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 analyzed and designed through classical model checking or discrete event supervisory control techniques. The designed paths 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 paths 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 support from NSF-CNS-1239222 and NSF-ECCS-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: h.lin1@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 behavio-

or with multi-affine dynamics over the rectangular partition-
ed model and the original continuous model holds for a plant 
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 supervi-
or 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], rectangle [4] or triangulation [3] of 
the motion space, or polar and spherical partitioning [8] of 
the space. Here we adopt the rectangularization 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 \(x = x_i|0 \leq x_i \leq x_N\), such that for \(i < j\) :
\(x_i < x_j\), \(i, j = 1, \ldots, N_x, x_1 = 0, x_N = x_N\) and \(y = y_j|0 \leq y_j \leq y_N\) such that for \(i < j\) :
\(y_i < y_j\), \(i, j = 1, \ldots, N_y, y_1 = 0, y_N = y_N\). Further-

more, the element \(R_{i,j}\) has four edges \((E_{x1}, E_{x2}, E_{y1}, E_{y2})\) 
and correspondingly, four outer normal vectors \(\{n_x, n_x, n_y, n_y\}\) 
with \(n_x = [1, 0]^T, n_x = [0, 1]^T, n_y = [0, 1]^T, n_y = [0, -1]^T\).

For this partitioned space, \(S(\hat{v}) = *\) relates the label \(\hat{v}\) to the 
set *. This partitioned space can be captured by the 
equivalence relation \(Q = \{[x_1, x_2] \mid s.t. x_1, x_2 \in S(\hat{v})\}\), 
where * is one of the above-mentioned partitioning elements. 
Correspondingly, \(\pi_Q(x) = \hat{v} \ast s.t. x \in * \text{ and } S(\hat{v}) = *\), 
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 ex-
terior 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 \(1 \leq i, i' \leq N_x - 1, 1 \leq j, j' \leq N_y - 1\).

Correspondingly, consider \(V_r, R_{i,j}, d([i, j], [i', j']), P, \text{ and } W\) as the labels for these 
partitioning elements.

III. PROBLEM FORMULATION

Consider a robot with the dynamics \(X(t) = f(X(t), u(t))\) 
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 \(R_1, R_2, \ldots, R_n\) as 
the elements of the partitioned space, and correspondingly 
\(R_1, R_2, \ldots, R_n\) as the finite set of symbols that label these 
elements, where \(\Sigma(R_i) = R_i\). 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 \(P = \{R_1, R_2, \ldots, R_n\}\) can be constructed 
using the combination of traditional logical operators including 
negation (¬), disjunction (∨), conjunction (∧), and the
temporal operators including next (O), until (U), eventually (□), always (□), and release (R). For example the formula
\[ \phi R_1 \land \phi R_2 \] 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 \( \phi \), 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 \( \phi \).

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
\( \phi \) 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 con-
tinuous 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.

IV. ROBOT MOTION CONTROL OVER A PARTITIONED
SPACE

To address the above mentioned problem over the parti-
tioned 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 bisimilarity abstracted to a
finite state machine and the reachability problem for such
a system becomes decidable [16]. The decidability prop-
erty desponds on both the system dynamics and the par-
titioning style. For a rectangulated partitioned space, a
system with multi-affine dynamics is decidable [17]. A
multi-affine function \( f : \mathbb{R}^n \rightarrow \mathbb{R}^m \), has the property
that for any \( 1 \leq i \leq n \) and any \( a_1, a_2 \geq 0 \) with
\[ a_1 + a_2 = 1, \] \[ f(x_1, ..., (a_1x_i + a_2x_{i+1}), x_{i+1}, ..., x_n) = a_1f(x_1, ..., x_{i+1}, ..., x_n) + a_2f(x_1, ..., x_{i+2}, ..., x_n). \] 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 \( g(X) \) de-

fined 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
\[ g(x) = \sum_{m=0}^{3} \lambda_m g(v_m), \] where \( v_m \),
\( m = 0, ..., 3 \) are the vertices of the element \( R_{i,j} \) and \( \lambda_m \),
can be obtained uniquely as follows:
\[ \lambda_m = \lambda_x^{m_x}(1 - \lambda_x)1^{-m_x}\lambda_y^{m_y}(1 - \lambda_y)1^{-m_y}, \] (1)
where \( m_x, m_y, \) are the corresponding binary digits of
the index \( m \), and
\[ \lambda_x = \frac{x - x_i}{x_{i+1} - x_i}, \quad \lambda_y = \frac{y - y_j}{y_{j+1} - y_j} \] (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 \( R_{i,j} \), 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 \( E^s_q \in E(v_m) \), and
corresponding outer normals \( n^s_q \) we have \( U_m(Inv) = \{ u \mid n^s_q \cdot g(v_m) < 0, \forall E^s_q \} \neq \emptyset \).

Lemma 3: [4] (Constructing an exit edge) For a con-
tinuous multi-affine vector field \( \dot{X} = h(X, u(X)) =
\) \( g(X) \), the edge \( E^q_q \) with the outer normal \( n^s_q \) is an exit
edge if there exists a controller \( u \), such that for each
vertex \( v_m \), \( m = 0, 1, 2, 3 \), we have \( U_m(Exit_q) = \{ u \mid n^s_q \cdot g(v_m) > 0, \forall E^s_q \} \neq \emptyset \).

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 \( E^q_q \) constructed by Lemma 3, all \( y \in E^s_q \setminus E \) 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 \( E^q_q \), we will have \( n^s_q(y)^T \cdot g(y) > 0, \)
\( \forall y \in E^s_q \). 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
\( y \in E^s_q \) is not reachable form an adjacent region or from
another point on \( E^s_q \). Then, considering \( n^s_q(y)^T \cdot g(y) > 0, \)
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 $T_Q = (X_Q, X_{Q_0}, U_Q, \rightarrow_Q, Y_Q, H_Q)$, where

- $X_Q = V_r \cup R_{i,j} \cup \{ [i,j],[i',j'] \} \cup P \cup W$ is the set of system states, where $1 \leq i, i' \leq N_x - 1$, and $1 \leq j, j' \leq N_y - 1$.
- $X_{Q_0} \subseteq R_{i,j}$ 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:
    - $r(C_x^+) = \{ u(X) | u(X) = \sum_{m=0,1,2,3} \lambda_m u(v_m), v_m \in V(R_{i,j}), u(v_m) \in U_m(m v = 0,1,2,3) \}$, and
    - $r(C_x^-) = \{ u(X) | u(X) = \sum_{m=0,1,2,3} \lambda_m u(v_m), v_m \in V(R_{i,j}), u(v_m) \in U_m(m v = 0,1,2,3) \}$, where $\lambda_m$ can be obtained by (1).
  - $U_d = \{ d^+(\{i,j\},[i',j']) \} \cup \{ d^-(\{i,j\},[i',j']) \} \cup \{ \bar{P} \}$ is the set of the detection events, where $1 \leq i, i' \leq N_x - 1$, and $1 \leq j, j' \leq N_y - 1$.
    - The events $d^+(\{i,j\},[i',j'])$, $d^-(\{i,j\},[i',j'])$, and $\bar{P}$ 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.

- $(X_1, X_2, v) \rightarrow_Q X_2$, denoted by $X_1 \xrightarrow{v} X_2$, if and only if one of the following conditions holds true:
  1. Actuation:
     - Exit edge: $v \in \{ C_x^+, C_x^-, C_y^+, C_y^- \}$; $\pi_Q(X_1) \neq \pi_Q(X_2)$; $\exists \bar{R}_{i,j}$ such that $\pi_Q(X_1) = \bar{R}_{i,j}$ and $\pi_Q(X_2) = d([i,j],[i',j'])$, or $\pi_Q(X_2) = \bar{P}$; Furthermore, $\exists r(\text{finite})$ and $\varepsilon > 0$ such that $\psi(t) : [0,\tau + \varepsilon] \rightarrow \mathbb{R}^2$ is the solution of $X = h(X, r(v)), \psi(0) = X_1; \psi(\tau) = X_2, \pi_Q(\psi(t)) = \pi_Q(X_1)$ 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: $v \in C_0$; $\exists \bar{R}_{i,j}$ such that $\pi_Q(X_1) = \pi_Q(X_2) = \bar{R}_{i,j}$; $\psi(t) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is the solution of $X = h(X, \psi(v)), \psi(0) = X_1, \psi(\tau) = X_2$, and $\pi_Q(\psi(t)) = \pi_Q(X_1) = \pi_Q(X_2)$ for all $t \geq 0$.
  2. Detection:
     - Crossing a detection element to enter to a new region:
\( N_y = 1 \) \( \cup \{ \hat{P}, W \} \). 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.

- \( X_{\xi'} \subseteq \{ \hat{R}_{i,j} \mid 1 \leq i, i' \leq N_x - 1, 1 \leq j, j' \leq N_y - 1 \} \).
- \( U_{\xi} = U_a \cup U_d \) is like what we have in \( T_Q \).
- \((r_1, r_2, v) \in X_{\xi} \), denoted by \( r_1 \xrightarrow{u} r_2 \), if \( \exists v \in U_{\xi} \), \( X_1 \in \mathcal{X}(r_1) \), \( X_2 \in \mathcal{X}(r_2) \) such that \( X_1 \xrightarrow{u} Q X_2 \).
- \( Y_{\xi} = X_{\xi} \).
- \( H_{\xi}(r) = r \) 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 \( T_{\xi} \) 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 \( (T_1 = (Q_1, \delta_1, Z_1, I_1, F_1), (i = 1, 2), R \) is a bisimulation relation between \( T_1 \) and \( T_2 \), denoted by \( T_1 \sim_R T_2 \), 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. \( \forall q_1 \rightarrow q_1' \), and \( (q_1, q_2) \in R \) then \( \exists q_2' \in Q_2 \) such that \( q_2 \rightarrow q_2' \) and \( (q_1', q_2') \in R \). Also, \( \forall q_1 \rightarrow q_1' \), and \( (q_1, q_2) \in R \) then \( \exists q_1' \in Q_1 \) such that \( q_1 \rightarrow q_1' \) 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 bisimilar as described in the following theorem:

**Theorem 1**: The original partitioned system, \( T_Q \), and the abstract model, \( T_{\xi} \), are bisimilar.

**Proof**: Consider the relation \( R = \{(q_0, q_\xi) | q_0 \in X_Q, q_\xi \in X_{\xi}, \text{ and } q_0 \in \mathcal{X}(q_\xi) \} \). We will show that this relation is a bisimulation relation between \( T_Q \) and \( T_{\xi} \). To prove this bisimulation 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 \( q_0 \in X_{Q_0} \) there exists a region \( R_{i,j} \) such that \( q_0 \in R_{i,j} \). For this reason, there exists a label, \( R_{i,j} \), such that \( R_{i,j} \subseteq \mathcal{X}(R_{i,j}) \) and \( R_{i,j} \subseteq X_{\xi} \). Hence, \( (q_0, R_{i,j}) \in R \). Conversely, it can be similarly shown that for any \( q_\xi \in X_{\xi} \), there exists a \( q_0 \in X_{Q_0} \) such that \( (q_\xi, q_0) \in R \).

To verify the second condition of the bisimulation relation, following from the definition of \( T_{\xi} \), we know that for any \( (q_0, q_\xi) \in R \) and \( q_\xi \xrightarrow{u} q_\xi' \), there exists a transition \( q_\xi \xrightarrow{u} q_\xi' \) where \( q_\xi' \in \mathcal{X}(q_\xi') \) or equivalently \( (q_0, q_\xi') \in R \). For the converse case, assume that \( q_\xi \xrightarrow{u} q_\xi' \). According to the definition of \( R \), all \( x \in \mathcal{X}(q_\xi') \) are related to \( q_\xi \). Hence, to prove the second condition of the bisimulation relation, we should investigate it for all \( x \in \mathcal{X}(q_\xi) \). Based on the control construction procedure, the labels \( u, q_\xi \), and \( q_\xi' \) can be 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 \( q_\xi \in \mathcal{X}(q_\xi) \) will remain inside the region \( \mathcal{X}(q_\xi) \). Therefore, for any \( q_0 \in \mathcal{X}(q_\xi) \), there exists a \( q_0' \in \mathcal{X}(q_\xi) \) such that \( q_0 \xrightarrow{u} q_0' \) and \( q_\xi' = \mathcal{X}(q_\xi') \).
2. \( u \in \{ C_x^+, C_y^+, C_y^- \}, q_\xi \in \{ R_{i,j} \mid 1 \leq i \leq N_x - 1, 1 \leq j \leq N_y - 1 \}, \) and \( q_\xi' \in \{ d([i,j],[i',j']) \mid 1 \leq i, i' \leq N_x - 1, 1 \leq j, j' \leq N_y - 1 \} \) or \( q_0 = \hat{P} \). In this case, based on Proposition 3 starting from any \( q_0 \in \mathcal{X}(q_\xi) \), the controller \( u \) drives the system trajectory towards the detection element \( \mathcal{X}(q_\xi) \). Therefore, for any \( q_\xi \in \mathcal{X}(q_\xi) \), there exists a \( q_0 \in \mathcal{X}(q_\xi) \) such that \( q_0 \xrightarrow{u} q_0' \) and \( q_\xi' = \mathcal{X}(q_\xi') \).
3. \( u \in \{ d^+([i,j],[i',j']) \mid 1 \leq i, i' \leq N_x - 1, 1 \leq j, j' \leq N_y - 1 \} \subseteq U_d \), \( q_\xi \in \{ R_{i,j} \mid 1 \leq i \leq N_x - 1, 1 \leq j \leq N_y - 1 \}, \) and \( q_\xi' \in \{ d([i,j],[i',j']) \mid 1 \leq i, i' \leq N_x - 1, 1 \leq j, j' \leq N_y - 1 \} \). In this case, based on Lemma 1, for any \( q_0 \in \mathcal{X}(q_\xi) \), there exists a controller \( v \in \{ C_x^+, C_y^+, C_y^- \} \) that has led the trajectory of the system from the region \( R_{i,j} \) to the point \( q_0 \) 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 \( u = d([i,j],[i',j']) \) is generated. Therefore, for any \( q_0 \in \mathcal{X}(q_\xi) \), there exists a \( q_0' \in \mathcal{X}(q_\xi) \) such that \( q_0 \xrightarrow{u} q_0' \) and \( q_\xi' = \mathcal{X}(q_\xi') \). Similar explanation can be provided for the case \( u \in \{ d^-(i,j), [i',j'] \mid 1 \leq i, i' \leq N_x - 1, 1 \leq j, j' \leq N_y - 1 \} \) or \( u = P \).

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 bisimulation relation hold, \( T_{\xi} \) 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 \( T_{\xi} \), the function \( \alpha \) can be defined as follows:

\[
\alpha(\tilde{R}_{i,j},\sigma) =
\begin{cases}
\tilde{R}_{i,j} & \text{if } \sigma = C_0 \\
\tilde{d}(i,j,[i+1,j]) & \text{if } \sigma = C_x^+ \text{ and } i \neq N_x - 1 \\
\tilde{d}(i,j,[i-1,j]) & \text{if } \sigma = C_x^- \text{ and } i \neq 1 \\
\tilde{d}(i,j,[i,j+1]) & \text{if } \sigma = C_y^+ \text{ and } j \neq N_y - 1 \\
\tilde{d}(i,j,[i,j-1]) & \text{if } \sigma = C_y^- \text{ and } j \neq 1 \\
P & \text{if } \sigma = C_x^+, i = n_x-1; \text{ or } \sigma = C_x^-, i = 1; \\
\sigma = C_y^+, j = n_y-1; \text{ or } \sigma = C_y^-, j = n_y-1
\end{cases}
\]

Theorem 2: Let \( G \) be the plant and \( K \subseteq \Sigma^* \) be a desired language. If \( \emptyset \neq K = \tilde{K} \subseteq L(G) \) and \( \tilde{K} \) is controllable, there exist a nonblocking supervisor \( \tilde{S} \) such that \( L(S/G) = L(\|G\|/\tilde{K}) \). In this case, \( \tilde{S} \) 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 shown to be bisimilar with the abstract model. This bimilarity 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