2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Decidability of Motion Planning with Differential Constraints

Peng Cheng George Pappas Vijay Kumar
GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104 USA
{chpeng, pappasg, kumar}@grasp.upenn.edu

***Abstract*— Classical path planning does not address many of the challenges of robotic systems subject to differential constraints. While there have been many recent efforts to develop motion planning algorithms for systems with differential constraints (MPD), very little has been said about the existence of exact algorithms. In other words, the decidability of MPD problems is still an open question. In this paper, we propose a partial answer to this question limiting ourselves to special cases where the trajectory functions of the systems under the finite-dimensional piecewise-continuous controls have a closed-form polynomial formulation. We define an abstract formulation for the MPD problem based on the concept of a control space. We provide an incremental decision algorithm to answer the decidability question and present sufficient conditions for problems to which this algorithm can be applied. Decidability results for several non trivial MPD problems are presented. For example, we show that the question of existence of a trajectory for a Dubin's car with a polygonal rigid body between two specified positions and orientations in a polygonal environment with a fixed and finite number of discontinuities in curvature is decidable.**

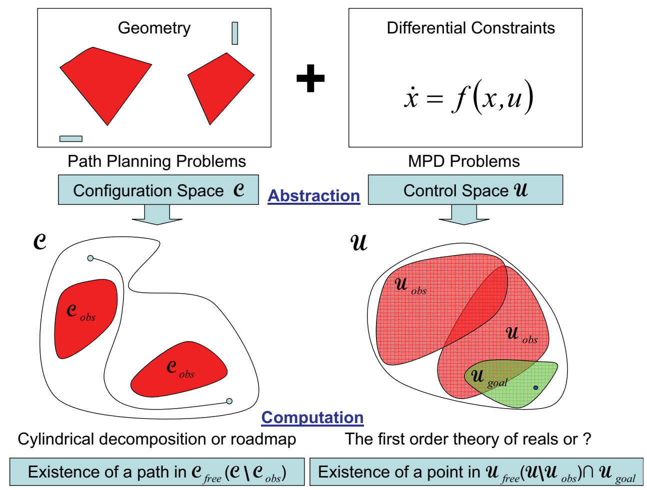

Fig. 1. Configuration space abstraction for path planning and control space abstraction for MPD

I. Introduction

Motion planning is important in many applications of robotic systems including manipulation, transportation, manufacturing, surveillance, and medical instrumentation. Because differential constraints, such as drift and especially nonholonomy, restrict admissible velocities and accelerations, a collision-free path from classical path planning, which ignores differential constraints, might be infeasible and inefficient for systems to follow. Recent motion planning algorithms start to directly consider differential constraints in the planning process, such as nonholonomic planning [13] and kinodynamic planing [7], which are together called *motion planning with differential constraints* (MPD) [4], [14]. While a large amount of practical MPD algorithms have obtained considerable success, there is very little known about existence of exact algorithms that decide whether a solution exists in finite time. The decidability of MPD problems is unknown, except for $1D$ [16] and $2D$ problems [2], and for a Dubin's car [8] with the point body moving in a polygonal environment [9]. These exact algorithms will improve understanding of the complexity of MPD problems, which provides a sound reason to pursue practical incomplete algorithms and helps design of novel verification algorithms [5] using motion planning techniques.

The concept of the *configuration space* established a fundamental abstract formulation for general path planning problems [15]. Workspace obstacles are mapped into configuration space obstacles, denoted as $\mathcal{C}_{obs}$'s, and a solution is a collision-free and continuous path between the initial and goal configurations (see Fig. 1 left). With this abstraction, a central scheme in designing exact algorithms is to first construct a finite and exact representation of the collision-free configuration space, such as cylindrical decomposition [19] or roadmap [3], and then use a graph search algorithm to query the existence of a solution.

In the spirit of the configuration space abstraction, this paper presents an abstract formulation for general MPD problems using the concept of a *control space*. In this formulation (Fig. 1 right), workspace obstacles are mapped into control space obstacles, denoted as $\mathcal{U}_{obs}$'s, which are subsets of the control space including all controls that drive the system into collision. The goal-reaching set $\mathcal{U}_{goal}$ includes all controls which drive the system to the goal state while ignoring obstacles. A solution is a point inside the intersection of the collision-free control set $\mathcal{U} \setminus \mathcal{U}_{obs}$ and $\mathcal{U}_{goal}$. Because the control space is universal for all robotic systems, any MPD problem can be incorporated into this formulation. The proposed formulation can be considered as a generalization of the unconventional "configuration space" specially designed for the Dubin's car in [9], [10] to characterize the trajectories connecting two vertices, instead of the conventional configurations.

In this paper, we design an incremental decision algorithm to check the existence of a solution for MPD problems for systems whose finite-dimensional controls are piecewise-continuous with a fixed and finite number of switches. We prove that if the sets in the abstract formulation are semi-algebraic, then the problem is decidable using the proposed decision algorithm. While this sufficient condition verifies decidability of MPD problems with a class of linear systems [12], we also obtain a new decidability result for a non trivial MPD problem using similar symbolic computation techniques in [12], [19], in which the Dubin's car with a polygonal rigid body moves between two given configurations in a polygonal environment using up to a fixed and finite number of control switches.

Besides of exact algorithms, local properties of the systems and planners, such as integrable differential constraints, small time local controllability, and topological properties of planners [20], can also be used to decide specific MPD problems by reducing their decidability into that of path planning problems. Semi-decidability results were also provided for general MPD problems by the resolution complete MPD algorithms [4], which will find an existing solution in finite time, but run forever when there is no solution.

The organization of the paper is as follows. In Section II, the MPD problem is formulated. Section III provides the abstract formulation and incremental decision algorithm. Decidable MPD problems are given in Section IV. Conclusion and several future directions are given in Section V.

II. PROBLEM FORMULATION

An MPD problem, denoted as $\mathcal{M}$, can be considered as an extension of the path planning problem, denoted as $\mathcal{P}$, with differential constraints, denoted as $\mathcal{F}$. Typical MPD includes both nonholonomic planning and kinodynamic planning.

A path planning problem $\mathcal{P}$ has an abstract formulation $(\mathcal{C}, \mathcal{C}_{obs}, q_{init}, q_{goal})$, in which $\mathcal{C} \subset \Re^k$ is the configuration space, $\mathcal{C}_{obs} \subset \mathcal{C}$ includes all configurations in collision, and $q_{init}, q_{goal} \in \mathcal{C}$ are respectively the initial and goal configurations. Set $\mathcal{C}_{obs}$ is normally assumed to be semi-algebraic such that the problem can be decided either by cylindrical decomposition [19] or roadmap [3]. Set $\mathcal{C}_{free} = \mathcal{C} \setminus \mathcal{C}_{obs}$ includes all collision-free configurations. A solution for $\mathcal{P}$ is a continuous path from q_{init} to q_{goal} in $\mathcal{C}_{free}$.

Differential constraints, denoted as $\mathcal{F}$, are characterized by the triple $(f, X, \mathcal{U})$, in which f represents a set of Ordinary Differential Equations (ODEs), $\dot{x} = f(x, u)$, in which x is a state in the state space $X \subset \Re^n$, and u is an input in the input space $U \subset \Re^m$. The trajectory $\tilde{x}$ of the system under a given control $\tilde{u} : [0, t_{\tilde{u}}] \to U$ ($t_{\tilde{u}}$ varies with $\tilde{u}$) from a state $x_0 \in X$ is

$$\tilde{x}(\tilde{u}, x_0, t) = x_0 + \int_0^t f\left(\tilde{x}(\tau), \tilde{u}(\tau)\right) d\tau, t \in [0, t_{\tilde{u}}]. \quad (1)$$

We use $\mathcal{U}$ to denote the *control space*, which includes all admissible controls for the system. For the purpose of analysis and computation, a control space is normally assumed to consist of a class of controls, such as finite-dimensional continuous functions including sinusoidals and polynomials, or piecewise-continuous controls. Note that the general control space is infinite dimensional because a control can either be a general continuous function or have an infinite number of switches.

Given an initial state x_{init} and goal state x_{goal}, a control $\tilde{u}$ is a solution if $\tilde{x}(\tilde{u}, x_{init}, t_{\tilde{u}}) = x_{goal}$ and $P_{\mathcal{C}}^X\left(\tilde{x}(\tilde{u}, x_0, t)\right) \notin \mathcal{C}_{obs}, \forall t \in [0, t_{\tilde{u}}]$, in which $P_{\mathcal{C}}^X$ is the projection from state space X to configuration space $\mathcal{C}$.

An exact algorithm will decide whether such a solution exists in finite time, i.e., if an exact algorithm exists, then the problem is decidable. In this paper, we only care about decidability, and the exact algorithm might not be able to return a solution if one does exist.

III. ABSTRACT FORMULATION AND AN INCREMENTAL DECISION ALGORITHM

In this section, we will first present the control space abstraction for the MPD problem, based on which an incremental decision algorithm is given. Finally, an example is provided to illustrate the decision process.

A. The control space abstraction

As the configuration space abstraction of path planning problems maps the workspace obstacles into the configuration space obstacles, we use the control space concept to map the workspace obstacles into the *control space obstacles*, or called *collision control set*,

$$\mathcal{U}_{obs} = \left\{\tilde{u} \in \mathcal{U} \mid \exists t \in [0, t_{\tilde{u}}], P_{\mathcal{C}}^X\left(\tilde{x}(\tilde{u}, x_{init}, t)\right) \in \mathcal{C}_{obs}\right\}. \quad (2)$$

We define *collision-free control set* as $\mathcal{U}_{free} = \mathcal{U} \setminus \mathcal{U}_{obs}$ to include all controls which will not cause collision. The *goal-reaching control set*, $\mathcal{U}_{goal} = \{\tilde{u} \in \mathcal{U} \mid \tilde{x}(\tilde{u}, x_{init}, t_{\tilde{u}}) = x_{goal}\}$, includes all the controls that drive the system to x_{goal} while ignoring the workspace obstacles. Thus, the MPD problem has the following abstract formulation $(\mathcal{U}, \mathcal{U}_{obs}, \mathcal{U}_{goal})$. A solution is a control $\tilde{u} \in \mathcal{U}_{free} \cap \mathcal{U}_{goal}$. An exact algorithm will decide whether $\mathcal{U}_{free} \cap \mathcal{U}_{goal}$ is empty in finite time.

B. An incremental decision algorithm

Our decision algorithm for MPD problems depends on the following three assumptions:

1. Semi-algebraic $\mathcal{C}_{obs}$ Let a configuration be parameterized by $q \in \Re^k$. Set $\mathcal{C}_{obs}$ is the following semi-algebraic set:

$$\{q \mid \vee_{i=1,2,\cdots,n_o} \left(\wedge_{m=1,2,\cdots,n_e} \mathcal{A}_{i,m}(q) \leq 0\right)\}, \quad (3)$$

in which $\vee$ and $\wedge$ are respectively the standard logical "or" and "and" operators, $\{\mathcal{A}_{m,i}(q)\}$ are polynomial functions [1] of q, and n_o and n_e are constant positive integers. For the details about computation of $\mathcal{C}_{obs}$ and semi-algebraic sets, please refer to [6], [14].

2. Finite-dimensional control space Generally, the control space can be formulated as a semigroup [2] generated from a finite-dimensional *control space generator set*, denoted as $\bar{\mathcal{U}}$ [4], which includes only continuous controls. The semigroup operation $\circ$ for any two controls $\tilde{u}_1 : [0, t_1] \to U$ and $\tilde{u}_2 : [0, t_2] \to U$ is simply a "concatenation" and formally defined as

$$(\tilde{u}_1 \circ \tilde{u}_2)(t) = \begin{cases} \tilde{u}_1(t) & t \in [0, t_1) \\ \tilde{u}_2(t - t_1) & t \in [t_1, t_1 + t_2]. \end{cases} \quad (4)$$

The *k-order expanded control set* is defined as

$$\hat{\mathcal{U}}^k = \{\tilde{u}_1 \circ \tilde{u}_2 \circ \cdots \circ \tilde{u}_k \mid \tilde{u}_i \in \bar{\mathcal{U}} \text{ for } i = 1, 2, \cdots, k\}. \quad (5)$$

[1] As the standard assumption to ensure exact computation in the computer, we also require in this paper that the coefficients of all polynomials defining the semi-algebraic sets are algebraic numbers.

[2] A semigroup is a set equipped with an operation whose elements satisfy closure and associativity properties.

IDA($\mathcal{M}$)
1 $\mathcal{U}^0_{free} = \emptyset$
2 $k = 1$
3 **while** $(k \le K)$
4 Construct $\mathcal{U}^k_{obs}$ from $\mathcal{U}^{k-1}_{free}$ and $\bar{\mathcal{U}}$
5 Construct $\mathcal{U}^k_{free} = (\mathcal{U}^{k-1}_{free} \circ \bar{\mathcal{U}}) \setminus \mathcal{U}^k_{obs}$
6 **if** $\mathcal{U}^k_{free} = \emptyset$
7 Return and report that there is no solution
8 Construct $\mathcal{U}^k_{goal}$
9 **if** $\mathcal{U}^k_{free} \cap \mathcal{U}^k_{goal} \neq \emptyset$
10 Return and report that there exists a solution
11 $k = k + 1$
12 Report that there is no solution

Fig. 2. The incremental decision algorithm (IDA) for MPD problems

The *k-order generated control set* is defined as $\bar{\mathcal{U}}^k = \cup_{i=1}^k \hat{\mathcal{U}}^i$. The general control space is defined as $\mathcal{U} = \bar{\mathcal{U}}^\infty$. If a control $\tilde{u}$ in the generator $\bar{\mathcal{U}}$ is parameterized by $l < \infty$ real numbers $c_1, c_2, \cdots, c_l$, then the general control space is infinite dimensional because of the infinite number of switches. To ensure finite computation time for exact algorithms, the control space in this paper only consists of piecewise-continuous controls with up to K pieces, that is, $\mathcal{U} = \bar{\mathcal{U}}^K$, which is $K \times l$ dimensional.

3. Closed-form trajectory functions We assume that the integral in (1), i.e., the trajectory function $\tilde{x}$, is a closed-form function of control parameters.

Remark: If the trajectory (1) is not closed-form, then its numerical computation is necessary and exact representations of $\mathcal{U}_{free}$ and $\mathcal{U}_{goal}$ are not available such that exact algorithms based on the control space abstraction in Section III-A do not exist. This remark highlights the importance of motion equation and control space for obtaining decidable problems. The combination of motion equation and control space must generate a closed-form trajectory function, which might be one of the reasons why exact algorithms for MPD problems with most robotic systems are unknown because closed-form integration only exists for few classes of simple functions.

Before describing the decision algorithm, let us first introduce the following notations. We use set

$$\mathcal{U}^k_{free} = \{\tilde{u} \in \hat{\mathcal{U}}^k \mid \tilde{u} \text{ causes no collision with } \mathcal{C}_{obs}\} \quad (6)$$

to denote all collision-free controls in $\hat{\mathcal{U}}^k$. Set

$$\begin{aligned}\mathcal{U}^k_{obs} = \{\tilde{u} \in \hat{\mathcal{U}}^k \mid \tilde{u} = \tilde{u}' \circ \tilde{u}'', \tilde{u}' \in \mathcal{U}^{k-1}_{free}, \tilde{u}'' \in \bar{\mathcal{U}},\\ \tilde{u} \text{ causes collision with } \mathcal{C}_{obs}\}\end{aligned} \quad (7)$$

denotes controls in $\hat{\mathcal{U}}^k$ whose first $k-1$ pieces are collision-free, but the last piece causes collision. Set

$$\mathcal{U}^k_{goal} = \{\tilde{u} \in \hat{\mathcal{U}}^k \mid \tilde{x}(\tilde{u}, x_{init}, t_{\tilde{u}}) = x_{goal}\} \quad (8)$$

includes all controls in $\hat{\mathcal{U}}^k$ which lead the system to x_{goal}.

It is easy to see in Fig. 3 that

$$\mathcal{U}^k_{free} = (\mathcal{U}^{k-1}_{free} \circ \bar{\mathcal{U}}) \setminus \mathcal{U}^k_{obs}, \quad (9)$$

$$\mathcal{U}_{obs} = \bigcup_{k=1,2,\cdots,K} \mathcal{U}^k_{obs}, \mathcal{U}_{free} = \bigcup_{k=1,2,\cdots,K} \mathcal{U}^k_{free}, \quad (10)$$

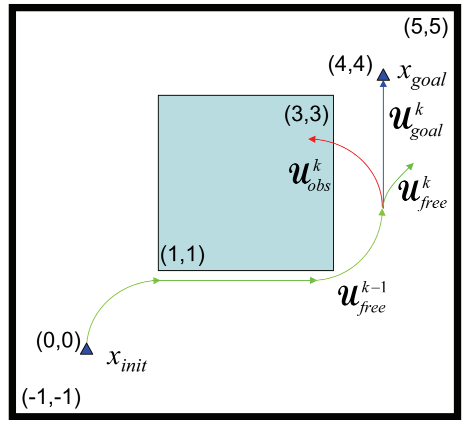

Fig. 3. An MPD problem for the Dubin's car

$$\mathcal{U}_{goal} = \bigcup_{k=1,2,\cdots,K} \mathcal{U}^k_{goal}. \quad (11)$$

The relationship between $\mathcal{U}^k_{free}$, $\mathcal{U}^k_{obs}$, $\mathcal{U}^k_{goal}$, and $\bar{\mathcal{U}}$ in (7), (8), and (9) naturally leads to an incremental decision algorithm, denoted as *IDA*, for a given MPD problem $\mathcal{M}$. The IDA iteratively constructs $\mathcal{U}^k_{obs}$, $\mathcal{U}^k_{free}$, and $\mathcal{U}^k_{goal}$ and checks the existence of a solution (see Fig. 2). Note that Steps 6 and 7 in Fig. 2 are optional, but might help the algorithm to terminate earlier before reaching iteration K.

C. An example of the IDA computation

The MPD problem in Fig. 3 for the Dubin's car is used to illustrate the IDA. The initial and goal states are respectively $(0, 0, \pi/2)$ and $(4, 4, \pi/2)$. An obstacle $(1, 3) \times (1, 3)$ is in the workspace $(-1, 5) \times (-1, 5)$. The car is assumed to be a point to simplify $\mathcal{C}_{obs}$. The differential constraint is given in the following ODEs: $\dot{x} = \cos(\theta)$, $\dot{y} = \sin(\theta)$, $\dot{\theta} = w$, in which $w \in \{-1, 0, 1\}$ is the turning rate of the car. The control space consists of piecewise-constant controls with up to three pieces (two switches). Each constant piece of a control has turning rate w and duration t, and is represented by (w, t). If $w \neq 0$, then $|w|t$ is in $(0, 2\pi)$ to avoid the car to stay static or come back to the starting configuration when it moves around a circle; otherwise, duration t is in $(0, \infty)$. Therefore, the generator set $\bar{\mathcal{U}}$ is

$$\{(w \neq 0 \wedge |w|t \in (0, 2\pi)) \vee (w = 0 \wedge t > 0)\}. \quad (12)$$

Trajectories of the car under a control with $w \neq 0$ or $w = 0$ from state (x_0, y_0, θ_0) are respectively:

$$\begin{aligned} x(w,t) &= (\sin(\theta_0 + wt) - \sin(\theta_0))/w + x_0 \\ y(w,t) &= (\cos(\theta_0) - \cos(\theta_0 + wt))/w + y_0 \\ \theta(w,t) &= \theta_0 + wt, \end{aligned} \quad (13)$$

$$\begin{aligned} x(w,t) &= \cos(\theta_0)t + x_0 \\ y(w,t) &= \sin(\theta_0)t + y_0 \\ \theta(w,t) &= \theta_0. \end{aligned} \quad (14)$$

The following gives the first iteration computation of the IDA in Fig. 2. The other iterations are computed similarly.

Step 4 in Fig. 2: Constructing $\mathcal{U}^1_{obs}$ According to (13) and (14), a control in $\bar{\mathcal{U}}$ will lead to collision if it is in:

$$\begin{aligned}\mathcal{U}^1_{obs} = \{(w,t) \mid \exists \tau \in (0,t], (1 \le x(w,\tau) \le 3 \wedge \\ 1 \le y(w,\tau) \le 3) \vee x(w,\tau) \le -1 \vee \\ x(w,\tau) \ge 5 \vee y(w,\tau) \le -1 \vee y(w,\tau) \ge 5\}.\end{aligned} \quad (15)$$

Step 5: Constructing $\mathcal{U}^1_{free}$ According to (9), because $\mathcal{U}^0_{free} = \emptyset$, we have $\mathcal{U}^1_{free} = \bar{\mathcal{U}} \setminus \mathcal{U}^1_{obs}$.

Step 6: Checking whether $\mathcal{U}^1_{free}$ is empty By manually checking, we know that $\mathcal{U}^1_{free}$ is not empty.

Step 8: Constructing $\mathcal{U}^1_{goal}$ According to (13) and (14),

$$\mathcal{U}^1_{goal} = \{(w,t) \mid x(w,t) = 4, y(w,t) = 4, \theta(w,t) = \pi/2\}. \quad (16)$$

Step 9: Checking whether $\mathcal{U}^1_{free} \cap \mathcal{U}^1_{goal}$ is empty It is easy to see that $\mathcal{U}^1_{goal}$ is empty such that there is no solution at this iteration. The computation will go to Step 4.

For this simple example, we can manually complete the above computation. However, for general MPD problems, as the trajectory function and geometry of the workspace become complicated and the number of control switches increases, computation in Steps 6 and 9 might not computable with existing tools such that decidability of the problem is unknown. Noticing that Step 6 is optional, the following theorem gives the sufficient condition for decidability of an MPD problem.

Theorem 1: If Step 9 of the IDA in Fig. 2 can be computed in finite time for a given MPD problem, then the existence of a solution for the problem can be decided by the IDA, i.e., the MPD problem is decidable.

Proof: Firstly, the IDA will terminate in finite time because it only has a finite number of iterations and each iteration costs finite time. Secondly, (10) and (11) show that the algorithm will report whether a solution exists. ■

Note that Theorem 1 does not restrict us to use a special computational method. However, no matter which method is used, if Step 9 can be computed in finite time, then the problem is always decidable.

IV. Decidable MPD Problems with Quantifier Elimination

Computation in Step 9 in Fig. 2 can be achieved through evaluation of the following quantified formula:

$$\exists \tilde{u}, \tilde{u} \in \mathcal{U}^k_{free} \cap \mathcal{U}^k_{goal}. \quad (17)$$

By Tarski's theorem [21], if both $\mathcal{U}_{free}$ and $\mathcal{U}_{goal}$ are semi-algebraic sets, then quantifier elimination will be able to compute (17) (i.e., Step 9) in finite time. Therefore, we have the following result.

Theorem 2: If both $\mathcal{U}_{free}$ and $\mathcal{U}_{goal}$ of a given MPD problem are semi-algebraic sets, then the MPD problem is decidable.

The following two types of MPD problems satisfy these conditions. The first one is the MPD problems for a class of linear systems [12], [11]. The second one is an MPD problem for the Dubin's car with a polygonal rigid body moving in a polygonal environment. The key to show the decidability of these problems is to find nontrivial transformations to construct a closed-form polynomial formulation of the trajectory functions of these systems under the given control space, such that $\mathcal{U}_{free}$ and $\mathcal{U}_{goal}$ can be shown to be semi-algebraic.

A. Decidable MPD problems with a class of linear systems

MPD problems have close relation with reachability problems. MPD problems compute collision-free trajectories from the initial states to the goal states, while reachability problems compute all states reachable from initial state sets. If a reachability problem is decidable, then the corresponding MPD problem is decidable. The reachability results in [12] have been used to design exact planning algorithms and show the MPD problems with a class of linear systems are decidable [11]. It can be verified that $\mathcal{U}_{free}$ and $\mathcal{U}_{goal}$ for these problems are semi-algebraic, and the results in [11] are shown in Theorem 3 as examples of applications of Theorem 2.

Assume that motion equation of the robotic system is:

$$\dot{x} = Ax + Bu, \quad (18)$$

which includes many robotic systems admitting feedback linearizable controls. The control space generator set is of the following form

$$\begin{aligned} \bar{\mathcal{U}} = \{\tilde{u} = [\tilde{u}_1, \cdots, \tilde{u}_m] \mid \tilde{u}_j = \textstyle\sum_{l=1}^r b_{jl} p_l(t), \\ \Phi_b(b_{jl}), 1 \le j \le m, 1 \le l \le r\}, \end{aligned} \quad (19)$$

in which $\Phi_b(b_{jl})$ is a semi-algebraic set, and $\{p_l(t)\}$ are some basis functions. The results in [12] are given as follow.

Theorem 3: [11] A linear system has motion equation (18) and control space generator (19). Let Λ be the spectrum of matrix A in (18). The reachability problem of the linear system is decidable if

1) A is nilpotent, and the basis functions are of the form $p_l(t) = t^l$, or
2) A is diagonalizable with real, rational eigenvalues, and the basis functions are of the form $p_l(t) = e^{\mu_l t}$ with $\mu_l \notin \Lambda$, or
3) A is diagonalizable, has purely imaginary eigenvalues of the form $\mathrm{i}r$ with r be rational number, and the basis functions are of form $p_l(t) = \sin(\mu_l t)$ or $p_l(t) = \cos(\mu_l t)$, with $\mu_l \notin \Lambda$.

It can be verified that if the linear system satisfies one of the above conditions, then its trajectory will be transformed into a closed-form polynomial function (see [12] for details). Therefore, the corresponding $\mathcal{U}_{obs}$, $\mathcal{U}_{free}$, and $\mathcal{U}_{goal}$ will be semi-algebraic. As a special case of [12], it can also be shown that $\mathcal{U}_{obs}$, $\mathcal{U}_{free}$, and $\mathcal{U}_{goal}$ of the problem in [22] are also semi-algebraic because the trajectory is piecewise-linear.

B. A decidable MPD problem for the Dubin's car

The MPD problem for the Dubin's car with a point body moving in a polygonal environment has been shown to be decidable [9], and an approximate algorithm was given in [10]. In this section, we will use the proposed IDA to give a more general decidable result for MPD problems in Theorem 4, in which the Dubin's car has a polygonal rigid body and moves in a polygonal environment. Furthermore, Corollary 2 shows that the results in [9], [10] can be also obtained from Theorem 4.

Theorem 4: If the Dubin's car with a polygonal rigid body moves in a polygonal environment and its control space is $\mathcal{U} = \bar{\mathcal{U}}^K$ for a fixed positive integer K and generator $\bar{\mathcal{U}}$ in (12), then the MPD problem for this car is decidable.

Proof: Semi-algebraic configuration space obstacles It is easy to check that parameterizing configuration (x, y, θ) of a planar polygonal rigid body as $q = (x, y, \alpha = \cos\theta, \beta = \sin\theta)$ with $\alpha^2 + \beta^2 = 1$ will make the configuration space obstacles be semi-algebraic sets in form (3).

Transforming trajectory functions into closed-form polynomials With respect to the above configuration space parameterization, trajectories (13) and (14) for $w \neq 0$ and $w = 0$ from the starting state $q_0 = (x_0, y_0, \alpha_0 = \cos\theta_0, \beta_0 = \sin\theta_0)$ are respectively transformed into

$$\begin{array}{rcl} x(w,t) & = & (\beta_0\cos(wt) + \alpha_0\sin(wt) - \beta_0)/w + x_0 \\ y(w,t) & = & (\alpha_0 - \alpha_0\cos(wt) + \beta_0\sin(wt))/w + y_0 \\ \alpha(w,t) & = & \alpha_0\cos(wt) - \beta_0\sin(wt) \\ \beta(w,t) & = & \beta_0\cos(wt) + \alpha_0\sin(wt), \end{array} \tag{20}$$

$$\begin{array}{rcl} x(w,t) & = & \cos(\theta_0)t + x_0 = \alpha_0 t + x_0 \\ y(w,t) & = & \sin(\theta_0)t + y_0 = \beta_0 t + y_0 \\ \alpha(w,t) & = & \cos(\theta_0) = \alpha_0 \\ \beta(w,t) & = & \sin(\theta_0) = \beta_0. \end{array} \tag{21}$$

It is easy to see that (21) is a polynomial function of time t and the starting state q_0.

Under the following substitution

$$l = wt, \eta = 1/w, z_1 = \cos(l), z_2 = \sin(l), z_1^2 + z_2^2 = 1, \tag{22}$$

the control $(w \neq 0, t)$ with $|wt| \in (0, 2\pi)$ has two equivalent parameterizations (l, η) and (z_1, z_2, η) such that (20) can be written in the following forms

$$\begin{array}{rcl} x(l,\eta) & = & (\beta_0\cos(l) + \alpha_0\sin(l) - \beta_0)\eta + x_0 \\ y(l,\eta) & = & (\alpha_0 - \alpha_0\cos(l) + \beta_0\sin(l))\eta + y_0 \\ \alpha(l,\eta) & = & \alpha_0\cos(l) - \beta_0\sin(l) \\ \beta(l,\eta) & = & \beta_0\cos(l) + \alpha_0\sin(l),, \end{array} \tag{23}$$

$$\begin{array}{rcl} x(z_1,z_2,\eta) & = & (\beta_0 z_1 + \alpha_0 z_2 - \beta_0)\eta + x_0 \\ y(z_1,z_2,\eta) & = & (\alpha_0 - \alpha_0 z_1 + \beta_0 z_2)\eta + y_0 \\ \alpha(z_1,z_2,\eta) & = & \alpha_0 z_1 - \beta_0 z_2 \\ \beta(z_1,z_2,\eta) & = & \beta_0 z_1 + \alpha_0 z_2. \end{array} \tag{24}$$

It is easy to see that the second form is a polynomial function of z_1, z_2, η, and the starting state q_0. It can be further checked that the trajectory function for a piecewise-constant control with $k < \infty$ switches is also a closed-form polynomial.

Construction of semi-algebraic $\mathcal{U}^1_{obs}$, $\mathcal{U}^1_{free}$, and $\mathcal{U}^1_{goal}$ Because trajectory functions are different for $w = 0$ and $w \neq 0$, we will construct their semi-algebraic sets respectively.

The set of constant controls with $w = 0$ and duration t is

$$\bar{\mathcal{U}}_{w=0} = \{(0,t) \mid t \in (0,\infty)\}. \tag{25}$$

The collision, collision-free, and goal-reaching control set are respectively

$$\mathcal{U}^1_{obs,w=0} = \{(0,t) \mid \exists\tau \in (0,t], \tilde{x}(0,\tau) \in \mathcal{C}_{obs}\}, \tag{26}$$

$$\mathcal{U}^1_{free,w=0} = \{(0,t) \mid \forall\tau \in (0,t], \tilde{x}(0,\tau) \notin \mathcal{C}_{obs}\}, \tag{27}$$

$$\mathcal{U}^1_{goal,w=0} = \{t \mid \tilde{x}(t) = q_{goal}\}. \tag{28}$$

Because $\tilde{x}(w,t)$ is a polynomial trajectory function (see (21)) from x_{init} and $\mathcal{C}_{obs}$ is semi-algebraic, all these three sets are semi-algebraic by Tarski's theorem.

The set of controls with $w \neq 0$ and duration t is parameterized as

$$\bar{\mathcal{U}}_{w\neq0} = \{(w,t) \mid t \in (0,\infty), |w|t \in (0,2\pi)\}. \tag{29}$$

The collision control set is

$$\mathcal{U}^1_{obs,w\neq0} = \{(w,t) \in \bar{\mathcal{U}}_{w\neq0} \mid \exists\tau \in (0,t], \tilde{x}(w,\tau) \in \mathcal{C}_{obs}\}, \tag{30}$$

in which $\tilde{x}(w,t)$ represents the trajectory function (13) from x_{init}. With the first two substitutions in (22), set $\mathcal{U}^1_{obs,w\neq0}$ also has the following equivalent parameterizations:

$$\mathcal{U}^1_{obs,w\neq0} = \{(l,\eta) \mid |l| \in (0,2\pi), \exists\tau \in (0,l], \tilde{x}(\tau,\eta) \in \mathcal{C}_{obs}\}, \tag{31}$$

in which $\tilde{x}(l,\eta)$ is the trajectory function in (23). In the above, we have shown that with substitution (22), (23) can be transformed into polynomial functions (24). However, because the sinusoidal functions are not bijections, simply applying the last three substitutions in (22) on (31) will not obtain an equivalent parameterization of (31).

Instead, we reparameterize a control with $w \neq 0$ and duration t with four consecutive constant controls, each of which has the same turning rate w and durations t_i with $|w|t_i \in (0, \pi/2)$ for $i = 1$, 2, 3 and 4. With the reparameterization, the control set (29) is changed into

$$\begin{array}{rcl} \bar{\mathcal{U}}_{w\neq0} = & \{(w,t_1,t_2,t_3,t_4) \mid t_i > 0, |w|t_i \in (0,\pi/2)\} \\ = & \bar{\mathcal{U}}_{w\neq0,\frac{\pi}{2}} \circ \cdots \circ \bar{\mathcal{U}}_{w\neq0,\frac{\pi}{2}} = \hat{\mathcal{U}}^4_{w\neq0,\frac{\pi}{2}}, \end{array} \tag{32}$$

in which

$$\begin{array}{rcl} \bar{\mathcal{U}}_{w\neq0,\frac{\pi}{2}} & = & \{(w,t) \mid t \in (0,\infty), |w|t \in (0,\pi/2)\} \\ & = & \{(l,\eta) \mid |l| \in (0,\pi/2)\}(\text{with } (22)) \\ & = & \{(z_1,z_2,\eta) \mid z_1^2 + z_2^2 = 1, z_1, z_2 \in (0,1)\}. \end{array} \tag{33}$$

Note that in (33), all three representations of $\bar{\mathcal{U}}_{w\neq0,\frac{\pi}{2}}$ are equivalent because the sinusoidal functions are bijections in the restricted domain $(0, \pi/2)$.

With the above reparameterization, if the collision, collision-free, and goal-reaching control sets in $\hat{\mathcal{U}}^k_{w\neq0,\frac{\pi}{2}}$ for $k = 1$, 2, 3, and 4 are semi-algebraic, then so are $\mathcal{U}^1_{obs,w\neq0}$, $\mathcal{U}^1_{free,w\neq0}$, and $\mathcal{U}^1_{goal,w\neq0}$ according to (10) and (11).

The collision control set in $\hat{\mathcal{U}}^1_{w\neq0,\frac{\pi}{2}} = \bar{\mathcal{U}}_{w\neq0,\frac{\pi}{2}}$ is

$$\begin{array}{c} \mathcal{U}^1_{obs,w\neq0,t_1} = \{(z_1,z_2,\eta) \mid z_1^2 + z_2^2 = 1, z_1, z_2 \in (0,1), \\ \exists\ \tau_1 \in [z_1, 1), \tau_2 \in (0, z_2], \tilde{x}(\tau_1,\tau_2,\eta) \in \mathcal{C}_{obs}\}. \end{array} \tag{34}$$

It is easy to check that set (34) is semi-algebraic because $\tilde{x}(\tau_1,\tau_2,\eta)$ is a closed-form polynomial function (see (24)). Similarly, the collision-free control set $\mathcal{U}^1_{free,w\neq0,t_1}$ and goal-reaching set $\mathcal{U}^1_{goal,w\neq0,t_1}$ are also semi-algebraic.

According to (7), the collision control set in $\hat{\mathcal{U}}^2_{w\neq0,\frac{\pi}{2}}$ is

$$\begin{array}{c} \mathcal{U}^1_{obs,w\neq0,t_1,t_2} = \{(k_1,k_2,z_1,z_2,\eta) \mid z_1^2 + z_2^2 = 1, \\ z_1, z_2 \in (0,1), (k_1,k_2,\eta) \in \mathcal{U}^1_{free,w\neq0,t_1} \exists\ \tau_1 \in [z_1,1), \\ \tau_2 \in (0,z_2], \tilde{x}_{k_1,k_2}(\tau_1,\tau_2,\eta) \in \mathcal{C}_{obs}\}, \end{array} \tag{35}$$

in which $\tilde{x}_{k_1,k_2}(\tau_1,\tau_2,\eta)$ denotes the trajectory under the control with duration t_2 from the final state of the control with duration t_1. Because $\tilde{x}_{k_1,k_2}(\tau_1,\tau_2,\eta)$ is a polynomial function and $\mathcal{U}^1_{free,w\neq0,t_1}$ is semi-algebraic, $\mathcal{U}^1_{obs,w\neq0,t_1,t_2}$ is semi-algebraic, and therefore, so are $\mathcal{U}^1_{free,w\neq0,t_1,t_2}$ and $\mathcal{U}^1_{goal,w\neq0,t_1,t_2}$. By induction, we can show that $\mathcal{U}^1_{obs,w\neq0}$, $\mathcal{U}^1_{free,w\neq0}$, and $\mathcal{U}^1_{goal,w\neq0}$ are semi-algebraic.

Construction of semi-algebraic $\mathcal{U}^k_{obs}$, $\mathcal{U}^k_{free}$, and $\mathcal{U}^k_{goal}$ Because $\mathcal{U}^k_{obs}$, $\mathcal{U}^k_{free}$, and $\mathcal{U}^k_{goal}$ are constructed from $\bar{\mathcal{U}}$ and semi-algebraic $\mathcal{U}^{k-1}_{free}$, it is easy to check that these sets are all semi-algebraic with the above techniques. Therefore, the MPD problem for the Dubin's car is decidable. ■

Corollary 1: The above MPD problem for the Dubin's car with its turning rate in a continuous set $[-1, 1]$ is also decidable by the IDA.

Proof: The above proof does not require that the turing rate has to be chosen from a discrete set. ■

Corollary 2: The MPD problem for the Dubin's car with a point body moving in a polygonal environment (without a bound on the number of control switches) is decidable by the IDA in Fig. 2.

Proof: As shown in [9], [10], the number of switches in all solutions is bounded by kn_v^2 for some positive constant k, in which n_v is the number of vertices of the polygonal environment. Therefore, the problem is decidable by the proposed IDA if we choose the control space to include piecewise-constant controls with up to kn_v^2 switches. ■

V. Conclusion and Future Research

In this paper, we attempt to answer the decidability question for motion planning with differential constraints (MPD). To this end we proposed a control space abstraction for MPD problems and developed an incremental decision algorithm that can be used to answer this question. We show that this decidability question can be answered for a class of linear systems and for the Dubin's car with a polygonal rigid body moving in a polygonal environment with a fixed (finite) number of curvature discontinuities. While the decision algorithm involves the solution to a quantifier elimination problem, which is computational expensive, our main goal is to provide a theoretical framework to address the decidability question.

There are many directions for future research. First, complexity of the IDA algorithm is worth investigating. Also, this paper only considers MPD problems for a narrow class of systems for which the control space is finite-dimensional. However, the general control space is a semigroup with an infinite number of switches, and therefore is infinite dimensional. A natural question is how to ensure that no solution will exist in the infinite-dimensional control space of a given MPD problem if there is no solution in a finite-dimensional control space. The answer to this question directly affects decidability checking for more general MPD problems.

Another area of investigation concerns sampling-based planning algorithms, which use a finite number of sample points to approximate solutions, and have had considerable success in the last two decades. A large amount of research has been done on sampling techniques in the state space. It is expected that research on sampling techniques in the control space will also improve design and analysis of MPD algorithms [1], [4], [14], [17], [18]. For example, the structure of the reachable set of quantized control systems under a given discrete set of sample controls [1] leads to an efficient MPD algorithm in [17]. Research on the relationship between control space sampling dispersion and trajectory variation leads to sufficient conditions for resolution complete MPD algorithms [4], [14].

As we compare Theorems 1 and 2, requirements for semi-algebraic sets in Theorem 2 come directly from the limitation of the existing computational tool. To be able to decide more general MPD problems using the IDA in Fig. 2, new computational tools are necessary.

VI. Acknowledgments

We thank Jean-Paul Laumond for helpful discussions on planning problems of the Dubin's car. We also thank Steve Lindemann, Steven LaValle, and anonymous reviewers for helpful comments. We gratefully acknowledge support from NSF grants CCR02-05336, CNS-0410514 and IIS-0427313 and ONR grant FA8650-04-C-7133.

References

[1] A. Bicchi, A. Marigo, and B. Piccoli. On the reachability of quantized control systems. *IEEE Transactions on Automatic Control*, 47(4):546–563, April 2002.

[2] J. Canny, A. Rege, and J. Reif. An exact algorithm for kinodynamic planning in the plane. *Discrete and Computational Geometry*, 6:461–484, 1991.

[3] J. F. Canny. *The Complexity of Robot Motion Planning*. MIT Press, Cambridge, MA, 1988.

[4] P. Cheng. *Sampling-Based Motion Planning with Differential Constraints*. PhD thesis, University of Illinois, Urbana, IL, August 2005.

[5] P. Cheng and V. Kumar. Sampling-based falsification and verification of controllers for continuous dynamic systems. In S. Akella, N. Amato, W. Huang, and B. Misha, editors, *Workshop on Algorithmic Foundations of Robotics VII*, 2006.

[6] 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*. MIT Press, Cambridge, MA, 2005.

[7] B. R. Donald, P. G. Xavier, J. Canny, and J. Reif. Kinodynamic planning. *Journal of the ACM*, 40:1048–66, November 1993.

[8] L. E. Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[9] S. Fortune and G. Wilfong. Planning constrained motion. In *Proceedings ACM Symposium on Theory of Computing*, pages 445–459, 1988.

[10] P. Jacobs and J. Canny. Planning smooth paths for mobile robots. In *Proceedings IEEE International Conference on Robotics & Automation*, pages 2–7, 1989.

[11] G. Lafferriere, G. Pappas, G. Schneider, and S. Yovine. Parameter synthesis in robot motion planning using symbolic reachability computations. In *Proceedings of the 8th IEEE Mediterranean Conference on Control and Automation*, Rio, Patras, Greece, 2000.

[12] G. Lafferriere, G. Pappas, and S. Yovine. A new class of decidable hybrid systems. In F. Vaandrager and J. van Schuppen, editors, *Hybrid Systems: Computation and Control*. Springer-Verlag, Berlin, 1999.

[13] J.-P. Laumond. Trajectories for mobile robots with kinematic and environment constraints. In *Proceedings International Conference on Intelligent Autonomous Systems*, pages 346–354, 1986.

[14] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at http://planning.cs.uiuc.edu/.

[15] T. Lozano-Pérez. Spatial planning: A configuration space approach. *IEEE Transactions on Computing*, C-32(2):108–120, 1983.

[16] C. O'Dunlaing. Motion planning with inertial constraints. *Algorithmica*, 2(4):431–475, 1987.

[17] S. Pancanti, L. Pallottino, D. Salvadorini, and A. Bicchi. Motion planning through symbols and lattices. In *Proceedings IEEE International Conference on Robotics & Automation*, pages 3914–3919, 2004.

[18] S. Ramamoorthy, R. Rajagopal, Q. Ruan, and L. Wenzel. Low-discrepancy curves and efficient coverage of space. In S. Akella, N. Amato, W. Huang, and B. Misha, editors, *Workshop on Algorithmic Foundations of Robotics VII*, 2006.

[19] J. T. Schwartz and M. Sharir. On the Piano Movers' Problem: II. General techniques for computing topological properties of algebraic manifolds. *Communications on Pure and Applied Mathematics*, 36:345–398, 1983.

[20] S. Sekhavat and J.-P. Laumond. Topological property for collision-free nonholonomic motion planning: The case of sinusoidal inputs for chained-form systems. *IEEE Transactions on Robotics & Automation*, 14(5):671–680, 1998.

[21] A. Tarski. *A decision method for elementary algebra and geometry*. University of California Press, 1951.

[22] V. Weispfenning. Semilinear motion planning in redlog. Technical Report MIP-9906, Universitat Passau, Germany, 1999.