

Sufficient Conditions for the Existence of Resolution Complete Planning Algorithms

Dmitry S. Yershov and Steven M. LaValle

Abstract This paper addresses theoretical foundations of motion planning with differential constraints in the presence of obstacles. We establish general conditions for the existence of resolution complete planning algorithms by introducing a functional analysis framework and reducing algorithm existence to a simple topological property. First, we establish metric spaces over the control function space and the trajectory space. Second, using these metrics and assuming that the control system is Lipschitz continuous, we show that the mapping between open-loop controls and corresponding trajectories is continuous. Next, we prove that the set of all paths connecting the initial state to the goal set is open. Therefore, the set of open-loop controls, corresponding to solution trajectories, must be open. This leads to a simple algorithm that searches for a solution by sampling a control space directly, without building a reachability graph. A dense sample set is given by a discrete-time model. Convergence of the algorithm is proven in the metric of a trajectory space. The results provide some insights into the design of more effective planning algorithms and motion primitives.

1 Introduction

We consider the general problem of motion planning under both differential constraints and obstacles. A control system, geometric robot model, and a model of obstacles in the workspace are given. The task is to compute a control signal that brings a robot along a trajectory from an initial state into a goal region in a state space that may represent configurations and possibly their time derivatives. This problem is a unification of several fundamental, classical problems in robotics: 1) *Nonholonomic planning*: In this case, the differential constraints may arise from wheeled mobile robots and planning occurs in the configuration space [1]; however, dynamics and

Dmitry S. Yershov
University of Illinois, Urbana, IL 61801, e-mail: yershov2@uiuc.edu

Steven M. LaValle
University of Illinois, Urbana, IL 61801 e-mail: lavalle@uiuc.edu

constraints due to angular momentum may also be included. Such problems usually arise from *underactuated* (less controls than the number of degrees of freedom) systems. 2) *Kinodynamic planning*: Here, there are both velocity and acceleration bounds, and the system is fully actuated [2]. 3) *Trajectory planning*: This problem has been pursued for several decades [3, 4, 5] and typically involves computing an open loop control for a manipulator while satisfying the kinematics and dynamics expressed as a control system. See Chapter 14 of [6] for a detailed presentation of this unified class of problems.

In spite of all of this effort, there is still no general characterization of the particular conditions under which an algorithmic solution exists. Since basic motion planning (without differential constraints) is already PSPACE hard [7], and particular instances of motion planning with differential constraints are even harder, there is not much hope for efficient, complete solutions. In this context, *complete* means that the planning algorithm must return a solution whenever one exists; otherwise, report failure. Therefore, virtually all approaches to the problem are *sampling-based*, which employ discretization and heuristics to incrementally explore the state space by concatenating pieces of control signals to obtain a search tree of collision-free trajectories. In this case, the most we can hope for is *resolution completeness* [8], which means that the algorithm correctly finds a solution whenever one exists; however, it may run forever if one does not. This is analogous to classical Turing decidability vs. Turing recognizability, which are comparable to completeness and resolution completeness, respectively. We believe that having general conditions for the existence of resolution complete algorithms may be useful in the formulation of solvable robotics problems, in the design of better sampling-based planning algorithms, in the design of motion primitives [9, 10, 11], and possibly for the verification problem [12, 13], which is a negated form of planning that establishes path *nonexistence*.

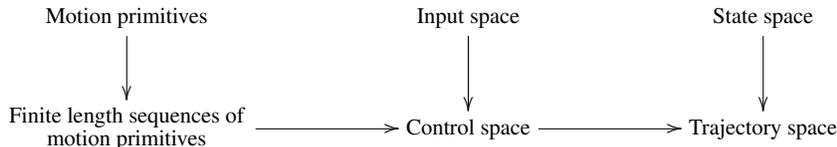


Fig. 1 Sets, spaces, and relations

In this paper, we determine simple, general requirements for the existence of a resolution complete planning algorithm based mainly on Lipschitz conditions on the control system mapping. This allows substantial generality and is inspired by analysis of the convergence of numerical dynamic programming algorithms [14, 15]. Our basic approach in this paper is to analyze the relationships between the six spaces shown informally in Fig. 1. The *input space* is the set of possible control system inputs and the *state space* is the configuration space or more generally the

phase space of the system. Considering inputs and states parametrized over time, we design suitable metric spaces for both the space of control signals (called the *control space*) and the space of trajectories (the *trajectory space*). By establishing the continuity of the mapping between these spaces induced by the control system, we show that resolution completeness boils down to a simple topological condition. Furthermore, we constructively prove existence by providing a resolution complete algorithm that systematically enumerates candidate solution trajectories by concatenating sequences of *motion primitives*, which form a discrete set, suitable for computation. This closely resembles the execution trace of most existing planning algorithms, which incrementally sample and search the state space (see [16, 6] for surveys).

Resolution completeness in this general setting is provided by ensuring that the set of all *computed* control signals is dense in the space of *all* control signals. One surprising observation, however, is that this may be achieved even where it is impossible to incrementally reduce the radius of the largest empty ball (*dispersion* [17]) in the space of control signals. This peculiar behavior is explained in Section 2, where basic sampling concepts are defined. Section 3 formally defines the general problem. Section 4 develops a continuous mapping from control space to trajectory space by carefully designing appropriate metric spaces. The main algorithmic constructions and theorems are presented in Section 5, which give sufficient conditions for the existence of a resolution complete planning algorithm. Conclusions appear in Section 6.

2 Precompactness and Sampling Convergence

Before providing the main technical results of the paper, a counterintuitive property regarding sampling and convergence needs to be addressed. Consider sampling-based planning algorithms for the basic motion planning problem (no differential constraints). Achieving resolution completeness (or alternatively, probabilistic completeness) amounts to assuring that the computed samples are dense in the limit as the number of iterations tends to infinity. Intuitively, the sampling resolution gradually increases over time. This notion can be nicely captured by defining *dispersion* of a sample set P in a subset of any metric space X [17]:

$$\delta(P) = \sup_{x \in X} \{ \inf_{p \in P} \{ \rho(x, p) \} \}, \quad (1)$$

in which ρ is the metric.

Typically, the sample set increases gradually during the execution of a planning algorithm, and the dispersion converges to zero in the limit. This is formalized by considering P as an infinite *sample sequence* P , which is a set together with a specified linear ordering. We say that P *converges* to X if and only if

$$\lim_{N \rightarrow \infty} \delta(P|_N) = 0, \quad (2)$$

in which $P|_N$ denotes the first N elements of P . Note that the convergence rate may depend on the ordering.

In the coming sections, we consider notions of denseness, dispersion, and convergence over the function spaces of control signals and trajectories. For these spaces, it might be surprising that there are dense sample sequences for which the dispersion does not converge to zero. In other words, the samples may eventually get arbitrarily close to every point in the space, even though they are not converging to that space. For a simple example of this behavior, let $X = \mathbb{R}$. For any finite set of samples in \mathbb{R} , the dispersion is always infinite. Nevertheless, any ordering placed on \mathbb{Q} , the set of rationals, yields a sequence that is dense¹ in \mathbb{R} . For a bounded space, such as S^1 and the configuration spaces arising in robotics, this behavior does not occur: A dense sequence must drive the dispersion to zero.

Here is a critical question for sampling-based planning with differential constraints: What property must X have to enable convergence? Define X to be *precompact* if and only if for any $\varepsilon > 0$ there exists a finite cover of X with open balls of radius ε . The following lemma answers the question.

Lemma 1. *A sequence P that is dense in X converges to X if and only if X is precompact.*

Proof. We prove necessary and sufficient conditions separately.

Necessary condition: Assume P is convergent, and let ε be greater than zero. Find N_ε such that $\delta(P|_{N_\varepsilon}) < \varepsilon$. Consider a set of open balls of radius ε centered at points of $P|_{N_\varepsilon}$. From the definition of dispersion, it follows that this set of balls is a finite cover of X . Since ε is arbitrary, X is precompact by definition.

Sufficient condition: Suppose, on the other hand, that X is precompact. Thus, for any $\varepsilon > 0$, there exists a finite cover of X with open balls of radius ε . Denote these balls as B_i . Since the sequence P is dense, the intersection of P with each B_i is nonempty. Take the smallest N_ε such that $P|_{N_\varepsilon}$ has at least one element in each B_i . By construction, the dispersion of $P|_{N_\varepsilon}$ in X is bounded by 2ε . Since ε is arbitrary and for $N > N_\varepsilon$ we have $\delta(P|_N) \leq \delta(P|_{N_\varepsilon})$, the sequence converges. \square

Note that this is a property of the *space* X , and not a particular sample sequence. Our analysis will demonstrate that under general conditions space of all control signals is not precompact. Therefore, it is impossible to achieve convergence in the control space. However, we will show precompactness of the space of finite-time trajectories. In this case, the denseness implies convergence, which means that if a solution is not found in a finite number of steps, it either does not exist or the goal set must be smaller than the reached dispersion of the sampled trajectories.

3 Problem Definition

Let the *state space*, $X \subset \mathbb{R}^m$, be a smooth manifold of dimension n . Let U be the *input space*, which is a compact subset of \mathbb{R}^k with $k \leq n$. A given mechanical system is expressed in local coordinates as²

¹ Here, dense means that the topological closure of \mathbb{Q} is all of \mathbb{R} .

² We may consider more general case of time-varying systems $\dot{x} = f(x, u, t)$, without changing further analysis in the paper. We choose the time-invariant case for notational convenience.

$$\dot{x} = f(x, u), \quad (3)$$

in which $\dot{x} = dx/dt$. Also, in the equation above, $x \in X$ and $u \in U$.

It is assumed that f is a *Lipschitz continuous* function in both x and u , which implies that there exists positive real-valued constants L_x and L_u such that

$$\|f(x, u) - f(x', u)\| \leq L_x \|x - x'\| \quad \text{and} \quad \|f(x, u) - f(x, u')\| \leq L_u \|u - u'\| \quad (4)$$

for all $x, x' \in X$, and $u, u' \in U$. The norms used here are defined on the ambient vector spaces \mathbb{R}^m and \mathbb{R}^k , respectively. Furthermore, it is assumed that there exists $M > 0$ such that

$$\|f(x, u)\| \leq M \quad (5)$$

for all $x \in X$ and $u \in U$.

Let \mathcal{U} be the set of all measurable functions, \tilde{u} defined on $[0, T]$, for all $T \in [0, \infty)$, with values in U . Similarly, denote \mathcal{X} to be the set of all Lipschitz continuous functions \tilde{x} defined on $[0, T]$, for $T \geq 0$, with values in X . We require that for all functions in \mathcal{X} the Lipschitz constant is bounded by M . In other words, for any element $\tilde{x} \in \mathcal{X}$ and any given t and t' in the domain of \tilde{x} , we have $\|\tilde{x}(t) - \tilde{x}(t')\| \leq M|t - t'|$. Also, define $\tau : \mathcal{U} \cup \mathcal{X} \rightarrow [0, \infty)$ to return the duration of a control or a trajectory, depending on the argument.

Constraints are imposed on X that account for mechanical limits due to kinematics and dynamics, and also to avoid collisions with static obstacles. Let X_{free} denote an open and bounded subset of X that consists of all states satisfying these constraints. Usually, X_{free} is defined only implicitly via representations of the kinematics and obstacles. Therefore, a *collision detection* algorithm is often needed to evaluate whether states lie in X_{free} .

The *planning problem* is as follows. Assume X, U , and f are given. Furthermore, an *initial state*, $x_1 \in X_{\text{free}}$, and open goal set $X_G \subseteq X_{\text{free}}$ are specified. The problem is to compute $\tilde{u} \in \mathcal{U}$ such that for the corresponding \tilde{x} , satisfying (3) with given \tilde{u} , the following is true: 1) $\tilde{x}(0) = x_1$, 2) $\tilde{x}(\tau(\tilde{u})) \in X_G$, and 3) the image of \tilde{x} lies in X_{free} .

To accomplish this task, we assume the existence of an *integration module* which integrates (3) to produce trajectory segments, and a *collision detection module* which determines whether a trajectory segment lies entirely in X_{free} .

4 Preliminary Concepts and Properties

In this section we introduce some preliminary concepts that are necessary to establish basic properties for an algorithm to be resolution complete and convergent. We define metrics on the space of controls and the trajectory space, and show that the relation between controls and trajectories is a well-defined continuous function.

4.1 Designing Metric Spaces on \mathcal{U} and \mathcal{X}

The control space, \mathcal{U} , can be made into a metric space as follows. Let α be the diameter of the smallest ball that contains U . We call α the *diameter* of U . It is easy to verify $\alpha < \infty$ because U is assumed to be compact. For two controls \tilde{u} and \tilde{u}' in \mathcal{U} define the L_1 -type metric

$$\rho_{\mathcal{U}}(\tilde{u}, \tilde{u}') = \int_0^T \|\tilde{u}(t) - \tilde{u}'(t)\| dt + \alpha |\tau(\tilde{u}) - \tau(\tilde{u}')|, \quad (6)$$

in which $T = \min(\tau(\tilde{u}), \tau(\tilde{u}'))$.

Note that this metric is different from a standard L_1 metric due to variable domain length of functions in \mathcal{U} , which is accounted by an additional term in (6). The extra term separates any meaningful controls from the control defined on the zero length time interval, we call it *zero control*.

The choice of the metric is motivated by Figs. 2a and 2b. Consider driving a car around a corner. The trajectory deviates only slightly if steering is applied with small delays. Moreover, the trajectory deviation depends on the delay continuously. The introduced L_1 -type metric (6) captures this behavior.

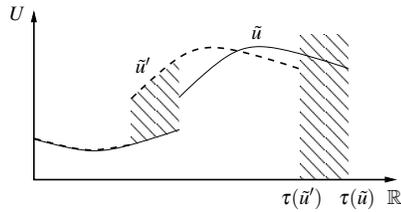


Fig. 2a Control signals. The area of the shaded regions corresponds to $\rho_{\mathcal{U}}(\tilde{u}, \tilde{u}')$

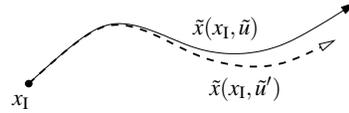


Fig. 2b Corresponding trajectories

To save space, we omit the (tedious) proof of the following lemma:

Lemma 2. *The control space \mathcal{U} is a metric space³ with respect to (6).*

Next we describe a metric on the space of all trajectories, \mathcal{X} . For two trajectories \tilde{x} and \tilde{x}' define the L_∞ -type metric

$$\rho_{\mathcal{X}}(\tilde{x}, \tilde{x}') = \sup_{0 \leq t \leq T} \left\{ \|\tilde{x}(t) - \tilde{x}'(t)\| \right\} + M |\tau(\tilde{x}) - \tau(\tilde{x}')|, \quad (7)$$

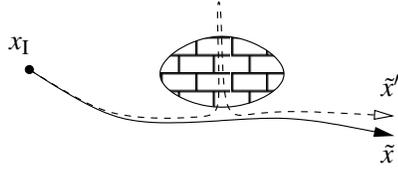
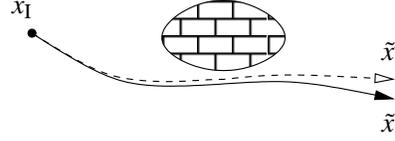
in which $T = \min(\tau(\tilde{x}), \tau(\tilde{x}'))$.

On the trajectory space, an L_1 -type metric cannot be used because it ignores “spikes” in the trajectory deviation. The two trajectories illustrated in Fig. 3a are “close” in terms of the L_1 -type metric. However, they exhibit qualitatively different behavior; the first one does not intersect the obstacle and the second one does. Moreover, it is possible to find two trajectories arbitrary close in L_1 -type metric, but with arbitrary large deviation between them. On the other hand, later we will show that two trajectories which are “close” in the L_∞ -type metric behave similarly; see Fig. 3b.

Lemma 3. *The trajectory space \mathcal{X} is a metric space with respect to (7).*

(The proof is omitted to save space.)

³ More precisely, it is a pseudometric space [18] because $\rho_{\mathcal{U}}(\tilde{u}, \tilde{u}') = 0$ for some $\tilde{u} \neq \tilde{u}'$. However, if two controls are identified in case their distance is zero, then the resulting space is a metric space.

Fig. 3a L_1 -type metricFig. 3b L_∞ -type metric

4.2 Relating Controls to Trajectories

Next, we analyze the mapping between a control $\tilde{u} \in \mathcal{U}$ and the corresponding trajectory originating from some $x_0 \in X$. Denote the mapping as a function $\tilde{x}(x_0, \tilde{u}) : X \times \mathcal{U} \rightarrow \mathcal{X}$. Note that for fixed \tilde{u} and x_0 , the trajectory $\tilde{x}(x_0, \tilde{u})$ is a function of time and it satisfies the integral equation

$$\tilde{x}(x_0, \tilde{u}, t) = x_0 + \int_0^t f(\tilde{x}(x_0, \tilde{u}, s), \tilde{u}(s)) ds, \quad (8)$$

which is equivalent to (3). In the integral equation, $\tilde{x}(x_0, \tilde{u}, t)$ denotes the point of the trajectory $\tilde{x}(x_0, \tilde{u})$ at time t .

Lemma 4 (Well-defined trajectories). *For any initial state $x_0 \in X$ and any control signal $\tilde{u} \in \mathcal{U}$, the trajectory $\tilde{x}(x_0, \tilde{u})$ belongs to \mathcal{X} .*

Proof. For any given $\tilde{u} \in \mathcal{U}$, function $f(x, \tilde{u}(t))$ as a function of x and t satisfies the Caratheodory conditions. Hence, the solution for the differential equation (3), with initial value at any x_0 , exists, is unique, and is absolutely continuous on $[0, \tau(\tilde{u})]$. We now use the integral equation (8) to prove that $\tilde{x}(x_0, \tilde{u}, t)$ is Lipschitz continuous as a function of time. Consider

$$\left\| \tilde{x}(x_0, \tilde{u}, t) - \tilde{x}(x_0, \tilde{u}, t') \right\| \leq \left\| \int_{t'}^t f(\tilde{x}(x_0, \tilde{u}, s), \tilde{u}(s)) ds \right\| \leq M|t - t'|. \quad (9)$$

Note that the Lipschitz constant is bounded by M ; therefore, $\tilde{x}(x_0, \tilde{u})$ is in \mathcal{X} . \square

To show the continuity of $\tilde{x}(x_0, \tilde{u})$, we first prove the following lemma.

Lemma 5 (Bounded trajectory deviation).

Let \tilde{u} and \tilde{u}' be two independent controls, with $\tau(\tilde{u}) = \tau(\tilde{u}') = T$. Assume further that $\tilde{x} = \tilde{x}(x_0, \tilde{u})$ and $\tilde{x}' = \tilde{x}(x'_0, \tilde{u}')$ are the corresponding trajectories. The deviation between trajectories \tilde{x} and \tilde{x}' at any time $t \in [0, T]$ is bounded by

$$\|\tilde{x}(t) - \tilde{x}'(t)\| \leq \left(\|x_0 - x'_0\| + L_u \int_0^t \exp(-L_x s) \|\tilde{u}(s) - \tilde{u}'(s)\| ds \right) \exp(L_x t). \quad (10)$$

Proof. Trajectories \tilde{x} and \tilde{x}' satisfy the integral equation (8), with \tilde{u} and \tilde{u}' , respectively. Using the integral equation and the Lipschitz continuity of f , we derive the integral inequality for the trajectory deviation

$$\begin{aligned} \|\tilde{x}(t) - \tilde{x}'(t)\| &= \left\| x_0 - x'_0 + \int_0^t \left[f(\tilde{x}(s), \tilde{u}(s)) - f(\tilde{x}'(s), \tilde{u}'(s)) \right] ds \right\| \\ &\leq \|x_0 - x'_0\| + \int_0^t L_u \|\tilde{u}(s) - \tilde{u}'(s)\| + L_x \|\tilde{x}(s) - \tilde{x}'(s)\| ds . \end{aligned} \quad (11)$$

From the integral form of Gronwall-Bellman inequality [19] a bound on the trajectory deviation at any time $t \in [0, T]$ follows as

$$\begin{aligned} \|\tilde{x}(t) - \tilde{x}'(t)\| &\leq \|x_0 - x'_0\| \exp(L_x t) + L_u \int_0^t \|\tilde{u}(s) - \tilde{u}'(s)\| ds \\ &\quad + L_x L_u \int_0^t \int_0^r \exp(L_x(t-r)) \|\tilde{u}(s) - \tilde{u}'(s)\| ds dr . \end{aligned} \quad (12)$$

The double integral is reduced to a single integral by applying Fubini's theorem [18] to obtain

$$\begin{aligned} &\int_0^t \int_0^r \exp(L_x(t-r)) \|\tilde{u}(s) - \tilde{u}'(s)\| ds dr \\ &= \int_0^t \int_s^t \exp(L_x(t-r)) \|\tilde{u}(s) - \tilde{u}'(s)\| dr ds \\ &= \frac{1}{L_x} \int_0^t \|\tilde{u}(s) - \tilde{u}'(s)\| (\exp(L_x(t-s)) - 1) ds \end{aligned} \quad (13)$$

The combination of the results above finishes the proof. \square

The next theorem establishes the continuity of the function $\tilde{x}(x_0, \tilde{u})$ with respect to both x_0 and \tilde{u} .

Theorem 1 (Continuity of $\tilde{x}(x_0, \tilde{u})$). *The mapping $\tilde{x}(x_0, \tilde{u})$ is continuous.*

Proof. Take two initial points x_0 and x'_0 in X and two control signals \tilde{u} and \tilde{u}' in \mathcal{U} . Let \tilde{x} and \tilde{x}' be defined as in Lemma 5, and let $T^* = \min(\tau(\tilde{x}), \tau(\tilde{x}'))$. It follows from Lemma 5 that for all $t \in [0, T^*]$

$$\|\tilde{x}(t) - \tilde{x}'(t)\| \leq \left(\|x_0 - x'_0\| + L_u \int_0^t \exp(-L_x s) \|\tilde{u}(s) - \tilde{u}'(s)\| ds \right) \exp(L_x t) . \quad (14)$$

Take the supremum over the interval $[0, T^*]$ on both sides and derive

$$\sup_{0 \leq t \leq T^*} \|\tilde{x}(t) - \tilde{x}'(t)\| \leq \exp(L_x T^*) \left(\|x_0 - x'_0\| + L_u \int_0^{T^*} \|\tilde{u}(s) - \tilde{u}'(s)\| ds \right) . \quad (15)$$

Using the derivation above, we bound the distance between trajectories in terms of the distances between initial values and control signals

$$\begin{aligned}
\rho_{\mathcal{X}}(\tilde{x}, \tilde{x}') &= \sup_{0 \leq t \leq T^*} \|\tilde{x}(x_0, \tilde{u}, t) - \tilde{x}(x'_0, \tilde{u}', t)\| + M|T - T'| \\
&\leq \exp(L_x T^*) \left(\|x_0 - x'_0\| + L_u \int_0^{T^*} \|\tilde{u}(s) - \tilde{u}'(s)\| ds \right) + \frac{M}{\alpha} |T - T'| \\
&\leq \exp(L_x T^*) \|x_0 - x'_0\| + \max \left(L_u \exp(L_x T^*), \frac{M}{\alpha} \right) \rho_{\mathcal{U}}(\tilde{u}, \tilde{u}'). \quad (16)
\end{aligned}$$

The inequality above proves continuity of the map. \square

Since \mathcal{X} and \mathcal{U} are metric spaces and $\tilde{x}(x_0, \tilde{u})$ is continuous, we may employ topological methods to analyze properties of subsets and sequences in control and trajectory spaces, as well as relations between them.

4.3 Topological Properties

We next address properties of the set of *collision-free* paths connecting x_1 with X_G . Consider \mathcal{X}_1 , the subset of \mathcal{X} containing all trajectories that originate from x_1 . Define the induced subset topology on \mathcal{X}_1 . Also, consider the set \mathcal{X}_{sol} of solutions to the path planning problem, which consists of all paths in \mathcal{X} that originate from x_1 , terminate in X_G , with the image contained in X_{free} (collision-free). Note that an element of \mathcal{X}_{sol} may not be necessarily a trajectory governed by the system (3).

Theorem 2. *If X_{free} and X_G are open in X , then \mathcal{X}_{sol} is an open subset of \mathcal{X}_1 .*

Proof. Let \tilde{x} be in \mathcal{X}_{sol} , and let $T = \tau(\tilde{x})$. According to the definition, the image of \tilde{x} is contained in X_{free} and the terminal point, $\tilde{x}(T)$, is in X_G . Since the image of \tilde{x} is compact and the complement of X_{free} is closed, the distance between these two sets, δ_1 , is well-defined and strictly positive. Moreover, the distance from the terminal point to the complement of X_G , δ_2 , is also well-defined and positive.

Consider $\delta = \min(\delta_1, \delta_2)$ and any trajectory \tilde{x}' from \mathcal{X}_1 such that $\rho_{\mathcal{X}}(\tilde{x}, \tilde{x}') < \delta$. First, we show that the image of \tilde{x}' is in X_{free} . Assume to the contrary that there exists some $t' \in [0, \tau(\tilde{x}')] such that $\tilde{x}'(t') \notin X_{\text{free}}$. Let $t = \min(t', T)$. The bound$

$$\begin{aligned}
\|\tilde{x}(t) - \tilde{x}'(t')\| &\leq \|\tilde{x}(t) - \tilde{x}'(t)\| + \|\tilde{x}'(t) - \tilde{x}'(t')\| \\
&\leq \|\tilde{x}(t) - \tilde{x}'(t)\| + M|t - t'| \\
&\leq \rho_{\mathcal{X}}(\tilde{x}, \tilde{x}') < \delta \leq \delta_1 \quad (17)
\end{aligned}$$

contradicts the definition of δ_1 .

Second, we prove that $\tilde{x}'(T') \in X_G$, in which $T' = \tau(\tilde{x}')$. Assume to the contrary that $\tilde{x}'(T') \notin X_G$. Let $T^* = \min(T, T')$ and consider the bound

$$\begin{aligned}
\|\tilde{x}(T) - \tilde{x}'(T')\| &\leq \|\tilde{x}(T) - \tilde{x}(T^*)\| + \|\tilde{x}(T^*) - \tilde{x}'(T^*)\| + \|\tilde{x}'(T^*) - \tilde{x}'(T')\| \\
&\leq M|T - T^*| + \|\tilde{x}(T^*) - \tilde{x}'(T^*)\| + M|T^* - T'| \\
&= \|\tilde{x}(T^*) - \tilde{x}'(T^*)\| + M|\max(T', T) - \min(T', T)| \\
&\leq \rho_{\mathcal{X}}(\tilde{x}, \tilde{x}') < \delta \leq \delta_2, \quad (18)
\end{aligned}$$

which contradicts the definition of δ_2 . By definition, \tilde{x}' belongs to \mathcal{X}_{sol} . \square

Denote a set of solutions to the motion planning problem as \mathcal{U}_{sol} . Clearly, for any given \tilde{u} in \mathcal{U}_{sol} , the trajectory $\tilde{x}(x_1, \tilde{u})$ belongs to \mathcal{X}_{sol} . Therefore, \mathcal{U}_{sol} is a preimage of \mathcal{X}_{sol} with respect to $\tilde{x}(x_1, \tilde{u}) : \mathcal{U} \rightarrow \mathcal{X}$. From the continuity of $\tilde{x}(x_1, \tilde{u})$ and Theorem 2 it follows that \mathcal{U}_{sol} is open. We state this result as a separate theorem:

Theorem 3. *Assume all the conditions of Section 3 are met, then \mathcal{U}_{sol} is open.*

5 Planning Algorithm and Resolution Completeness Conditions

In this section we establish sufficient conditions under which resolution complete algorithms to the motion planning problem exist. A simple resolution complete algorithm is constructed. We also prove the convergence of sampling-based algorithms in the trajectory space.

5.1 Controls Via Concatenation of Primitives

Let $\Sigma \subset \mathcal{U}$ be a countable set of *motion primitives*, each defined over a closed and bounded time interval. Borrowing concepts from the theory of computation, Σ can be interpreted as an *alphabet*. If motion primitives are applied in succession, a control that represents their *concatenation* is obtained. For example, if $\sigma_1, \sigma_2 \in \Sigma$, in which $\sigma_1 : [0, t_1] \rightarrow U$ and $\sigma_2 : [0, t_2] \rightarrow U$, are applied in succession, the resulting control, denoted by $\sigma_1 \sigma_2$ is

$$(\sigma_1 \sigma_2)(t) = \begin{cases} \sigma_1(t) & \text{if } t \in [0, t_1] \\ \sigma_2(t - t_1) & \text{if } t \in [t_1, t_1 + t_2] \end{cases}. \quad (19)$$

Allowing any finite number of concatenations, each resulting control can be expressed as a *string*, which is a finite sequence of motion primitives in Σ . Considering this, the set of all controls that can be formed from motion primitives is the *Kleene star* of Σ , which is denoted and defined as

$$\Sigma^* = \{\sigma_1 \sigma_2 \cdots \sigma_k \mid k \geq 0 \text{ and each } \sigma_i \in \Sigma\}. \quad (20)$$

Note that we do not allow infinite sequences of motion primitives to be applied. The definition of Σ^* allows the empty string, which is assumed to be zero control.

The following lemma establishes a useful property of Σ^* for the purposes of computation.

Lemma 6 (Rectangular enumeration argument). *For any set, Σ , of motion primitives, the set, Σ^* , of all strings is countable.*

Proof. Consider Σ_n^* , which consists of all strings of length not greater than n and composed of any characters of the alphabet $\Sigma|_n$. For example, $\Sigma_1^* = \{\sigma_1\}$, $\Sigma_2^* = \{\sigma_1, \sigma_2, \sigma_1 \sigma_1, \sigma_1 \sigma_2, \sigma_2 \sigma_1, \sigma_2 \sigma_2\}$, and so on; see Fig. 4. Verify that each Σ_n^* is finite, and $\Sigma^* = \bigcup_{n=1}^{\infty} \Sigma_n^*$. Hence, Σ^* is countable as a countable union of finite sets. \square

To facilitate the development of resolution complete planning algorithms, it will be helpful to introduce a set of motion primitives that is straightforward to describe and utilize. Moreover, assuming that all motion primitives in the set Σ are encoded digitally, it follows from the lemma above that all strings in Σ^* are computable.

Fig. 4 Even if Σ is countably infinite, Σ^* is countably infinite, as shown by rectangular enumeration argument. We show that $\Sigma^* = \bigcup_{n=1}^{\infty} \Sigma_n^*$, in which sets Σ_n^* correspond to regions bounded by dashed lines and finite.

	1	2	3
σ_1	{ σ_1 }	{ $\sigma_1 \sigma_1$ }	{ $\sigma_1 \sigma_1 \sigma_1$ }
σ_2	{ σ_2 }	{ $\sigma_1 \sigma_2,$ $\sigma_2 \sigma_1,$ $\sigma_2 \sigma_2$ }	...
σ_3	{ σ_3 }

Suppose that a system is defined as in (3). First, replace U with a countable subset. If U is uncountably infinite, then choose a countable, dense subset $U_d \subset U$. For example, U_d could be the set of all $u \in U$ for which all coordinates are rational numbers. If U is already countable, then we may simply let $U_d = U$.

Let $\Sigma_{dt} \subset \mathcal{U}$ be called the *discrete-time model*, and be defined as the set of all constant functions, $\tilde{u} : [0, t] \rightarrow U_d$, in which $t = 1/2^i$ for all $i \in \mathbb{N}$. Thus, the duration of motion primitives can be $1/2, 1/4, 1/8$, and so on. Any sequence of time intervals that converges to zero may alternatively be used. The set of all strings that utilizes the alphabet Σ_{dt} and the concatenation rule (19) consists of piecewise constant functions. We denote this set of strings as Σ_{dt}^* .

The discrete-time model is just one of numerous possible sets of motion primitives. However, the particular choice of Σ depends heavily on the considered system, the intended application, and the efficiency of the planning algorithm. Virtually any definition of Σ is allowed, provided that Σ^* is dense in \mathcal{U} . We will show that this requirement is sufficient for a resolution complete algorithm to exist.

5.2 Planning algorithm

Based on the background results of Section 4, we are now ready to establish the existence of resolution complete algorithms in a very general setting. The existence is demonstrated by construction of a simple string enumeration algorithm; refer to Algorithm 1.

The selection operation 4 is required to be *systematic*, which means that strings are selected so that all of Σ^* is enumerated as the number of iterations tends to infinity. This is always possible because Σ^* is countable.

For the integration line 5, validation line 6, and termination line 7, we would ideally like to have them executed in constant time with perfect precision. In practice, however, this is usually not possible. Most often, a *numerical* integration is necessary, which causes errors to propagate to the remaining two operations. Due to these limitations, an alternative will be defined: 1) An exact computation model, and 2) a numerical computation model.

Let the *exact computation model* refer to the case in which all operations are performed exactly in finite time without errors. This model is the simplest to analyze;

Algorithm 1 String enumeration algorithm

Input : a set of motion primitives Σ , the initial state x_1 , a goal set X_G , a collision detection module, an integration module.

Output : a solution control $\tilde{\sigma} \in \Sigma^*$.

```

1:  $n \leftarrow 0$ 
2: loop
3:    $\varepsilon_i \leftarrow 1/2^n ; n \leftarrow n + 1$ 
4:    $\tilde{\sigma} \leftarrow$  select the  $n$ th string from  $\Sigma^*$ 
5:    $\hat{x}(x_1, \tilde{\sigma}) \leftarrow$  integrate the trajectory, starting from  $x_1$  under the control  $\tilde{\sigma}$ , with tolerance  $\varepsilon_i$ 
6:    $C_1 \leftarrow$  (true or false) determine whether the image of  $\hat{x}(x_1, \tilde{\sigma})$  is contained in  $X_{\text{free}}$ 
7:    $C_2 \leftarrow$  (true or false) determine whether the final state of  $\hat{x}(x_1, \tilde{\sigma})$  lies in  $X_G$ 
8:   if  $C_1 \wedge C_2$  then
9:     return  $\tilde{\sigma}$ 
10:  end if
11: end loop

```

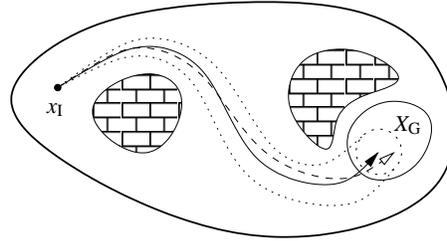
however, it is also less realistic. Note that for the exact computational model the parameter ε_i is ignored in line 5 of the algorithm.

To account for errors arising from a numerical integration error, let the *numerical computation model* be defined as shown in Fig. 5. Assume that the precision of the numerical integration algorithm can be tuned using a parameter $\varepsilon_i \in (0, \infty)$, and the error of the numerical trajectory is bounded by

$$\|\tilde{x}(x_0, \tilde{u}, t) - \hat{x}(x_0, \tilde{u}, t)\| \leq \varepsilon_i F(t), \quad (21)$$

in which F is a strictly positive, monotonic function of time, and $\hat{x}(x_0, \tilde{u})$ is the numerical trajectory.

Fig. 5 The numerical computation model. The dotted region is the error cone constructed around the numerically integrated trajectory (solid line), which contains the exact trajectory (dashed line).



The proposed numerical error model allows us to construct an error cone (around the numerical trajectory), which contains the exact trajectory; see Fig. 5. The error cone is then used for validation and termination purposes. It will be assumed that validation and termination algorithms are conservative, that is, the algorithm must *reject* a candidate solution if there is any possibility that the trajectory leaves X_{free} or fails to terminate in X_G . Moreover, it may also be possible to reject strings under the numerical computation model by determining that *all* possible trajectories either leave X_{free} or fail to terminate in X_G . However, this will not be considered in detail in this paper.

5.3 Sufficient Conditions for Resolution Completeness

We derive a set of sufficient conditions which guarantee that the proposed algorithm is resolution complete. The result is demonstrated for both exact and numerical computational models. Furthermore, we demonstrate that the discrete-time model satisfies these sufficient conditions.

Theorem 4 (Sufficient conditions for the exact computational model). *If Σ^* is dense in \mathcal{U} , then Algorithm 1 is resolution complete under the exact computational model.*

Proof. We have already shown that the set $\mathcal{U}_{\text{sol}} \subset \mathcal{U}$ is open, and in case a solution exists, \mathcal{U}_{sol} is nonempty. Since Σ^* is dense, its intersection with \mathcal{U}_{sol} is also nonempty. Hence, the algorithm, selecting systematically strings from Σ^* , eventually finds an element that is also in \mathcal{U}_{sol} . Under the exact computational model, the selected string is accepted. \square

Theorem 5 (Sufficient conditions for the numerical computational model). *If Σ^* is dense in \mathcal{U} , then Algorithm 1 is resolution complete under the numerical computational model.*

Proof. Let $\tilde{u} \in \mathcal{U}_{\text{sol}}$ be a solution to the motion planning problem and ε be a positive constant. Furthermore, let $\delta = \varepsilon / (2 \max [L_u \exp(L_x \tau(\tilde{u})), M/\alpha])$ and $N \in \mathbb{N}$ be such that $1/2^N < \varepsilon / (4F(\tau(\tilde{u}) + \delta/\alpha))$. Find $\tilde{\sigma} \in \Sigma^*$, such that its number is greater than N and $\rho_{\mathcal{U}}(\tilde{u}, \tilde{\sigma}) < \delta$. It is always possible to do so because Σ^* is assumed to be dense in \mathcal{U} . With such choice of $\tilde{\sigma}$, it follows that $|\tau(\tilde{u}) - \tau(\tilde{\sigma})| < \delta/\alpha$, and, following the proof of Theorem 1, the distance $\rho_{\mathcal{X}}(\tilde{x}(x_1, \tilde{u}), \tilde{x}(x_1, \tilde{\sigma})) < \varepsilon/2$.

Consider the distance between the exact trajectory, $\tilde{x}(x_1, \tilde{u})$, and the numerically integrated trajectory, $\hat{x}(x_1, \tilde{\sigma})$:

$$\begin{aligned} \rho_{\mathcal{X}}(\tilde{x}(x_1, \tilde{u}), \hat{x}(x_1, \tilde{\sigma})) &\leq \rho_{\mathcal{X}}(\tilde{x}(x_1, \tilde{u}), \tilde{x}(x_1, \tilde{\sigma})) + \rho_{\mathcal{X}}(\tilde{x}(x_1, \tilde{\sigma}), \hat{x}(x_1, \tilde{\sigma})) \\ &\leq \frac{\varepsilon}{2} + \varepsilon_1 F(\tau(\tilde{\sigma})) \leq \frac{\varepsilon}{2} + \frac{\varepsilon F(\tau(\tilde{\sigma}))}{4F(\tau(\tilde{u}) + \delta/\alpha)} \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{4} \leq \frac{3\varepsilon}{4}. \end{aligned} \quad (22)$$

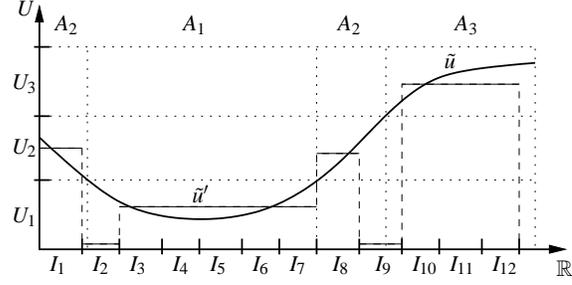
Furthermore, note that the error cone around $\hat{x}(x_1, \tilde{\sigma})$ stays in $\varepsilon/4$ -neighborhood of the numerical trajectory. Hence, the error cone is contained in the ε -neighborhood of $\tilde{x}(x_1, \tilde{u})$. By adjusting ε , which is arbitrary, we can ensure that the error cone is in \mathcal{X}_{sol} . Therefore, $\tilde{\sigma}$ is accepted by Algorithm 1. \square

Now that we have a simple test to determine whether Algorithm 1 is resolution complete, we apply it to the discrete-time model.

Theorem 6 (Denseness theorem). *The set Σ_{dt}^* is dense in \mathcal{U} .*

Proof. Intuitively, the proof follows from the well known fact that piecewise constant functions are dense in the space of measurable functions [20]. Unfortunately, the discrete-time model is only a proper subset of the set of all piecewise constant functions. Here we outline our proof that overcomes this difficulty. Refer to Fig. 6 for details.

Fig. 6 Collection $\{U_i\}$ is a partition of U . Sets A_i s are preimages of U_i s under the map \tilde{u} . Intervals I_j s approximate A_i s from within. The function \tilde{u}' is a piecewise constant approximation of \tilde{u} , defined on the collection $\{I_j\}$.



Consider any \tilde{u} in \mathcal{U} and ε greater than zero. Assume that U is partitioned with a collection of measurable sets $\{U_i\}$ with nonempty interior such that the diameter of each set is less than $\varepsilon/(2\tau(\tilde{u}))$. Let A_i be a preimage of U_i with respect to \tilde{u} . Since \tilde{u} is a measurable function and U_i is a measurable set, A_i is measurable.

Next, consider the approximation of A_i s from within by intervals of length $1/2^N$; denote these intervals as $I_j = [(j-1)/2^N, j/2^N]$. Assume the tolerance of the approximation is less than $\varepsilon/(2\alpha)$, collectively for all A_i s. This means that the measure of the difference between A_i and all intervals which are subsets of A_i does not exceed $\varepsilon/(2\alpha)$, collectively for all sets A_i .

Finally, define u_i to be any element of U_d that is also in U_i (it is possible to find such u_i because U_i has nonempty interior and U_d is dense in U), and let u_0 to be any in U_d . Construct the approximation

$$\tilde{u}'(t) = \begin{cases} u_i & \text{if } \exists i \text{ and } \exists j \text{ such that } t \in I_j \subseteq A_i \\ u_0 & \text{otherwise} \end{cases} \quad (23)$$

By construction, \tilde{u}' is in Σ_{dt}^* . Now, compute the distance

$$\rho_{\mathcal{U}}(\tilde{u}, \tilde{u}') \leq \sum_j' \int_{I_j} \|\tilde{u}(t) - u_i\| dt + \alpha \frac{\varepsilon}{2\alpha} \leq \tau(\tilde{u}) \frac{\varepsilon}{2\tau(\tilde{u})} + \frac{\varepsilon}{2} = \varepsilon. \quad (24)$$

Here, \sum_j' denotes the summation over j such that $I_j \subseteq A_i$ for some i .

In conclusion, we note that for any $\tilde{u} \in \mathcal{U}$ and $\varepsilon > 0$ we found $\tilde{u}' \in \Sigma_{\text{dt}}^*$ such that $\rho_{\mathcal{U}}(\tilde{u}, \tilde{u}') \leq \varepsilon$. Therefore, Σ_{dt}^* is dense in \mathcal{U} . \square

Finally, we have established the main result of the paper:

Theorem 7. *Under exact and numerical computation models, there exists a set of motion primitives and a resolution complete planning algorithm.*

Proof. The result follows from Theorems 4, 5, and 6. \square

Perhaps surprisingly, the algorithm may be resolution complete without actually causing convergence in dispersion. As discussed in Section 2, a dense sample sequence on a precompact set converges. However, \mathcal{U} is generally not precompact. Therefore, our proposed algorithm does not converge in \mathcal{U} . Nevertheless, it converges in the trajectory space for the bounded time problem, assuming X_{free} is bounded:

Theorem 8 (Convergence of Finite Length Trajectories). *Assuming that execution time is bounded by T , the space $\mathcal{X}_{1,T} = \{\tilde{x} \in \mathcal{X}_1 \mid \tau(\tilde{x}) \leq T\}$ is precompact. Hence, a dense sequence of controls corresponds to a dense sequence of trajectories that converges to the set of all feasible trajectories.*

Proof. For the set $\mathcal{X}_{1,T}$, the following two conditions hold:

1. **Uniform boundedness:** There exists $C_1 > 0$ such that $\|\tilde{x}(t)\| \leq C_1$ for all \tilde{x} in $\mathcal{X}_{1,T}$. The condition is easy to verify. Let $C_1 = MT + \|x_1\|$ and consider

$$\|\tilde{x}(t)\| \leq \|\tilde{x}(t) - \tilde{x}(0)\| + \|\tilde{x}(0)\| \leq Mt + \|x_1\| \leq C_1; \quad (25)$$

2. **Uniform equicontinuity:** There exists $C_2 > 0$ such that for all $\tilde{x} \in \mathcal{X}_{1,T}$ and any t and t' in $[0, \tau(\tilde{x})]$, we have $\|\tilde{x}(t) - \tilde{x}(t')\| \leq C_2|t - t'|$. The above follows directly from the definition of \mathcal{X} , by letting $C_2 = M$.

Using the Arzelà–Ascoli⁴ theorem [20], it follows that $\mathcal{X}_{1,T}$ is precompact. The convergence of sampled trajectories follows from Lemma 1. \square

6 Conclusions

To summarize, we revisit Fig. 1 with the precise terminology from the paper to obtain:

$$\begin{array}{ccccc} \Sigma & & U & & X \\ \downarrow & & \downarrow & & \downarrow \\ \Sigma^* & \xrightarrow{\subseteq} & \mathcal{U} & \xrightarrow{\tilde{x}(x_t, \bar{u})} & \mathcal{X}. \end{array} \quad (26)$$

The most important step was to relate the space of controls \mathcal{U} to the trajectory space \mathcal{X} topologically by forming metric spaces. Furthermore, we have relied on the fact that \mathcal{U} can be sufficiently sampled using primitives Σ to obtain Σ^* , yielding a computational approach. Using this formulation, we have introduced a resolution-complete algorithm for motion planning with differential constraints in the most general setting known to date, requiring only Lipschitz continuity of the system.

More importantly, the functional analysis framework provides some practical implication. For example, consider a system which is symmetric under Lie group transformations [10]. Moreover, suppose the dispersion of a given set of motion primitives is limited by $\varepsilon > 0$ in the set of all trajectories with the final time bounded by Δt . Consider further the set of all strings of length N composed of these primitives and the set of all trajectories with the final time bounded by $N\Delta t$. It follows from Lemma 5 that the dispersion of the former set in the later set is bounded by $K\varepsilon$, in which K depends only on N , Δt , and L_x , however, it is independent of a particular trajectory (we leave it to the reader to verify this fact). Hence, to reach the desired dispersion in the reachable set, it suffice to build primitives that approximate short time trajectories with a given resolution. This example illustrates one of the possible practical implications of our study. Furthermore, we believe that the metric space

⁴ In the provided reference the theorem is called Arzelà’s theorem, and the synonymous term “relatively compact” is used instead of “precompact”

formulations and resulting analysis may be practically useful for a broader class of systems and/or motion planning algorithms.

Acknowledgements This work was supported in part by NSF grants 0904501 (IIS Robotics), NSF grant 0535007 (IIS Robotics) and 1035345 (Cyberphysical Systems), DARPA SToMP grant HR0011-05-1-0008, and MURI/ONR grant N00014-09-1-1052.

References

1. J.-P. Laumond, S. Sekhavat, and F. Lamiroux, "Guidelines in nonholonomic motion planning for mobile robots," in *Robot Motion Planning and Control* (J.-P. Laumond, ed.), pp. 1–53, Berlin: Springer-Verlag, 1998.
2. B. R. Donald, P. G. Xavier, J. Canny, and J. Reif, "Kinodynamic planning," *Journal of the ACM*, vol. 40, pp. 1048–66, Nov. 1993.
3. J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.
4. J. Hollerbach, "Dynamic scaling of manipulator trajectories," tech. rep., MIT A.I. Lab Memo 700, 1983.
5. J.-J. E. Slotine and H. S. Yang, "Improving the efficiency of time-optimal path-following algorithms," *IEEE Transactions on Robotics & Automation*, vol. 5, no. 1, pp. 118–124, 1989.
6. S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006. Also available at <http://planning.cs.uiuc.edu/>.
7. J. H. Reif, "Complexity of the mover's problem and generalizations," in *Proceedings IEEE Symposium on Foundations of Computer Science*, pp. 421–427, 1979.
8. J.-C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer, 1991.
9. M. S. Branicky, R. A. Knepper, and J. J. Kuffner, "Path and trajectory diversity: Theory and algorithms," in *Proceedings IEEE International Conference on Robotics & Automation*, pp. 1359–1364, 2008.
10. E. Frazzoli, M. A. Dahleh, and E. Feron, "Robust hybrid control for autonomous vehicles motion planning," Tech. Rep. LIDS-P-2468, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 1999.
11. L. H. Erickson and S. M. LaValle, "Survivability: Measuring and ensuring path diversity," in *Proceedings IEEE International Conference on Robotics and Automation*, 2009.
12. A. Bhatia and E. Frazzoli, "Sampling-based resolution-complete algorithms for safety falsification of linear systems," in *Hybrid Systems: Computation and Control* (M. Egerstedt and B. Mishra, eds.), pp. 606–609, Berlin: Springer-Verlag, 2008. Lecture Notes in Computer Science, 4981.
13. E. Haghverdi, P. Tabuada, and G. J. Pappas, "Bisimulation relations for dynamical, control, and hybrid systems," *Theoretical Computer Science*, vol. 342, pp. 229–261, Sept. 2005.
14. D. P. Bertsekas, "Convergence in discretization procedures in dynamic programming," *IEEE Transactions on Automatic Control*, vol. 20, pp. 415–419, June 1975.
15. S. M. LaValle and P. Konkimalla, "Algorithms for computing numerical optimal feedback motion strategies," *International Journal of Robotics Research*, vol. 20, pp. 729–752, Sept. 2001.
16. H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, 2005.
17. H. Niederreiter, *Random Number Generation and Quasi-Monte-Carlo Methods*. Philadelphia: Society for Industrial and Applied Mathematics, 1992.
18. H. Royden, *Real Analysis*. Collier-Macmillan Limited, London: The Macmillan Company, 1988.
19. M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equation*. Boston, Basel, Berlin: Birkhäuser, 2008.
20. A. N. Kolmogorov and S. V. Fomin, *Introductory Real Analysis*. Dover Publications, Inc., New York: Dover, 1975.