} \tag{2}$$

The unit ellipsoid $\mathcal{E}_{\mathbf{Q}[\mathbf{v}]}(\mathbf{x}, 1)$ centered at $\mathbf{x}$ generated by a directional matrix $\mathbf{Q}[\mathbf{v}]$ is elongated in the direction of $\mathbf{v}$.

Lemma 1. *For any vector $\mathbf{v}$, the directional matrix $\mathbf{Q}[\mathbf{v}]$ is symmetric positive definite.*

Proof. Since $\mathbf{v}\mathbf{v}^T$ is symmetric, $\mathbf{Q}[\mathbf{v}]^T = \mathbf{Q}[\mathbf{v}]$. If $\mathbf{v} = \mathbf{0}$, $\mathbf{Q}[\mathbf{v}] = c_1\mathbf{I}$ is positive definite. If $\mathbf{v} \neq \mathbf{0}$ and $\mathbf{q}$ is arbitrary:

$$\mathbf{q}^T\mathbf{Q}[\mathbf{v}]\mathbf{q} = c_2\mathbf{q}^T\mathbf{q} + (c_1 - c_2)\frac{(\mathbf{q}^T\mathbf{v})^2}{\|\mathbf{v}\|^2}$$

$$\geq c_2\mathbf{q}^T\mathbf{q} + (c_1 - c_2)\frac{\|\mathbf{q}\|^2\|\mathbf{v}\|^2}{\|\mathbf{v}\|^2} = c_1\|\mathbf{q}\|^2,$$

which follows from $c_2 > c_1$ and the Cauchy-Schwarz inequality. The proof is completed by noting that $c_1 > 0$. $\square$

B. Trajectory Bounds using SDDM

Using a directional matrix, one can define an SDDM to adaptively evaluate the risk of surrounding obstacles. We will show how to use an SDDM to obtain bounds on the closed-loop trajectory of the constrained state $\mathbf{x}(t)$ in (1). Assume the robot is stabilized by a feedback controller $\mathbf{u} = -\mathbf{K}\mathbf{s}$. The closed-loop dynamics are:

$$\dot{\mathbf{s}} = \bar{\mathbf{A}}\mathbf{s} \qquad \mathbf{z} = \mathbf{C}\mathbf{s} \qquad (3)$$

where $\bar{\mathbf{A}} := (\mathbf{A} - \mathbf{B}\mathbf{K})$ is Hurwitz. Any initial state $\mathbf{s}_0 := \mathbf{s}(t_0)$ will converge exponentially to the equilibrium point at origin. An output $\mathbf{z}$ is introduced to consider the constrained state $\mathbf{x}$. We are interested in measuring the maximum deviation of $\mathbf{x}(t)$ for $t \geq 0$ from the origin using a directional measure determined by the orientation of initial state $\mathbf{x}_0 := \mathbf{x}(t_0)$ with respect to $\mathbf{0}$. Define an SDDM using the directional matrix:

$$\mathbf{Q} := \mathbf{Q}\left[\mathbf{0} - \mathbf{x}_0\right] \in \mathbb{S}_{>0}^n \qquad (4)$$

and choose output $\mathbf{z}(t) = \mathbf{Q}^{\frac{1}{2}}\mathbf{x}(t)$ so that $\mathbf{C} := \mathbf{Q}^{\frac{1}{2}}\mathbf{P}$, where $\mathbf{P} := [\mathbf{I}, \mathbf{0}]$ is the projection matrix from $\mathbf{s}$ to $\mathbf{x}$. Note that $\mathbf{z}(t)^T\mathbf{z}(t) = \mathbf{x}(t)^T\mathbf{Q}\mathbf{x}(t) = \|\mathbf{x}(t)\|_{\mathbf{Q}}^2$. Thus, measuring the maximum deviation of $\mathbf{x}(t)$ in the SDDM is equivalent to finding the output peak along the robot trajectory.

$$\eta(t_0) := \max_{t \geq t_0}\|\mathbf{x}(t)\|_{\mathbf{Q}}^2 = \max_{t \geq t_0}\|\mathbf{z}(t)\|^2 \qquad (5)$$

We outline two approaches to solve this problem.

1) Exact solution: The output peak $\eta(t_0)$ can be computed exactly by comparing the values of $\|\mathbf{z}(t)\|^2$ at the boundary point $t = t_0$ and all critical points $\left\{t > t_0 \mid \frac{d}{dt}\|\mathbf{z}(t)\|^2 = 0\right\}$. Since the closed-loop system in (3) is linear time-invariant, $\mathbf{s}(t)$ can be obtained in closed form. Let $\bar{\mathbf{A}} = \mathbf{V}\mathbf{J}\mathbf{V}^{-1}$ be the Jordan decomposition of $\bar{\mathbf{A}}$, where $\mathbf{J}$ is block diagonal. The critical points satisfy:

$$0 = \frac{d}{dt}\mathbf{z}(t)^T\mathbf{z}(t) = 2\mathbf{z}(t)^T\dot{\mathbf{z}}(t) \qquad (6)$$
$$= 2\left(\mathbf{P}\mathbf{V}e^{\mathbf{J}(t-t_0)}\mathbf{V}^{-1}\mathbf{s}_0\right)^T\mathbf{Q}\left(\mathbf{P}\mathbf{V}\mathbf{J}e^{\mathbf{J}(t-t_0)}\mathbf{V}^{-1}\mathbf{s}_0\right).$$

In general, an exact solution may be hard to compute due to the complicated expression of $e^{\mathbf{J}t}$.

2) Approximate solution: When an exact solution to (5) is hard to obtain, we may instead compute a tight upper bound on $\eta(t_0)$. Given a $\mathbf{U} \in \mathbb{S}_{>0}^{n_s}$,

$$\mathcal{E}_{inv} := \left\{\boldsymbol{\xi} \in \mathbb{R}^{n_s} \mid \boldsymbol{\xi}^T\mathbf{U}\boldsymbol{\xi} \leq 1\right\} \qquad (7)$$

is an invariant ellipsoid for the robot dynamics (3), i.e., $\mathbf{s}(t) \in \mathcal{E}_{inv}$ for all $t \geq t_0$. Instead of finding the peak value of $\|\mathbf{z}(t)\|^2$ along the state trajectory, we can compute it over the invariant ellipsoid $\mathcal{E}_{inv}$. Since $\mathcal{E}_{inv}$ contains the system trajectory, we have for all $t \geq t_0$:

$$\|\mathbf{z}(t)\|^2 \leq \eta(t_0) \leq \max_{\boldsymbol{\xi} \in \mathcal{E}_{inv}}\boldsymbol{\xi}^T\mathbf{C}^T\mathbf{C}\boldsymbol{\xi} \qquad (8)$$

Obtaining the upper bound above is equivalent to solving the following semi-definite program [24, Ch.6]:

$$\begin{aligned}\underset{\mathbf{U},\delta}{\text{minimize}} \quad & \delta \\ \text{subject to} \quad & \bar{\mathbf{A}}^T\mathbf{U} + \mathbf{U}\bar{\mathbf{A}} \preceq \mathbf{0}, \quad \mathbf{s}_0^T\mathbf{U}\mathbf{s}_0 \leq 1 \\ & \begin{bmatrix}\mathbf{U} & \mathbf{C}^T \\ \mathbf{C} & \delta\mathbf{I}\end{bmatrix} \succeq \mathbf{0}, \quad \mathbf{U} \succ \mathbf{0}.\end{aligned} \qquad (9)$$

Lemma 2. *For any initial condition $\mathbf{s}_0$ and associated constant directional matrix $\mathbf{Q}$ in (4), the trajectory $\mathbf{x}(t)$ under system dynamics (3) admits a tight ellipsoid bound, $\mathbf{x}(t) \in \mathcal{E}_{\mathbf{Q}}(\mathbf{0}, \eta(t_0)) \subseteq \mathcal{E}_{\mathbf{Q}}(\mathbf{0}, \delta(t_0))$, for all $t \geq t_0$, where $\eta(t_0)$ is the solution to (5) and $\delta(t_0)$ is the solution to (9).*

Proof. By definition, $\mathbf{x}(t) \in \mathcal{E}_{\mathbf{Q}}(\mathbf{0}, \eta(t_0))$ is equivalent to $d_{\mathbf{Q}}^2(\mathbf{0}, \mathbf{x}(t)) \leq \eta(t_0)$. Since $\delta(t_0) = \max_{\boldsymbol{\xi} \in \mathcal{E}_{inv}}\boldsymbol{\xi}^T\mathbf{C}^T\mathbf{C}\boldsymbol{\xi}$, inequality (8) yields $\delta(t_0) \geq \eta(t_0) \geq \|\mathbf{z}(t)\|^2 = \|\mathbf{x}(t)\|_{\mathbf{Q}}^2 = d_{\mathbf{Q}}^2(\mathbf{0}, \mathbf{x}(t))$. Hence, $\mathbf{x}(t) \in \mathcal{E}_{\mathbf{Q}}(\mathbf{0}, \eta(t_0)) \subseteq \mathcal{E}_{\mathbf{Q}}(\mathbf{0}, \delta(t_0))$. $\square$

Now, we know how to find an accurate outer approximation of the system trajectory in the SDDM defined by (4). We are ready to develop a feedback controller that utilizes the trajectory bounds to quantify the safety of the system with respect to surrounding obstacles, while following the navigation path towards the goal.

C. Structure of the Robot-Governor Controller

The problem of collision checking is simple for first-order kinematic systems since they can stop instantaneously to avoid collisions. We introduce a *reference governor* [20], [21], a virtual first-order system: $\dot{\mathbf{g}} = \mathbf{u}_{\mathbf{g}}$ with state $\mathbf{g} \in \mathbb{R}^n$ and control input $\mathbf{u}_{\mathbf{g}} \in \mathbb{R}^n$, which will serve to simplify the conditions for maintaining stability and safety concurrently. Our proposed structure of a path-following control design is shown in Fig. 3. The reference governor behaves as a real-time reactive trajectory generator that continuously regulates a reference signal for the real robot dynamics depending on risk level evaluation using SDDM. More precisely, we choose the real system's control input $\mathbf{u}$ so that the robot tracks the governor state $\mathbf{g}$, while $\mathbf{g}$ is regulated via $\mathbf{u}_{\mathbf{g}}$ to ensure collision avoidance and stability for the joint robot-governor system.

In detail, let $\tilde{\mathbf{s}} := \mathbf{s} - \mathbf{P}^T\mathbf{g}$ be the system state with the first element changed from $\mathbf{x}$ to $(\mathbf{x} - \mathbf{g})$ to make $(\mathbf{g}, \mathbf{0})$ an equilibrium point. Choose a local controller for (1) that tracks the governor state $\mathbf{g}$:

$$\mathbf{u} = -\mathbf{K}\tilde{\mathbf{s}}. \qquad (10)$$

Consider the augmented *robot-governor system* with state $\hat{\mathbf{s}} = (\tilde{\mathbf{s}}, \mathbf{g}) \in \mathbb{R}^{(n_s+n)}$, coupling the real states with the governor state:

$$\dot{\hat{\mathbf{s}}} = \begin{bmatrix}\dot{\tilde{\mathbf{s}}} \\ \dot{\mathbf{g}}\end{bmatrix} = \begin{bmatrix}\bar{\mathbf{A}}\tilde{\mathbf{s}} \\ \mathbf{u}_{\mathbf{g}}\end{bmatrix}. \qquad (11)$$

Before proposing the design of the governor controller $\mathbf{u}_{\mathbf{g}}(t)$, we analyze the behavior of the robot-governor system in the case of static governor.
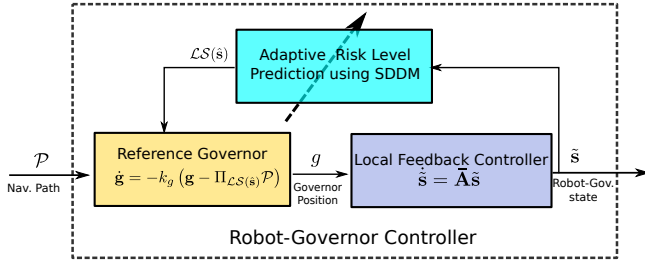
Fig. 3: Structure of the proposed controller. A virtual reference governor adaptively conveys global navigation information to a local feedback controller based on the prediction of robot position trajectory. A local safe zone $\mathcal{LS}(\hat{\mathbf{s}})$, depending only on the current system state $\hat{\mathbf{s}}$, is constructed from an SDDM trajectory bound. A time-varying local goal $\Pi_{\mathcal{LS}(\hat{\mathbf{s}})}\mathbf{r}$ is obtained by projecting the navigation path $\mathbf{r}$ onto the local safe zone $\mathcal{LS}(\hat{\mathbf{s}})$. The governor chases the local goal and continuously sends its updated state $\mathbf{g}$ as a reference signal to guide the local controller.

Lemma 3. *If the governor is static, i.e., $\mathbf{u_g}(t) \equiv 0$ so that $\mathbf{g}(t) \equiv \mathbf{g}_0$, then the robot-governor system in* (11) *is globally exponentially stable with respect to the equilibrium* $(\mathbf{g}_0, \mathbf{0}, \mathbf{g}_0)$.

Proof. The subsystem $\dot{\tilde{\mathbf{s}}} = \bar{\mathbf{A}}\tilde{\mathbf{s}}$ has an equilibrium at $(\mathbf{g}_0, \mathbf{0})$, which is globally exponentially stable because $\bar{\mathbf{A}}$ is Hurwitz, while $\mathbf{g}(t) \equiv \mathbf{g}_0$ by assumption. $\square$

In addition to guaranteeing stability for a static governor, we can use the ellipsoidal trajectory bounds from Lemma 2 to ensure safety.

Theorem 1. *Let $(\mathbf{x}_0 - \mathbf{g}_0, \mathbf{y}_0, \mathbf{g}_0)$ be any initial state for the robot-governor system in* (11) *with $\mathbf{x}_0, \mathbf{g}_0 \in \mathcal{F}$. Suppose that $\mathbf{u_g}(t) \equiv \mathbf{0}$ so that $\mathbf{g}(t) \equiv \mathbf{g}_0$. Let $\mathbf{Q} := \mathbf{Q}\,[\mathbf{g}_0 - \mathbf{x}_0] \in \mathbb{S}^n_{>0}$ be a constant directional matrix and suppose that the following safety condition is satisfied:*

$$\delta(t_0) \leq d_{\mathbf{Q}}^2(\mathbf{g}_0, \mathcal{O}), \tag{12}$$

where $\delta(t_0)$ is an upper bound for $\|\mathbf{x}(t) - \mathbf{g}_0\|_{\mathbf{Q}}^2$ obtained according to Lemma 2. Then, the robot-governor system is globally exponentially stable with respect to the equilibrium $(\mathbf{g}_0, \mathbf{0}, \mathbf{g}_0)$ and, moreover, the robot trajectory is collision free, i.e., $\mathbf{x}(t) \in \mathcal{F}$, for all $t \geq t_0$.

Proof. Since $(\mathbf{g}_0, \mathbf{0})$ is an equilibrium for $\dot{\tilde{\mathbf{s}}} = \bar{\mathbf{A}}\tilde{\mathbf{s}}$, by Lemma 2, $\mathbf{x}(t) \in \mathcal{E}_{\mathbf{Q}}(\mathbf{g}_0, \delta(t_0))$ for all $t \geq t_0$. From the safety condition in (12), $\mathbf{x}(t) \in \mathcal{E}_{\mathbf{Q}}(\mathbf{g}_0, \delta(t_0)) \subseteq \mathcal{E}_{\mathbf{Q}}(\mathbf{g}_0, d_{\mathbf{Q}}^2(\mathbf{g}_0, \mathcal{O})) \subseteq \mathcal{F}$ for all $t \geq t_0$. Stability is ensured by Lemma 3. $\square$

D. Local Projected Goal and Governor Control Policy

We established that the robot-governor system is stable and safe as long as the governor is static and (12) holds. Next, we consider how to move the governor without violating these properties. Based on (10), we know that the robot will attempt to track the governor state. The main idea is to choose a time varying directional matrix $\mathbf{Q}(t) = \mathbf{Q}\,[\mathbf{g}(t) - \mathbf{x}(t)]$ to measure the system's safety. Since $\mathbf{Q}(t)$ is positive definite (by Lemma 1), it can still be used to define an SDDM $\|\cdot\|_{\mathbf{Q}(t)}$. Then, Lemma 2, can still provide

an accurate robot trajectory bound $\mathcal{E}_{\mathbf{Q}(t)}(\mathbf{g}(t), \delta(t))$, which takes the robot's direction of motion into account. We need to design the governor control policy $\mathbf{u_g}(t)$ so that $\delta(t)$ never violates a time-varying version of the safety condition in (12).

Our approach is to define an ellipsoid $\mathcal{LS}(\hat{\mathbf{s}})$, called a *local safe zone*, centered at the governor state $\mathbf{g}$, and have the size of $\mathcal{LS}(\hat{\mathbf{s}})$ determine how fast the governor can move. In the worst case, if system safety or stability are endangered, $\mathcal{LS}(\hat{\mathbf{s}})$ should shrink to a point, forcing the governor to remain static. Once there is enough leeway in the safety conditions in (12), $\mathcal{LS}(\hat{\mathbf{s}})$ can grow, allowing the governor to move without endangering the safety or stability.

Definition 3. A *local safe zone* is a time-varying set that at time t depends on the robot-governor state $\hat{\mathbf{s}}(t)$ as follows:

$$\mathcal{LS}(\hat{\mathbf{s}}) := \left\{ \mathbf{q} \in \mathcal{F} \mid d_{\mathbf{Q}}^2(\mathbf{q}, \mathbf{g}) \leq \max\left(0, \Delta E(\hat{\mathbf{s}})\right) \right\}, \tag{13}$$

where $\mathbf{Q}(t) := \mathbf{Q}\,[\mathbf{g}(t) - \mathbf{x}(t)]$ is a directional matrix, determined by $\hat{\mathbf{s}}(t)$, and $\Delta E(\hat{\mathbf{s}}(t)) := d_{\mathbf{Q}(t)}^2(\mathbf{g}(t), \mathcal{O}) - \delta(t)$ is a measure of leeway to safety violation, determined by an upper bound $\delta(t)$ on $\|\mathbf{x}(\tau) - \mathbf{g}(\tau)\|_{\mathbf{Q}(t)}$ for all $\tau \geq t$ and the directional distance $d_{\mathbf{Q}(t)}^2(\mathbf{g}(t), \mathcal{O})$ from the governor to the nearest obstacle.

The term $\Delta E(\hat{\mathbf{s}})$ estimates safety of the system based on local environment geometry and robot activeness. The requirement $\Delta E(\hat{\mathbf{s}}) \geq 0$ only places a constraint on the magnitude $\|\dot{\mathbf{g}}\|$, so $\dot{\mathbf{g}}/\|\dot{\mathbf{g}}\|$ is a degree of freedom that can be utilized to make $\mathbf{g}$ asymptotically tend to a desired goal. We define a local goal for the governor.

Definition 4. A *local projected goal* is the farthest point along the path $\mathbf{r}$ contained in the local safe zone $\mathcal{LS}(\hat{\mathbf{s}})$:

$$\alpha^* := \max_{\alpha \in [0,1]} \left\{ \alpha \mid \mathbf{r}(\alpha) \in \mathcal{LS}(\hat{\mathbf{s}}) \right\} \quad \bar{\mathbf{g}}(\hat{\mathbf{s}}) = \mathbf{r}(\alpha^*) \tag{14}$$

The informal notation $\bar{\mathbf{g}}(\hat{\mathbf{s}}) = \Pi_{\mathcal{LS}(\hat{\mathbf{s}})}\mathbf{r}$ will be used for the local projected goal to emphasize that $\bar{\mathbf{g}}(\hat{\mathbf{s}})$ is determined by projecting the path $\mathbf{r}$ onto the local safe zone $\mathcal{LS}(\hat{\mathbf{s}})$. The structure of the complete closed-loop control policy is illustrated in Fig. 3, while the definitions of a local safe zone and a local projected goal are visualized in Fig. 5. We are finally ready to define the *governor control policy*:

$$\mathbf{u_g} := -k_g\left(\mathbf{g} - \bar{\mathbf{g}}(\hat{\mathbf{s}})\right) \tag{15}$$

where $k_g > 0$ is a control gain for the governor controller. We prove that the closed-loop system is stable, safe, and asymptotically reaches the goal specified by the path $\mathbf{r}$. We also informally claim that the path-following controller is fast due to the use of directional information for safety verification. This claim is supported empirically in Sec. V.

Theorem 2. *Given a path $\mathbf{r}$, the closed-loop robot-governor system* (11) *with move-to-projected-goal* (15) *is asymptotically steered from any safe initial state $\hat{\mathbf{s}}_0$, i.e., $\Delta E(\hat{\mathbf{s}}_0) > 0$ and $\mathbf{r}(\alpha) \in \mathcal{LS}(\hat{\mathbf{s}}_0)$ for some $\alpha \in [0,1]$, to the goal state $\hat{\mathbf{s}}^* = (\mathbf{r}(1), \mathbf{0}, \mathbf{r}(1))$ and the robot trajectory is collision-free for all time, i.e., $\mathbf{x}(t) \in \mathcal{F}$ for all $t \geq 0$.*
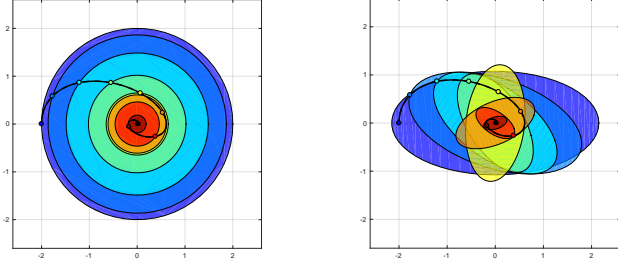
Fig. 4: Trajectory bounds comparison between a Euclidean metric (left) and an SDDM (right). The governor is fixed at the origin while the robot's initial conditions are $\mathbf{x}_0 = (-2, 0)$ and $\dot{\mathbf{x}}_0 = (0, 2)$. The change of the trajectory bounds over time is illustrated via ellipsoids with different colors, starting from cold/blue and converging towards warm/red.

Proof. From Thm. 1, we know that the robot-governor system (11) will asymptotically converge to the equilibrium point $(\mathbf{g}, 0, \mathbf{g})$ without collisions if the governor is static. The governor control policy in (15) allows the governor to move only when the interior of $\mathcal{LS}(\hat{\mathbf{s}})$ is nonempty. From Def. 3, this happens only if the safety condition is strictly satisfied, i.e., $\Delta E(\hat{\mathbf{s}}) = d^2_{\mathbf{Q}(t)}(\mathbf{g}(t), \mathcal{O}) - \delta(t) > 0$. Since $\mathbf{x}$ approaches $\mathbf{g}$, $\Delta E(\hat{\mathbf{s}})$ eventually becomes strictly positive and the set $\mathcal{LS}(\hat{\mathbf{s}})$ becomes an ellipsoid in free space with non-empty interior. Since initially $\mathbf{r}(\alpha) \in \mathcal{LS}(\hat{\mathbf{s}}_0)$ for some $\alpha \in [0, 1]$, the local projected goal in (14) will be well defined and when $\mathcal{LS}(\hat{\mathbf{s}})$ grows, the projected goal will move further along the path $\mathbf{r}$, i.e., the path length parameter α will increase. Since the system dynamics are continuous, $\Delta E(\hat{\mathbf{s}})$ cannot suddenly become negative without crossing zero. If $\Delta E(\hat{\mathbf{s}}) \downarrow 0$, the local energy zone $\mathcal{LS}(\hat{\mathbf{s}})$ shrinks to a point, i.e., $\mathcal{LS}(\hat{\mathbf{s}}) = \{\mathbf{g}\}$, and hence the governor stops moving and waits until the robot catches up. When the governor is static, and since the safety condition in (12) is satisfied, Thm. 1 again guarantees that the robot can approach the governor without collisions, increasing $\Delta E(\hat{\mathbf{s}})$ in the process. Once $\Delta E(\hat{\mathbf{s}})$ goes above 0, the governor starts moving towards the goal again by chasing the projected goal. Note that the local projected goal always lies on the navigation path inside the free space, i.e., $\bar{\mathbf{g}} \in \mathbf{r} \subset \mathring{\mathcal{F}}$, so the robot-governor system cannot remain stuck at any configuration except $\hat{\mathbf{s}}^* = (\mathbf{r}(1), 0, \mathbf{r}(1))$. Using LaSalle's Invariance Principle [25], we can conclude that the largest invariant set is the point where both the robot and the governor are stationary at the goal location $\mathbf{r}(1)$. □

V. EVALUATION

Consider an acceleration-controlled robot, stabilized by a proportional-derivative (PD) controller:

$$\ddot{\mathbf{x}} = \mathbf{u} := -2k\mathbf{x} - \zeta\dot{\mathbf{x}}. \tag{16}$$

The closed-loop robot-governor system is:

$$\dot{\hat{\mathbf{s}}} = \begin{bmatrix} \dot{\mathbf{x}} \\ \ddot{\mathbf{x}} \\ \dot{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{x}} \\ -2k(\mathbf{x} - \mathbf{g}) - \zeta\dot{\mathbf{x}} \\ -k_g(\mathbf{g} - \bar{\mathbf{g}}(\hat{\mathbf{s}})) \end{bmatrix} \tag{17}$$
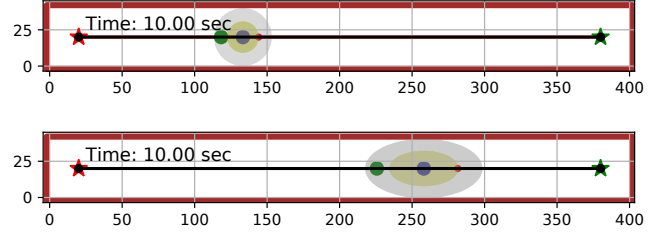

Fig. 5: Comparison of controller 1 (top) and 2 (bottom) in a corridor simulation. A snapshot is shown at the same time instant for both controllers. The local energy zone (yellow) resulting from the proposed SDDM trajectory bounds fits the corridor environment well, leading to fast, yet safe, movement.

Several experiments will be shown to compare the performance of two path-following controllers:

- **Controller 1** [22]: uses a Lyapunov function to ensure stability and collision avoidance. The robot's kinetic and potential energy, i.e., $E(\hat{\mathbf{s}}) := k\|\mathbf{x} - \mathbf{g}\|^2 + \frac{1}{2}\|\dot{\mathbf{x}}\|^2$ is used to define a spherical local safe zone.
- **Controller 2** (ours): uses SDDM trajectory bounds (Lemma 2) to define an ellipsoidal local safe zone.

In visualizations, the governor and robot positions are shown by a blue and green dot, respectively. A light-gray ellipse/ball indicates the distance from the governor to the nearest obstacles, while the local safe zone $\mathcal{LS}(\hat{\mathbf{s}})$ is indicated by a yellow ellipse/ball. The projected goal $\bar{\mathbf{g}}$ is shown as a small red dot at the boundary of $\mathcal{LS}(\hat{\mathbf{s}})$. The controller parameters were $k = k_g = 1$, $\zeta = 2\sqrt{2}$, $c_1 = 1, c_2 = 4$.

Trajectory Prediction using Different Metrics. First, we demonstrate that predicting the robot trajectory using our directional metric has some desirable properties for enforcing safety constraints. Fig. 4 compares trajectory bounds obtained from Lemma 2 for (16) using a Euclidean metric and an SDDM. Since the robot dynamics are simple, a tight directional trajectory bound $\mathcal{E}_{\mathbf{Q}_t}(\mathbf{0}, \eta(t))$ can be obtained from an exact computation of the critical points according to (6). It is clear that the ellipsoid bounds on the system trajectory are less conservative (smaller area/volume) than the spherical bounds at beginning. Unlike a Lyapunov function, the ellipse $\mathcal{E}_{\mathbf{Q}(t)}(\mathbf{0}, \eta(t))$ bounding the robot trajectory is not forward invariant. It can be shown that requiring invariance of directional ellipsoids ($\mathcal{E}_{\mathbf{Q}_{(t_1)}} \subset \mathcal{E}_{\mathbf{Q}_{(t_2)}}$ $\forall t_2 \geq t_1$) would need infinite damping unless $\mathbf{Q} = k\mathbf{I}$ for some $k > 0$, causing the metric to lose directionality. In contrast to control designs based on Lyapunov function invariance, we make an interesting observation that a safe and stable controller can be defined even if the sets containing the system trajectory are not strictly shrinking over time.

Corridor Environment. We show that utilizing a directional metric in the control design alleviates the corridor effect discussed in Sec. I. We setup a simulation requiring a robot to navigate through a corridor (Fig. 5). The results show that controller 1, using a Lyapunov function with spherical level sets, suffers from the corridor effect while the proposed controller 2, making directional predictions about the system trajectory, does not.
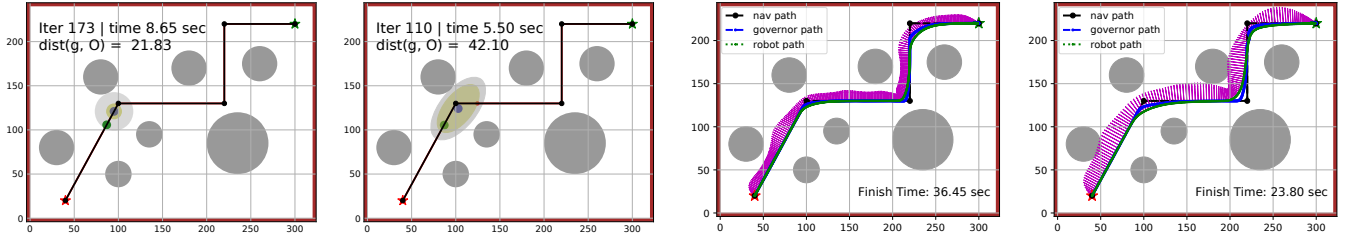
Fig. 6: Simulation of the robot-governor system tracking a piecewise-linear path (black) in an environment with circular obstacles (dark gray circles). The start and end of the path are indicated by a red and green star, respectively. The two plots on the left show that the robot at around the same location behaves differently due to different distance measures. The controller using Euclidean distance is overly cautious with respect to lateral obstacles resulting in conservative motion. The system employing SDDM trajectory bounds has a larger local safe zone, which helps the robot turn fast and smoothly. The two plots on the right show the trajectories followed by the systems employing the two controllers. The velocity profiles are shown as magenta arrows perpendicular to the robot path. The controller based on SDDM trajectory bounds (rightmost) results in higher velocities compared to the controller using Euclidean ball invariant sets. Note that the path followed by the robot (green line) is also smoother, especially when turning, for the directional controller despite the higher velocity.
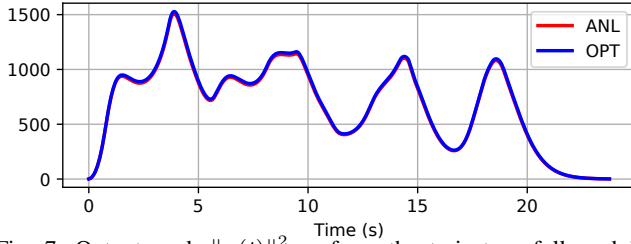

Fig. 7: Output peak $\|\mathbf{x}(t)\|^2_{\mathbf{Q}(t)}$ from the trajectory followed in Fig. 6. The red curve is $\eta(t, \mathbf{Q}(t))$ obtained analytically from eq. (6). The blue curve is $\delta(t)$ computed from the SDP optimization in eq. (9). It is clear that $\delta(t)$ is an upper bound for $\eta(t, \mathbf{Q}(t))$, and the bound is tight at certain moments. Analytical bounds are used in simulation, while optimization bounds are computed for comparison purpose.

Sparse Environment with Circular Obstacles. This experiment compares the two controllers in a longer path-following task in an environment with circular obstacles. Snapshots illustrating how the two controllers judge distances to obstacles and define a local energy zone are shown in Fig. 6. It can be seen that the controller equipped with a directional sensing ability has a better understanding of the local environment geometry, leading to a larger, elongated local safe zone set. As a result, controller 2 does not need to slow down for low-risk lateral obstacles, leading to smoother and faster navigation. The directional bounds on the robot trajectory obtained analytically, according to eq. (6), and from the SDP in eq. (9) are compared in Fig. 7.

Unknown Environment with Arbitrary Obstacles. This experiment demonstrates that our controller can work in a complex unknown environment, shown in Fig. 1, relying only on local onboard measurements. The directional distance $d^2_{\mathbf{Q}(t)}(\mathbf{g}(t), \mathcal{O})$ from the governor to the obstacles is computed from the latest lidar scans. The path $\mathcal{P}$ is re-planned from the current governor position to the goal using an occupancy grid map constructed from the lidar scans over time, as illustrated in Fig. 8.

VI. CONCLUSION

This paper presented a path-following controller relying on a state-dependent directional metric for trajectory predic-
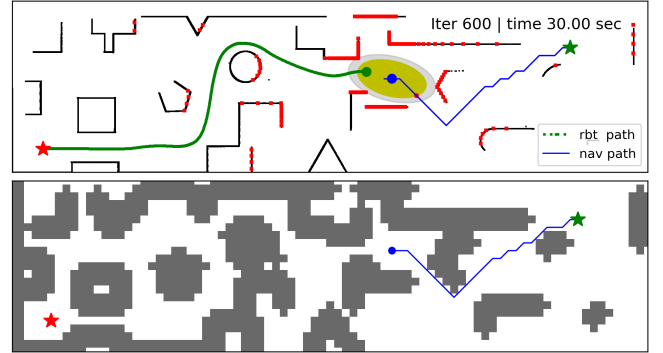

Fig. 8: Snapshot of the robot-governor system navigating the environment shown in Fig. 1. Streaming lidar scan measurements (red dots) are used to update an occupancy grid map (black lines and white regions in top plot) of the unknown environment. An acceleration-controlled robot (green dot) follows a virtual governor (blue dot) whose motion is modulated based on the local energy zone (yellow ellipse) and the directional distance to obstacles (gray ellipse). A navigation path (blue line) is periodically replanned using an A^* planner and an inflated occupancy grid map (bottom plot).

tion and safety quantification. The controller achieves fast tracking in unknown complex environments, mitigating the corridor effect, while providing safety and stability guarantees. The approach offers a promising direction for ensuring the safety of mobile autonomous systems operating in dynamically changing environments. Our design places very minimal requirements on the navigation path (piecewise-continuity) but the overall system performance depends on the path quality. Future work will focus on incorporating safety and stability considerations in path planning, applying our results to complex robot dynamics, and demonstrating the effectiveness of our design in hardware experiments.

REFERENCES

[1] M. Likhachev, G. J. Gordon, and S. Thrun, "ARA*: Anytime A* with provable bounds on sub-optimality," in *Advances in neural information processing systems (NeurIPS)*, 2004, pp. 767–774.

[2] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research (IJRR)*, vol. 30, no. 7, pp. 846–894, 2011.

[3] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 2537–2542.

[4] D. J. Webb and J. van den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 5054–5061.

[5] V. Pacelli, O. Arslan, and D. E. Koditschek, "Integration of local geometry and metric information in sampling-based motion planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3061–3068.

[6] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, no. 3, 2017.

[7] F. Gao, W. Wu, Y. Lin, and S. Shen, "Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 344–351.

[8] S. Liu, N. Atanasov, K. Mohta, and V. Kumar, "Search-based motion planning for quadrotors using linear quadratic minimum time control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[9] M. T. Mason and J. K. Salisbury Jr, *Robot Hands and the Mechanics of Manipulation*. The MIT Press, Cambridge, MA, 1985.

[10] R. Burridge, A. Rizzi, and D. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *The International Journal of Robotics Research (IJRR)*, vol. 18, no. 6, pp. 534–555, 1999.

[11] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification," *The International Journal of Robotics Research (IJRR)*, 2009.

[12] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 8, pp. 947–982, 2017.

[13] T. Gawron and M. M. Michalek, "VFO feedback control using positively-invariant funnels for mobile robots travelling in polygonal worlds with bounded curvature of motion," in *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2017, pp. 124–129.

[14] ——, "Algorithmization of constrained motion for car-like robots using the VFO control strategy with parallelized planning of admissible funnels," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 6945–6951.

[15] W. Tan, A. Packard, *et al.*, "Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming," *IEEE Transactions on Automatic Control (TAC)*, vol. 53, no. 2, p. 565, 2008.

[16] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[17] G. Wu and K. Sreenath, "Safety-critical and constrained geometric control synthesis using control lyapunov and control barrier functions for systems evolving on manifolds," in *American Control Conference (ACC)*, 2015, pp. 2038–2044.

[18] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[19] G. Wu and K. Sreenath, "Safety-critical control of a planar quadrotor," in *American Control Conference (ACC)*, 2016, pp. 2252–2258.

[20] E. Garone and M. M. Nicotra, "Explicit reference governor for constrained nonlinear systems," *IEEE Transactions on Automatic Control (TAC)*, vol. 61, no. 5, pp. 1379–1384, 2016.

[21] I. Kolmanovsky, E. Garone, and S. Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *American Control Conference (ACC)*, 2014.

[22] O. Arslan and D. E. Koditschek, "Smooth extensions of feedback motion planners via reference governors," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[23] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[24] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. Siam, 1994, vol. 15.

[25] H. Khalil, *Nonlinear systems*. Prentice Hall, 2002.