

# Integration of Local Geometry and Metric Information in Sampling-Based Motion Planning

Vincent Pacelli<sup>1</sup> and Omur Arslan<sup>2</sup> and Daniel E. Koditschek<sup>2</sup>

**Abstract**—The efficiency of sampling-based motion planning algorithms is dependent on how well a steering procedure is capable of capturing both system dynamics and configuration space geometry to connect sample configurations. This paper considers how metrics describing local system dynamics may be combined with convex subsets of the free space to describe the local behavior of a steering function for sampling-based planners. Subsequently, a framework for using these subsets to extend the steering procedure to incorporate this information is introduced. To demonstrate our framework, three specific metrics are considered: the LQR cost-to-go function, a Gram matrix derived from system linearization, and the Mahalanobis distance of a linear-Gaussian system. Finally, numerical tests are conducted for a second-order linear system, a kinematic unicycle, and a linear-Gaussian system to demonstrate that our framework increases the connectivity of sampling-based planners and allows them to better explore the free space.

## I. INTRODUCTION

Sampling-based planners, e.g. probabilistic roadmaps (PRM) [1] and rapidly-exploring random trees (RRTs) [2], have enjoyed wide success in motion planning literature with application in a diverse range of systems including automobiles [3], manipulators [4], and multi-agent aerial systems [5]. The popularity of these algorithms is a consequence of the fact that they are often probabilistically complete even when the configuration space of a robot is difficult to represent [2], [6]. However, the rate at which these algorithms explore the free space depends on how well their distance metrics—and the steering procedure they attempt to describe—capture the true distance between configurations [7]. Unfortunately, determining the shortest path between configurations with respect to both dynamical and free space constraints is often as difficult to compute as solving the original path planning problem. As such, dynamical and free space information are typically incorporated in separate algorithmic steps. In this paper, we instead consider a framework for supplementing steering procedures in sampling-based planners to account for local properties of both the configuration space and underlying dynamical system.

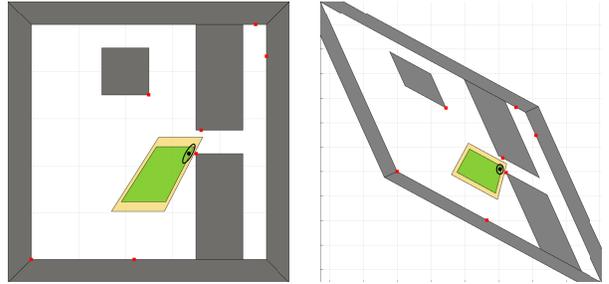
### A. Motivation and Related Work

Traditionally, system dynamics is often incorporated into a sampling-based planner by using a steering policy that

This work was supported in part by AFRL grant FA865015D1845 (subcontract 669737-1).

<sup>1</sup>Vincent Pacelli is with the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544. E-mail: vpacelli@princeton.edu

<sup>2</sup>Omur Arslan and Daniel E. Koditschek are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA. E-mail: {omur, kod}@seas.upenn.edu



**Fig. 1:** (Left) An example of the generalized local workspace  $\mathcal{LW}(x; \mathbf{S}, 1)$  (yellow) and free space  $\mathcal{LF}(x; \mathbf{S}, 1)$  (green) constructed about  $x = [6.7 \ 4.5]^T$  (black circle) using the metric defined by  $\mathbf{S} = [0.0680 \ 0.0854; 0.0854 \ 0.1655]$ . The boundary of ellipsoid  $\mathcal{E}(x; \mathbf{S}, 1)$  is shown in black. The workspace obstacles are shown in gray while the metric projection computed using  $\|\cdot\|_{\mathbf{S}}$  onto each obstacle is shown in red. (Right) The generalized local workspace / free space, workspace obstacles, and metric projections under the transformation  $h(x) = \mathbf{S}^{-\frac{1}{2}}x$ . The transformed generalized local free space is equivalent to the Euclidean local free space computed with respect to the transformed obstacles.

also provides a notion of cost-to-go between states to use as a metric. Two examples include, linear-quadratic regulators [8]–[10] and control Lyapunov functions [11]. Another approach taken in [12], [13], is to develop a statistical model of a cost-to-go function, when not available explicitly from a steering function, using example trajectories produced offline. Such a cost-to-go function allows the motion planner to estimate the geodesic distance between states and to determine which state to expand at each iteration.

However, these metrics do not incorporate any information about the free space geometry and do not offer any guidance on how to steer a dynamical system to reflect it. Instead, geometric information about configuration space obstacles is primarily exploited in a global manner through modification of the distribution used in the sampling step. For example, in [14], the sample distribution was modified by replacing samples in collision near the boundary of the free space with nearby collision-free ones. Similarly, in [15], [16] sampling was biased to occur more often in regions with a high ratio of samples in and out of collision. Finally, [17] considers midpoints between configurations that are deemed to not be connected, thereby biasing the sample distribution in obstacle dense regions. Overall, these global approaches are not able to incorporate any local dynamical information and, therefore, offer no guidance for local steering.

In contrast to these biased sampling approaches, a steering procedure was considered in [18] that was defined in terms of the *Euclidean local free space* to exploit local geometric aspects of the free space. The local free space for a con-

figuration is a polytope defined by the set of maximum-margin separating hyperplanes between the configuration and each obstacle, and is a convex approximation of the free space surrounding the configuration [19], [20]. The planning algorithm then attempts to steer the system toward the *projected goal*—the Euclidean projection of a sample goal configuration onto the local free space—effectively incorporating local geometry into the steering procedure. However, when system dynamics are present, defining the local free space in terms of the Euclidean distance is no longer suitable because the shortest path length to the free space boundary from a configuration will often be greater than the Euclidean distance between the configuration and the point of collision. This work is motivated by this fact and attempts to combine local dynamics with local geometry, through the local free space, in order to design effective steering motion for dynamical systems.

### B. Contributions and Organization

The contributions of this paper are threefold. First, we propose a natural generalization of the definition of the local free space used in [18]–[20]. Specifically, we relax the inner product used in the definition of the local free space to any inner product defined by a positive-definite matrix. The specified inner product also induces a metric which is used to incorporate local information. A simple framework is then established that uses the generalized local free space to create steering procedures that reflect both the local behavior of the dynamical system and the local geometry of the free space. Furthermore, we provide two example metrics from the literature that describe the behavior of the dynamical system locally: the cost-to-go function from an infinite-horizon LQR policy [8] and the Mahalanobis distance for a linear-Gaussian system [21]–[23]. We also introduce a novel metric based on the Gram matrix formed by the linearized dynamics of an autonomous system. Additionally, in the case of the Mahalanobis distance metric, we demonstrate that the steering procedure provides bounds on the probability of collision while following the planned trajectory. Finally, these steering procedures are demonstrated to improve the connectivity of a sampling-based planner for unicycle and linear-Gaussian systems in numerical examples when compared to using the local control policy directly without the local free space.

The rest of this paper is organized as follows. In Section II, we provide an overview of the RRT algorithm, which is used as the sampling-based motion planner in our numerical simulations in Section V. In Section III, we formally introduce a new notion of generalized local free space and state its properties. Then, in Section IV we present example metrics for describing the local behaviour of dynamical systems.

## II. BACKGROUND: RAPIDLY-EXPLORING RANDOM TREES

The rapidly-exploring random tree (RRT) algorithm is a common sampling-based planner. It can be applied regardless of whether the robot is modeled as a discrete-time system  $x_{t+1} = f(x_t, u_t)$  or a continuous-time system

$\dot{x}(t) = f(x(t), u(t))$  and is defined in terms of the following functions

- 1)  $x = \text{Sample}(\mathcal{C})$  randomly samples independent, identically distributed configurations from the compact set  $\mathcal{C}$ . While sampling may be biased toward certain regions of interest, e.g. [14]–[17], all examples presented in this paper are based on uniform sampling.
- 2)  $d = \text{Dist}(x_1, x_2)$  returns an estimate of the geodesic distance between configurations  $x_1$  and  $x_2$ . Since computing the exact geodesic distance between configurations can be challenging, this function often encodes some notion of an approximate “cost-to-go” value related to how a chosen local controller steers the system from  $x_1$  to  $x_2$  [8], [10], [11]. It is not required that  $\text{Dist}$  be a proper metric because, when dynamical constraints are considered, the geodesic distance from  $x_1$  to  $x_2$  is not necessarily the same as from  $x_2$  to  $x_1$ .
- 3)  $x^* = \text{NearestNeighbor}(x, \mathcal{X})$  produces the closest member of the finite set  $\mathcal{X}$  to  $x$ , i.e. if  $\mathcal{X} = \{x_1, \dots, x_m\}$ , then  $x^* = \min_{i=1, \dots, m} \text{Dist}(x, x_i)$  is returned.
- 4)  $\mathcal{T} = \text{Steer}(x_1, x_2)$  computes a finite-time segment of the trajectory that results from applying a local control policy on the system from  $x_1$  toward  $x_2$ . It is not required that the trajectory segment terminates at  $x_2$ . The set of states, denoted as  $\mathcal{T}$ , traversed by the system over the finite-time horizon is returned. The final state in the trajectory segment is denoted by  $\mathcal{T}(\text{end})$ .
- 5)  $\text{CollisionFree}(\mathcal{T}, \mathcal{C})$  determines whether or not the trajectory  $\mathcal{T}$ , produced by  $\text{Steer}$ , is collision-free with regard to the free space  $\mathcal{C}$ . How “collision-free” is defined depends on the robot model under consideration. In the three examples considered in this paper, specific definitions of this function will be provided. If  $\mathcal{T}$  is collision-free, then the function returns `true` otherwise it returns `false`.

Algorithm 1 describes how to construct an RRT over the robot’s configuration space  $\mathcal{C}$ , which is assumed to be compact. The algorithm approximates the connectivity of the free space  $\mathcal{C}$  by growing a tree, represented as graph  $G = (\mathcal{X}, \mathcal{E})$ , over a finite number of iterations  $K \in \mathbb{N}_+$ . At each step, a new configuration is sampled from the free space and an attempt is made to connect it to the tree via the closest member of  $\mathcal{X}$  as measured by  $\text{Dist}$ . If  $\text{Steer}$  produces a trajectory,  $\mathcal{T}$ , that is collision-free, then the final configuration of the trajectory,  $\mathcal{T}(\text{end})$ , is added to the tree.

The probabilistic completeness of this algorithm largely depends on the chosen  $\text{Steer}$  procedure for the underlying system dynamics. Some situations where the algorithm is not complete are examined in [24]. In [2], the algorithm is demonstrated to be complete when  $\text{Steer}$  samples from a finite set of control actions. In an unconstrained kinematic setting, a basic condition for completeness is presented in [1]. In [6], a sufficient condition of weak local controllability is established for general nonlinear systems. Informally, weak local controllability means that the steering procedure is able to connect configurations that are close together. The steering

---

**Algorithm 1:** Rapidly-Exploring Random Tree (RRT) [2]

---

**Input:**  $x_0 \in \mathcal{C}, K \in \mathbb{N}_+$   
**Output:**  $G = (\mathcal{V}, \mathcal{E})$

- 1  $\mathcal{V} \leftarrow \{x_0\}, \mathcal{E} \leftarrow \emptyset;$
- 2 **for**  $i = 1, \dots, K$  **do**
- 3      $x_{rand} \leftarrow \text{Sample}(\mathcal{F});$
- 4      $x_{near} \leftarrow \text{NearestNeighbor}(x_{rand}, \mathcal{V});$
- 5      $\mathcal{T} \leftarrow \text{Steer}(x_{near}, x_{rand});$
- 6     **if**  $\text{CollisionFree}(\mathcal{T}, \mathcal{F})$  **then**
- 7          $\mathcal{V} \leftarrow \mathcal{V} \cup \{\mathcal{T}(\text{end})\};$
- 8          $\mathcal{E} \leftarrow \mathcal{E} \cup \{(x_{near}, \mathcal{T}(\text{end}))\};$
- 9     **end**
- 10 **end**

---

procedure we introduce in Section III maintains this local controllability property as our framework only modifies a given steering policy when the goal configuration is far away.

The emphasis of this paper is how a given *Steer* function can be modified to simultaneously incorporate both local geometric and dynamical information. It will then be demonstrated experimentally that the new steering procedure will increase the connectivity of the RRT algorithm. The framework we introduce will also be applicable to other sampling-based algorithms, such as the probabilistically optimal RRT\* and the PRM (through the formulation in [18]) algorithms, that do not make additional assumptions on the properties of the *Steer* procedure.

### III. GENERALIZED LOCAL FREE SPACE STEERING

We now discuss how the concept of the *local free space* can be generalized from the definitions used in [18]–[20] to incorporate additional local information measured by a metric and discuss how this new construct can be used to define a *Steer* procedure that reflects both this metric and local workspace geometry.

#### A. Generalized Local Free Space

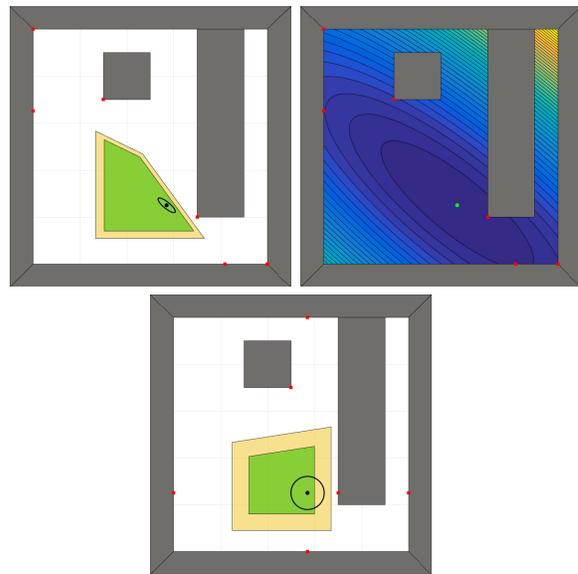
Consider a compact configuration space  $\mathcal{C} \subset \mathbb{R}^n$  populated with a finite family of closed, convex obstacles  $\mathcal{O}_1, \dots, \mathcal{O}_M \subset \mathbb{R}^n$ , i.e.  $\mathcal{C} = \mathbb{R}^n \setminus \bigcup_{i=1}^M \mathcal{O}_i$ .<sup>1</sup> For a robot configuration  $x \in \mathcal{C}$ , with a spherical safety zone of radius  $r > 0$  around it, the *Euclidean local work space* [19], [20] is defined as

$$\mathcal{LW}(x; r) := \left\{ x' \mid n_i^T(x' - (m_i + rn_i)) \leq 0, \quad \forall i \right\}, \quad (1)$$

$$s_i = \Pi_{\mathcal{O}_i}(x), \quad m_i = \frac{x + s_i}{2}, \quad n_i = \frac{s_i - x}{2\|s_i - x\|_2},$$

where  $\Pi_{\mathcal{A}}(x) = \arg \min_{y \in \mathcal{A}} \|x - y\|_2$  is the metric projection of  $x$  onto the closed convex set  $\mathcal{A}$ . The set  $\mathcal{LW}(x; r)$  is a polytope contained in  $\mathcal{C}$  whose faces are defined by the maximum-margin separating hyperplane between each

<sup>1</sup>It is a common practice to represent obstacles as a union of convex polytopes because the surface features that define the closest point between convex polytopes persist under small perturbations and so the closest point between convex polytopes can be computed incrementally [25].



**Fig. 2:** (Upper Left) The generalized local workspace and free space for a point robot at  $x = [5.7, 2.5]^T$  with  $\mathbf{S} = [0.1367, -0.0963; -0.0963, 0.0968]$ . (Right) Level sets of the metric centered at  $x$  (green dot) in the configuration space. The boundary of each halfspace constraint lies tangent to a level set. Due to the alignment of these level sets, the projection of  $x$  onto the obstacles produces points within the narrow passage in the lower right. This behavior, in conjunction with the rotation of each constraint to align with the level sets of  $\|\cdot\|_{\mathbf{S}}$ , causes the generalized local free space to extend more into the passage than its Euclidean counterpart (Bottom).<sup>3</sup>

obstacle and the open ball  $\mathcal{B}(x; r)$  centered at  $x$  with radius  $r$ . Additionally, the *Euclidean local free space*  $\mathcal{LF}(x; r)$  is defined to be the set of configurations in  $\mathcal{LW}(x; r)$  at least a distance of  $r$  away from the free space boundary [19], [20]:

$$\mathcal{LF}(x; r) := \mathcal{LW}(x; r) \setminus (\partial \mathcal{LW}(x; r) \oplus \mathcal{B}(0; r)), \quad (2)$$

$$= \left\{ x' \mid n_i^T(x' - (m_i - rn_i)) \leq 0, \quad \forall i \right\}, \quad (3)$$

where  $\partial \mathcal{A}$  is the boundary of  $\mathcal{A}$  and  $\oplus$  is the Minkowski sum. A natural property of  $\mathcal{LF}(x; r)$  is that for  $x' \in \mathcal{LF}(x; r)$ ,  $\mathcal{B}(x'; r)$  is collision-free, i.e.  $\mathcal{B}(x'; r) \subset \mathcal{C}$  [19], [20].

Since  $(s_i - x)$  is normal to a face of  $\mathcal{LF}(x; r)$ , this halfspace primarily limits motion from  $x$  in the direction that produces the shortest path to collision with  $\mathcal{O}_i$  as measured by  $\|\cdot\|_2$ . However, when the system is subject to dynamical constraints, the closest point on an obstacle to  $x$  in terms of the geodesic distance may not be the same as the closest point on the obstacle as measured by  $\|\cdot\|_2$ . This fact motivates the consideration of a wider class of metrics for construction of  $\mathcal{LW}(x; r)$  and  $\mathcal{LF}(x; r)$  than the Euclidean distance. To this end, we define the *generalized local work space* to be

$$\mathcal{LW}(x; \mathbf{S}, r) := \left\{ x' \mid -n_i^T \mathbf{S}^{-1}(x' - (m_i + rn_i)) \leq 0, \quad \forall i \right\}, \quad (4)$$

$$s_i = \Pi_{\mathcal{O}_i}(x; \mathbf{S}), \quad m_i = \frac{x + s_i}{2}, \quad n_i = \frac{s_i - x}{2\|s_i - x\|_{\mathbf{S}}},$$

where  $\mathbf{S} \in \mathbb{S}_{++}^n$  (the set of positive-definite matrices),  $\Pi_{\mathcal{O}_i}(x; \mathbf{S}) = \arg \min_{y \in \mathcal{O}_i} \|x - y\|_{\mathbf{S}}$ , and  $\|x\|_{\mathbf{S}} = \sqrt{x^T \mathbf{S}^{-1} x}$ . This definition replaces the Euclidean inner product  $y^T x$  and norm with the inner product  $y^T \mathbf{S}^{-1} x$  and its induced norm.

Similarly, we define the *generalized local free space* to be

$$\begin{aligned} \mathcal{LF}(x; \mathbf{S}, r) &:= \mathcal{LW}(x; r) \setminus (\partial \mathcal{LW}(x; \mathbf{S}, r) \oplus \mathcal{E}(0; \mathbf{S}, r)), \quad (5) \\ &= \left\{ x' \mid -n_i^T \mathbf{S}^{-1} (x' - (m_i - rn_i)) \leq 0 \quad \forall i \right\}, \quad (6) \end{aligned}$$

where  $\mathcal{E}(x; \mathbf{S}, r) = \{x' \mid \|x' - x\|_{\mathbf{S}} < r\}$ . Note that  $\mathcal{LF}(x; \mathbf{S}, 0) = \mathcal{LW}(x; \mathbf{S}, 0)$ .

There are multiple ways to interpret these generalizations. Under this construction, each halfspace constraint limits the robot's motion from  $x$  most in directions that might produce a collision for the robot, with a ellipsoidal safety zone  $\mathcal{E}(x; \mathbf{S}, r)$ , most quickly with its defining obstacle as measured by  $\|\cdot\|_{\mathbf{S}}$ . Alternatively, these generalizations may be interpreted as a change of coordinates via an affine transformation as summarized in the following lemma.

**Lemma 1.** *For obstacle sets  $\mathcal{O}_1, \dots, \mathcal{O}_M$ , the generalized local workspace  $h(\mathcal{LF}(x; \mathbf{S}, r))$  about  $x$  is equal to  $\mathcal{LF}(h(x); r)$ , where  $h(x) = \mathbf{S}^{-\frac{1}{2}}x + g$ ,  $g \in \mathbb{R}^n$ , and  $\mathcal{LF}(h(x); r)$  is computed in a transformed environment whose obstacles are  $h(\mathcal{O}_1), \dots, h(\mathcal{O}_M)$ , the image of each obstacle through  $h$ , as illustrated in Figure 1.<sup>2</sup>*

*Proof.* Since  $\mathbf{S} \in \mathbb{S}_{++}^n$ , a positive-definite choice of  $\mathbf{S}^{-\frac{1}{2}}$  exists and is given by  $\mathbf{U}(\text{diag}(\frac{1}{\sqrt{\sigma_1}}, \dots, \frac{1}{\sqrt{\sigma_n}}))\mathbf{U}^T$  where  $\mathbf{U}(\text{diag}(\sigma_1, \dots, \sigma_n))\mathbf{U}^T$  is the singular-value decomposition of  $\mathbf{S}$ . Next, some consequences of the linearity of  $\mathbf{S}^{-\frac{1}{2}}$  are the identities

$$\Pi_{\mathcal{A}}(x; \mathbf{S}) = h^{-1}(\Pi_{h(\mathcal{A})}(h(x))), \quad (7)$$

$$h(x) - h(x') = \mathbf{S}^{-\frac{1}{2}}(x - x'), \quad (8)$$

$$\frac{h(x) + h(y)}{2} = h\left(\frac{x + y}{2}\right). \quad (9)$$

Plugging each of these into the definition of  $\mathcal{LF}(h(x); r)$  computed with respect to obstacles  $h(\mathcal{O}_1), \dots, h(\mathcal{O}_M)$  will produce  $h(\mathcal{LF}(x; \mathbf{S}, r))$ .  $\square$

A consequence of this perspective is that properties proved for  $\mathcal{LF}(x; r)$  that are preserved under an affine transformation of the space can typically be extended to  $\mathcal{LF}(x; \mathbf{S}, r)$ . One example is that  $\mathcal{LF}(x; \mathbf{S}, r)$  defines a safe zone, at any collision free  $x \in \mathcal{C}$ , for a robot with a ellipsoidal safety constraint  $\mathcal{E}(x; \mathbf{S}, r) \subset \mathcal{C}$ .

**Theorem 2.** *Let  $x \in \mathbb{R}^n$ . Then,  $\mathcal{E}(x; \mathbf{S}, r) \subset \mathcal{C}$  if and only if  $\mathcal{LF}(x; \mathbf{S}, r) \neq \emptyset$ .*

*Proof.* The forward direction follows from the fact that if  $\mathcal{E}(x; \mathbf{S}, r) \subset \mathcal{C}$ , then  $\mathcal{E}(x; \mathbf{S}, r) \cap \mathcal{O}_i = \emptyset \quad \forall i$  and therefore a separating hyperplane exists between  $\mathcal{E}(x; \mathbf{S}, r)$  and each  $\mathcal{O}_i$  due to the convexity of both objects [26]. The reverse direction is a consequence of the fact that a  $\mathcal{LF}(x; \mathbf{S}, r)$  being nonempty means there exists a separating hyperplane between  $\mathcal{E}(x; \mathbf{S}, r)$  and each obstacle  $\mathcal{O}_i$  and therefore  $\mathcal{E}(x; \mathbf{S}, r) \cap \mathcal{O}_i = \emptyset$ , which implies  $\mathcal{E}(x; \mathbf{S}, r) \subset \mathcal{C}$ .  $\square$

<sup>2</sup>In general  $\mathbf{S}^{\frac{1}{2}}$  is not a unique matrix and, for  $\mathbf{S} \in \mathbb{S}_{++}^n$ , not all choices of  $\mathbf{S}^{\frac{1}{2}}$  may be symmetric. One choice of  $\mathbf{S}^{\frac{1}{2}}$  is  $\mathbf{S}^{\frac{1}{2}} = \mathbf{U}(\text{diag}(\sqrt{\sigma_1}, \dots, \sqrt{\sigma_n}))\mathbf{U}^T$  where  $\mathbf{U}(\text{diag}(\sigma_1, \dots, \sigma_n))\mathbf{U}^T$  is the singular-value decomposition of  $\mathbf{S}$ .

Finally, the following proposition highlights the fact that  $\mathcal{LF}(x; \mathbf{S}, r)$  only contains configurations that do not violate the ellipsoidal safety constraint.

**Proposition 3.** *For any  $x' \in \mathcal{LF}(x; \mathbf{S}, r)$ ,  $\mathcal{E}(x'; \mathbf{S}, r) \subset \mathcal{W}$ .*

*Proof.* This statement is true due to the facts that  $\mathcal{LF}(x; \mathbf{S}, r)$  for obstacles  $\mathcal{O}_1, \dots, \mathcal{O}_M$  is equivalent to  $\mathcal{LW}(x; \mathbf{S}, 0)$  for obstacles  $\mathcal{E}(0; \mathbf{S}, r) \oplus \mathcal{O}_1, \dots, \mathcal{E}(0; \mathbf{S}, r) \oplus \mathcal{O}_M$ , that  $\mathcal{LW}(x; \mathbf{S}, 0) = \mathcal{LF}(x; \mathbf{S}, 0)$ , and that  $\mathcal{LW}(x; \mathbf{S}, 0) \subset \mathcal{W}$ .  $\square$

## B. Steering Using Local Free Space

We are now ready to state the chief focus of this work. Namely, a simple framework for incorporating local information provided by a quadratic metric and the local geometric structure of the configuration space into a generic steering function. For a given steering procedure  $\text{Steer}(x_1, x_2)$ , we define the new steering procedure

$$\text{GLFSteer}(x_1, x_2) = \text{Steer}(x_1, \Pi_{\mathcal{LF}(x_1; \mathbf{S}, r)}(x_2)), \quad (10)$$

where  $\mathbf{S}$  defines a locally informative metric. This new procedure uses the given  $\text{Steer}$  procedure to drive the system toward the projection of  $x_2$  onto the local free space computed about  $x_1$  instead of  $x_2$  directly. As is demonstrated in Figure 2, the halfspace constraints defining  $\mathcal{LF}(x; \mathbf{S}, r)$  are rotated to align with the level sets of  $\|\cdot\|_{\mathbf{S}}$ . As a result, projecting  $x_2$  onto  $\mathcal{LF}(x; \mathbf{S}, r)$  naturally encourages steering of the system in directions of low cost as measured by  $\|\cdot\|_{\mathbf{S}}$ . Moreover, because the halfspace constraints are designed to limit motion in the direction of the shortest path of collision of each obstacle as measured by  $\|\cdot\|_{\mathbf{S}}$ , the projection favors expansion into areas that do not as readily produce collisions.

## IV. EXAMPLE LOCAL METRICS

In this section, we present three choices of  $\mathbf{S}$  to use in construction of the generalized local free space: the LQR cost-to-go function, a metric derived from the linearized dynamics of a nonlinear system, and the Mahalanobis measuring uncertainty in the context of linear-Gaussian dynamical system. We briefly remark that the Euclidean local free space and Euclidean local workspace are equivalent to their generalized counterparts when  $\mathbf{S} = \mathbf{I}$ .

### A. LQR Cost-to-Go Metric

In [8], [9], the LQR cost-to-go function was used to define a local metric,  $\text{Dist}$ . This metric is shown to be effective because it accurately encapsulates the geodesic distance for a linear system and, through linearization, can locally estimate the geodesic distance of a nonlinear dynamical system when the infinite-horizon LQR control policy is used for steering.

Specifically, consider a robot modeled by a continuous-time linear system of the form

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t), \quad (11)$$

the control policy  $u(x)$  that minimizes the cost functional

$$J(x(t)) = \int_{t_0}^{\infty} x(t)^T \mathbf{Q}x(t) + u(t)^T \mathbf{R}u(t), \quad (12)$$

over the time horizon  $t \in [t_0, \infty)$  with  $\mathbf{Q}, \mathbf{R} \in \mathbb{S}_{++}^n$  can be computed exactly. This policy has the form  $\mathbf{u}(x) = -\mathbf{K}x$  where  $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}\mathbf{S}$  and  $\mathbf{S}$  is the solution to the continuous algebraic Riccati equation (CARE) [27]

$$\mathbf{A}^T\mathbf{S} + \mathbf{S}\mathbf{A} - \mathbf{S}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T + \mathbf{Q} = \mathbf{0} \quad (13)$$

When (11) is controllable [28], the CARE may be solved efficiently through numerical methods [27]. The function  $V(x) = x^T\mathbf{S}x$  is the *LQR cost-to-go function* and describes the value of  $J(x(t))$  computed for the system in (11) with initial condition  $x_0$ . It also serves as a Lyapunov function for the closed-loop system.

The infinite-horizon LQR can be used to define a steering procedure  $\text{Steer}(x_1, x_2)$  by integrating (11), with initial condition  $x_1$  and control policy  $\mathbf{u}(t) = -\mathbf{K}(x(t) - x_d)$  over a finite time interval. When this steering policy is used,  $V(x_2 - x_1) = \|x_2 - x_1\|_{\mathbf{S}^{-1}}$  defines a natural choice of local metric due to its quadratic form and direct relationship to the local control policy used to steer the system.

### B. Gram Metric Constructed via System Linearization

A limitation of the LQR cost-to-go as a local metric is that common robot models, such as the unicycle dynamical system described in Section V, are not controllable when linearized. As a result, the CARE in (13) may not have a solution [27]. Therefore, we propose an alternative local metric based on the Gram matrix of the linearization of a dynamical system.

Let the dynamics for a point robot be modeled by the general nonlinear system  $\dot{x}(t) = f(x(t), u(t))$  and assume that the feedback policy  $u(x)$  provided. Then, the autonomous system  $\dot{x}(t) = f(x(t)) = f(x(t), u(t))$  may be linearized at a state  $x_0$  to produce the linear system

$$\begin{aligned} \dot{x}(t) &\approx \underbrace{\frac{\partial f(x_0)}{\partial x}}_{\mathbf{A}} x(t) + \underbrace{f(x_0) - \frac{\partial f(x_0)}{\partial x} x_0}_{\mathbf{b}} \quad (14) \\ &= \mathbf{A}x(t) + \mathbf{b} \quad (15) \end{aligned}$$

where  $\frac{\partial f(x_0)}{\partial x}$  is the Jacobian of  $f(x)$  evaluated at  $x_0$ . The matrix  $\mathbf{A}$  is often full rank and, when it is,  $\mathbf{A}x(t) + \mathbf{b}$  is of the form of  $h(x)$  in Lemma 1. This structure suggests the choice of  $\mathbf{S} = (\mathbf{A}^T\mathbf{A})^{-1}$ . The affine term is dropped because the purpose of  $\|\cdot\|_{\mathbf{S}}$  is to quantify how the dynamical system is evolving locally whereas the affine term represents a global property (with respect to this particular linearization) and does not vary with  $x$ .

However, the choice of  $\mathbf{S} = (\mathbf{A}^T\mathbf{A})^{-1}$  induces a metric which behaves like similarity measure while our convention requires  $\mathbf{S}$  to induce a dissimilarity measure. Specifically, the metric will increase rapidly in directions that the system evolves in locally. However, we would like  $\Pi_{\mathcal{A}}(x; \mathbf{S})$  to return the point in  $\mathcal{A}$  closest to  $x$  in terms of the geodesic distance. Therefore, we instead choose the metric defined by the Gram matrix  $\mathbf{S} = \mathbf{A}^T\mathbf{A}$ , which will interpret points in the direction the system is evolving as closer than those in the orthogonal directions.

### C. Mahalanobis Metric over Belief Space

Another common setting in motion planning literature is the *belief space*. In belief space planning, the algorithm reasons over distributions of states defined by a stochastic process instead of individual trajectories. For arbitrary dynamical systems with a continuous state space, these distributions are not tractably representable. A notable exception is the linear-Gaussian system [21], [23], [29], [30]

$$\begin{aligned} x_{t+1} &= \mathbf{A}x_t + \mathbf{B}u_t + w_t, & w_t &\sim N(0, \mathbf{W}_t), \\ y_t &= \mathbf{C}x_t + v_t, & v_t &\sim N(0, \mathbf{V}_t), \end{aligned} \quad (16)$$

with  $x_0 \sim N(\bar{x}_0, \Sigma_0)$ , where  $N(\bar{x}, \Sigma)$  is the multivariate Gaussian distribution with mean  $\bar{x}$  and covariance  $\Sigma$ . Due to the linear dynamics and measurement model, at any time step  $t$ , the distribution of the system is  $N(\bar{x}_t, \Sigma_t)$  where

$$\bar{x}_{t+1} = \mathbf{A}\bar{x}_t + \mathbf{B}u_t, \quad \Sigma_{t+1|t} = \mathbf{A}\Sigma_t\mathbf{A}^T + \mathbf{W}_t \quad (17)$$

$\Sigma_{t+1} = \Sigma_{t+1|t} - \Sigma_{t+1|t}\mathbf{C}^T(\mathbf{C}\Sigma_{t+1|t}\mathbf{C}^T + \mathbf{V}_t)^{-1}\mathbf{C}\Sigma_{t+1|t}$  is derived from the Kalman dynamics [31]. Here,  $\Sigma_{t+1|t}$  represents the dynamical uncertainty prior to the arrival of the measurement  $y_t$  while  $\Sigma_t$  is the uncertainty of the system after processing  $y_t$ . The *belief state* of the system at time step  $t$  is defined to be  $b_t = (\bar{x}_t, \Sigma_t)$ .

In this setting, the planner's goal is to produce a finite trajectory  $(\bar{x}_1, \Sigma_1), \dots, (\bar{x}_T, \Sigma_T)$  such that  $x_{t|t-1} \sim N(\bar{x}_t, \Sigma_{t|t-1})$  is collision-free with some probability  $\delta$ , i.e.,

$$\Pr\{x_{t|t-1} \notin \mathcal{O}_i\} \geq \delta, \quad \forall i = 1, \dots, M. \quad (18)$$

For example, in [21], a belief space variant of the RRT algorithm is presented. An alternative approach is used in [30], where a trajectory is computed for a noiseless system using an RRT, and then modified afterward using an iterative optimization procedure to incorporate belief space constraints such as probability of collision.

Due to the computational difficulty of integrating the multivariate Gaussian distribution over obstacle sets, conservative approximations are typically used to guarantee (18). A common approximation is to ensure that the *confidence region* [32] lies entirely within the configuration space [21], [23]. A confidence region of a distribution with mean  $\bar{x}_t$  and covariance  $\Sigma_t$  is of the form

$$\mathcal{E}_{\delta}(b_t) = \left\{ x \mid (x - \bar{x}_t)^T(\gamma\Sigma_t)^{-1}(x - \bar{x}_t) \leq 1 \right\}, \quad (19)$$

where  $\gamma = \chi_n^{-1}(\delta)$  and  $\chi_n(\cdot)$  is the cumulative distribution function of the chi-squared distribution with  $n$  degrees of freedom and is constructed such that  $\Pr\{x_t \in \mathcal{E}_{\delta}(b_t)\} = \delta$ .

A natural choice of  $\mathbf{S}$  in this context is  $\gamma\Sigma_{t|t-1}$ . When  $\mathbf{S}$  represents a covariance,  $\|\cdot\|_{\mathbf{S}}$  is typically referred to as the Mahalanobis distance. This choice of  $\mathbf{S}$  confers two benefits. First,  $\|\cdot\|_{\Sigma_{t|t-1}}$  is weighted to increase in directions of high variance—and therefore high uncertainty—thereby favoring motion in directions in which motion is more certain. Second, the local free space can be used to enforce the chance of collision constraint (18) via the following proposition

**Proposition 4.** *For any belief state  $b_t = (\bar{x}_t, \Sigma_t)$ , if  $\bar{x}_{t+1} \in \mathcal{L}\mathcal{F}(\bar{x}_t; \Sigma_{t+1|t}, \sqrt{\gamma})$  then  $\Pr\{x_{t+1} \notin \mathcal{O}_i\} \geq \delta$  for all  $i$ .*

*Proof.* The results follows from the definition of the confidence interval, and Proposition 3 and the facts that  $\mathcal{LF}(\bar{x}_t; \Sigma_{t|t+1}, \sqrt{\gamma})$  is a generalized local free space, and that  $\mathcal{E}_\delta((x_t, \Sigma_{t+1|t})) = \mathcal{E}(\bar{x}_{t+1}, \Sigma_{t|t+1})$ .  $\square$

If the steering procedure is chosen such that Proposition 4 holds, then the chance-of-collision constraint in (18) is actively enforced during planning.

## V. NUMERICAL SIMULATIONS

We now demonstrate the efficacy of the framework proposed in Section III through a series of numerical simulations of the RRT algorithm described in Section II using each of the metrics described in Section IV.

### A. Double Integrator System

We first consider the cont.-time double integrator system

$$\underbrace{\begin{bmatrix} \dot{p}(t) \\ \dot{\dot{p}}(t) \end{bmatrix}}_{\dot{x}(t)} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \underbrace{\begin{bmatrix} p(t) \\ \dot{p}(t) \end{bmatrix}}_{x(t)} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} u(t), \quad (20)$$

which is a common benchmark in motion-planning literature [6], [10]. Here, configuration space constraints are only checked with regard to the position variables  $p(t)$  and not velocity variables  $\dot{p}(t)$ , i.e.  $\text{CheckCollision}(\mathcal{J}, \mathcal{C})$  returns `true` if and only if  $p(t) \in \mathcal{C}$  for all  $t$  in the finite time horizon. To highlight the impact of the different steering functions,  $\text{Dist}(x_1, x_2) = \|x_1 - x_2\|_2$  is used for all trials. As suggested in Section IV-A, the function  $\text{Steer}(x_1, x_2)$  integrates (20) over a finite-time horizon using the infinite-horizon LQR control policy to steer from  $x_1$  to  $x_2$ . In addition, the steering policy will also terminate before the end of the time horizon if the trajectory leaves the ball  $\mathcal{B}(x_1, r_{max})$  (for a choice of  $r_{max}$ ) and returns the trajectory segment computed up to that point. This condition regularizes the length of trajectory computed at each step and improves the performance of the planner because long trajectories are likely to intersect obstacles in cluttered environments.

Simulation parameters and results are presented in Figure 3. The matrix  $\mathbf{Q}$  was chosen so that level sets of the LQR cost-to-go function were ellipsoidal and not spherical for this demonstration. The original steering procedure by itself is only able to explore roughly half of the space before the planner terminates with  $|\mathcal{V}| = 465$  vertices. The steering based on the Euclidean local free space offers some improvement and allows the planner to explore roughly a third of the space and terminates with  $|\mathcal{V}| = 668$  vertices. Finally, The generalized local free space steering, using both the LQR cost-to-go metric and linearization Gram matrix, allow the system to fully explore the space. The former spends more iterations exploring the lower right quadrant and terminates with  $|\mathcal{V}| = 688$  vertices while the latter focuses on the upper left quadrant.

### B. Kinematic Unicycle System

A second commonly use example in the literature is the kinematic unicycle system [11], [21], [33] with nonlinear dynamics

$$\dot{x}(t) = \underbrace{\begin{bmatrix} v(t) \cos \theta \\ v(t) \sin \theta \\ \omega(t) \end{bmatrix}}_{f(x(t), u(t))}, \quad x(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix}, \quad u(t) = \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}. \quad (21)$$

The robot is assumed to be a point robot. This system is underactuated and therefore not controllable when linearized. As a result, the infinite-horizon LQR steering policy and cost-to-go metric cannot be used for this system. The same definitions for `Dist` and `CollisionFree` are used in this setting as in the previous example. Steering is done by sampling a point  $x_d = [x_d \ y_d]^T$  in the workspace (as opposed to a full configuration, which includes an orientation) and applying the following feedback control policy [19]

$$u(x, d) = \begin{bmatrix} (x - x_d) \cos(\theta) + (y - y_d) \sin(\theta) \\ \text{atan} \left( \frac{-(x - x_d) \sin(\theta) + (y - y_d) \cos(\theta)}{(x - x_d) \cos(\theta) + (y - y_d) \sin(\theta)} \right) \end{bmatrix} \quad (22)$$

for a finite time horizon. We here use the same regularization method to limit the length of a steering trajectory as in the previous example.

The parameters and the results for experiments using this model are shown in Figure 4. The original steering procedure by itself is unable to explore the right half of the environment before terminating with  $|\mathcal{V}| = 285$  vertices. The use of the Euclidean local free space managed to increase the number of vertices to  $|\mathcal{V}| = 681$ , but is still not able to traverse the narrow passage in the middle of the workspace. Finally, while the use of the metric linearization Gram matrix only increased the number of vertices in the final graph to  $|\mathcal{V}| = 718$ , the planner is able to traverse the central passage and fully explore the space.

### C. Linear-Gaussian System

Finally, we consider the linear-Gaussian system

$$\begin{aligned} x_{t+1} &= x_t + u_t + w_t, & w_t &\sim N(0, \mathbf{W}_t), \\ y_t &= x_t + v_t, & v_t &\sim N(0, \mathbf{V}_t), \end{aligned} \quad (23)$$

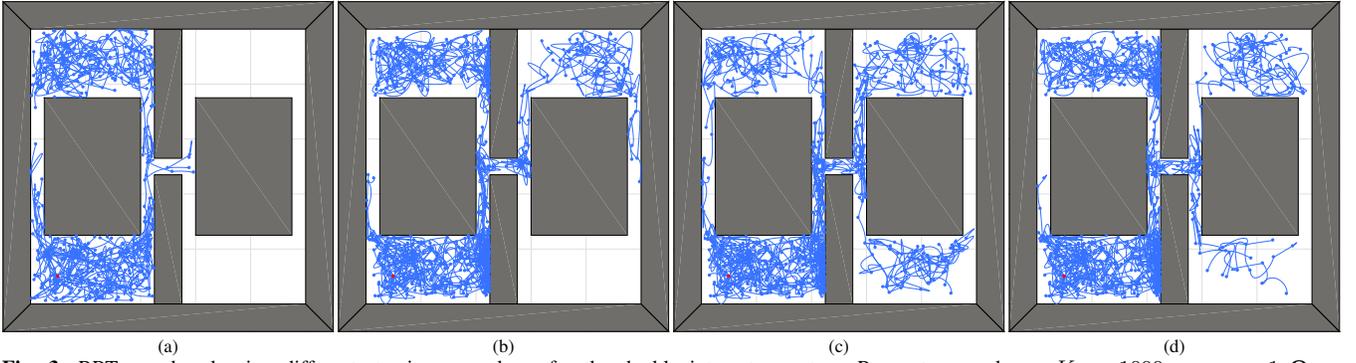
in a light-dark domain [8], [30], [34]. In this scenario, the covariance of the process noise  $\mathbf{W}_t$  is set to a fixed value  $\mathbf{W}$  and the measurement covariance  $\mathbf{V}_t$  is a function of  $x_t$ . Specifically, we consider a measurement covariance function  $\mathbf{V}(x) = \sigma(x)\mathbf{I}$ , where

$$\sigma(x) = 0.005 \min \left( \left\| x - \begin{bmatrix} 1 \\ 9 \end{bmatrix} \right\|_2^2, \left\| x - \begin{bmatrix} 9 \\ 1 \end{bmatrix} \right\|_2^2 \right). \quad (24)$$

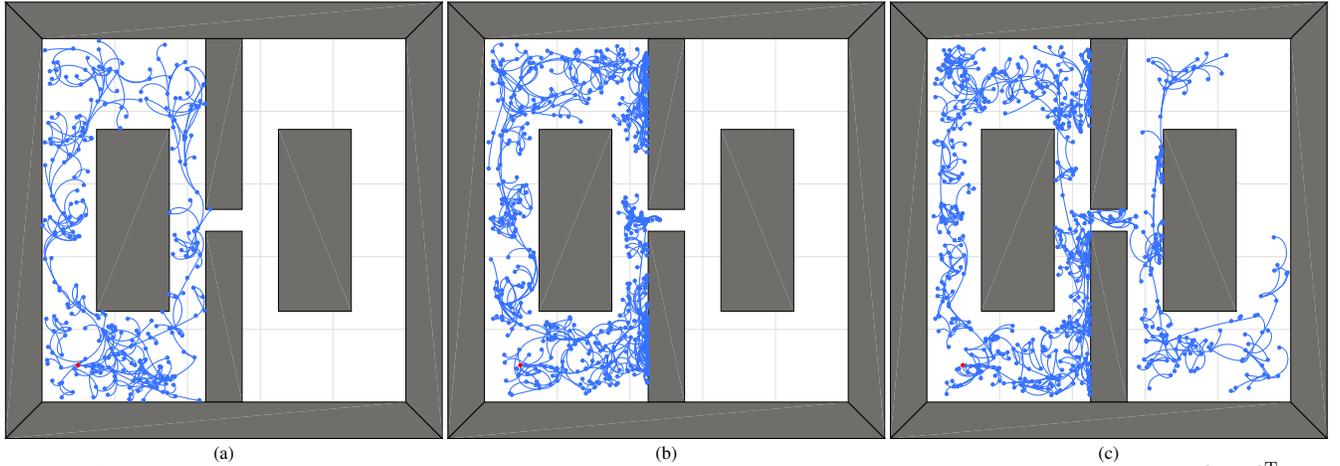
Again,  $\text{Dist}(x_1, x_2) = \|x_1 - x_2\|_2$ . The function  $\text{CollisionFree}(\mathcal{J}, \mathcal{F})$  is defined to be the following for the finite trajectory of states  $\mathcal{J} = \{\bar{x}_1, \dots, \bar{x}_p\}$

$$\bigwedge_{i=2}^p (\text{Line}(\bar{x}_{i-1}, \bar{x}_i) \oplus \mathcal{E}_\delta((0, \Sigma_{i|i-1})) \subset \mathcal{W}) \quad (25)$$

where  $\text{Line}(x_1, x_2) = \{\alpha x_1 + (1 - \alpha)x_2 \mid \alpha \in [0, 1]\}$  and  $\Sigma_{i|i-1}$  is defined in (17). This procedure verifies that the chance-of-collision constraint holds at any intermediate point



**Fig. 3:** RRTs produced using different steering procedures for the double integrator system. Parameters used are:  $K = 1000$ ,  $r_{max} = 1$ ,  $\mathbf{Q} = \text{diag}(2, 1, 1, 1)$ ,  $\mathbf{R} = \mathbf{I}$ ,  $\mathbf{x}_0 = [1 \ 1 \ 0 \ 0]^T$ . States were sampled from the domain  $[0, 10] \times [0, 10] \times [-1, 1] \times [-1, 1]$ . From left to right, the steering procedures used were: (a) the unaltered Steer procedure, (b) GLFSteer using the Euclidean metric, (c) GLFSteer using the LQR cost-to-go metric, and (d) GLFSteer using the linearized dynamics Gram matrix.<sup>3</sup>



**Fig. 4:** RRTs (blue) produced using different steering procedures for the kinematic unicycle system. Parameters used were:  $\mathbf{x}_0 = [1 \ 1 \ 0]^T$ ,  $K = 1500$ ,  $r_{max} = 1$ . From left to right, the steering procedures used were: (a) the unaltered Steer procedure, (b) GLFSteer using the Euclidean metric, and (c) GLFSteer using the linearized dynamics Gram matrix.<sup>3</sup>

between  $\mathbf{x}_{i-1}$  and  $\mathbf{x}_i$  with respect to the process covariance of the system so that it cannot “skip” over obstacles.

The  $\text{Steer}(\mathbf{x}_t, \mathbf{x})$  is computed by solving the linearly constrained convex quadratic optimization problem

$$\begin{aligned} & \underset{\mathbf{u}}{\text{minimize}} && (\mathbf{x}_{t+1} - \mathbf{x})^T (\mathbf{x}_{t+1} - \mathbf{x}) + \mathbf{u}^T \mathbf{u} && (26) \\ & \text{subject to} && \mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{u}, \mathbf{x}_{t+1} \in \mathcal{LF}(\mathbf{x}_t, \Sigma_{t+1|t}) \end{aligned}$$

If this program is feasible, it guarantees the chance-of-collision constraint defined by  $\text{CollisionFree}$  is upheld. When it is infeasible, then it is treated as a collision. In a similar manner to the previous examples, if  $\|\mathbf{u}\|_2 > r_{max}$ , its magnitude is reduced such that  $\|\mathbf{u}\|_2 = r_{max}$ .

The simulation parameters and results for this example are summarized in Figure 5. When steering without a local free space, the planner is unable to leave the upper-left quadrant and terminates with only  $|\mathcal{V}| = 65$  vertices. Meanwhile, when the generalized local free space computed using the process covariance  $\Sigma_{t|t-1}$  is used, the planner explores all quadrants and terminates with  $|\mathcal{V}| = 301$  vertices.

## VI. CONCLUSION

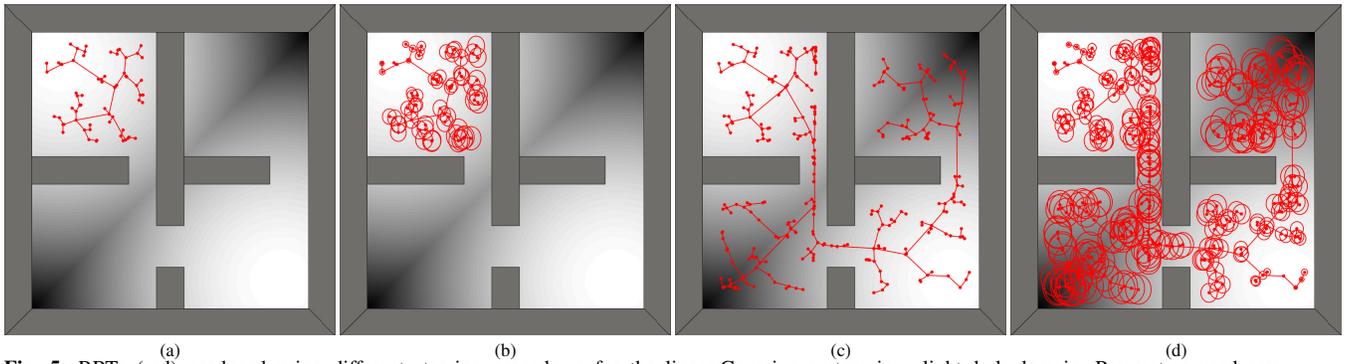
In this paper, we introduced a framework for incorporating local dynamical and geometric information into steering procedures for sampling-based planners. Our methodology

generalizes the concept of the local free space used in [18]–[20] to allow the specification of different inner products. These inner products induce norms that locally represent informative metrics about the dynamical nature of the system. Furthermore, we discuss three examples of such metric—the LQR cost-to-go function, a novel metric derived from the Gram matrix of a linearized autonomous system, and the Mahalanobis distance for a linear-Gaussian system. Finally, the efficacy of the steering framework is demonstrated in numerical examples in different dynamical settings.

We conclude with some potential extensions of this work. A key assumption of this paper is that obstacle sets are explicitly representable in terms of robot states so that the maximum-margin separating hyperplane between a configuration and obstacle may be computed. However, these representations are not possible for a number of robots, such as manipulators. One future direction is to explore how the ideas presented here can be adapted to such a setting.

A second extension in the same vein is to determine how a local metric can be approximated when it is not a quadratic form. Many cost-to-go functions for nonlinear systems are not available as simple quadratic functions [11],

<sup>3</sup>Please see the accompanying video for an animated demonstration.



**Fig. 5:** RRTs (red) produced using different steering procedures for the linear-Gaussian system in a light-dark domain. Parameters used are:  $\mathbf{x}_0 = [1 \ 1 \ 0 \ 0]^T$ ,  $K = 1000$ ,  $r_{max} = 1$ ,  $\mathbf{Q} = \mathbf{I}$ ,  $\mathbf{R} = \mathbf{I}$ ,  $\gamma = 0.95$ . Figure (a) was steered using the unaltered *Steer* procedure while Figure (c) was steered using *GLFSteer* with the Mahalanobis distance metric. Figures (b) and (d) show the confidence region of each vertex as well. Regions of the workspace with low measurement covariances are white while dark regions represent states with high measurement covariances.<sup>3</sup>

[33]. Determining how more general metrics may be used to construct local free spaces will create steering procedures that are even more capable of exploiting local information.

## REFERENCES

- [1] L. E. Kavraki, M. N. Kolountzakis, and J. C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 166–171, 1998.
- [2] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.
- [3] P. Cheng, Z. Shen, and S. La Valle, "RRT-based trajectory design for autonomous automobiles and spacecraft," *Archives of Control Sciences*, vol. 11, no. 3/4, pp. 167–194, 2001.
- [4] D. Berenson, S. S. Srinivasa, D. Ferguson, A. Collet, and J. J. Kuffner, "Manipulation planning with workspace goal regions," in *IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 618–624.
- [5] M. Kothari, I. Postlethwaite, and D.-W. Gu, "Multi-UAV path planning in obstacle rich environments using rapidly-exploring random trees," in *IEEE Conf. Decision and Control*, 2009, pp. 3069–3074.
- [6] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *IEEE Conference on Decision and Control*, 2010, pp. 7681–7687.
- [7] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo, "Choosing good distance metrics and local planners for probabilistic roadmap methods," in *IEEE International Conference on Robotics and Automation*, vol. 1, 1998, pp. 630–637.
- [8] 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*, 2012, pp. 2537–2542.
- [9] E. Glassman and R. Tedrake, "A quadratic regulator-based heuristic for rapidly exploring state space," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 5021–5028.
- [10] D. J. Webb and J. van den Berg, "Kinodynamic RRT\*: asymptotically optimal motion planning for robots with linear dynamics," in *IEEE Int. Conf. Robotics and Automation*, 2013, pp. 5054–5061.
- [11] J. J. Park and B. Kuipers, "Feedback motion planning via non-holonomic RRT\* for mobile robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 4035–4040.
- [12] L. Palmieri and K. O. Arras, "Distance metric learning for RRT-based motion planning with constant-time inference," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 637–643.
- [13] M. Bharatheesha, W. Caarls, W. J. Wolfslag, and M. Wisse, "Distance metric approximation for state-space RRTs using supervised learning," in *IEEE/RSJ Int. Conf. on Intel. Robots and Systems*, 2014, pp. 252–257.
- [14] D. Hsu, L. E. Kavraki, J.-C. Latombe, R. Motwani, and S. Sorkin, "On finding narrow passages with probabilistic roadmap planners," in *The Int. Workshop on the Algorithmic Foundations of Robotics*, 1998.
- [15] S. Rodríguez, S. Thomas, R. Pearce, and N. M. Amato, "Resampl: A region-sensitive adaptive motion planner," in *Algorithmic Foundation of Robotics VII*. Springer, 2008, pp. 285–300.
- [16] S. W. H. Wong and M. Jenkin, "Exploiting collision information in probabilistic roadmap planning," in *IEEE International Conference on Mechatronics*, 2009, pp. 1–5.
- [17] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "The bridge test for sampling narrow passages with probabilistic roadmap planners," in *IEEE Int. Conf. Robot. Autom.*, vol. 3, 2003, pp. 4420–4426.
- [18] O. Arslan, V. Pacelli, and D. E. Koditschek, "Sensory steering for sampling-based motion planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, In Press.
- [19] O. Arslan and D. E. Koditschek, "Exact robot navigation using power diagrams," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 1–8.
- [20] —, "Sensor-based reactive navigation in unknown convex sphere worlds," in *The 12th International Workshop on the Algorithmic Foundations of Robotics*, 2016.
- [21] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 723–730.
- [22] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 849–882, 2015.
- [23] L. Blackmore and M. Ono, "Convex chance constrained predictive control without sampling," in *AIAA Guidance, Navigation and Control Conference*, 2009, pp. 7–21.
- [24] T. Kunz and M. Stilman, "Kinodynamic RRTs with fixed time step and best-input extension are not probabilistically complete," in *Algorithmic foundations of robotics XI*. Springer, 2015, pp. 233–244.
- [25] M. C. Lin and J. F. Canny, "A fast algorithm for incremental distance calculation," in *IEEE Int. Conf. Robot. Autom.*, 1991, pp. 1008–1014.
- [26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [27] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*. Clarendon press, 1995.
- [28] W. J. Rugh, *Linear System Theory*. Prentice Hall, 1996, vol. 2.
- [29] M. P. Vitus and C. J. Tomlin, "Closed-loop belief space planning for linear, Gaussian systems," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2152–2159.
- [30] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The Inter. J. Robot. Res.*, vol. 31, no. 11, pp. 1263–1278, 2012.
- [31] W. H. Fleming and R. W. Rishel, *Deterministic and Stochastic Optimal Control*. Springer Science & Business Media, 2012, vol. 1.
- [32] J. Berger, "A robust generalized bayes estimator and confidence region for a multivariate normal mean," *Ann. Stat.*, pp. 716–761, 1980.
- [33] L. Palmieri and K. O. Arras, "A novel RRT extend function for efficient and smooth mobile robot motion planning," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2014, pp. 205–211.
- [34] R. Platt Jr, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Robotics: Science and Systems Conference*, 2010.