2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Optimal Control of Robotic Systems with Logical Constraints: Application to UAV Path Planning

Shangming Wei, Miloš Žefran, and Raymond A. DeCarlo

Abstract— **Optimal control of robotic systems with logical constraints is an instance of a hybrid optimal control problem. It has been traditionally treated as a mixed-integer programming problem (MIP) which is of combinatorial complexity. This paper proposes a new approach for transforming logical constraints into inequality and equality constraints involving only continuous variables. In this way the hybrid optimal control problem is converted to a smooth optimal control problem that can in turn be solved using traditional nonlinear programming methods, thereby dramatically reducing the computational complexity of finding the solution. We illustrate the techniques by solving an optimal path planning problem for multiple unmanned aerial vehicles (UAVs) with collision avoidance. Simulation results are given to show the effectiveness of the approach.**

I. Introduction

Because many robotic systems are characterized by a combination of continuous and discrete (symbolic) variables, they can be modeled as hybrid systems. Differential equations typically describe the lower level behavior, whereas logical/Boolean formulas describe the effect of valves, switches, gears, and logical controllers, as well as reasoning and planning operations, constraint enforcement, and conflict resolution that occur at the higher or supervisory level.

A number of frameworks have been proposed to model and control hybrid systems described by interacting continuous dynamics and logic rules (see [1]–[4] among many other) while other researchers consider the optimal control of hybrid systems (e.g. [5]–[7]). A common approach for solving hybrid optimal control problems is to formulate them as a mixed integer programming (MIP) problems. For instance, in [7], propositional logic is transformed into linear inequalities involving real and integer variables. Then the optimization of the resulting mixed logical dynamical (MLD) system is solved through mixed integer quadratic programming (MIQP). Although the performance of this approach is satisfactory, in general it does not scale well as the solution time increases exponentially with the number of integer variables.

The authors of [8]–[10] converted collision avoidance constraints which are *or*-constraints into mixed integer linear constraints. But the resulting mixed integer linear programming (MILP) still suffers from combinatorial complexity. Although there has been a dramatic increase in the quantity and quality of software designed to solve MIPs or MILPs in recent years, the fundamental limitations of the methodology remain.

An extension of the embedding technique of [11] was developed in [12] to transform hybrid optimal control problems into traditional smooth optimal control problems. The solution methodology relies on traditional nonlinear programming techniques such as sequential quadratic programming (SQP); thus the computational complexity is dramatically reduced over other approaches. The approach is applicable to hybrid systems that exhibit autonomous and controlled switches, both resulting in discontinuous jumps in the vector fields governing the evolution of the continuous dynamics of the system. In other words, the discrete aspect of the system arises only from switches in the dynamic equations. However, there is no discussion in [12] about how to deal with the logical/discrete components which also appear as constraints.

This paper presents a new method to numerically solve hybrid optimal control problems with logical constraints. Specifically, we transform logical expressions in the constraints into inequality and equality constraints involving only continuous variables. The resulting optimal control problems can then be solved utilizing standard nonlinear programming methods such as sequential quadratic programming (SQP); MIP methods are not necessary. The approach is demonstrated

The first two authors have been supported in part by NSF grants IIS-0093581, CCR-0330342, and CMS-0600658.

S. Wei and M. Žefran are with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA. Email: {swei3,mzefran}@uic.edu.

R. A. DeCarlo is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA. Email: decarlo@ecn.purdue.edu.

on the application to optimal path planning problem of multiple autonomous vehicles with collision avoidance. Model predictive control (MPC) solutions to the example are computed and simulation results are presented to show the effectiveness of the approach.

II. Methodology

The robotic systems considered in this paper can be modeled by a set of differential equations:

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t_0) = x_0 \tag{1}$$

where $x(t) \in \mathbb{R}^p$ is the system's state at time t, $u(t) \in \mathbb{R}^r$ is the control input constrained to a convex and compact set Ω, and $f : \mathbb{R}^p \times \mathbb{R}^r \to \mathbb{R}^p$ is a real vector-valued function of class $\mathcal{C}^1$. We assume that $u(t)$ is a measurable function on the interval $[t_0, t_f]$ that has to be chosen so that the appropriate boundary conditions are satisfied. Formally, the initial and terminal constraints are described by requirements $(t_0, x(t_0)) \in \mathcal{T}_0 \times \mathcal{B}_0$ and $(t_f, x(t_f)) \in \mathcal{T}_f \times \mathcal{B}_f$, where the endpoint constraint set $\mathcal{B} = \mathcal{T}_0 \times \mathcal{B}_0 \times \mathcal{T}_f \times \mathcal{B}_f$ is contained in a compact set in $\mathbb{R}^{2p+2}$.

As mentioned in Section I, the focus of this paper are systems that must also satisfy a set of logical constraints that are frequently an integral part of a robotic system. Logical constraints are utilized to represent symbolic behavior such as planning and reasoning rules, obstacle avoidance, no-fly zones, safety requirements, etc. Since every Boolean expression can be converted into conjunctive normal form (CNF) [13], any logical constraint can in general be expressed as a CNF formula:

$$D_1 \wedge D_2 \wedge \ldots \wedge D_n \tag{2}$$

where $D_i = P_i^1 \vee P_i^2 \vee \ldots \vee P_i^{m_i}$ and P_i^j is either X_i^j or $\neg X_i^j$, with X_i^j being a literal that is either True or False. We use X_i^j to represent statements such as "position $x \leq 2$" or "velocity in y direction $v_y \geq 0$". So P_i^j is of the form

$$P_i^j \triangleq [g_i^j(x(t)) \leq 0] \tag{3}$$

where $g_i^j : \mathbb{R}^p \to \mathbb{R}$ is a $\mathcal{C}^1$ function.

We define the cost functional of the system

$$J = \Phi(t_0, x_0, t_f, x_f) + \int_{t_0}^{t_f} L(x(t), u(t))dt \tag{4}$$

where Φ is a real-valued $\mathcal{C}^1$ function defined on a neighborhood of $\mathcal{B}$, and $L : \mathbb{R}^p \times \mathbb{R}^r \to \mathbb{R}$ is of class $\mathcal{C}^1$.

The optimal control problem with logical constraints is defined as:

$$\min_u J$$

subject to the constraints: (i) $x(\cdot)$ satisfies Eq. (1); (ii) $(t_0, x(t_0), t_f, x(t_f)) \in \mathcal{B}$; (iii) for each $t \in [t_0, t_f]$, $u(t) \in \Omega$; (iv) the logical formula Eq. (2) is satisfied.

To solve this problem, we need to transform the constraint (iv) into equality or inequality constraints. Furthermore, we don't want to use any Boolean variables in order to avoid combinatorial complexity of integer programming. Satisfying the conjunction is easy as $D_1 \wedge D_2 \wedge \ldots \wedge D_n =$ is simply equivalent to $\forall i \in [1, \ldots, n] : \ D_i$. Therefore, we have the following constraints:

$$\forall i \in [1, \ldots, n] : \quad P_i^1 \vee P_i^2 \vee \ldots \vee P_i^{m_i}. \tag{5}$$

To translate the disjunctions into inequality constraints, we associate with each P_i^j a continuous variable $\alpha_i^j \in [0, 1]$. Then we have the following equivalent expression of Eq. (5)

$$\begin{aligned} \forall i \in [1, \ldots, n] : \ & \alpha_i^j \cdot g_i^j(x(t)) \leq 0, \\ \text{and } \ & 0 \leq \alpha_i^j \leq 1, \\ \text{and } \ & \sum_j \alpha_i^j = 1. \end{aligned} \tag{6}$$

Note that when $\alpha_i^j = 0$, the constraint $g_i^j(x(t)) \leq 0$ is not satisfied, and when $0 < \alpha_i^j \leq 1$, $\alpha_i^j \cdot g_i^j(x(t)) \leq 0$ is the same as $g_i^j(x(t)) \leq 0$, which means that the constraint is enforced. The last constraint of Eq. (6) makes sure that at least one of P_i^j holds.

After replacing the logical constraints with the inequality and equality constraints in Eq. (6), we obtain the following optimal control problem:

$$\min_u J$$

subject to the constraints: (i) $x(\cdot)$ satisfies Eq. (1); (ii) $(t_0, x(t_0), t_f, x(t_f)) \in \mathcal{B}$; (iii) for each $t \in [t_0, t_f]$, $u(t) \in \Omega$; (iv) for $i = 1, \ldots, n$ and $j = 1, \ldots, m_i$, $\alpha_i^j \in [0, 1]$; (v) for $i = 1, \ldots, n$, $\sum_j \alpha_i^j = 1$; (vi) for each $t \in [t_0, t_f]$, $i = 1, \ldots, n$, and $j = 1, \ldots, m_i$, $\alpha_i^j \cdot g_i^j(x(t)) \leq 0$. We can see that the combinatorial aspect of the original problem is effectively eliminated. The resulting problem is a smooth problem which can be solved using traditional nonlinear programming techniques. Thus the overall computational complexity is dramatically reduced [14], [15].

III. MULTIPLE UAVS PATH PLANNING PROBLEM

In this section we illustrate the approach by applying it to a multiple UAVs path planning example. The problem studied here is to find an energy-optimal path for a single or a group of autonomous vehicles on a two-dimensional plane. Each vehicle is required to move from an initial state to a target state without colliding with the obstacles or other vehicles. In order to establish the effectiveness of our methodology, the problem is the same as the one addressed in [8]. Logical constraints are the consequence of the obstacle and collision avoidance requirements. Using the methodology proposed in Section II, we solve the problem utilizing traditional nonlinear programming techniques, making our approach an alternative to that of [8] where mixed integer/linear programming (MILP) is needed.

We describe the case of a single vehicle first and then discuss how the solution methodology generalizes to multiple vehicles.

A. Model

A vehicle is modeled as a point moving on the horizontal plane. We choose the state vector s and the control input vector u as

$$\begin{aligned} s &= [x, y, v_x, v_y]^T, \\ u &= [a_x, a_y]^T \end{aligned}$$

where (x, y) is the position of the vehicle, v_x and v_y are its velocity in the x and y direction respectively, and a_x and a_y are its acceleration in the x and y direction respectively.

A variation of direct collocation [16] is used to numerically solve the optimal control problem. In this case, $u(t)$ and $s(t)$ are chosen from finite-dimensional spaces. Given basis functions $\{\phi^j\}_{j=0}^N$ and $\{\psi^j\}_{j=0}^M$,

$$\begin{aligned} s_i &= \sum_{j=0}^{N} \sigma_i^j \phi^j(t), \quad \sigma_i^j \in \mathbb{R}, \quad i = 1, \ldots, n, \\ u_i &= \sum_{j=0}^{M} \mu_i^j \psi^j(t), \quad \mu_i^j \in \mathbb{R}, \quad i = 1, \ldots, m. \end{aligned}$$

Since in general $s(t)$ can be nonsmooth, the basis functions $\{\phi^j\}_{j=0}^N$ are chosen to be nonsmooth. Similarly, since the control $u(t)$ can be discontinuous, the basis functions $\{\psi^j\}_{j=0}^M$ are chosen to be discontinuous. Partition the time interval $[0, T]$ into N subintervals with the endpoints $0 = t_0 < t_1 < \ldots < t_{N-1} < t_N = T$.

The state trajectory is approximated by a piecewise-linear function:

$$\begin{aligned} &\widehat{s}_i(t) = s_i(t_j) + \frac{t - t_j}{t_{j+1} - t_j} \left(s_i(t_{j+1}) - s_i(t_j) \right), \\ &t_j \le t < t_{j+1}, \quad i = 1, \ldots, n. \end{aligned}$$

This approximation corresponds to $\sigma_i^j = s_i(t_j)$ and the triangular basis functions:

$$\phi^j(t) = \begin{cases} \frac{t - t_{j-1}}{t_j - t_{j-1}} & t_{j-1} \le t < t_j, \\ \frac{t_{j+1} - t}{t_{j+1} - t_j} & t_j \le t < t_{j+1}, \\ 0 & \text{otherwise.} \end{cases}$$

The control input is chosen to be piecewise constant so that

$$\widehat{u}_i(t) = u_i(t_j), \quad t_j \le t < t_{j+1}, \quad i = 1, \ldots, m.$$

This approximation corresponds to $\mu_i^j = u_i(t_j)$ and the square basis functions:

$$\psi^j(t) = \begin{cases} 1 & t_j \le t < t_{j+1}, \\ 0 & \text{otherwise.} \end{cases}$$

Note that the above choice of basis functions implies $M = N - 1$.

The system equations are enforced at the midpoints:

$$\begin{aligned} &\dot{\widehat{s}}(t) - f(\widehat{s}(t), \widehat{u}(t)) = 0, \\ &\text{for} \;\; t = \frac{t_j + t_{j+1}}{2}, \quad j = 0, \ldots, N - 1. \end{aligned} \tag{7}$$

For our problem, this results in the following dynamic equations:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \\ v_{x,i+1} \\ v_{y,i+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & h & 0 \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ v_{x,i} \\ v_{y,i} \end{bmatrix} + \begin{bmatrix} \frac{h^2}{2} & 0 \\ 0 & \frac{h^2}{2} \\ h & 0 \\ 0 & h \end{bmatrix} \begin{bmatrix} a_{x,i} \\ a_{y,i} \end{bmatrix}$$

for $i = 0, \ldots, N - 1$. We can rewrite these equations as

$$s_{i+1} = As_i + Bu_i, \quad i = 0, \ldots, N - 1 \tag{8}$$

where $s_i = [x_i, y_i, v_{x,i}, v_{y,i}]^T$ is the state vector at t_i, and $u_i = [a_{x,i}, a_{y,i}]^T$ is the piecewise constant acceleration (input) vector on $[t_i, t_{i+1}]$.

The objective is to find a path which uses the least energy and at the end of the horizon T we want to move the vehicle as close to the target state as possible. With the chosen representation of s and u, the integral cost function is replaced with a finite sum. Therefore, the desired performance index is

$$J = \sum_{i=0}^{N-1} c_1 \left\| u_i \right\|^2 + c_2 \left\| s_N - s_f \right\|^2 \tag{9}$$

where c_1 and c_2 are constant positive weights, s_N is the state at the end of the horizon, and s_f is the target state.

B. Obstacle Avoidance

In this paper we assume that there are a set of stationary rectangular obstacles in the environment, but the approach is quite general and can handle any obstacle described by Eqs. (2)-(3). Each rectangular obstacle is described by its lower left corner (x_{min}, y_{min}) and its upper right corner (x_{max}, y_{max}). Obstacle avoidance means that at every endpoint of the subintervals, the position of the vehicle (x_i, y_i) must be in the area outside of the obstacle. This leads to the following logical constraints:

$$\begin{aligned} \forall i \in [1, \ldots, N]: \quad & x_i - x_{min} \leq 0, \\ \text{or} \quad & x_{max} - x_i \leq 0, \\ \text{or} \quad & y_i - y_{min} \leq 0, \\ \text{or} \quad & y_{max} - y_i \leq 0. \end{aligned} \tag{10}$$

The constraints are not enforced at the initial point t_0 because the initial positions are fixed. Note also that the constraints are not enforced between the endpoints of this discretized system, so small penetration into obstacles are possible. Therefore, in practice the obstacle regions should be appropriately enlarged to provide sufficient safety margin.

The techniques in Section II are used to deal with the obstacle avoidance constraints. Introduce new variables $\alpha_i^j \in [0, 1]$, for $i = 1, \ldots, N, j = 1, \ldots, 4$, that satisfy

$$\sum_{j=1}^{4} \alpha_i^j = 1. \tag{11}$$

Then the constraints (10) can be replaced by the following constraints:

$$\begin{aligned} \forall i \in [1, \ldots, N]: \quad & \alpha_i^1 (x_i - x_{min}) \leq 0, \\ \text{and} \quad & \alpha_i^2 (x_{max} - x_i) \leq 0, \\ \text{and} \quad & \alpha_i^3 (y_i - y_{min}) \leq 0, \\ \text{and} \quad & \alpha_i^4 (y_{max} - y_i) \leq 0, \\ \text{and} \quad & 0 \leq \alpha_i^j \leq 1, j = 1, 2, 3, 4, \\ \text{and} \quad & \sum_{j=1}^{4} \alpha_i^j = 1. \end{aligned} \tag{12}$$

At the i^{th} endpoint t_i, if the j^{th} constraint in Eq. (10) is not satisfied, the corresponding α_i^j equals 0, otherwise α_i^j satisfies $0 < \alpha_i^j \leq 1$. The constraint (11) ensures that at least one of the constraints in Eq. (10) is satisfied, which means that the vehicle stays outside of the rectangular obstacle.

Besides the obstacle avoidance constraints, there are other constraints such as the bounds on the control inputs:

$$\forall i \in [0, \ldots, N-1]: \quad U_{min} \leq u_i \leq U_{max},$$

where U_{min} and U_{max} are constant vectors.

The resulting optimal control problem is as follows:

$$\begin{aligned} & \min_{u_i} \sum_{i=0}^{N-1} c_1 \|u_i\|^2 + c_2 \|s_N - s_f\|^2 \\ & \text{subject to} \quad \forall i \in [1, \ldots, N]: \\ & \quad \text{(i)} \quad s_i = A s_{i-1} + B u_{i-1}, \\ & \quad \text{(ii)} \quad U_{min} \leq u_{i-1} \leq U_{max}, \\ & \quad \text{(iii)} \quad \alpha_i^1 (x_i - x_{min}) \leq 0, \\ & \quad \text{(iv)} \quad \alpha_i^2 (x_{max} - x_i) \leq 0, \\ & \quad \text{(v)} \quad \alpha_i^3 (y_i - y_{min}) \leq 0, \\ & \quad \text{(vi)} \quad \alpha_i^4 (y_{max} - y_i) \leq 0, \\ & \quad \text{(vii)} \quad 0 \leq \alpha_i^j \leq 1, \quad j = 1, \ldots, 4, \\ & \quad \text{(viii)} \quad \sum_{j=1}^{4} \alpha_i^j = 1. \end{aligned} \tag{13}$$

Note that there is no combinatorial aspect in this optimization problem. Hence it can be readily solved by a nonlinear programming solver.

C. Multiple Vehicles

The optimal control problem of Eq. (13) can be easily extended to the multi-vehicle case by modifying the cost function and the constraints to include all vehicles.

The additional constraint that needs to be dealt with in the multi-vehicle case is collision avoidance between the vehicles. In this paper we require that at each endpoint every pair of vehicles is a safe distance apart in the x or y direction, but the approach can easily be adapted to other safety constraints. Suppose thus that the positions of vehicles p and q at the i^{th} endpoint t_i are $(x_{p,i}, y_{p,i})$ and $(x_{q,i}, y_{q,i})$, respectively. Denote the safety distance in x and y directions as d_x and d_y, respectively. Then the constraints are

$$\begin{aligned} & \forall q > p, \forall i \in [1, \ldots, N]: \\ & \qquad |x_{p,i} - x_{q,i}| \geq d_x, \\ & \qquad \text{or} \quad |y_{p,i} - y_{q,i}| \geq d_y. \end{aligned} \tag{14}$$

The condition $q > p$ is to avoid duplication of the constraints. Eq. (14) is equivalent to

$$\begin{aligned}
&\forall q > p, \forall i \in [1, \dots, N]: \\
&\qquad x_{p,i} - x_{q,i} \geq d_x, \\
&\text{or } \ x_{q,i} - x_{p,i} \geq d_x, \\
&\text{or } \ y_{p,i} - y_{q,i} \geq d_y, \\
&\text{or } \ y_{q,i} - y_{p,i} \geq d_y.
\end{aligned} \tag{15}$$

These constraints can be treated in the same way as the obstacle avoidance constraints. After introducing new variables $\beta^j_{p,q,i} \in [0, 1]$, for $i = 1, \dots, N, j = 1, \dots, 4$, that satisfy

$$\sum_{j=1}^{4} \beta^j_{p,q,i} = 1,$$

we obtain the following constraints:

$$\begin{aligned}
&\forall q > p, \forall i \in [1, \dots, N]: \\
&\qquad \beta^1_{p,q,i}(x_{p,i} - x_{q,i} - d_x) \geq 0, \\
&\text{and } \ \beta^2_{p,q,i}(x_{q,i} - x_{p,i} - d_x) \geq 0, \\
&\text{and } \ \beta^3_{p,q,i}(y_{p,i} - y_{q,i} - d_y) \geq 0, \\
&\text{and } \ \beta^4_{p,q,i}(y_{q,i} - y_{p,i} - d_y) \geq 0, \\
&\text{and } \ 0 \leq \beta^j_{p,q,i} \leq 1, j = 1, 2, 3, 4, \\
&\text{and } \ \sum_{j=1}^{4} \beta^j_{p,q,i} = 1.
\end{aligned} \tag{16}$$

Again, the last constraint of (16) ensures that at least one of the constraints of (15) is satisfied. Therefore, the vehicles p and q are guaranteed to be a safe distance apart.

D. Model Predictive Control

Model predictive control (MPC) has been a popular approach for tracking control of system with constraints (see e.g. [17], [18]). The basic idea of MPC is to solve a finite horizon constrained optimal control problem online at each time step. Only the first control input of the optimal sequence is implemented, and then the optimization is repeated starting from the reached state. Because of the re-planning, the MPC approach can account for external disturbances and modeling errors.

The MPC scheme is adopted to turn the path planning approach into a control strategy. The procedure (for one vehicle case)is as follows:

1) Let h be the interval length h, N the horizon length, and let $k = 1$.
2) Solve the optimization problem (13) over the horizon $[t_{k-1}, t_{k-1} + Nh]$ with the initial state s_{k-1} and obtain the (look ahead) control sequence $\{u_k, \dots, u_k + N\}$.
3) Apply the control input u_{k-1} for the time interval $[t_{k-1}, t_{k-1} + h]$ to the system to compute s_k, the initial condition for the next iteration.
4) Let $t_k = t_{k-1} + h$, increment k and repeat steps 2 and 3 until the goal is reached.

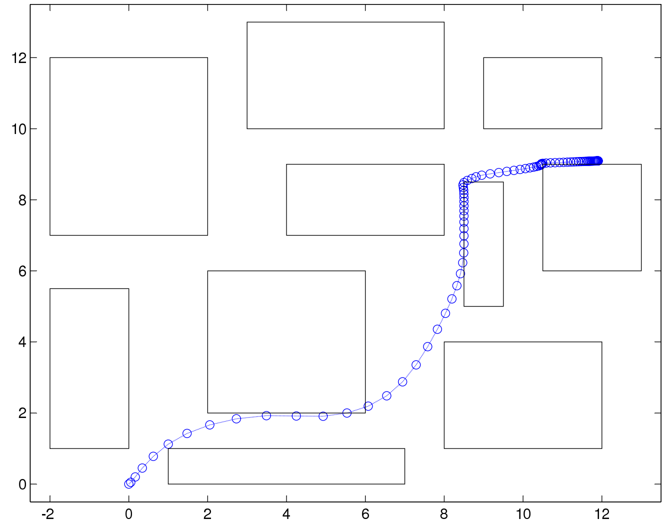

Fig. 1. Trajectory of the vehicle in Setting 1.

In the MPC approach, the arrival time of the vehicles is no longer fixed. The optimization is repeatedly applied over a moving time window of length $N \cdot h$ (h is a predefined small time step, for example, $0.1s$) until the goal is reached. Within each iteration, a short locally optimal segment of trajectory is designed to move towards the goal, but not necessarily reach it. In this way, we obtain a series of short trajectories instead of one long trajectory, and the computation load is greatly reduced.

E. Simulation Results

We implemented the path planning example using the NAG C Library from Numerical Algorithms Group. Fig. 1 and Fig. 2 show the simulation results under two settings. For both settings, there is one vehicle and the number of the obstacles is 10. In Setting 1, the initial state is $[0, 0, 0, 0]^T$, and the target state is $[12, 9.1, 0, 0]^T$. In Setting 2, the initial state is $[15, 9, 0, 0]^T$, and the target state is $[-0.5, -6, 0, 0]^T$. The discretization step h is $0.1s$ for both cases. The vehicle reaches the target state at $8.7s$ in Setting 1, and at $8.5s$ in Setting 2. The computation times for Setting 1 and Setting 2 are $937s$ and $1232s$ respectively.

From the results we can see that the performance of the proposed approach is satisfactory. All the con-

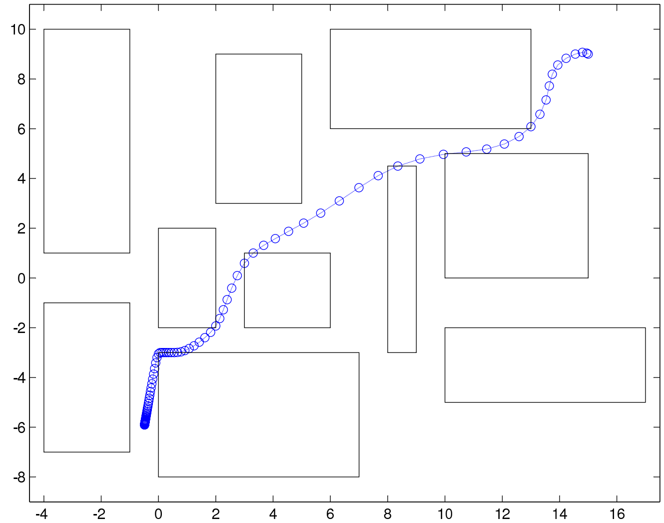

Fig. 2. Trajectory of the vehicle in Setting 2.

straints, especially the obstacle avoidance constraints, are enforced without using mixed integer programming (MIP) and our computation time is comparable to the times reported in [8]. However, there are many systems in which some items of dynamic equations, performance index or constraints are not linear so that MIP as opposed to MILP has to be used, thereby significantly increasing the computational time. For these cases, our approach has significant advantages over the approaches using MIP.

The simulation results with a single UAV demonstrate that our approach compares favorably to the approach in [8] in cases where both of them are applicable, and is in addition applicable to a much broader class of problems. However, as was the case there, due to the long computation times we did not evaluate our method in a multiple UAV scenario.

IV. Conclusions

This paper presents a new method for converting logical constraints in an optimal control problem into inequality and equality constraints involving only continuous variables. The resulting problem can be solved using traditional nonlinear programming techniques, in contrast to many current approaches whose formulations involve mixed integer programming (MIP). The proposed approach dramatically reduces the computation complexity, thus making optimal control (and model predictive control) more appealing for hybrid systems. We demonstrate the techniques by applying them to an UAV path planning example. Simulation results are quite encouraging and show the potential of the approach.

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, and T. A. Henzinger, "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, no. 1, p. 3, 1995.

[2] E. Klavins and D. E. Koditschek, "A formalism for the composition of concurrent robot behaviors," *Proceedings / 4*, pp. 3395–3402, 2000.

[3] M. S. Branicky, V. S. Borkar, and S. K. Mitter, "A unified framework for hybrid control: model and optimal control theory," *IEEE Trans. Automat. Control*, vol. 43, no. 1, pp. 31–45, 1998.

[4] J. Lygeros, D. N. Godbole, and S. Sastry, "A game-theoretic approach to hybrid system design," *Lecture notes in computer science*, vol. 1066, p. 1, 1996.

[5] C. G. Cassandras, D. L. Pepyne, and Y. Wardi, "Optimal control of a class of hybrid systems," *IEEE Trans. Automat. Control*, vol. 46, no. 3, pp. 398–415, 2001.

[6] S. Hedlund and A. Rantzer, "Optimal control of hybrid systems," in *The 38th IEEE Conference on Decision and Control (CDC), Dec 7-Dec 10 1999*, ser. Proceedings of the IEEE Conference on Decision and Control, vol. 4. Phoenix, AZ, USA: IEEE, Piscataway, NJ, USA, 1999, pp. 3972–3977.

[7] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica J. IFAC*, vol. 35, no. 3, pp. 407–427, 1999.

[8] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *European control conference; ECC 2001*, Sep 2001, pp. 2603–2608.

[9] J. Bellingham, A. Richards, and J. P. How, "Receding horizon control of autonomous aerial vehicles," *Proceedings of the American Control Conference*, vol. 5, pp. 3741–3746, 2002.

[10] L. Pallottino, E. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 3, no. 1, pp. 3–11, Mar 2002.

[11] S. C. Bengea and R. A. DeCarlo, "Optimal control of switching systems," *Automatica*, vol. 41, no. 1, pp. 11–27, 2005.

[12] S. Wei, K. Uthaichana, M. Žefran, R. A. DeCarlo, and S. Bengea, "Applications of numerical optimal control to nonlinear hybrid systems," *Nonlinear Analysis: Hybrid Systems*, vol. 1, no. 2, pp. 264–279, 2007.

[13] T. M. Cavalier, P. M. Pardalos, and A. L. Soyster, "Modeling and integer programming techniques applied to propositional calculus," *Computers and Operations Research*, vol. 17, no. 6, pp. 561–570, 1990.

[14] G. Nemhauser and L. Wolsey, *Integer and combinatorial optimization*. New York, NY, USA: Wiley-Interscience, 1988.

[15] C. Floudas, *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. New York, NY, USA: Oxford University Press, 1995.

[16] O. von Stryk, "Numerical solution of optimal control problems by direct collocation," *Optimal control (Freiburg, 1991), International Series of Numerical Mathematics*, vol. 111, pp. 129–143, 1993.

[17] E. F. Camacho and C. Bordons, *Model Predictive Control*. Springer Verlag, 2004.

[18] D. Gu and H. Hu, "Model predictive control for simultaneous robot tracking and regulation," in *International Conference on Intelligent Mechatronics and Automation*, Chengdu, China, 2004, pp. 212–217.