Hybrid Symbolic Control for Robot Motion Planning*

Ali Karimoddini and Hai Lin

***Abstract*— This paper addresses the symbolic motion planning and control of robots to meet high level specifications through hybrid supervisory control. The basic idea is to partition the motion space of robots into logically equivalent regions, based on which a bisimulation quotient transition system is derived and supervisor is synthesized. The bisimulation relation between the abstracted model and the original continuous dynamics is formally proved, which guarantees the existence of feasible continuous control signals and closed-loop trajectories for robots to satisfy the high level specifications as well. The main contribution of the paper lies in the development of a unified hybrid hierarchical control framework whose top layer is a discrete supervisor that is responsible for decision making to satisfy the assigned specification. This discrete supervisor is connected to the low level continuous dynamics of the system via an interface layer. The interface layer is responsible for translating discrete commands of the supervisor to a continuous control signals implementable by the continuous plant and vice versa.**

***Index Terms*— Hybrid systems, Supervisory control, Robot motion planning**

I. Introduction

Robots are inherently hybrid systems since they have to make logic decisions in uncertain environments and adapt to changing circumstances so as to achieve non-trivial tasks individually or collectively, such as visiting particular regions in order under certain conditions while avoiding obstacles and collisions. Meanwhile, these logic decisions made by robots, like hovering over a target, turn to neighboring regions, back to the base station etc., will unavoidably influence their continuous dynamics and control laws respectively. To comprehensively analyze and design such a system, one has to turn to hybrid modeling and control theory [1] and consider the discrete and continuous dynamics of the system, simultaneously and within a unified framework.

Actually, a current trend in the robotics literature is to study the robot motion planning problem in the framework of hybrid supervisory control, which is known as robot symbolic planning and control [2]. The basic idea is to partition the motion space of robots into logically equivalent regions, say a room itself can be considered as a region if we are just interested in the fact whether there are robots visiting the room or not. Then, a quotient transition system can be derived accordingly with these partitioned regions as its states and the existence of continuous trajectories from one region to another as its transitions. The partitioned regions are usually of finite number, so the quotient transition system can be analysized and designed through classical model checking or discrete event supervisory control techniques. The designed pathes in the quotient transition system satisfying the requested logic specifications, in the form of sequences of partitioned regions in the motion space, are then mapped back to the continuous motion space and being used to synthesize continuous control signals driving robots' physical motions or coordinations. The key challenge here is how to guarantee that there always exist physically feasible continuous trajectories and control signals for robots with respect to a discrete path in the quotient transition system. The feasibility here means twofold. No only can the robots really follow the mapped continuous trajectories using the synthesized control signals, but the continuous trajectories that robots actually exhibit need to also satisfy the logic specifications as the pathes in the quotient transition system.

To guarantee the feasibility, most efforts in the literature [3], [4], [5], [6], [7], [8] have been devoted to partitioning the motion space and obtaining an equivalent abstracted model in the sense of bisimulation, approximate bisimulation or language equivalent quotient systems. For instance, in [9] a complicated search and rescue and in [10], the motion control of robot swarms are addressed using symbolic control methods and abstraction techniques. These schemes reduce the system to a finite state transition system [2], [11], [12], for which one can design a proper discrete supervisor [13] to achieve certain properties expressed in high-level specifications such as linear or branching temporal logics. In [3], by triangulation and in [4] by the rectangulation of the motion space, continuous motion planning and control problems are mapped to a finite state transition systems. In [5] and [6], the robot motion is controlled to satisfy temporal logic specifications over convex cells.

In this paper, we intend to unify some existing results and propose a computationally effective hybrid approach for the robot motion control so that the closed-loop system satisfies the discrete logic of the decision making unit. In particular, we adopt the bisimulation-based abstraction of multi-affine dynamics on rectangular regions to obtain a equivalent quotient transition systems, by which the equivalent behaviors of the abstract model and the original plant allows the designer to synthesis the discrete supervisor for the abstract model and then apply it to the original plant. Therefore, the main contribution of this paper lies in the developing a unified hierarchical hybrid framework for symbolic motion planning and control of robots based on a bisimulation-based abstraction technique. Starting from the low level continuous dynamics of the system, it can be abstracted to a finite state machine

* Financial supports from NSF-CNS-1239222 and NSF-EECS-1253488 for this work are greatly acknowledged. Both authors are from the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556. Corresponding author: H. Lin, email: hlin1@nd.edu, Tel. 574-6313177.

over the partitioned motion space, for which we design a discrete supervisor to achieve the desired specification. We prove that the bisimulation relation between the abstracted model and the original continuous model holds for a plant with multi-affine dynamics over the rectangular partitioned space. This bisimulation relation implies the same behavior of the plant and its abstract model and therefore, the discrete supervisor designed for the abstracted model can be applied to the original continuous plant so that the closed-loop system's behavior does not change. To implement the idea, a hierarchical hybrid control structure is proposed whose lowest layer is a plant with continuous dynamics and its top layer is a discrete supervisor which controls the system to satisfy the given specification. To connect the discrete supervisor to the continuous plant, an interface layer is introduced by which the discrete commands of the supervisor can be converted to a continuous form applicable to the plant. Furthermore, when the system trajectory crosses the partitioning curves, the interface layer generates detection events which inform the supervisor about the current state of the system based on which the supervisor can issue new commands.

The rest of this paper is organized as follows. After explaining the preliminaries and notations in Section II, in Section III, the symbolic motion planning and control problem is described. Then, in Section IV, the partitioning of the motion space will be described. Several controllers will be introduced to drive the system trajectory over the partitioning elements. In Section V, the partitioned system will be bisimilarly abstracted to a finite state machine and the bisimulation relation will be proven. For the resulting finite state machine one can design a discrete supervisor as explained in Section VI. Finally, the paper is concluded in Section VII.

II. PRELIMINARIES

In the literature, there are several methods that can be used for partitioning of the space such as using natural invariants of the plants [14], rectangulation [4] or triangulation [3] of the motion space, or polar and spherical partitioning [8] of the space. Here we adopt the rectangulation of motion space for convenience, while the basic ideas can be extended to other abstraction schemes. Consider that the motion space is a $[0, x_N] \times [0, y_N]$ rectangle which is partitioned by the curves $\{x = x_i \,|\, 0 \le x_i \le x_N$, such that for $i < j$: $x_i < x_j$, $i, j = 1, ..., N_x$, $x_1 = 0$, $x_{N_x} = x_N\}$ and $\{y = y_i \,|\, 0 \le y_i \le y_N$ such that for $i < j$: $y_i < y_j$, $i, j = 1, ..., N_y$, $y_1 = 0$, $y_{N_y} = y_N\}$ into $(N_x - 1) \times (N_y - 1)$ rectangles.

In this partitioned space, the region $R_{i,j} = \{(x, y) |\, x_i \le x \le x_{i+1}, y_j \le y \le y_{j+1}\}$ is a rectangular partitioning element, which is surrounded by the curves $x = x_i$, $x = x_{i+1}$, $y = y_j$, and $y = y_{j+1}$. The interior of the region $R_{i,j}$ is denoted by $\bar{R}_{i,j}$. Each region has four vertices v_m, $m = (m_x, m_y)_2$ where m_x and m_y are the binary indices, which refer to the partitioning curves that have generated the vertex v_m. Hence, we have $v_0 = v_{(00)_2} = [x_i, y_j]^T$, $v_1 = v_{(01)_2} = [x_{i+1}, y_j]^T$, $v_2 = v_{(10)_2} = [x_i, y_{j+1}]^T$, and $v_3 = v_{(11)_2} = [x_{i+1}, y_{j+1}]^T$ as the vertices of the region $R_{i,j}$ as shown in Fig. 1.

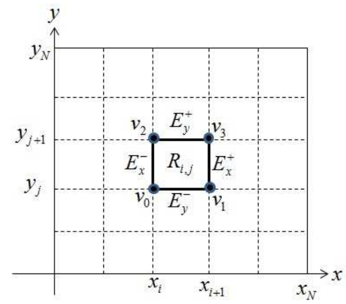

Fig. 1. Vertices and edges of the region $R_{i,j}$.

The set $V(*)$ stands for the vertices that belong to $*$, and $E(v_m)$ is the set of edges that touch the vertex v_m. Furthermore, the element $R_{i,j}$ has four edges $\{\mathrm{E}_\mathrm{x}^+, \mathrm{E}_\mathrm{x}^-, \mathrm{E}_y^+, \mathrm{E}_y^-\}$ and correspondingly, four outer normal vectors $\{\mathrm{n}_\mathrm{x}^+ = [1, 0]^T, \mathrm{n}_\mathrm{x}^- = [-1, 0]^T, \mathrm{n}_y^+ = [0, 1]^T, \mathrm{n}_y^- = [0, -1]^T\}$. For this partitioned space, $\Im(\tilde{*}) = *$ relates the label $\tilde{*}$ to the set $*$. This partitioned space can be captured by the equivalence relation $Q = \{(x_1, x_2) | \exists \tilde{*} \; s.t. \; x_1, x_2 \in \Im(\tilde{*})\}$, where $*$ is one of the above-mentioned partitioning elements. Correspondingly, $\pi_Q(x) = \tilde{*} \; s.t. \; x \in *$ and $\Im(\tilde{*}) = *$, where $\pi_Q(x)$ is a projection map.

In this partitioned space, let's define V_r is the set of all vertices of the rectangles, P as the perimeter of the motion space in which the vertices are excluded, and W is the exterior of the motion space. Also consider the *detection element* $d([i, j], [i', j']) = R_{i,j} \cap R_{i',j'} - V_r$, which is defined for two adjacent regions $R_{i,j}$ and $R_{i',j'}$ (the order is not important). With this procedure, the whole space has been partitioned into $V_r \cup R_{i,j} \cup d([i, j], [i', j') \cup P \cup W$, where $1 \le i, i' \le N_x - 1$, $1 \le j, j' \le N_y - 1$. Correspondingly, consider $\tilde{V}_r$, $\tilde{R}_{i,j}$, $\tilde{d}([i, j], [i', j'])$, $\tilde{P}$, and $\tilde{W}$ as the labels for these partitioning elements.

III. PROBLEM FORMULATION

Consider a robot with the dynamics $\dot{X}(t) = f(X(t), u(t))$ where X is the robot position and u is the control input. For the motion control of this robot, the motion space can be partitioned into several disjoint regions which are separated by hypersurfaces. Our objective here is to construct a hybrid controller to drive the robot through the partitioned space to satisfy a given specification. Let R_1, R_2,...,R_n, as the elements of the partitioned space, and correspondingly $\tilde{R}_1$, $\tilde{R}_2$,...,$\tilde{R}_n$ as the finite set of symbols that label these elements, where $\Im(\tilde{R}_i) = R_i$. The motion planning objective may require the robot to visit particular regions with a specific order while avoiding some other regions which can be specified by a LTL formula [15]. A LTL formula over the set of propositions $P = \{\tilde{R}_1, \tilde{R}_2, ..., \tilde{R}_n\}$ can be constructed using the combination of traditional logical operators including negation ($\neg$), disjunction ($\bigvee$), conjunction ($\bigwedge$), and the temporal operators including next (O), until(U), eventually ($\diamond$), always ($\square$), and release (R). For example the formula $\diamond \tilde{R}_1 \bigwedge \diamond \tilde{R}_2$ means that the robot will eventually reach region R_1 and will eventually reach region R_2. Now, the robot motion planning and control problem can be described as follows:

Problem 1: Given the system dynamics as $\dot{X}(t) = f(X(t), u(t))$ and the desired specification in terms of an LTL formula ϕ, construct the hybrid controller to generate the control signal $u(t)$ such that starting form any point inside the set of initial states X_0, visited regions by the robot trajectory $X(t)$ satisfy the formula ϕ.

To address this problem, we propose a hierarchical hybrid controller (Fig. 2) in which a discrete supervisor commands the system such that closed-loop system satisfies the formula ϕ over the partitioned space. This discrete supervisor cannot be directly connected to the plant with continuous dynamics. Hence, an interface layer is introduced which converts the discrete commands of the supervisor, u_d, to the continuous form, $u(t)$, to be applied to the plant, and translates the continuous signals of the plant, $X(t)$, to discrete symbols, x_d, understandable by the supervisor. To construct this control hierarchy, we first need to rigorously describe the partitioning of the motion space, and then, bisimilarly abstract the system to a finite state machine to be able to design the discrete supervisor.

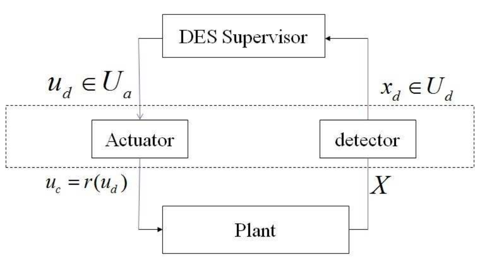

Fig. 2. The hierarchical hybrid control structure.

IV. Robot Motion Control over a Partitioned Space

To address the above mentioned problem over the partitioned space, we will develop a control mechanism that starting from any point inside a region, the robot moves to a unique destination region on its neighborhood. In this case, the system can be bisimilarly abstracted to a finite state machine and the reachability problem for such a system becomes decidable [16]. The decidability property desponds on both the system dynamics and the partitioning style. For a rectangularly partitioned space, a system with multi-affine dynamics is decidable [17]. A multi-affine function $f : \mathbb{R}^n \to \mathbb{R}^m$, has the property that for any $1 \leq i \leq n$ and any $a_1, a_2 \geq 0$ with $a_1 + a_2 = 1$, $f(x_1, ..., (a_1 x_{i_1} + a_2 x_{i_2}), x_{i+1}, ...x_n) = a_1 f(x_1, ..., x_{i_1}, x_{i+1}, ...x_n) + a_2 f(x_1, ..., x_{i_2}, x_{i+1}, ...x_n)$. In a rectangular partitioned space, this property allows us to find the value of a multi-affine vector field at any point inside a partition just based on the values of the vector field at its vertices. This property has been formally described in the following proposition.

Lemma 1: [18] Given a multi-affine function $g(X)$ defined over a rectangle $R_{i,j}$, the function g can be uniquely described based on the values of g at vertices of $R_{i,j}$ as $\forall X = (x, y) \in \bar{R}_{i,j} : g(X) = \sum_{m=0}^{3} \lambda_m g(v_m)$, where v_m, $m = 0, ..., 3$ are the vertices of the element $R_{i,j}$ and λ_m, can be obtained uniquely as follows:

$$\lambda_m = \lambda_x^{m_x} (1 - \lambda_x)^{1-m_x} \lambda_y^{m_y} (1 - \lambda_y)^{1-m_y}, \qquad (1)$$

where m_x, m_y, are the corresponding binary digits of the index m, and

$$\lambda_{\mathrm{x}} = \frac{x - x_i}{x_{i+1} - x_i} \qquad \lambda_y = \frac{y - y_j}{y_{j+1} - y_j} \qquad (2)$$

In this theorem, it can be verified that $\lambda_m \geq 0$, and $\sum_m \lambda_m = 1$. Also, since the above theorem holds true for all points in $\bar{R}_{i,j}$, it can be also applied for the points on the edge.

Now, using these properties, for a system with multi-affine dynamics it is possible to construct multi-affine controllers to either keep the system's trajectory inside the region (invariant region) or to push it out from the desired edge (exit edge) as it is described in the following two propositions.

Lemma 2: [4] (Constructing an invariant region) For a continuous multi-affine vector field $\dot{X} = h(X, u(X)) = g(X)$, the region $R_{i,j}$ is an invariant region if there exists a controller u, such that for each vertex v_m, $m = 0, 1, 2, 3$, with incident edges $E_q^s \in E(v_m)$, and corresponding outer normals n_q^s we have $U_m(Inv) = \{u | \, n_q^{sT} . g(v_m) < 0, \text{ for all } E_q^s \in E(v_m)\} \neq \emptyset$.

Lemma 3: [4] (Constructing an exit edge) For a continuous multi-affine vector field $\dot{X} = h(X, u(X)) = g(X)$, the edge E_q^s with the outer normal n_q^s, is an exit edge if there exists a controller u, such that for each vertex v_m, $m = 0, 1, 2, 3$, we have $U_m(Ex(F_q^s)) = \{u \in \mathbb{R}^2 | \, n_q^{sT} . g(v_m) > 0, \text{ for all } \; v_m \text{ and } n_{q'}^{s'T} . g(v_m) < 0, \text{ for all } E_{q'}^{s'} \neq E_q^s, \; v_m \in V(E_{q'}^{s'})\} \neq \emptyset$.

Next proposition shows that if we construct an controller based on Lemma 3, all of the points on an exit edge are reachable.

Proposition 1: For a continuous multi-affine vector field $\dot{X} = h(X, u(X)) = g(X)$, in a region $R_{i,j}$ with the exit edge E_q^s constructed by Lemma 3, all $y \in E_q^s \setminus E$ are reachable from a point inside the region $R_{i,j}$.

Proof: Respecting the second condition of Lemma 3 for the points on the exit edge E_q^s, we will have $n_q^s(y)^T.g(y) > 0$, $\forall y \in E_q^s$. This strictly positive inequality guarantees that the trajectories that leave the region do not return back any more. In addition, it shows that the points on the exit edge are not reachable from other points on the edge. Therefore, any $y \in E_q^s$ is not reachable form an adjacent region or from another point on E_q^s. Then, considering $n_q^s(y)^T.g(y) > 0$, by continuity of g, it can be concluded that there is a point inside the region $R_{i,j}$ on the neighborhood of y from which y is reachable. ■

With these controllers defined over the partitioned space, it is possible to drive the system's trajectory to one of the adjacent regions or to keep it inside the current region. This system can be captured by a transition system $T_Q = (X_Q, X_{Q_0}, U_Q, \to_Q, Y_Q, H_Q)$, where

- $X_Q = V_r \cup R_{i,j} \cup d([i,j],[i',j']) \cup P \cup W$ is the set of system states, where $1 \le i, i' \le N_x - 1,\ 1 \le j, j' \le N_y - 1$.
- $X_{Q_0} \subseteq R_{i,j}$ is the set of initial states. Here we assume that the system initially starts from some of the regions $R_{i,j}$.
- $U_Q = U_a \cup U_d$, where
  - $U_a = \{C_x^+, C_x^-, C_y^+, C_y^-, C_0\}$ is the set of labels corresponding to the controllers that can make the region $R_{i,j}$ an invariant region or can make one of its edges an exit edge. For these control labels, the sets of control actions that can be activated in this region are : $r(C_q^s) = \{u(X) | u(X) = \sum_m \lambda_m u(v_m),\ m = 0,1,2,3,\ v_m \in V(R_{i,j}),\ u(v_m) \in U_m(Ex(F_q^s))\}$, and $r(C_0) = \{u(X) | u(X) = \sum \lambda_m u(v_m),\ v_m \in V(R_{i,j}),\ u(v_m) \in U_m(Inv)\}$, where λ_m can be obtained by (1).
  - $U_d = \{\hat{d}^+([i,j],[i',j'])\} \cup \{\hat{d}^-([i,j],[i',j'])\} \cup \{\hat{P}\}$ is the set of the detection events, where $1 \le i, i' \le N_x - 1$, and $1 \le j, j' \le N_y - 1$. The events $\hat{d}^+([i,j],[i',j'])$, $\hat{d}^-([i,j],[i',j'])$, and $\hat{P}$ respectively show that the detection element $d([i,j],[i',j'])$ is crossed in positive direction of x or y axis, the detection element $d([i,j],[i',j'])$ is crossed in negative direction of x or y axis, and the perimeter of the partitioned motion space is crossed.
- $(X_1, X_2, v) \in \to_Q$, denoted by $X_1 \xrightarrow{v}_Q X_2$, if and only if one of the following conditions holds true:
  1) Actuation:
     - Exit edge: $v \in \{C_x^+, C_x^-, C_y^+, C_y^-\}$; $\pi_Q(X_1) \neq \pi_Q(X_2)$; $\exists\, i,\ j,\ i',\ j'$ such that $\pi_Q(X_1) = \tilde{R}_{i,j}$ and $\pi_Q(X_2) = \tilde{d}([i,j],[i',j'])$, or $\pi_Q(X_2) = \tilde{P}$; Furthermore, $\exists \tau(finite)$ and $\varepsilon > 0$ such that $\psi(t) : [0, \tau + \varepsilon] \to \mathbb{R}^2$ is the solution of $\dot{X} = h(X, r(v))$, $\psi(0) = X_1$; $\psi(\tau) = X_2$, $\pi_Q(\psi(t)) = \pi_Q(X_1)$ for $t \in [0, \tau)$, and $\pi_Q(\psi(t)) \neq \pi_Q(X_1)$ for $t \in [\tau, \tau + \varepsilon]$. Here, $r(v)$ is the continuous controller corresponding to the control label v, which can be constructed as discussed above.
     - Invariant region: $v = C_0$; $\exists \tilde{R}_{i,j}$ such that $\pi_Q(X_1) = \pi_Q(X_2) = \tilde{R}_{i,j}$; $\psi(t) : \mathbb{R}^+ \to \mathbb{R}^2$ is the solution of $\dot{X} = h(X, r(v))$, $\psi(0) = X_1$, $\psi(\tau) = X_2$, and $\pi_Q(\psi(t)) = \pi_Q(X_1) = \pi_Q(X_2)\ for\ all\ t \ge 0$.
  2) Detection:
     - Crossing a detection element to enter to a new region:
       a) $v \in \{\hat{d}^+([i,j],[i',j'])\} \subseteq U_d$; $\pi_Q(X_1) \neq \pi_Q(X_2)$; $\exists \tilde{R}_{i,j}$, $\tilde{R}_{i',j'}$, $\tilde{d}([i,j],[i',j'])$, $i' \ge i$, and $j' \ge j$ such that $\pi_Q(X_1) = \tilde{d}([i,j],[i',j'])$ and $\pi_Q(x_2) = \tilde{R}_{i',j'}$; $\exists 0 < \varepsilon < \tau$ and $\exists w \in \{C_x^+, c_y^+\}$ such that $\psi(t) : [0, \tau] \to \mathbb{R}^2$ is the solution of $\dot{X} = h(X, r(w))$, $\psi(\varepsilon) = X_1$; $\psi(\tau) = X_2, \pi_Q(\psi(t)) = \tilde{R}_{i,j}\ for\ t \in (0, \varepsilon)$, and $\pi_Q(\psi(t)) = \tilde{R}_{i',j'}\ for\ t \in (\varepsilon, \tau]$.
       b) $v \in \{\hat{d}^-([i,j],[i',j'])\} \subseteq U_d$; $\pi_Q(X_1) \neq \pi_Q(X_2)$; $\exists \tilde{R}_{i,j}$, $\tilde{R}_{i',j'}$, $\tilde{d}([i,j],[i',j'])$ $i' \le i$, and $j' \le j$ such that $\pi_Q(X_1) = \tilde{d}([i,j],[i',j'])$ and $\pi_Q(x_2) = \tilde{R}_{i',j'}$; $\exists 0 < \varepsilon < \tau$ and $\exists w \in \{C_x^-, c_y^-\}$ such that $\psi(t) : [0, \tau] \to \mathbb{R}^2$ is the solution of $\dot{X} = h(X, r(w))$, $\psi(\varepsilon) = X$; $\psi(\tau) = X_2, \pi_Q(\psi(t)) = \tilde{R}_{i,j}\ for\ t \in (0, \varepsilon)$, and $\pi_Q(\psi(t)) = \tilde{R}_{i',j'}\ for\ t \in (\varepsilon, \tau]$.
     - Crossing the motion space's boundary: $v = \hat{P}$; $\pi_Q(X_1) = \tilde{P}$ and $\pi_Q(X_2) = \tilde{W}$; $\exists \tilde{R}_{i,j}$ and $\exists 0 < \varepsilon < \tau$ and $\exists w \in \{C_x^+, C_x^-, C_y^+, C_y^-\}$ such that $\psi(t) : [0, \tau] \to \mathbb{R}^2$ is the solution of $\dot{X} = h(X, r(w))$, $\psi(\varepsilon) = X_1$; $\psi(\tau) = X_2, \pi_Q(\psi(t)) = \tilde{R}_{i,j}\ for\ t \in (0, \varepsilon)$, and $\pi_Q(\psi(t)) = \tilde{W}\ for\ t \in (\varepsilon, \tau]$.
- $Y_Q = X_Q$ is the output space.
- $H_Q : X \to Y_Q$ is the output map. Here, we have chosen $H_Q(X) = \pi_Q(X)$.

Analogous with [14], to model this partitioned system, we can define an interface layer which connects this partitioned system to a higher discrete supervision layer. The interface layer has two main blocks: Detector and Actuator. The detector converts continuous time signals to a sequence of symbols. Upon crossing partitioning hypersurfaces, plant symbols, $\hat{d}^+([i,j],[i',j'])$,$\hat{d}^-([i,j],[i',j'])$, and $\hat{P}$, will be generated, which inform the current situation of the plant to the supervisor. Based on the observed plant symbols, the supervisor decides which control signal should be injected to the plant to satisfy the desired specification. This command has a discrete nature and the control commands to the plant are continuous. The actuator, will translate these discrete commands to continuous signals. The block diagram of this control structure is shown in Fig. 2.

V. Abstraction over the partitioned space

In the partitioned system T_Q, although we captured all of important transitions, this transition system still has infinite number of states which makes the control synthesis problem very difficult or even impossible. Abstraction [19] is a technique that reduces the number of states by aggregating similar states. Hence, using this strategy, and considering each partitioning element as one of states in the abstracted model, the resulting model will be :
$T_\xi = (X_\xi, X_{\xi_0}, U_\xi, \to_\xi, Y_\xi, H_\xi)$, where

- $X_\xi = \{\tilde{R}_{i,j} | 1 \le i \le N_x - 1,\ 1 \le j \le N_y - 1\} \bigcup \{\tilde{d}([i,j],[i',j']) | 1 \le i, i' \le N_x - 1,\ 1 \le j, j' \le$

$N_y - 1\} \bigcup \{\tilde{P}, \tilde{W}\}$. Note that since the system starts from a point inside the regions $R_{i,j}$ and due to strictly negative inequalities in Lemmas 2 and 3, the system trajectory never crosses the vertices, and hence, the set V_r does not need to be considered in the abstracted system.

- $X_{\xi_0} \subseteq \{\tilde{R}_{i,j}|\ 1 \le i, i' \le N_x - 1,\ \ 1 \le j, j' \le N_y - 1\}$.
- $U_\xi = U_a \cup U_d$ is like what we have in T_Q.
- $(r_1, r_2, v) \in \rightarrow_\xi$, denoted by $r_1 \xrightarrow{v}_\xi r_2$, if $\exists v \in U_\xi$, $X_1 \in \Im(r_1)$, $X_2 \in \Im(r_2)$ such that $X_1 \xrightarrow{v}_Q X_2$.
- $Y_\xi = X_\xi$.
- $H_\xi(r) = r$ is the output map.

With this method, the partitioned system, T_Q which previously was modelled by the regulation layer and the interface layer, now is abstracted to a finite state transition system T_ξ for which we can design a discrete supervisor [13] to achieve the desired specification. Then, with the aid of the interface layer, the designed supervisor for the abstract model can be applied to the original continuous model. To guarantee that the discrete supervisor for the abstract model can also work for the original continuous model, it is necessary that the abstract model and the original continuous model represent the same behavior which requires them to be bisimilar. A bisimulation relation between two transition systems can be formally defined as follows:

Definition 1: [19] Given T_i=$(Q_i, Q_i^0, U_i, \rightarrow_i, Y_i, H_i)$, $(i = 1, 2)$, R is a bisimulation relation between T_1 and T_2, denoted by $T_1 \approx_R T_2$, iff:

1) $\forall q_1 \in Q_1^0$ then $\exists q_2 \in Q_2^0$ that $(q_1, q_2) \in R$. Also, $\forall q_2 \in Q_2^0$ then $\exists q_1 \in Q_1^0$ that $(q_1, q_2) \in R$.
2) $\forall q_1 \rightarrow_1 q_1'$, and $(q_1, q_2) \in R$ then $\exists q_2' \in Q_2$ such that $q_2 \rightarrow_2 q_2'$ and $(q_1', q_2') \in R$. Also, $\forall q_2 \rightarrow_2 q_2'$, and $(q_1, q_2) \in R$ then $\exists q_1' \in Q_1$ such that $q_1 \rightarrow_1 q_1'$ and $(q_1', q_2') \in R$.

For multi-affine functions defined over a rectangular partitioned model, and with the controllers which we defined to construct exit edges or to make a region invariant, the abstract model and the original partitioned system are bisimialr as described in the following theorem:

Theorem 1: The original partitioned system, T_Q, and the abstract model, T_ξ, are bisimilar.

Proof: Consider the relation $R = \{(q_Q, q_\xi)|q_Q \in X_Q,\ q_\xi \in X_\xi$, and $q_Q \in \Im(q_\xi)\}$. We will show that this relation is a bisimulation relation between T_Q and T_ξ. To prove this bismulation relation we should verify both conditions of Definition 1.

To verify the first condition of the bisimulation relation in Definition 1, we know that for any $q_Q \in X_{Q_0}$ there exists a region $R_{i,j}$ such that $q_Q \in R_{i,j}$. For this region, there exists a label, $\tilde{R}_{i,j}$ such that $R_{i,j} = \Im(\tilde{R}_{i,j})$ and $\tilde{R}_{i,j} \in X_{\xi_0}$. Hence, $(q_Q, \tilde{R}_{i,j}) \in R$. Conversely, it can be similarly shown that for any $q_\xi \in X_{\xi_0}$, there exists a $q_Q \in X_{Q_0}$ such that $(q_\xi, q_Q) \in R$.

To verify the second condition of the bisimulation relation, following from the definition of T_ξ, we know that for any $(q_Q, q_\xi) \in R$ and $q_Q \xrightarrow{u}_Q q_Q'$, there exists a transition $q_\xi \xrightarrow{u}_\xi q_\xi'$, where $q_Q' \in \Im(q_\xi')$ or equivalently $(q_Q', q_\xi') \in R$. For the converse case, assume that $q_\xi \xrightarrow{u}_\xi q_\xi'$. According to the definition of R, all $x \in \Im(q_\xi)$ are related to q_ξ. Hence, to prove the second condition of the bisimulation relation, we should investigate it for all $x \in \Im(q_\xi)$. Based on the control construction procedure, the labels u, q_ξ, and q_ξ' can be the one of the following cases:

1) $u = C_0$ and $q_\xi = q_\xi'$. In this case, since the controller C_0 makes the region an invariant region (Proposition 2), all of the trajectories starting from any $q_Q \in \Im(q_\xi)$ will remain inside the region $\Im(q_\xi)$. Therefore, for any $q_Q \in \Im(q_\xi)$, there exists a $q_Q' \in \Im(q_\xi)$ such that $q_Q \xrightarrow{u}_Q q_Q'$ and $q_Q' = \Im(q_\xi')$.
2) $u \in \{C_x^+, C_x^-, C_y^+, C_y^-\}$, $q_\xi \in \{\tilde{R}_{i,j}|\ 1 \le i \le N_x - 1,\ \ 1 \le j \le N_y - 1\}$, and $q_\xi' \in \{\tilde{d}([i,j],[i',j'])|\ 1 \le i, i' \le N_x - 1,\ \ 1 \le j, j' \le N_y - 1\}$ or $q_\xi = \tilde{P}$. In this case, based on Proposition 3 starting from any $q_Q \in \Im(q_\xi)$, the controller u drives the system trajectory towards the detection element $\Im(q_\xi')$. Therefore, for any $q_Q \in \Im(q_\xi)$, there exists a $q_Q' \in \Im(q_\xi')$ such that $q_Q \xrightarrow{u}_Q q_Q'$ and $q_Q' \in \Im(q_\xi')$.
3) $u \in \{\hat{d}^+([i,j],[i',j'])|1 \le i, i' \le N_x - 1,\ \ 1 \le j, j' \le N_y - 1\} \subseteq U_d$, $q_\xi' \in \{\tilde{R}_{i',j'}|\ 1 \le i' \le N_x - 1,\ \ 1 \le j' \le N_y - 1\}$, and $q_\xi \in \{\tilde{d}([i,j],[i',j'])|\ 1 \le i, i' \le N_x - 1,\ \ 1 \le j, j' \le N_y - 1\}$ such that $i' \ge i$ and $j' \ge j$. In this case, based on Lemma 1, for any $q_Q \in \Im(q_\xi) = d([i,j],[i',j'])$, there exists a controller $v \in \{C_x^+, C_y^+\}$ that has led the trajectory of the system from the region $R_{i,j}$ to the point q_Q on the detection element $d([i,j],[i',j'])$. Since $R_{i',j'}$ is the unique adjacent region of the element $R_{i,j}$, common in the detection element $d([i,j],[i',j'])$, based on the definition of the controller for the exit edge and Proposition 3, the controller v leads the trajectory of the system to a point inside the region $R_{i',j'}$ so that the detection event $u = \hat{d}([i,j],[i',j'])$ is generated. Therefore, for any $q_Q \in \Im(q_\xi)$, there exists a $q_Q' \in \Im(q_\xi')$ such that $q_Q \xrightarrow{u}_Q q_Q'$ and $q_Q' \in \Im(q_\xi')$. Similar explanation can be provided for the case $u \in \{\hat{d}^-([i,j],[i',j'])|1 \le i, i' \le N_x - 1,\ \ 1 \le j, j' \le N_y - 1\}$ or $u = \hat{P}$.

In all of the above mentioned cases, the second condition of the bisimulation relation for the converse case holds true. Since both conditions of the bismulation relation hold, T_ξ and T_Q are bisimilar. ■

VI. Adopting the DES supervisory control to the abstracted model

For the abstracted model with finite number of states we can design a discrete supervisor using Discrete Event Systems (DES) supervisory control theory [13]. Formally, the finite state machine model of the abstracted system can be represented by an automaton $G = (Q, \Sigma, \alpha, Q_0, Q_m)$, where $Q = Q_\xi$ is the set of states; $Q_0 = Q_{\xi_0} \subseteq Q$ is the set of initial states; $\Sigma = U_a \cup U_d$ is the (finite) set of events; $Q_m \subseteq Q$ is the set of final (marked) states, and

$\alpha : Q \times \Sigma \rightarrow Q$ is the transition function which is a partial function and determines the possible transitions in the system caused by an event. Based on the transitions in T_ξ, the function α can be defined as follows:

$$\begin{cases} \alpha(\tilde{R}_{i,j}, \sigma) = \\ \quad \begin{cases} \tilde{R}_{i,j} & if\ \sigma = C_0 \\ \tilde{d}([i,j],[i+1,j]) & if\ \sigma = C_x^+\ and\ i \neq N_x - 1 \\ \tilde{d}([i,j],[i-1,j]) & if\ \sigma = C_x^-\ and\ i \neq 1 \\ \tilde{d}([i,j],[i,j+1]) & if\ \sigma = C_y^+\ and\ j \neq N_y - 1 \\ \tilde{d}([i,j],[i,j-1]) & if\ \sigma = C_y^-\ and\ j \neq 1 \\ \tilde{P} & if\ \sigma = C_x^+,\ i = n_{x-1};\ \sigma = C_x^-,\ i = 1; \\ & \sigma = C_y^+,\ j = n_{y-1},\ or\ \sigma = C_y^+,\ j = n_{y-1} \end{cases} \\ \alpha(\tilde{d}([i,j],[i',j',]),\sigma) = \tilde{R}_{i',j'} \\ \qquad if\ \sigma = \hat{d}^+([i,j],[i',j']), i' \geq i,\ j' \geq j, \\ \qquad or\ \sigma = \hat{d}^-([i,j],[i',j']), i' \leq i,\ j' \leq j \\ \alpha(\tilde{P},\hat{P}) = \tilde{W} \end{cases}$$

In this automaton, the sequence of events composes a string. ε is an empty string, and Σ^* is the set of all possible strings over the set Σ including ε. The function α can be extended from acting on events to acting on the strings as $\alpha_{ext} : Q \times \Sigma^* \rightarrow Q$ in which $\alpha_{ext}(q,\varepsilon) = q$ and $\alpha_{ext}(q, s\sigma) = \alpha(\alpha_{ext}(q,s),\sigma)\ \forall\ s \in \Sigma^*$ and $\sigma \in \Sigma$. The language of the automaton is a sequence of strings that can be generated by G and can be defined as $L(G) = \{s \in \Sigma^* | \exists q_0 \in Q_0\, s.t.\ \alpha_{ext}(q_0,s)\ is\ defined.\}$. The marked language, denoted by $L_m(G)$ consists of the strings that can be generated by the automaton G and end with the marked states which formally can be defined as $L_m(G) = \{s \in \Sigma^* | \exists q_0 \in Q_0\, s.t.\ \alpha_{ext}(q_0,s)\ is\ defined\ and\ \alpha_{ext}(q_0,s) \in Q_m\}$. The event set Σ consists of two types of events: the controllable event set $\Sigma_c = U_a$ and the uncontrollable event set $\Sigma_{uc} = U_d$. The controllable events are those that can be disabled or enabled by an external supervisor; however, the uncontrollable events cannot be affected by the supervisor. Playing with the controllable events, the supervisor can modify the plant's generable language so that $\varepsilon \in L(S/G)$ and $[(s \in L(S/G)) \wedge (s\sigma \in L(G)) \wedge (\sigma \in L(S))] \Leftrightarrow [s\sigma \in L(S/G)]$. Correspondingly, the closed-loop marked language will be $L_m(S/G) = L(S/G) \bigcap L_m(G)$. This supervisor can be used to achieve a controllable language specification. A language specification K is said to be controllable with respect to the language of the plant G and set of uncontrollable events E_{uc} if $\forall s \in \overline{K},\ e \in E_{uc}, se \in L(G) \Rightarrow se \in \overline{K}$. To realize this control strategy and to combine the plant discrete model and the supervisor, we can use parallel composition which is a binary operation between two automata. Next theorem from DES literature shows how the parallel composition can be used to modify the plant language to achieve a desirable specification given in terms of a controllable language.

Theorem 2: Let G be the plant and $K \subseteq \Sigma^*$ be a desired language. If $\emptyset \neq K = \bar{K} \subseteq L(G)$ and K is controllable, there exist a nonblocking supervisor S such that $L(S/G) = L(S||G) = K$. In this case, S could be any automaton that satisfies $L_m(S) = L(S) = K$.

VII. Conclusion

In this paper, a hybrid framework was proposed for the symbolic motion planning and control of robots. The approach was based on rectangular partitioning of the motion space and then, abstracting the original continuous system with infinite number of states to a finite state machine. To implement the idea, a multi-layer control structure was proposed in which the discrete supervisor was connected to the plant via an interface layer. The continuous plant and the interface layer together were shown to be bisimilar with the abstract model. This bismilarity let us apply the discrete supervisor which was designed for the abstract model to the continuous plant while the closed-loop behavior does not change.

References

[1] P. Antsaklis, A. Nerode, Hybrid control systems: An introductory discussion to the special issue, IEEE Transactions on Automatic Control, 43 (4) (1998) 457–460.

[2] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, G. Pappas, Symbolic planning and control of robot motion, Robotics and Automation Magazine, 14 (1) (2007) 61–70.

[3] G. Fainekos, H. Kress-Gazit, G. Pappas, Temporal logic motion planning for mobile robots, in: IEEE International Conference on Robotics and Automation, 2005, pp. 2020–2025.

[4] C. Belta, L. Habets, Controlling a class of nonlinear systems on rectangles, IEEE Transactions on Automatic Control, 51 (11) (2006) 1749–1759.

[5] C. Belta, V. Isler, G. Pappas, Discrete abstractions for robot motion planning and control in polygonal environments, IEEE Transactions on Robotics, 21 (5) (2005) 864–874.

[6] G. Fainekos, H. Kress-Gazit, G. Pappas, Hybrid controllers for path planning: A temporal logic approach, in: 44th IEEE Conference on Decision and Control, 2005, pp. 4885–4890.

[7] A. Karimoddini, H. Lin, B. M. Chen, T. H. Lee, Hybrid three-dimensional formation control for unmanned helicopters, Automatica 49 (2) (2013) 424–433.

[8] A. Karimoddini, H. Lin, B. Chen, T. Heng Lee, Hybrid formation control of the unmanned aerial vehicles, Mechatronics, 21 (5) (2011) 886–898.

[9] H. Kress-Gazit, G. Fainekos, G. Pappas, Where's waldo? sensor-based temporal logic motion planning, in: IEEE International Conference on Robotics and Automation, 2007, pp. 3116–3121.

[10] M. Kloetzer, C. Belta, Temporal logic planning and control of robotic swarms by hierarchical abstractions, Robotics, IEEE Transactions on 23 (2) (2007) 320–330.

[11] P. Tabuada, An approximate simulation approach to symbolic control, IEEE Transactions on Automatic Control, 53 (6) (2008) 1406–1418.

[12] G. Pola, P. Tabuada, Symbolic models for linear control systems with disturbances, in: 46th IEEE Conference on Decision and Control, 2007, pp. 432–437.

[13] P. Ramadge, W. Wonham, The control of discrete event systems, Proceedings of the IEEE 77 (1) (1989) 81–98.

[14] X. Koutsoukos, P. Antsaklis, J. Stiver, M. Lemmon, Supervisory control of hybrid systems, Proceedings of the IEEE, 88 (7) (2000) 1026–1049.

[15] E. Emerson, Temporal and modal logic, Handbook of theoretical computer science, 2 (1990) 995–1072.

[16] T. Henzinger, P. Kopke, A. Puri, P. Varaiya, What's decidable about hybrid automata?, Journal of Computer and System Sciences, 57 (1) (1998) 94–124.

[17] G. Lafferriere, G. Pappas, S. Yovine, A new class of decidable hybrid systems, Hybrid Systems: Computation and Control, (1999) 137–151.

[18] L. Habets, M. Kloetzer, C. Belta, Control of rectangular multi-affine hybrid systems, in: 45th IEEE Conference on Decision and Control, 2006, pp. 2619–2624.

[19] R. Alur, T. Henzinger, G. Lafferriere, G. Pappas, Discrete abstractions of hybrid systems, Proceedings of the IEEE, 88 (7) (2000) 971 –984.