PARAMETER SYNTHESIS IN ROBOT MOTION PLANNING USING SYMBOLIC REACHABILITY COMPUTATIONS

G. LAFFERRIERE†,G. PAPPAS‡[1],G. SCHNEIDER*[2],S. YOVINE*[2]

†Department of Mathematical Sciences, Portland State University, P.O. Box 751, Portland, OR 97207-0751, gerardo@mth.pdx.edu

‡Department of Electrical Engineering, University of Pennsylvania, 200 South 33rd Street, Philadelphia, PA 19104, pappasg@ee.upenn.edu

∗VERIMAG, Centre Equation, 2 Ave Vignate, 38610 Gières, France, gerardo@imag.fr, yovine@imag.fr

Abstract. A well known problem in robotics is the motion planning problem in the presence of static obstacles. The trajectory of the robot must satisfy a linear differential equation as well as possible input and state constraints. In this paper, we explore the use of symbolic reachability algorithms to decide whether the motion planning problem is feasible or not. In the case where it is feasible, it computes a feasible nominal input profile satisfying all system constraints. Our algorithm is based on quantifier elimination techniques in the ordered field of the reals, which have been recently applied to compute the reachable space for classes of linear hybrid systems.

Key Words. Motion planning, robotics, hybrid systems, reachable sets, quantifier elimination

1. INTRODUCTION

The robot motion planning problem asks whether a mobile robot can reach a desired final configuration from a given initial configuration while avoiding all static obstacles. During the past three decades, the motion planning problem has received, in various forms, the attention of the robotics community [13].

Similar reachability computations have been very recently the focus of much research in the emerging area of hybrid systems. Hybrid systems combine both digital and analog systems, in a way that is useful for the analysis and design of distributed, embedded control systems. Many of the motivating applications, which include automated highway systems [16] and air traffic management systems [15], are *safety critical*, and require guarantees of safe operation. Consequently, much research focuses on computing reachable sets for hybrid systems in order to ensure that these systems avoid unsafe regions of the state space. In particular, in [9, 10, 11, 12] it has been shown that the reachable set of several useful classes of hybrid systems that contain linear control systems of the form $\dot{x} = Ax + Bu$ can be characterized as a predicate in the theory of the ordered field of real numbers, yielding a decidability result for the reachability problem. A quantifier-free characterization of the reachable sets can then be obtained by quantifier elimination using state-of-the-art tools such as Redlog [4] and Qepcad [3].

In this paper, we explore the application of the above reachability results and tools for the motion planning problem. We consider point mass robots whose dynamics are described by linear integrators, have velocity and acceleration constraints, and must avoid semialgebraic, static obstacles. Furthermore, the class of allowable inputs is parametrically defined as the set of polynomials of a fixed order, whose rational coefficient can satisfy semialgebraic constraints. This results in a motion planning algorithm that is *complete*, in the sense that the algorithm will return whether the problem is solvable or not, and if it is solvable, it will return the *exact set* of parameter values for which the motion planning problem has a solution. This set of feasible parameter values can then be used to perform secondary objectives, such as minimization of path, time, or control input.

Quantifier elimination has been recently applied in the area of motion planning [14, 17]. However, these works are restricted to semilinear translational motions, that is, continuous paths consisting in finitely many translations along straight lines. In this paper, we use the

[1]Partially supported by DARPA Grant N66001-99-C-8510 and by DARPA Grant F33615-00-C-1707.

[2]Partially supported by European ESPRIT-LTR Project 26270 VHS.

recent results in [9, 10, 11, 12] to extend the framework of [17] to deal with more general motions, characterized by linear control systems, at the price of increased computational complexity.

Other applications of quantifier elimination in control theory go as back as [1], where it was used to obtain an algorithmic solution to the problem of stabilization by static output feedback. More recently, a number of researchers have used quantifier elimination in testing stability of linear systems [7], robust feedback control [6], and trajectory tracking of nonlinear control systems [8].

The structure of this paper is as follows: In Section 2, we present a simplified, two-dimensional version of the motion planning problem. At the price of complexity, all the results and computations can be lifted to higher dimensions. In Section 3, we review the reachability results of [9, 10] and apply them to obtain decidability results for classes of motion planning problems. In Section 4, we use the state-of-the-art tools REDLOG [4] and QEPCAD [3] in order to solve simple motion planning problems by performing reachability computations. Section 5 concludes with areas for future research.

2. PROBLEM STATEMENT

A mobile robot is located at an initial position (x_0, y_0) with initial velocity (v_{x_0}, v_{y_0}) and it is desired that it reaches the target position (x_F, y_F) with velocity (v_{x_F}, v_{y_F}). The motion of the robot is modeled by the following two chains of integrators

$$\begin{aligned} \dot{x} &= v_x \\ \dot{v}_x &= a_x \\ \dot{y} &= v_y \\ \dot{v}_y &= a_y \end{aligned} \tag{1}$$

where $a_x, a_y \in \mathbb{R}$ is the acceleration control input in the x and y direction respectively. Furthermore, due to velocity and acceleration constraints, the trajectories generated by the above equations must satisfy constraints of the form

$$\begin{aligned} & v_x^{min} \leq v_x \leq v_x^{max} \\ & v_y^{min} \leq v_y \leq v_y^{max} \\ & a_x^{min} \leq a_x \leq a_x^{max} \\ & a_y^{min} \leq a_y \leq a_y^{max} \end{aligned} \tag{2}$$

The inputs are typically constrained to belong in a class of functions. We will constrain the inputs $a_x(t)$, $a_y(t)$ to be polynomial functions of time. Splines, special concatenations of polynomial functions, have been extremely popular in the robotics community. For simplicity, consider second order polynomial functions of time. Thus

$$\begin{aligned} a_x(t) &= a_2 t^2 + a_1 t + a_0 \\ a_y(t) &= b_2 t^2 + b_1 t + b_0 \end{aligned} \tag{3}$$

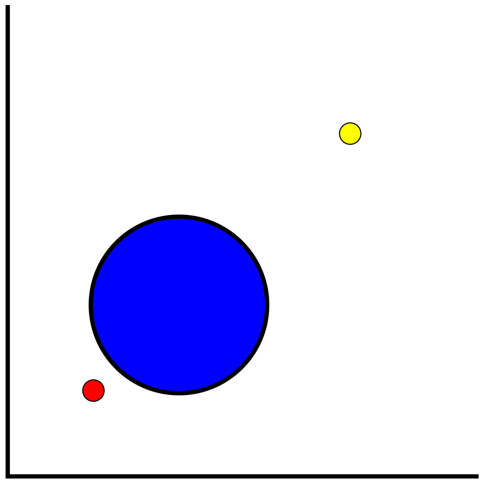

Figure 1. Motion Planning Problem

where a_i and b_i, $i = 0, 1, 2$, are control parameters.

The environment contains several obstacles O_i which must be avoided by the robot. The obstacles can have one of the following forms

$$\begin{aligned} O_i &= \{(x, y) \in \mathbb{R}^2 \mid p_i(x, y) \leq 0\} \\ O_i &= \{(x, y) \in \mathbb{R}^2 \mid p_i(x, y) \geq 0\} \\ O_i &= \{(x, y) \in \mathbb{R}^2 \mid p_i(x, y) = 0\} \end{aligned} \tag{4}$$

where $p_i(x, y)$ is some polynomial in x, y with rational coefficients. We are now ready to formally state the motion planning problem in the presence of obstacles.

Problem 1 (Motion planning) *Does there exist an instantiation of all the parameters a_i and b_i, such that the corresponding trajectory of the differential equation (1) connects the initial state $(x_0, y_0, v_{x_0}, v_{y_0})$ with the final state $(x_F, y_F, v_{x_F}, v_{y_F})$ while avoiding all obstacles O_i?*

Of course, Problem 1 can be made more general in many ways. One can have motion in three or n dimensions, the order of the integrators and the polynomial inputs could be larger than two, and the obstacles could be in general semialgebraic. Such generalizations pose no conceptual difficulty in the following constructions.

3. REACHABILITY COMPUTATIONS

A large class of continuous systems, including equation (1) of the point mass robot, are modeled by linear control systems of the following form

$$\dot{x} = Ax + Bu \tag{5}$$

where $x \in \mathbb{R}^n$ is the state of the system, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ are the system matrices, and $u(t) : \mathbb{R} \longrightarrow \mathbb{R}^m$ is a piecewise continuous control input. In our setting, we will further assume that the control input u belongs in a set $\mathcal{U}$ of input functions, where

$$\mathcal{U} = \{u = [u_1, \ldots, u_n]^T \mid u_j = \sum_{l=1}^{r} b_{jl} p_l(t)$$

$$\text{and}\quad \Phi_b(b_{jl})\quad 1 \le j \le n,\ 1 \le l \le r\} \tag{6}$$

where $\Phi_b(b_{jl})$ is a first order formula that defines a semialgebraic set, and $p_l(t)$ are some basis functions (to be determined later). Therefore, $\mathcal{U}$ consists of linear combinations of these basis functions, where the *rational* coefficients of the linear combination satisfy some semialgebraic constraint.

A *family* of linear vector fields is defined as a tuple $\mathcal{F} = (A,\mathcal{U})$. Given a family $\mathcal{F}$ we say that a state y is reachable from a state x, if there exists a control input $u(t) \in \mathcal{U}$, and a $t \ge 0$ such that $y = \Phi(x,u,t)$, where $\Phi(x,u,t)$ denotes the trajectory of system (5) with input $u(t)$. Our motion planning computations rely on the solution of the following reachability problem.

Problem 2 (Reachability Problem) *Given a family $\mathcal{F} = (A,\mathcal{U})$ of linear vector fields, compute all states that are reachable from a semi-algebraic set of initial states.*

Even though there is a wealth of results in control theory regarding the reachable set from a given point, only recently have people looked at computing reachable sets of differential equations or differential inclusions starting from an initial *set* of states. The first *exact* solution to Problem 2, for certain families of linear differential equations was given in [12] based on the results of [9, 10, 11].

Theorem 1 (Decidable Reachability Computation) *Let $\mathcal{F} = (A,\mathcal{U})$ be a family of linear control vector fields with $A \in \mathbb{Q}^{n\times n}$, let the control set $\mathcal{U}$ be defined by equation (6), and let Λ denote the spectrum of the matrix A. Furthermore, assume that one of the following cases holds :*

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 $\mathbf{i}r$ with $r \in \mathbb{Q}$, and the basis functions are of the form $p_l(t) = \sin(\mu_l t)$ or $p_l(t) = \cos(\mu_l t)$, with $\mu_l \notin \Lambda$.*

Then, Reachability Problem 2 is decidable for any $\mathcal{F} = (A,\mathcal{U})$ in the above classes.

In addition to reachability of linear systems, Theorem 1 can be trivially extended to hybrid systems which contain a fixed, finite number of discrete switches. This allows us to consider reachability computations of linear control systems whose inputs are *fixed, finite concatenations* of functions belonging in $\mathcal{U}$.

For the case of nilpotent A matrices, Theorem 1 allows us to consider splines, which are special concatenations of polynomials. This is particularly relevant for motion planning computations as the robot dynamics (1) belong to this class.

4. MOTION PLANNING COMPUTATIONS

To illustrate the approach consider the example shown in Figure 1. The robot is modeled by a differential equation (1) along with the following (lack of) constraints

$$v_x^{min} = v_y^{min} = -\infty \quad v_x^{max} = v_y^{max} = +\infty$$
$$a_x^{min} = a_y^{min} = -\infty \quad a_x^{max} = a_y^{max} = +\infty$$

The initial state is $(x_0, y_0, v_{x_0}, v_{y_0}) = (1,1,0,0)$, and the final state is $(x_F, y_F, v_{x_F}, v_{y_F}) = (4,4,0,0)$. The static obstacles are

$$\begin{aligned} O_1 &= \{(x,y) \in \mathbb{R}^2 \mid x \ge 0\} \\ O_2 &= \{(x,y) \in \mathbb{R}^2 \mid y \ge 0\} \\ O_3 &= \{(x,y) \in \mathbb{R}^2 \mid (x-2)^2 + (y-2)^2 \le 1\} \end{aligned} \tag{7}$$

Assume that the control inputs have the following form,

$$\begin{aligned} a_x(t) &= a_2t^2 + a_1t + a_0 \\ a_y(t) &= b_2t^2 + b_1t + b_0 \end{aligned}$$

Higher order polynomials can also be considered at the price of complexity.

We would like to know whether there exist values for the coefficients of the input polynomials that will steer the robot from the initial state to the final state while respecting all system dynamics and avoiding all obstacles. After integrating the control input, the robot state trajectories will have the following form

$$\begin{aligned} v_x(t) &= \psi_x(t) = \frac{1}{3}a_2t^3 + \frac{1}{2}a_1t^2 + a_0t + v_{x_0} \\ v_y(t) &= \psi_y(t) = \frac{1}{3}b_2t^3 + \frac{1}{2}b_1t^2 + b_0t + v_{y_0} \\ x(t) &= \phi_x(t) = \frac{1}{12}a_2t^4 + \frac{1}{6}a_1t^3 + \frac{1}{2}a_0t^2 + v_{x_0}t + x_0 \\ y(t) &= \phi_y(t) = \frac{1}{12}b_2t^4 + \frac{1}{6}b_1t^3 + \frac{1}{2}b_0t^2 + v_{y_0}t + y_0 \end{aligned}$$

Let I and F be the predicates characterizing the initial and final points

$$\begin{aligned} I &= (x_0 = y_0 = 1) \wedge (v_{x_0} = v_{y_0} = 0) \\ F &= (x = y = 4) \wedge (v_x = v_y = 0) \end{aligned}$$

For simplicity, let us first consider the problem without obstacles. We define the predicate $Post_I$ characterizing *all the values of the parameters* that correspond to trajectories starting at the initial point

$$\begin{aligned} &Post_I(a_0,a_1,a_2,b_0,b_1,b_2,x,y,v_x,v_y) = \\ &\quad \exists t, x_0, y_0, v_{x_0}, v_{y_0} \ : \\ &\qquad t \ge 0 \wedge x = \phi_x(t) \wedge y = \phi_y(t) \\ &\qquad \wedge\ v_x = \psi_x(t) \wedge v_y = \psi_y(t) \\ &\qquad \wedge\ I(x_0, y_0, v_{x_0}, v_{y_0}) \end{aligned} \tag{8}$$

After eliminating the existentially quantified variables

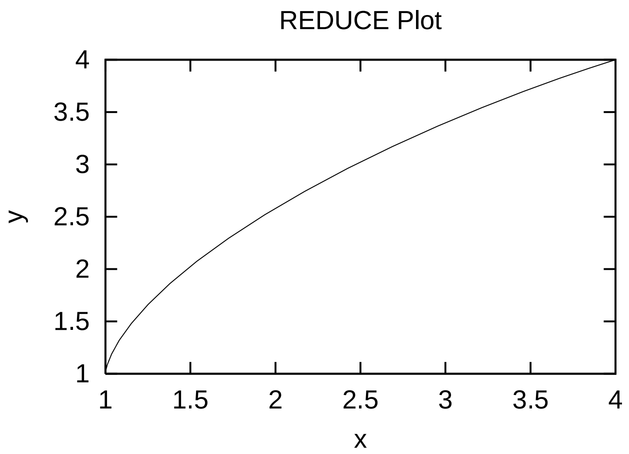

Figure 2. Parameter Synthesis without Obstacles

t, x_0, y_0, v_{x_0} and v_{y_0}, $Post_I$ is a quantifier free formula in the coefficients and the final position. We can decide whether the final configuration can be reached by simply checking whether

$$\begin{array}{l} \exists a_0, a_1, a_2, b_0, b_1, b_2, x, y, v_x, v_y \ : \\ \quad F(x, y, v_x, v_y) \\ \quad \wedge \ Post_I(a_0, a_1, a_2, b_0, b_1, b_2, x, y, v_x, v_y) \end{array} \tag{9}$$

is true by eliminating the variables. This can be done using the generic quantifier elimination with answer of REDLOG, which reduces the formula to true, yielding the following possible instantiation of the eliminated variables

$$\begin{array}{rcl} a_0 & = & -inf \\ a_1 & = & (2inf\sqrt{inf})/3 \\ a_2 & = & (-inf^2)/12 \\ b_0 & = & (15eps\sqrt{inf} - \sqrt{5}inf^2 - 5inf^2)/(15inf) \\ b_1 & = & (\sqrt{inf}\sqrt{5}inf - 15eps)/15 \\ b_2 & = & (30\sqrt{inf}eps - 2\sqrt{5}inf^2 + 5inf^2)/180 \end{array}$$

where eps and inf are dummy variables introduces by REDLOG [5]. The fact that one can *exactly compute the set of parameter values* for which the motion planning problem is feasible, is one of the main advantages of using symbolic reachability computations. In the case above, REDLOG returns this feasible set of parameters parameterized by eps and inf which are free parameters to explore in secondary objectives. For example, instantiating eps to 0 and inf to 1/16 we obtain the trajectory shown in Figure 2.

In the presence of obstacles, we define the predicate $Post_I$ characterizing all the values of the parameters that correspond to trajectories starting at the initial point and avoiding the obstacles as follows

$$\begin{array}{l} Post_I(a_0, a_1, a_2, b_0, b_1, b_2, x, y, v_x, v_y) = \\ \quad \exists t, x_0, y_0, v_{x_0}, v_{y_0} \ : \ t \geq 0 \\ \qquad \wedge \ x = \phi_x(t) \wedge y = \phi_y(t) \\ \qquad \wedge \ v_x = \psi_x(t) \wedge v_y = \psi_y(t) \\ \qquad \wedge \ I(x_0, y_0, v_{x_0}, v_{y_0}, a_0, b_0) \\ \qquad \wedge \ \forall t' \ : \ 0 \leq t' \leq t \Longrightarrow \\ \qquad\qquad \phi_x(t') \geq 0 \wedge \phi_y(t') \geq 0 \\ \qquad\qquad \wedge \ (\phi_x(t') - 2)^2 + (\phi_y(t) - 2)^2 > 1 \end{array} \tag{10}$$

After eliminating the existentially quantified variables, $Post_I$ is a quantifier free formula in the coefficients and the final position. Again, we can decide whether the final configuration can be reached by eliminating the variables.

Even though in theory the problem can be solved, in practice this is not the case. For instance, in formula (10) there are polynomials of degree eight ($\phi_x(t')$ and $\phi_y(t')$, that are polynomials of degree four, are under a quadratic power). Unfortunately, state-of-the-art tools in quantifier elimination such as QEPCAD [3] and REDLOG [4] are unable to handle such a predicate. In particular, REDLOG can feasibly handle more variables than QEPCAD, but QEPCAD can handle higher order polynomials than REDLOG. In this computational tradeoff, one approach to try to solve the problem is to avoid the universal quantification on the intermediate positions in (10) by imposing additional constraints over the velocity. These additional constraints impose decomposing the problem to smaller subproblems of manageable complexity.

Consider again the scenario illustrated in Figure 1 and let $(x_i, y_i) = (\frac{51}{16}, \frac{3}{4})$. The idea is to try to avoid going through the obstacle by passing by the intermediate point (x_i, y_i), thus decomposing the trajectory in two segments, the first one with x increasing and y decreasing, the second one with both x and y increasing. Let

$$\begin{array}{rcl} V^{++}(t) & = & \psi_x(t) \geq 0 \wedge \psi_y(t) \geq 0 \\ V^{+-}(t) & = & \psi_x(t) \geq 0 \wedge \psi_y(t) \leq 0 \\ V^{-+}(t) & = & \psi_x(t) \leq 0 \wedge \psi_y(t) \geq 0 \\ V^{--}(t) & = & \psi_x(t) \leq 0 \wedge \psi_y(t) \leq 0 \end{array}$$

For $p \in \{++, +-, -+, --\}$, we define

$$\begin{array}{l} Post_I^p(a_0, a_1, a_2, b_0, b_1, b_2, x, y, v_x, v_y) = \\ \quad \exists t, x_0, y_0, v_{x_0}, v_{y_0} \ : \ t \geq 0 \\ \qquad \wedge \ x = \phi_x(t) \wedge y = \phi_y(t) \\ \qquad \wedge \ v_x = \psi_x(t) \wedge v_y = \psi_y(t) \\ \qquad \wedge \ I(x_0, y_0, v_{x_0}, v_{y_0}, a_0, b_0) \\ \qquad \wedge \ \forall t' \ : \ 0 \leq t' \leq t \Longrightarrow V^p(t') \end{array} \tag{11}$$

Let

$$\begin{array}{rcl} I_0 & = & x_0 = y_0 = 1 \wedge v_{x_0} = v_{y_0} = a_0 = b_0 = 0 \\ I_i & = & x_i = \dfrac{51}{16} \wedge y_i = \dfrac{3}{4} \wedge v_{x_i} = v_{y_i} = a_0^i = b_0^i = 0 \\ F_0 & = & I_i \end{array}$$

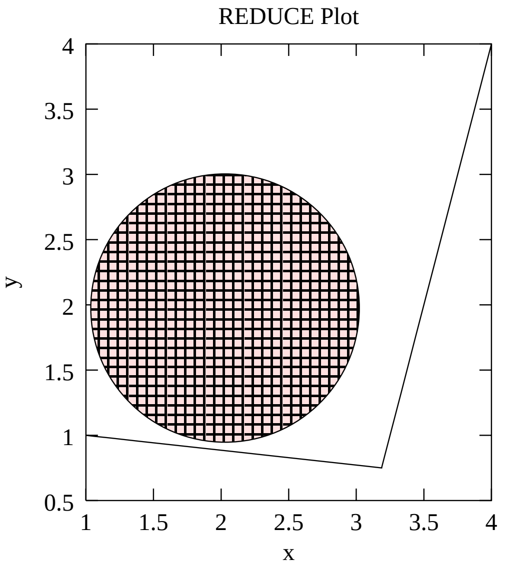

Figure 3. $(x_i, y_i) = (\frac{51}{16}, \frac{3}{4})$

$$F \quad = \quad x = y = 4 \wedge v_x = v_y = 0$$

We can check whether the final configuration can be reached from the initial one by passing through the intermediate point by simply checking whether the following predicate is true

$$\begin{array}{l} \exists a_1, a_2, b_1, b_2, a_1^i, a_2^i, b_1^i, b_2^i \;: \\ \quad Post_{I_0}(a_1, a_2, b_1, b_2) \wedge Post_{I_i}(a_1^i, a_2^i, b_1^i, b_2^i) \;: \\ \quad \wedge F(x, y, v_x, v_y) \wedge F_0(x_i, y_i, v_{x_i}, v_{y_i}) \end{array} \quad (12)$$

More generally, this leads us to consider the following problem.

Problem 3 *Let $n \in \mathbb{N}$ and $(x_i, y_i) \in \mathbb{Q}^2$ (for all $i, 0 \leq i \leq n$) be a set of points. Is there a piecewise polynomial trajectory of (1), where each segment has acceleration defined as in (3), that starts at (x_0, y_0), reaches (x_n, y_n) and passes through all the intermediate points (x_i, y_i)?*

In order to consider the practical restrictions of the tools used, for each segment we have the following assumptions:

- Initial and final speed are set to 0.
- Initial acceleration is set to 0.
- The absolute value of the acceleration is bounded by 1.

With these restrictions we have that the *Post* operator for each segment is

$$\begin{array}{l} Post_I^p(a_1, a_2, b_1, b_2, x, y) = \\ \quad \exists t, x_0, y_0 \;:\; t \geq 0 \\ \quad\quad \wedge\; x = \phi_x(t) \wedge y = \phi_y(t) \\ \quad\quad \wedge\; 0 = \psi_x(t) \wedge 0 = \psi_y(t) \\ \quad\quad \wedge\; \forall t' \;:\; 0 \leq t' \leq t \Longrightarrow V^p(t') \\ \quad\quad\quad\quad \wedge\; -1 \leq a_x(t') \leq 1 \wedge -1 \leq a_y(t') \leq 1 \end{array}$$

We use the generic quantifier elimination of REDLOG to eliminate a_1, a_2, t, x and y. The result is a predicate Q^p on the parameters b_1, b_2 and the initial conditions which cannot be further reduced by REDLOG, with the additional condition $b_1 \neq 0 \wedge b_2 \neq 0$. We use QEPCAD to prove that is equivalent to true. The answers obtained are the following.
For $p = ++$

$$\begin{array}{rcl} a_1 & = & (-4b_2)/(3b_1) \\ a_2 & = & (-4b_2 2)/(3b_1^2) \\ t & = & (-3b_1)/(2b_2) \\ x & = & (3b_1^2 + 16b_2^2 x_0)/(16b_2^2) \\ y & = & (-9b_1^4 + 64b_2^3 y_0)/(64b_2^3) \end{array}$$

For $p = -+$

$$\begin{array}{rcl} a_1 & = & (4b_2)/(3b_1) \\ a_2 & = & (4b_2^2)/(3b_1^2) \\ t & = & (-3b_1)/(2b_2) \\ x & = & (-3b_1^2 + 16b_2^2 x_0)/(16b_2^2) \\ y & = & (-9b_1^4 + 64b_2^3 y_0)/(64b_2^3) \end{array}$$

For $p = +-$

$$\begin{array}{rcl} a_1 & = & -b_1 \\ a_2 & = & -b_2 \\ t & = & (-3b_1)/(2b_2) \\ x & = & (9b_1^4 + 64b_2^3 x_0)/(64b_2^3) \\ y & = & (-9b_1^4 + 64b_2^3 y_0)/(64b_2^3) \end{array}$$

For $p = --$

$$\begin{array}{rcl} a_1 & = & b_1 \\ a_2 & = & b_2 \\ t & = & (-3b_1)/(2b_2) \\ x & = & (-9b_1^4 + 64b_2^3 x_0)/(64b_2^3) \\ y & = & (-9b_1^4 + 64b_2^3 y_0)/(64b_2^3) \end{array}$$

We have therefore reduced the problem to finding the solutions of the system of nonlinear equations in b_1 and b_2, and then checking that these solutions satisfy the predicate Q^p. Actually, the problem can be further simplified by setting a value for t and putting one of the parameters in terms of the other, yielding a system of nonlinear equations on a single variable.

Consider the following example. Let $(x_0, y_0) = (1, 1)$, $(x_1, y_1) = (\frac{3}{2}, \frac{1}{2})$, $(x_2, y_2) = (\frac{19}{12}, 2)$, $(x_3, y_3) = (\frac{3}{2}, 4)$. This yields the following equations.

1. From (x_0, y_0) to (x_1, y_1) with $p = +-$:

   $$\begin{array}{rcl} 3/2 & = & (9b_1^4 + 64b_2^3)/(64b_2^3) \\ 1/2 & = & (-9b_1^4 + 64b_2^3)/(64b_2^3) \end{array}$$

   By setting $t = 1$ we get $b_2 = -\frac{3}{2}b_1$. Solving the equations for b_1 gives $b_1 = -12$. By replacing in the other equations we get $a_1 = 12$, $a_2 = -18$, and $b_2 = 18$.

2. From (x_1, y_1) to (x_2, y_2) with $p = ++$:

$$\begin{aligned} 19/12 &= (3 * b21^2 + 24 * b22^2)/(16 * b22^2) \\ 2 &= (-9 * b21^4 + 32 * b22^3)/(64 * b22^3) \end{aligned}$$

Solving in the same way as before we obtain $a_1 = 2$, $a_2 = -3$, $b_1 = 36$, and $b_2 = -54$.

3. From (x_2, y_2) to (x_3, y_3) with $p = -+$:

$$\begin{aligned} 3/2 &= (-9b_1^2 + 76b_2^2)/(48b_2^2) \\ 4 &= (-9b_1^4 + 128b_2^3)/(64b_2^3) \end{aligned}$$

Solving in the same way as before we obtain $a_1 = -2$, $a_2 = 3$, $b_1 = 48$, and $b_2 = -72$.

5. CONCLUSIONS

In this paper, we considered the application of symbolic reachability computations to the motion planning problem. Advantages of this approach include the ability to compute the set of parameter values for which the motion planning problem is feasible. Such set computations are based on quantifier elimination techniques which have been implemented in the computer algebra tools REDLOG and QEPCAD. Unfortunately, the complexity of quantifier elimination forces the application of the method to problems of small size. Much faster quantifier elimination methods have been discovered recently [2]. Even though such algorithms can be of great use for exact reachability computations, they have not been implemented yet. Approximate reachability computations which trade precision for complexity should be pursued further.

References

[1] B.D.O Anderson, N.K. Bose, and E.I. Jury. Output feedback stabilization and related problems - solution via decision methods. *IEEE Transactions on Automatic Control*, 20(1):53–65, 1975.

[2] S. Basu, R. Pollack, and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM*, 43(6):1002 – 1046, 1996.

[3] G.E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, 12:299–328, September 1991.

[4] A. Dolzman and T. Sturm. REDLOG : Computer algebra meets computer logic. *ACM SIGSAM Bulletin*, 31(2):2–9, June 1997.

[5] A. Dolzmann and T. Sturm. Redlog user manual, edition 2.0, for redlog version 2.0. Technical Report MIP-9905, Universität Passau, Germany, April 1999.

[6] P. Dorato, W. Yang, and C. Abdallah. Robust multi-objective feedback design by quantifier elimination. *Journal of Symbolic Computation*, 24(2):153–159, August 1997.

[7] H. Hong, R. Liska, and S. Steinberg. Testing stability by quantifier elimination. *Journal of Symbolic Computation*, 24(2):161–187, August 1997.

[8] Mats Jirstrand. Nonlinear control system design by quantifier elimination. *Journal of Symbolic Computation*, 24(2):137–152, Aug 1997.

[9] G. Lafferriere, G. J. Pappas, and S. Yovine. A new class of decidable hybrid systems. In *Hybrid Systems : Computation and Control*, volume 1569 of *Lecture Notes in Computer Science*, pages 137–151. Springer Verlag, 1999.

[10] G. Lafferriere, G. J. Pappas, and S. Yovine. Reachability computation for linear hybrid systems. In *Proceedings of the 14th IFAC World Congress*, volume E, pages 7–12, Beijing, P.R. China, July 1999.

[11] Gerardo Lafferriere, George J. Pappas, and Shankar Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals, and Systems*, 13(1):1–21, 2000.

[12] Gerardo Lafferriere, George J. Pappas, and Sergio Yovine. Symbolic reachability computations for families of linear vector fields. *Journal of Symbolic Computation*. Submitted.

[13] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Press, Boston, MA, 1991.

[14] S. McCallum. Partial solution to path finding problems using the cad method. Electronic Proceedings of the IMACS ACA 1995, http://math.unm.edu/ACA/1995.html, 1995.

[15] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management : A study in muti-agent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, April 1998.

[16] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, 38(2):195–207, 1993.

[17] V. Weispfenning. Semilinear motion planning in REDLOG. Technical Report MIP-9906, Universität Passau, Germany, May 1999.