International Journal of Control
Vol. 85, No. 10, October 2012, 1464–1476

An input–output simulation approach to controlling multi-affine systems for linear temporal logic specifications

Yajuan Sun[a]*, Hai Lin[b] and Ben M. Chen[a]

[a]*Department of Electrical and Computer Engineering, National University of Singapore, Singapore;*
[b]*Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA*

(*Received 4 February 2012; final version received 26 April 2012*)

This article presents an input–output simulation approach to controlling multi-affine systems for linear temporal logic (LTL) specifications, which consists of the following steps. First, the state space is partitioned into rectangles, each of which satisfies atomic LTL propositions. Then, we study the control of multi-affine systems on rectangles, including the control based on the exit sub-region to drive all trajectories starting from a rectangle to exit through a facet and the control to stabilise the multi-affine system towards a desired point. With the proposed controllers, a finitely abstracted transition system is constructed which is shown to be input–output simulated by the rectangular transition system of the multi-affine system. Since the input–output simulation preserves LTL properties, the controller synthesis of the multi-affine system for LTL specifications is achieved by designing a nonblocking supervisor for the abstracted transition system and by implementing the resulting supervisor to the original multi-affine system.

Keywords: automatic synthesis; hybrid systems; multi-affine functions; linear temporal logic

1. Introduction

Due to the integration of embedded computers and communications, high-level specifications like sequencing tasks, system synchronisation and network adaptability naturally emerge in the engineering applications, which goes beyond the traditional control tasks such as stabilisation, output regulation and so on. To address such a challenge, temporal logic, especially linear temporal logic (LTL), has been adopted from computer science to the control and robotics society (Thistle and Wonham 1986; Knight and Passino 1990; Belta et al. 2007; Ulusoy, Smith, Xu, and Belta 2012). Temporal logic can be used to form complicated specifications in a succinct and unambiguous manner. In addition, temporal logic is similar to natural languages and can be easily interpreted by human operators (Eker et al. 2002). Therefore, recent years have seen increasing activities in controller design to satisfy temporal logic specifications.

The basic idea to solve the controller design for LTL specifications is to abstract finite-state transition systems from continuous systems. The resulting finite-state transition systems preserve LTL properties, therefore enabling the controller synthesis through discrete algorithm techniques. Fainekos, Kress-Gazit, and Pappas (2005) studied the control of robots with second-order linear dynamics in a polygonal workspace to fulfil LTL specifications, where the discrete abstraction can be obtained by a triangulation of polygon and vector fields assigned in each triangles drive the produced trajectories to satisfy an LTL formula over the triangles. This work was refined in Tabuada and Pappas (2006) by approaching arbitrary-dimensional discrete-time linear system. It was shown that an equivalent discrete transition system exists for the controllable system with properly chosen observables. Specifically, it builds up the framework for generating the runs of the discrete transition system satisfying the LTL specifications. As opposed to discrete-time linear systems in Tabuada and Pappas (2006) and Kloetzer and Belta (2008) studied the control problem for the LTL specifications with respect to continuous-time linear systems. Based on the results of controlling linear systems on polytopes (Habets and van Schuppen 2004), a computational approach was provided to controller design consisting of polyhedral operator and searches on graphs. Other related work includes the control of a planar robot to achieve sensor-based LTL specifications (Kress-Gazit, Fainekos, and Pappas 2009) and robust LTL specifications (Fainekos, Girard, Kress-Gazit, and Pappas 2009). Although many of these works

*Corresponding author. Email: sunyajuan@nus.edu.sg

ISSN 0020–7179 print/ISSN 1366–5820 online

http://dx.doi.org/10.1080/00207179.2012.689862
http://www.tandfonline.com

provide valuable inspiration, they are only applicable to linear systems.

In this article, we consider a particular class of nonlinear systems-multi-affine systems. This kind of continuous dynamics is widely used for system modelling in practice, such as the celebrated Ogawa (1993), Volterra (1926) and Lotka-Volterra (1925) equations, the control systems for aircraft and underwater vehicles (Belta 2004) and the models of genetic regulatory networks (Sastry 1999). Formal analysis and control of such systems were investigated in the literature (Belta and Habets 2006; Habets, Kloetzer, and Belta 2006; Kloetzer and Belta 2006; Berman, Halász, and Kumar 2007). Different from their works, we propose an input–output simulation approach so that the controlled multi-affine systems fulfil the LTL specifications. It consists of the following steps. First, we partition the state space into several rectangles consistent with the coordinates. Each rectangle satisfies atomic LTL propositions. Second, we investigate the control of multi-affine systems on rectangles. A control method is provided based on the exit sub-region to drive all trajectories starting from a rectangle to exit only through a facet. In addition, we investigate the control of stabilising the system towards a desired point. Third, by using the proposed control methods, a finitely abstracted transition system of the multi-affine system is constructed. Then, we formalise the notion of input–output simulation as a behaviour inclusion between transition systems and show that the abstracted transition system is input–output simulated by the rectangular transition system of the original multi-affine system. Since input–output simulation preserves LTL properties, the controller synthesis for the original multi-affine system to enforce the linear temporal specification is achieved by designing a nonblocking supervisor for the abstracted transition system and by implementing the resulting supervisor to the original multi-affine system.

Compared with the literature, the contributions of this article mainly lie on the following aspects. First, a novel control method is proposed based on the exit sub-region to drive the system to exit through a desired facet. It is shown that this method covers more classes of systems than those are addressed in Belta and Habets (2006) and Habets et al. (2006). Furthermore, we provide a solution for the convergence problem by stabilising the system towards a fixed point. Second, we formalise the notion of input–output simulation. Since this notion requires input equivalence as well as output equivalence, it is stronger than the conventional simulations which need either of them (Milner 1989; Tabuada and Pappas 2006). It is shown that there exists an input–output simulation between the abstracted transition system and the rectangular transition system of the multi-affine system. Therefore, the multi-affine map of the control input, enforcing LTL specifications with respect to the abstracted transition system, is also implementable for the original multi-affine system. Third, a nonblocking supervisor is designed for the abstracted transition system in order to prevent blocking in the execution and to implement the control strategy effectively. Moreover, multiple feasible paths can be automatically chosen by using this nonblocking supervisor.

The rest of this article is organised as follows. Section 2 gives the preliminary results. Section 3 presents the control of multi-affine systems on rectangles. Section 4 investigates the finitely abstracted transition system of the multi-affine system. The controller synthesis for LTL specifications is studied in Section 5. An illustrative example is presented in Section 6. This article concludes with Section 7.

2. Preliminary results

2.1 *Multi-affine systems on rectangles*

We start by reviewing the notions of multi-affine function and multi-affine control system.

Definition 2.1 (Belta and Habets 2006)**:** A function $f=(f_1,f_2,\dots,f_m)$: $\mathbb{R}^n\to\mathbb{R}^m$ (with $n,m\in\mathbb{N}$) is said to be multi-affine, if every $f_i(x)$: $\mathbb{R}^n\to\mathbb{R}$, where $x=(x_1,x_2,\dots,x_n)$ and $i=1,\dots,m$, is a polynomial in the indeterminates $x_1,x_2,\dots,x_n$, with the property that the degree of f_i in any of indeterminates $x_1,x_2,\dots,x_n$ is less or equal to 1. That is, f has the form

$$\begin{aligned} f(x) &= f(x_1,x_2,\dots,x_n)\\ &= \sum_{i_1,\dots,i_n\in\{0,1\}} c_{i_1i_2\cdots i_n}(x_1)^{i_1}(x_2)^{i_2}\cdots(x_n)^{i_n}\end{aligned}$$

where $c_{i_1i_2\cdots i_n}\in\mathbb{R}^m$ for all $i_1,i_2,\dots,i_n\in\{0,1\}$.

For example, for $n=2$ and arbitrary m, all multi-affine functions have the form $f(x_1,x_2)=c_{00}+c_{10}x_1+c_{01}x_2+c_{11}x_1x_2$, where $c_{ij}\in\mathbb{R}^m$ for $i,j\in\{0,1\}$.

Definition 2.2: A control system $\Sigma: \dot{x}=f(x,u)=g(x)+Bu$ with $B\in\mathbb{R}^{n\times m}$ is said to be multi-affine if g: $\mathbb{R}^n\to\mathbb{R}^n$ is a multi-affine function.

For a multi-affine control system, we write $\chi_{x_0,u}(t)$ to denote the point reached at time t under the control input u from initial condition x_0. In this article, the state space of the multi-affine system is assumed to be bounded and rectangular, which holds in lots of engineering applications (Belta and Habets 2006; Berman et al. 2007). Given such a state space, we would like to rectangularly partition it with respect to the coordinates. Then, the following concepts are provided.

An n-rectangle is described by $E=\prod_{i=1}^{n}(a_i, b_i)$, where $a_i,\ b_i \in \mathbb{R}$ satisfy $a_i < b_i$ for $i=1,2,\ldots,n$. The closure of E is defined as $\overline{E}=\prod_{i=1}^{n}[a_i, b_i]$. A facet of E is the intersection of $\overline{E}$ with one of its supporting hyperplanes. The set of facets of E is denoted by $F(E)$. The set of vertices of E, denoted by $V(E)$, is $V(E)=\{(x_1, x_2, \ldots, x_n) \mid x_i \in \{a_i, b_i\}, \quad i=1,2,\ldots,n\}$. Given $v \in V(E)$, we denote $F(v)$ the set of all facets containing v.

The state space can be partitioned into $\prod_{i=1}^{n} n_i$ rectangles as follows. Let $x_i \in \bigcup_{j=1}^{n_i}(a_i^j, b_i^j)$, where $a_i^j < b_i^j$ and $a_i^{j+1} = b_i^j$. Then, $R_{k_1k_2\cdots k_n} = \prod_{i=1}^{n}(a_i^{k_i}, b_i^{k_i})$ is a rectangle in the partitioned state space, where $1 \le k_i \le n_i$. The facet of $R_{k_1k_2\cdots k_n}$ is described by

$$F_{k_1k_2\cdots k_n}^{j,d} = \begin{cases} \overline{R_{k_1k_2\cdots k_n}} \bigcap \left\{x \in \mathbb{R}^n \mid x_j = b_j^{k_j}\right\} & \text{if } d=+ \\ \overline{R_{k_1k_2\cdots k_n}} \bigcap \left\{x \in \mathbb{R}^n \mid x_j = a_j^{k_j}\right\} & \text{if } d=- \end{cases}$$

where $d \in \{+,-\}$ and $j=1,\ldots,n$.

The outer normal of $F_{k_1k_2\cdots k_n}^{j,d}$ is given by

$$n^{j,d} = \begin{cases} e_j^{\top} & \text{if } d=+ \\ -e_j^{\top} & \text{if } d=- \end{cases}$$

where $d \in \{+,-\}$, $j=1,\ldots,n$ and e_j is the Euclidian basis of $\mathbb{R}^n$.

Given $w=(w_1, w_2, \ldots, w_n) \in V(R_{k_1k_2\cdots k_n})$, the vertex membership function S: $\{w_1, \ldots, w_n\} \to \{0,1\}$ is defined as

$$S(w_j) = \begin{cases} 1 & \text{if } w_j = b_j^{k_j} \\ 0 & \text{if } w_j = a_j^{k_j} \end{cases}.$$

Denote ξ as the set of rectangles generated by rectangularly partitioning the state space. The rectangular projection map π_Q: $\mathbb{R}^n \to \xi$ is defined as $\pi_Q(x) = \{R_{k_1k_2\cdots k_n} \in \xi \mid x \in R_{k_1k_2\cdots k_n}\}$. Subsequently, the property of the multi-affine function on rectangles is presented as follows.

Lemma 2.3 (Belta and Habets 2006)**:** *Consider a multi-affine function f and a rectangle $R_{k_1k_2\cdots k_n}$. In every point $x \in R_{k_1k_2\cdots k_n}$, the value $f(x)$ is uniquely determined by the values of f at vertices of $R_{k_1k_2\cdots k_n}$:*

$$f(x) = \sum_{w \in V(R_{k_1k_2\cdots k_n})} \lambda_w(x) f(w) \tag{1}$$

where for any $w=(w_1,\ldots,w_n) \in V(R_{k_1k_2\cdots k_n})$ and $x=(x_1,x_2,\ldots,x_n) \in R_{k_1k_2\cdots k_n}$, the coefficient $\lambda_w(x)$ is defined as

$$\lambda_w(x) = \prod_{j=1}^{n} \left(\frac{x_j - a_j^{k_j}}{b_j^{k_j} - a_j^{k_j}}\right)^{S(w_j)} \left(\frac{b_j^{k_j} - x_j}{b_j^{k_j} - a_j^{k_j}}\right)^{(1-S(w_j))} \tag{2}$$

By using this property, we review the results on the existence of a multi-affine feedback controller for a multi-affine system to keep the system in a rectangular invariant (Lemma 2.4) and to drive all initial states in a rectangle through a desired fact in finite time (Lemma 2.5).

Lemma 2.4 (Belta and Habets 2006)**:** *Given a multi-affine control system Σ: $\dot{x} = g(x) + Bu$ and a rectangle $R_{k_1k_2\cdots k_n}$, there exists a multi-affine feedback controller $K(x)$ such that $u = K(x)$ and all trajectories of the closed-loop system that start from $R_{k_1k_2\cdots k_n}$ remain in $R_{k_1k_2\cdots k_n}$ for all times if and only if for any $w \in V(R_{k_1k_2\cdots k_n})$, the following set is nonempty*:

$$U_I(w) = \bigcap_{F_{k_1k_2\cdots k_n}^{j,d} \in F(w)} \{v \in \mathbb{R}^m \mid n^{j,d}(g(w) + Bv) \le 0\}. \tag{3}$$

Lemma 2.5 (Belta and Habets 2006)**:** *Given a multi-affine control system $\Sigma: \dot{x} = g(x) + Bu$ and a rectangle $R_{k_1k_2\cdots k_n}$, there exists a multi-affine feedback controller $K(x)$ such that $u = K(x)$ and all trajectories of the closed-loop system that start from $R_{k_1k_2\cdots k_n}$ are driven only through $F_{k_1k_2\cdots k_n}^{j,d}$ in finite time if for any $w \in V(R_{k_1k_2\cdots k_n})$, the following set is nonempty*:

$$U_E(w) = \bigcap_{F_{k_1k_2\cdots k_n}^{j',d'} \in F(w),\,(j,d)\ne(j',d')} \{v \in \mathbb{R}^m \mid n^{j,d}(g(w) + Bv) > 0 \wedge n^{j',d'}(g(w) + Bv) \le 0\}. \tag{4}$$

2.2 *Transition system and LTL*

A transition system is a tuple $S=(E, E_0, U, \to, E_m, Y, H)$, where E is a set of states, $E_0 \subseteq E$ is a set of initial states, U is a set of control inputs, $\to \subseteq E \times U \times E$ is a transition relation, E_m is a set of marked states, Y is a set of outputs and H: $E \to Y$ is an output function. The evolution of a system is captured by the transition relation. A transition $(e,u,e') \in \to$ is denoted as $e \xrightarrow{u} e'$. Let U^* be a set of all finite strings over U, including the empty string ϵ. The transition relation $\to \subseteq E \times U \times E$ can be extended to $\to \subseteq E \times U^* \times E$ in a natural way: $e \xrightarrow{su} e'$ if there exists an e'' such that $e \xrightarrow{s} e''$ and $e'' \xrightarrow{u} e'$, where $s \in U^*$ and $u \in U$. For $E_1 \subseteq E$, the notation $\to|_{E_1 \times U \times E_1}$ means $\to$ is restricted to a smaller domain E_1. Consider a set of propositions Π, the label function L: $Y \to 2^{\Pi}$ assigns each output a set of atomic propositions satisfied by this output. Consider $e_1 \xrightarrow{u_1} e_2 \xrightarrow{u_2} \cdots e_n \xrightarrow{u_n} e_{n+1}$. A finite path generated from e_1, denoted as P_{e_1}, is a finite alternating sequence of outputs and inputs: $P_{e_1} = H(e_1)u_1H(e_2)u_2 \cdots H(e_n)u_nH(e_{n+1})$. A finite run generated from e_1, denoted as R_{e_1}, is a finite sequence of outputs: $R_{e_1} = H(e_1)H(e_2)\cdots H(e_n)$. If the lengths of the above sequences are infinite, they are called to be an infinite path and an infinite run, respectively. Denote $P(S)$, $P^w(S)$, $R(S)$ and $R^w(S)$ as the set of all finite paths generated by S, the set of all infinite paths generated by S, the set of all finite runs generated by S and the set of all infinite runs generated by S, respectively. Given $B \subseteq R^w(S)$, the prefix of B is defined as $\overline{B} = \{s \in R(S) \mid \exists t \in R^w(S) : st \in B\}$.

A transition system defines different languages. The finite language of S is defined as $L(S)=\{R_e \in R(S) \mid e \in E_0\}$. The infinite language of S is defined as $L^w(S)=\{R_e \in R^w(S) \mid e \in E_0\}$. Let $Y_m = \{y \mid y = H(e), e \in E_m\}$. The accepted language of S is defined as $L^w_A(S) = \{r \in R^w(S) \mid \inf(r) \cap Y_m \neq \emptyset\}$, where $\inf(r)$ denotes the set of outputs appearing infinitely often in run r. The finite path language of S is defined as $L_P(S) = \{P_e \in P(S) \mid e \in E_0\}$. The infinite path language of S is defined as $L^w_P(S)=\{P_e \in P^w(S) \mid e \in E_0\}$. Given a label function L: $Y \to 2^{\Pi}$, an infinite run $R = R(1) \times R(2)R(3)\cdots$ defines a word $W = W(1)W(2)W(3)\cdots$, where $W(i) = L(R(i))$ for $i = 1, 2, 3, \ldots$.

The syntax and semantics of LTL formulas over the words of the transition system are introduced (Kloetzer and Belta 2008).

Definition 2.6 (Syntax of LTL formulas)**:** An LTL formula over Π is recursively defined as:

- Every proposition $\pi \in \Pi$ is a formula.
- If φ_1 and φ_2 are formulas, then $\varphi_1 \wedge \varphi_2$, $\neg\varphi_1$, $\circ\varphi$ and $\varphi_1 \mathcal{U} \varphi_2$ are also formulas.

Definition 2.7 (Semantics of LTL formulas)**:** The satisfaction of an LTL formula φ at position $i = 1, 2, 3, \ldots$ of the word W, denoted by $W(i) \vDash \varphi$, is recursively defined as:

- $W(i) \vDash \pi$, if $\pi \in W(i)$;
- $W(i) \vDash \neg\varphi$, if $W(i) \nvDash \varphi$, where $\nvDash$ denotes the negation of $\vDash$;
- $W(i) \vDash \circ\varphi$ if $W(i+1) \vDash \varphi$;
- $W(i) \vDash \varphi_1 \wedge \varphi_2$, if $W(i) \vDash \varphi_1$ and $W(i) \vDash \varphi_2$;
- $W(i) \vDash \varphi_1 \mathcal{U} \varphi_2$, if there exists a $j > i$ such that $W(j) \vDash \varphi_2$ and for all $i \leq k < j$ we have $W(k) \vDash \varphi_1$.

If $W(1) \vDash \varphi$, we say that the word W satisfies φ, written as $W \vDash \varphi$. The symbols $\wedge$ and $\neg$ stand for conjunction and negation, respectively. The other Boolean connectors $\vee$ (disjunction), $\Rightarrow$ (implication), and $\Leftrightarrow$ (equivalence) are defined in the usual way. The temporal operator $\circ$ is called the next operator. Formula $\circ\varphi$ specifies that φ will be true in the next step. The temporal operator $\mathcal{U}$ is called the until operator. Formula $\varphi_1 \mathcal{U} \varphi_2$ means that φ_1 must hold until φ_2 holds. Two additional operators, 'eventually' and 'always' are defined as $\diamond\varphi = \mathbf{true}\mathcal{U}\phi$ and $\square\varphi = \neg\diamond\neg\varphi$. Formula $\diamond\varphi$ means that φ becomes eventually true whereas $\square\varphi$ indicates that φ is true at all positions of W. This set of operators can be employed to express many interesting specifications such as system synchronisation (Tabuada and Pappas 2006) and obstacle avoidance (Example 1).

3. Control of multi-affine systems on rectangles

In the previous section, several rectangles have been produced by a rectangular partition of the state space. Now, we investigate the control of multi-affine systems on rectangles. First, the notion of state-based switch multi-affine function is introduced.

Definition 3.1: Given multi-affine functions U: $\mathbb{R}^n \to \mathbb{R}^m$ and U': $\mathbb{R}^n \to \mathbb{R}^m$, $x_f \in \mathbb{R}^n$ and $\varepsilon \in \mathbb{R}^+$, a function $U \diamond U'$: $\mathbb{R}^n \to \mathbb{R}^m$ is said to be a state-based switch multi-affine function from U to U' with respect to x_f and ε if

$$U \diamond U'(x) = \begin{cases} U(x) & \text{if } x \notin B_\varepsilon(x_f) \\ U'(x) & \text{if } x \in B_\varepsilon(x_f) \end{cases}$$

where $B_\varepsilon(x_f) = \{x \mid \|x - x_f\| \leq \varepsilon\}$ with $\| \, \|$ denoting the Euclidean norm.

In this article, the control input for a multi-affine system $\dot{x} = g(x) + Bu$ is in terms of $u = K(x)$, where K is multi-affine function or a state-based switch multi-affine function. Therefore, the feedback law is automatically bounded on $R_{k_1k_2\cdots k_n}$. In the rest of this section, we propose a control method based on the exit sub-region to drive all trajectories of the closed-loop system starting from $R_{k_1k_2\cdots k_n}$ to exit through a desired facet of $R_{k_1k_2\cdots k_n}$, where the exit sub-region is defined as follows.

Definition 3.2: Let $\Sigma : \dot{x} = g(x) + Bu$ be a multi-affine control system, $K(x)$ be a multi-affine feedback controller, $R_{k_1k_2\cdots k_n}$ be a rectangle and $F^{j,d}_{k_1k_2\cdots k_n}$ be a facet of $R_{k_1k_2\cdots k_n}$. A sub-region of $R_{k_1k_2\cdots k_n}$ is called to an exit sub-region with respect to $F^{j,d}_{k_1k_2\cdots k_n}$ and $K(x)$, denoted as $[K]^{j,d}_{k_1k_2\cdots k_n}$, if for any $x_0 \in [K]^{j,d}_{k_1k_2\cdots k_n}$, there exists a $\tau \in \mathbb{R}^+$ such that

(1) $\chi_{x_0,K(x)}(t_1) \in R_{k_1k_2\cdots k_n}$ for $t_1 \in [0, \tau)$;
(2) $\chi_{x_0,K(x)}(t_2) \in F^{j,d}_{k_1k_2\cdots k_n}$ for $t_2 = \tau$;
(3) $\chi_{x_0,K(x)}(t_3) \notin R_{k_1k_2\cdots k_n} \bigcup F^{j,d}_{k_1k_2\cdots k_n}$ for $t_3 \in (\tau, \tau + \varepsilon)$ and $\varepsilon \in \mathbb{R}^+$.

We can see that all trajectories of the closed-loop system $\dot{x} = g(x) + BK(x)$ originating in the sub-region $[K]^{j,d}_{k_1k_2\cdots k_n}$ will leave $R_{k_1k_2\cdots k_n}$ only through $F^{j,d}_{k_1k_2\cdots k_n}$. It implies that if we can find a controller $K'(x)$ such that all trajectories of the closed-loop system $\dot{x} = g(x) + BK'(x)$ starting from $R_{k_1k_2\cdots k_n}$ can reach the exit sub-region $[K]^{j,d}_{k_1k_2\cdots k_n}$ in finite time, then the control of multi-affine systems with respect to the exit facet $F^{j,d}_{k_1k_2\cdots k_n}$ can be realised by using $K(x)$ together with $K'(x)$. That is, we can first apply the controller $K'(x)$ to the multi-affine system and then update the controller to $K(x)$ once the trajectories arrive in $[K]^{j,d}_{k_1k_2\cdots k_n}$. To implement this idea, the following problems should be addressed. Problem 1: how to find a controller $K(x)$ to guarantee the existence of an exit sub-region $[K]^{j,d}_{k_1k_2\cdots k_n}$? Problem 2: if there exists an exit sub-region $[K]^{j,d}_{k_1k_2\cdots k_n}$, how to compute it? Problem 3: how to design a controller $K'(x)$ to drive all trajectories of the closed-loop system starting from $R_{k_1k_2\cdots k_n}$ towards $[K]^{j,d}_{k_1k_2\cdots k_n}$? For Problem 1, we provide the following proposition.

Proposition 3.3: *Given a multi-affine control system* $\Sigma : \dot{x} = g(x) + Bu$, *a multi-affine feedback controller* $K(x)$, *a rectangle* $R_{k_1k_2\cdots k_n}$ *and a facet* $F^{j,d}_{k_1k_2\cdots k_n}$ *of* $R_{k_1k_2\cdots k_n}$, *there exists an exit sub-region* $[K]^{j,d}_{k_1k_2\cdots k_n}$ *with respect to* $F^{j,d}_{k_1k_2\cdots k_n}$ *and* $K(x)$ if

(1) $\exists w \in V(F^{j,d}_{k_1k_2\cdots k_n})$:

$$n^{j,d}[g(w) + BK(w)] > 0; \tag{5}$$

(2) $\forall v \in V(R_{k_1k_2\cdots k_n}) \setminus V(F^{j,d}_{k_1k_2\cdots k_n})$, $\forall F^{j',d'}_{k_1k_2\cdots k_n} \in F(v)$:

$$n^{j',d'}[g(v) + BK(v)] \le 0; \tag{6}$$

(3) $\forall x \in R_{k_1k_2\cdots k_n}$:

$$g(x) + BK(x) \neq 0. \tag{7}$$

Proof: We have $n^{j,d}[g(w) + \mathrm{BK}(w)] > 0$ at the vertex $w \in V(F^{j,d}_{k_1k_2\cdots k_n})$. Because the vector field is continuous, there exist some points at the neighbourhood of w that have strictly positive vector field outwards $R_{k_1k_2\cdots k_n}$ through $F^{j,d}_{k_1k_2\cdots k_n}$. Moreover, (6) implies that the trajectories of the closed-loop system cannot leave through the facets whose vertices all satisfy the condition (6), and (7) implies there does not exist an equilibrium point inside $R_{k_1k_2\cdots k_n}$. We conclude that some trajectories of the closed-loop system starting from $R_{k_1k_2\cdots k_n}$ will leave through $F^{j,d}_{k_1k_2\cdots k_n}$. That is, there is an exit sub-region $[K]^{j,d}_{k_1k_2\cdots k_n}$ of $R_{k_1k_2\cdots k_n}$ with respect to $F^{j,d}_{k_1k_2\cdots k_n}$ and $K(x)$. □

It intuitively states that there exists an exit sub-region $[K]^{j,d}_{k_1k_2\cdots k_n}$ with respect to $F^{j,d}_{k_1k_2\cdots k_n}$ and $K(x)$ if the multi-affine feedback controller $K(x)$ is such that: (1) there exists a vertex w on the exit facet such that the velocity of the closed-loop system $g(w) + BK(w)$ at w has a strictly positive projection along the outer normal of the exit facet; (2) for any vertex v which is not on the exit facet, the velocity of the closed-loop system $g(v) + BK(v)$ at v has a negative projection along the outer normal of the facet containing v; (3) there does not exist an equilibrium point inside $R_{k_1k_2\cdots k_n}$. Thus, Problem 1 is solved. Then, we consider Problem 2, i.e. the computation of the exit sub-region. Before presenting the calculation algorithm, we need the concept of time-elapse cone.

Definition 3.4 (Berman et al. 2007)**:** Given a multi-affine control system $\Sigma\colon \dot{x} = g(x) + Bu$, a multi-affine feedback controller $K(x)$ and a rectangle $R_{k_1k_2\cdots k_n}$, the time-elapse cone for $R_{k_1k_2\cdots k_n}$ with respect to $K(x)$, denoted by $C_{R_{k_1k_2\cdots k_n}, K(x)}$, is defined as

$$C_{R_{k_1k_2\cdots k_n}, K(x)} = \left\{ \sum_{w \in V(R_{k_1k_2\cdots k_n})} \mu_w \, [g(w) + BK(w)] \mid \mu_w \ge 0 \right\}. \tag{8}$$

The following lemma shows that the reachability of multi-affine systems can be estimated by the time-elapse cone.

Lemma 3.5 (Berman et al. 2007)**:** *Given a multi-affine control system* $\Sigma\colon \dot{x} = g(x) + Bu$, *a multi-affine feedback controller* $K(x)$, *a rectangle* $R_{k_1k_2\cdots k_n}$, *a state set* $B \subseteq R_{k_1k_2\cdots k_n}$ *and a reachable set of trajectories* $X_{R_{k_1k_2\cdots k_n}, K(x)}(B) = \{\chi_{x_0,K(x)}(t) \mid x_0 \in B \wedge t \in [0, \tau]\}$ *for* $\tau \in \mathbb{R}^+$ *with respect to* $K(x)$, *then* $X_{R_{k_1k_2\cdots k_n}, K(x)}(B) \subset B \oplus C_{R_{k_1k_2\cdots k_n}, K(x)}$, where $\oplus$ *is the Minkowski sum.*

Similarly, the exit sub-region can be calculated, as it is illustrated in Algorithm 3.6.

Algorithm 3.6 (Computation of exit sub-regions)

Input: a multi-affine control system $\Sigma\colon \dot{x} = g(x) + Bu$, a multi-affine feedback controller $K(x)$, a rectangle $R_{k_1k_2\cdots k_n}$, a facet $F^{j,d}_{k_1k_2\cdots k_n}$ of $R_{k_1k_2\cdots k_n}$ and an accuracy limitation ε.
Output: an exit sub-region $[K]^{j,d}_{k_1k_2\cdots k_n}$ with respect to $F^{j,d}_{k_1k_2\cdots k_n}$ and $K(x)$.
For any $R_{k'_1k'_2\cdots k'_n} = \prod_{i=1}^{n}(a_i^{k'_i}, b_i^{k'_i})$, we define the following functions:

$$\mathcal{L}(R_{k'_1k'_2\cdots k'_n}) := \max_{i\in\{1,2,\ldots,n\}} \left(b_i^{k'_i} - a_i^{k'_i}\right);$$

$$\mathcal{P}(R_{k'_1k'_2\cdots k'_n}) := \bigcup_{m,p,\cdots l\in\{1,2\}} R_{k'_{1_m}k'_{2_p}\cdots k'_{n_l}} = \bigcup_{m,p,\cdots l\in\{1,2\}} \left(a_1^{k'_{i_m}}, a_1^{k'_{i_{m+1}}}\right) \times \left(a_2^{k'_{i_p}}, a_2^{k'_{i_{p+1}}}\right) \times \cdots \times \left(a_n^{k'_{i_l}}, a_n^{k'_{i_{l+1}}}\right),$$

where $a_i^{k'_{i_1}} = a_i^{k'_i}$, $a_i^{k'_{i_2}} = \frac{a_i^{k'_i} + b_i^{k'_i}}{2}$, and $a_i^{k_{i_3}} = b_i^{k'_i}$.
Let $pR_{exit} = \phi$;
if $(\exists w \in V(F^{j,d}_{k_1k_2\cdots k_n}) : n^{j,d}[g(w) + BK(w)] > 0$ and $\forall v \in V(R_{k_1k_2\cdots k_n}) \setminus V(F^{j,d}_{k_1k_2\cdots k_n})$, $\forall F^{j',d'}_{k_1k_2\cdots k_n} \in F(v) : n^{j',d'}[g(v) + BK(v)] \le 0)$

if $(\forall v \in V(R_{k_1k_2\cdots k_n}) \setminus V(F^{j,d}_{k_1k_2\cdots k_n}) : n^{j,d}[g(v)+BK(v)] > 0$ and $\forall w \in V(F^{j,d}_{k_1k_2\cdots k_n})$, $\forall F^{j',d'}_{k_1k_2\cdots k_n} \in F(w)$ with $(j,d) \neq (j',d')$: $n^{j,d}[g(w)+BK(w)] > 0 \wedge n^{j',d'}[g(w)+BK(w)] \leq 0)$
$pR_{Exit} = \{R_{k_1k_2\cdots k_n}\}$;
else if $(g(x)+K(x) \neq 0$ for any $x \in R_{k_1k_2\cdots k_n})$
Let $Rect = \{R_{k_1k_2\cdots k_n}\}$;
if $(\mathcal{L}(R) > \varepsilon$ for any $R \in Rect)$
$Rect = \bigcup_{R \in Rect} \mathcal{P}(R)$;
end if
for all $R' \in Rect$, **do** $S = R' \oplus C_{R_{k_1k_2\cdots k_n}, K(x)}$
if $(S \cap F^{j,d}_{k_1k_2\cdots k_n} \neq \emptyset \wedge S \cap (F(R_{k_1k_2\cdots k_n}) \setminus F^{j,d}_{k_1k_2\cdots k_n}) = \emptyset)$
$pR_{Exit} = pR_{Exit} \bigcup R'$;
end if
end for
end if
$pR_{Exit} = [K]^{j,d}_{k_1k_2\cdots k_n}$ is an exit sub-region with respect to $F^{j,d}_{k_1k_2\cdots k_n}$ and $K(x)$.
end if

Proposition 3.7: *Algorithm* 3.6 *is correct*.

Proof: Since (5)–(7) are satisfied, there exists an exit sub-region with respect to $F^{j,d}_{k_1k_2\cdots k_n}$ and $K(x)$ from Proposition 3.3. Let R' be a rectangle obtained by dividing $R_{k_1k_2\cdots k_n}$, i.e. $R' \in \mathcal{P}(R_{k_1k_2\cdots k_n})$ and $X_{R_{k_1k_2\cdots k_n}, K(x)}(R')$ be all trajectories of the closed-loop system $\dot{x} = g(x) + BK(x)$ starting from R'. We have $X_{R_{k_1k_2\cdots k_n}, K(x)}(R') \subset R' \oplus C_{R_{k_1k_2\cdots k_n}, K(x)}$ according to Lemma 3.5. Here we use the facts: (1) $(R' \oplus C_{R_{k_1k_2\cdots k_n}, K(x)}) \cap F^{j,d}_{k_1k_2\cdots k_n} \neq \emptyset$; (2) $(R' \oplus C_{R_{k_1k_2\cdots k_n}, K(x)}) \cap (F(R_{k_1k_2\cdots k_n}) \setminus F^{j,d}_{k_1k_2\cdots k_n}) = \emptyset$; (3) there does not exist an equilibrium point inside $R_{k_1k_2\cdots k_n}$. It follows that all trajectories of the closed-loop system starting from R' exit only through $F^{j,d}_{k_1k_2\cdots k_n}$. Therefore, $pR_{Exit} = [K]^{j,d}_{k_1k_2\cdots k_n}$ is an exit sub-region with respect to $F^{j,d}_{k_1k_2\cdots k_n}$ and $K(x)$. □

Next, we present the result for Problem 3.

Proposition 3.8 (Control to a fixed point)**:** *Given a multi-affine control system* $\Sigma: \dot{x} = g(x) + Bu$, *a rectangle* $R_{k_1k_2\cdots k_n}$ *and a desired point* $x_f \in R_{k_1k_2\cdots k_n}$, *there exists a multi-affine feedback controller* $K'(x)$ *such that* $u = K'(x)$ *and all trajectories of the closed-loop system starting from* $R_{k_1k_2\cdots k_n}$ *remain in* $R_{k_1k_2\cdots k_n}$ *for all times and converge to* x_f *if for any* $w \in V(R_{k_1k_2\cdots k_n})$, $U_I(w) \neq \emptyset$ *holds and there exists an* $u'(w) \in U_I(w)$ *such that* x_f *is a unique point in* $R_{k_1k_2\cdots k_n}$:

$$g(x_f) + B \sum_{w \in V(R_{k_1k_2\cdots k_n})} \lambda_w(x_f) u'(w) = 0. \tag{9}$$

Proof: Because $U_I(w) \neq \emptyset$ for any $w \in V(R_{k_1k_2\cdots k_n})$, there exists a multi-affine feedback controller such that all trajectories of the closed-loop system starting from $R_{k_1k_2\cdots k_n}$ remain in $R_{k_1k_2\cdots k_n}$ for all times by Lemma 2.4. Let $u'(w) \in U_I(w)$ be the control input at w such that x_f is a unique point in $R_{k_1k_2\cdots k_n}$ satisfying (9). Then, we design $K'(x) = \sum_{w \in V(R_{k_1k_2\cdots k_n})} \lambda_w(x) u'(w)$. For all rectangle $\alpha R_{k_1k_2\cdots k_n}$, where $\alpha \in [0,1]$, the vertex set $V(\alpha R_{k_1k_2\cdots k_n}) = \{\alpha w + (1-\alpha)x_f\}$. It can be seen that $\alpha R_{k_1k_2\cdots k_n}$ is just a shrunken version of $R_{k_1k_2\cdots k_n}$ by multiplying $R_{k_1k_2\cdots k_n}$ from x_f by the factor α. Thus, the velocity vector of the closed-loop system at the vertex of $\alpha R_{k_1k_2\cdots k_n}$ is just α-multiple the velocity vector at the corresponding vertex of $R_{k_1k_2\cdots k_n}$. Since the vector field of the closed-loop system in all vertices of $\alpha R_{k_1k_2\cdots k_n}$ is pointing inside to $\alpha R_{k_1k_2\cdots k_n}$, there exist $t_0 > 0$ and $\alpha' \in [0, 1)$ such that $\chi_{w,K'(x)}(t_0) \in \alpha' R_{k_1k_2\cdots k_n}$. Then, $\chi_{x_0,K'(x)}(t) \in \alpha' R_{k_1k_2\cdots k_n}$ for all $x_0 \in R_{k_1k_2\cdots k_n}$ and $t \geq t_0$. Similarly, we obtain $\chi_{x_0,K'(x)}(t) \in (\alpha')^n R_{k_1k_2\cdots k_n}$ for $t \geq \mathrm{n}t_0$. Therefore, $\lim_{t\to\infty} \chi_{x_0,K'(x)}(t) = x_f$. □

It indicates that if we can construct a controller of the form $u = K'(x) = \sum_{w \in V(R_{k_1k_2\cdots k_n})} \lambda_w(x) u'(w)$, where $u'(w) \in U_I(w) \neq \emptyset$, such that x_f is a unique equilibrium point inside $R_{k_1k_2\cdots k_n}$, then all trajectories of the closed-loop system starting from $R_{k_1k_2\cdots k_n}$ are driven towards x_f. This kind of multi-affine function K' is called a fixed point controller with respect to x_f. By putting x_f inside the exit sub-region $[K]^{j,d}_{k_1k_2\cdots k_n}$, the fixed point controller yields a solution for Problem 3. Now, we are ready to present the result on the control with respect to a desired exit facet.

Proposition 3.9 (Control to an exit facet)**:** *Given a multi-affine control system* $\Sigma: \dot{x} = g(x) + Bu$, *a rectangle* $R_{k_1k_2\cdots k_n}$ *and a facet* $F^{j,d}_{k_1k_2\cdots k_n}$ *of* $R_{k_1k_2\cdots k_n}$, *there exists a feedback controller such that all trajectories of the closed-loop system starting from* $R_{k_1k_2\cdots k_n}$ *are driven only through* $F^{j,d}_{k_1k_2\cdots k_n}$ *in finite time if any of the following two conditions is satisfied*:

(1) $U_E(w) \neq \emptyset$ *holds for any* $w \in V(R_{k_1k_2\cdots k_n})$;
(2) $U_E(w) \neq \emptyset$ *does not hold for any* $w \in V(R_{k_1k_2\cdots k_n})$ *and there exist* $x_f \in R_{k_1k_2\cdots k_n}$, $\varepsilon \in \mathbb{R}^+$ *and multi-affine functions* U *and* U' *such that* $B_\varepsilon(x_f) \subseteq [U]^{j,d}_{k_1k_2\cdots k_n}$ *and* U' *is a fixed point controller with respect to* x_f.

Proof: As for condition (1), it obviously guarantees the existence of a controller with respect to an exit facet according to Lemma 2.5. As for condition (2), because U' is a fixed point controller with respect to x_f, all trajectories of the closed-loop system $\dot{x} = g(x) + BU'(x)$ starting from $R_{k_1k_2\cdots k_n}$ will converge towards x_f. Moreover, there is an $\varepsilon \in \mathbb{R}^+$ such that $B_\varepsilon(x_f) \subseteq [U]^{j,d}_{k_1k_2\cdots k_n}$, where $[U]^{j,d}_{k_1k_2\cdots k_n}$ is an exit sub-region with respect to $F^{j,d}_{k_1k_2\cdots k_n}$ and $U(x)$. By using the state-based switch multi-affine feedback controller $U' \diamond U(x)$ (w.r.t. x_f and ε), all trajectories of the corresponding closed-loop system starting from $R_{k_1k_2\cdots k_n}$ will exit only through $F^{j,d}_{k_1k_2\cdots k_n}$ in finite time. □

Proposition 3.9 provides two different ways to drive the trajectories of the corresponding closed-loop system starting from $R_{k_1k_2\cdots k_n}$ to exit only through a desired facet. One (condition (1)) is based on the result of Lemma 2.5 and the other (condition (2)) is based on the exit sub-region. Thus, the proposed control method for an exit facet covers more classes of systems than those and addressed in Belta and Habets (2006) and Habets et al. (2006). We call the multi-affine function or the state-based switch multi-affine function U, which drives all trajectories of the closed-loop system starting from $R_{k_1k_2\cdots k_n}$ to exit only through $F^{j,d}_{k_1k_2\cdots k_n}$ as an exit controller with respect to $F^{j,d}_{k_1k_2\cdots k_n}$. Such an exit controller can be obtained by the following algorithm.

Algorithm 3.10 (Synthesis of exit controllers)

Input: a multi-affine control system Σ: $\dot{x} = g(x) + Bu$, a rectangle $R_{k_1k_2\cdots k_n}$, a facet $F^{j,d}_{k_1k_2\cdots k_n}$ of $R_{k_1k_2\cdots k_n}$ and $|u| \le \eta$.
Output: an exit controller with respect to $F^{j,d}_{k_1k_2\cdots k_n}$.
Let $V(R_{k_1k_2\cdots k_n}) = \{w_j \mid j = 1, 2, \ldots, 2^n\}$.
if $(U_E(w_j) \neq \emptyset$ for any $w_j \in V(R_{k_1k_2\cdots k_n}) \setminus V(F^{j,d}_{k_1k_2\cdots k_n}))$
  Let $V_1 := \{j \in \{1, 2, \ldots, 2^n\} \mid U_E(w_j) = \emptyset, w_j \in V(R_{k_1k_2\cdots k_n})\}$.
  if $(V_1 = \emptyset)$

$$U_1 := \left\{ \{U^1_{R_{k_1k_2\cdots k_n}}(w_j) \mid j = 1, 2, \ldots, 2^n\} \;\middle|\; w_j \in V(R_{k_1k_2\cdots k_n}) \wedge U^1_{R_{k_1k_2\cdots k_n}}(w_j) \in U_E(w_j) \wedge \left| \sum_{j=1,2,\ldots,2^n} \lambda_{w_j}(x) U^1_{R_{k_1k_2\cdots k_n}}(w_j) \right| \le \eta, x \in R_{k_1k_2\cdots k_n} \right\},$$

  if $(U_1 \neq \emptyset)$

$$U^1_{R_{k_1k_2\cdots k_n}}(x) = \sum_{j=1,2,\ldots,2^n} \lambda_{w_j}(x) U^1_{R_{k_1k_2\cdots k_n}}(w_j),$$

   where $\{U^1_{R_{k_1k_2\cdots k_n}}(w_j) \mid j = 1, 2, \ldots, 2^n\} \in U_1$.
   The multi-affine function $U^1_{R_{k_1k_2\cdots k_n}}$ is an exit controller with respect to $F^{j,d}_{k_1k_2\cdots k_n}$
  end if
 else if $(V_1 \subset \{1, 2, \ldots, 2^n\})$
  if $(U_I(w_j) \neq \emptyset$ for any $w_j \in V(R_{k_1k_2\cdots k_n}))$

$$U_3 := \left\{ \{U^3_{R_{k_1k_2\cdots k_n}}(w_j) \mid j = 1, 2, \ldots, 2^n\} \;\middle|\; w_j \in V(R_{k_1k_2\cdots k_n}) \wedge U^3_{R_{k_1k_2\cdots k_n}}(w_j) \in U_I(w_j) \wedge \left| \sum_{j=1,2,\ldots,2^n} \lambda_{w_j}(x) U^3_{R_{k_1k_2\cdots k_n}}(w_j) \right| \le \eta, \quad x \in R_{k_1k_2\cdots k_n} \right\};$$

   if $(U_3 \neq \emptyset)$

$$U_2 := \left\{ \{U^2_{R_{k_1k_2\cdots k_n}}(w_j) \mid j = 1, 2, \ldots, 2^n\} \;\middle|\; n^{j,d}[g(w_m) + BU^2_{R_{k_1k_2\cdots k_n}}(w_m)] > 0 \wedge w_j \in V(R_{k_1k_2\cdots k_n}) \wedge U^2_{R_{k_1k_2\cdots k_n}}(w_l) \in U_E(w_l) \wedge \left| \sum_{j=1,2,\ldots,2^n} \lambda_{w_j}(x) U^2_{R_{k_1k_2\cdots k_n}}(w_j) \right| \ge \eta, m \in V_1, l \in \{1, 2, \ldots, 2^n\} \setminus V_1 \text{ and } x \in R_{k_1k_2\cdots k_n} \right\},$$

    if $(U_2 \neq \emptyset)$
     for all $\{U^2_{R_{k_1k_2\cdots k_n}}(w_j) \mid j = 1, 2, \ldots, 2^n\} \in U_2$
     do

$$U^2_{R_{k_1k_2\cdots k_n}}(x) = \sum_{j=1,2,\ldots,2^n} \lambda_{w_j}(x) U^2_{R_{k_1k_2\cdots k_n}}(w_j);$$

      Obtain the exit sub-region $[U^2_{R_{k_1k_2\cdots k_n}}]^{j,d}$ w.r.t. $F^{j,d}_{k_1k_2\cdots k_n}$ and $U^2_{R_{k_1k_2\cdots k_n}}(x)$;
      for all $\{U^3_{R_{k_1k_2\cdots k_n}}(w_j) \mid j = 1, 2, \ldots, 2^n\} \in U_3$
      do

$$U^3_{R_{k_1k_2\cdots k_n}}(x) = \sum_{j=1,2,\ldots,2^n} \lambda_{w_j}(x) U^3_{R_{k_1k_2\cdots k_n}}(w_j);$$

       if $(\exists\, \varepsilon \in \mathbb{R}^+$ and a unique point $x' \in R_{k_1k_2\cdots k_n}$ s.t. $g(x') + BU^3_{R_{k_1k_2\cdots k_n}}(x') = 0$ and $B_\varepsilon(x') \subseteq [U^2_{R_{k_1k_2\cdots k_n}}]^{j,d})$;
        The state-based switch multi-affine function $U^3_{R_{k_1k_2\cdots k_n}} \diamond U^2_{R_{k_1k_2\cdots k_n}}$ w.r.t. x_f and ε is an exit controller for $F^{j,d}_{k_1k_2\cdots k_n}$.
       end if
      end for
     end for
    end if
   end if
  end if
 end if
end if

Proposition 3.11: *Algorithm* 3.10 *is correct*.

Proof: The proof is obvious according to Proposition 3.9. □

4. Finitely abstracted transition systems of multi-affine systems

The control of multi-affine systems on rectangles enables the construction of a finitely abstracted transition system for the multi-affine system, as illustrated in Definition 4.1. Here we assume that any initial state of the multi-affine system is inside the rectangles and the duration of the trajectories staying on the boundary of the rectangle is ignored.

These assumptions result in no loss of generality since they always hold in the implementation.

Definition 4.1: Given a multi-affine control system $\Sigma: \dot{x} = g(x) + Bu$ and a rectangle set ξ generated by rectangularly partitioning the state space, the abstracted transition system of Σ associated with ξ, denoted as $S_{\Sigma,\xi}$, is a tuple

$$S_{\Sigma,\xi} = (X_\xi, X_{\xi 0}, U_\xi, \rightarrow_\xi, X_{m\xi}, Y_\xi, H_\xi)$$

- $X_\xi = \xi = X_{m\xi}$;
- $X_{\xi 0} = \{R_{k_1 k_2 \cdots k_n} \in \xi \mid R_{k_1 k_2 \cdots k_n}$ contains an initial state of the multi-affine control system$\}$;
- $U_\xi = \{U_{R_{k_1 k_2 \cdots k_n}} \mid U_{R_{k_1 k_2 \cdots k_n}}$ is a multi-affine function or a state-based switch multi-affine function, $R_{k_1 k_2 \cdots k_n} \in \xi\}$;
- $R_{k_1 k_2 \cdots k_n} \overset{U_{R_{k_1 k_2 \cdots k_n}}}{\rightarrow}_\xi R_{k'_1 k'_2 \cdots k'_n}$ if any of the following two conditions is satisfied:

  (1) $R_{k_1 k_2 \cdots k_n} = R_{k'_1 k'_2 \cdots k'_n}$ holds and for any $w \in V(R_{k_1 k_2 \cdots k_n})$, $U_I(w) \neq \emptyset$ and $U_{R_{k_1 k_2 \cdots k_n}}(w) \in U_I(w)$.

  (2) $R_{k_1 k_2 \cdots k_n} \neq R_{k'_1 k'_2 \cdots k'_n}$ with $\overline{R_{k_1 k_2 \cdots k_n}} \cap \overline{R_{k'_1 k'_2 \cdots k'_n}} = F^{j,d}_{k_1 k_2 \cdots k_n}$ holds and $U_{R_{k_1 k_2 \cdots k_n}}$ is an exit controller with respect to $F^{j,d}_{k_1 k_2 \cdots k_n}$.

- $Y_\xi = \xi$;
- $H_\xi(R_{k_1 k_2 \cdots k_n}) = R_{k_1 k_2 \cdots k_n}$.

An abstracted transition system is a finite-state system, therefore it facilitates the synthesis of the controller for finite-state requirements while accommodating to infinite-state dynamics. Next, a rectangular transition system of the multi-affine control system is established, and it can be understood as a transition system form of the multi-affine control system over a rectangularly partitioned state space.

Definition 4.2: Given a multi-affine control system $\Sigma: \dot{x} = g(x) + Bu$, a rectangle set ξ generated by rectangularly partitioning the state space and a rectangular project map π_Q defined by ξ, the rectangular transition system of Σ associated with ξ, denoted as $S_{\Sigma,Q}$, is a tuple

$$S_{\Sigma,Q} = (X_Q, X_{Q0}, U_Q, \rightarrow_Q, X_{mQ}, Y_Q, H_Q)$$

- $X_Q = \mathbb{R}^n = X_{mQ}$;
- $X_{Q0} = \{x \mid x$ is an initial state of the multi-affine control system$\}$;
- $U_Q = \{k \mid k(x)$ is a feedback control law$\}$;
- $x \overset{k}{\rightarrow}_Q x'$ if any of the following two conditions is satisfied:

(1) $\pi_Q(x) = \pi_Q(x')$ holds and there exists $\tau \in \mathbb{R}^+$ such that $\chi_{x,k(x)}(\tau) = x'$ and $\pi_Q(\chi_{x,k(x)}(t)) = \pi_Q(x)$, where $t \in [0, +\infty)$.

(2) $\pi_Q(x) \neq \pi_Q(x')$ holds and there exist $\tau, \epsilon \in \mathbb{R}^+$ such that $\chi_{x,k(x)}(\tau) = x'$, $\pi_Q(\chi_{x,k(x)}(t_1)) = \pi_Q(x)$ and $\pi_Q(\chi_{x,k(x)}(t_2)) = \pi_Q(x')$, where $t_1 \in [0, \epsilon)$ and $t_2 \in [\epsilon, \tau]$.

- $Y_Q = \xi$;
- $H_Q = \pi_Q$.

It can be seen that the construction of $S_{\Sigma,Q}$ relies on π_Q to define both the transitions and the outputs. To describe the relationship between the rectangular transition system and the abstracted transition system, we provide the notion of input–output simulation relation.

Definition 4.3: Given transition systems $S_a = (X_a, X_{a0}, U_a, \rightarrow_a, X_{ma}, Y_a, H_a)$ and $S_b = (X_b, X_{b0}, U_b, \rightarrow_b, X_{mb}, Y_b, H_b)$, an input–output simulation relation is a binary relation $\phi \subseteq X_a \times X_b$ such that $(x_a, x_b) \in \phi$ implies

(1) $H_a(x_a) = H_b(x_b)$;
(2) $(\forall u \in U_a)[x_a \overset{u}{\rightarrow}_a x'_a \Rightarrow \exists x'_b \; s.t. \; x_b \overset{u}{\rightarrow}_b x'_b$ and $(x'_a, x'_b) \in \phi]$.

A transition system S_a is said to be input–output simulated by S_b, denoted as $S_a \prec_{Io(\phi)} S_b$, if there is an input–output simulation relation ϕ from S_a to S_b such that for any $x_a \in X_{a0}$, there exists an $x_b \in X_{b0}$ with $(x_a, x_b) \in \phi$. The subscript (ϕ) is sometimes omitted from $\prec_{Io(\phi)}$ when it is clear from the context. The introduced input–output simulation relation requires input equivalence as well as output equivalence, which is stronger than the simulation relations requiring either of them (Milner 1989; Tabuada and Pappas 2006). However, it has the following advantages. First, it is natural since the observation of the system depends on the output. Second, it suggests that the control input, enforcing a desired behaviour with respect to the transition system S_a, is also applicable to its input–output similar transition system S_b. When S_a is input–output simulated by S_b, the behaviours of S_a such as finite/infinite language, accepted language and finite/infinite path language are included in the respective behaviours of S_b, which is shown in the following lemma.

Lemma 4.4: *If there exists an input–output simulation relation ϕ such that $S_a \prec_{Io(\phi)} S_b$, then $L(S_a) \subseteq L(S_b)$, $L^w(S_a) \subseteq L^w(S_b)$, $L^w_A(S_a) \subseteq L^w_A(S_b)$, $L_P(S_a) \subseteq L_P(S_b)$ and $L^w_P(S_a) \subseteq L^w_P(S_b)$.*

Besides language inclusion, input–output simulation preserves properties expressed in LTL, which will be discussed in Section 5. Next, we illustrate that the abstracted transition system is input–output simulated by the rectangular transition system.

Theorem 4.5: *Given a multi-affine control system $\Sigma : \dot{x} = g(x) + Bu$, a rectangle set ξ generated by rectangularly partitioning the state space and a rectangular project map π_Q defined by ξ, the relation ϕ defined as*

$$\phi = \{(R_{k_1k_2\cdots k_n}, x) \in \xi \times \mathbb{R}^n \mid x \in R_{k_1k_2\cdots k_n}\}$$

*is an input–*output *simulation relation from $S_{\Sigma,\xi}$ to $S_{\Sigma,Q}$.*

Proof: For any $(R_{k_1k_2\cdots k_n}, x) \in \phi$, we have $H_\xi(R_{k_1k_2\cdots k_n}) = R_{k_1k_2\cdots k_n} = H_Q(x) = \pi_Q(x)$. Further, if there is a transition $R_{k_1k_2\cdots k_n} \overset{U_{R_{k_1k_2\cdots k_n}}}{\rightarrow}{}_\xi R_{k_1'k_2'\cdots k_n'}$, we have the following two cases: (a) $R_{k_1k_2\cdots k_n} \neq R_{k_1'k_2'\cdots k_n'}$ with $F^{j,d}_{k_1k_2\cdots k_n} = \overline{R_{k_1k_2\cdots k_n}} \cap \overline{R_{k_1'k_2'\cdots k_n'}}$. According to the construction of $S_{\Sigma,\xi}$, there exists a controller $U_{R_{k_1k_2\cdots k_n}}$ such that all trajectories of the closed-loop system $\dot{x} = g(x) + BU_{R_{k_1k_2\cdots k_n}}(x)$ starting from $R_{k_1k_2\cdots k_n}$ are driven only through $F^{j,d}_{k_1k_2\cdots k_n}$. Then, for any $x \in R_{k_1k_2\cdots k_n}$, there is $x' \in R_{k_1'k_2'\cdots k_n'}$ such that $x \overset{U_{R_{k_1k_2\cdots k_n}}}{\rightarrow}{}_Q x'$ and $(R_{k_1'k_2'\cdots k_n'}, x') \in \phi$. (b) $R_{k_1k_2\cdots k_n} = R_{k_1'k_2'\cdots k_n'}$. The controller $U_{R_{k_1k_2\cdots k_n}}$ satisfying $U_{R_{k_1k_2\cdots k_n}}(w) \in U_I(w) \neq \emptyset$ for any $w \in V(R_{k_1k_2\cdots k_n})$ drives all trajectories of the closed-loop system $\dot{x} = g(x) + BU_{R_{k_1k_2\cdots x_n}}(x)$ starting from $R_{k_1k_2\cdots k_n}$ to remain in $R_{k_1k_2\cdots k_n}$ for all times (Belta and Habets 2006). Therefore, there exists an $x' \in R_{k_1k_2\cdots k_n}$ such that $x \overset{U_{R_{k_1k_2\cdots k_n}}}{\rightarrow}{}_Q x'$ and $(R_{k_1'k_2'\cdots k_n'}, x') \in \phi$. Moreover, the definition of $X_{\xi 0}$ indicates that for any $R_{k_1k_2\cdots k_n} \in X_{\xi 0}$, there exists an $x \in X_{Q0}$ such that $(R_{k_1k_2\cdots k_n}, x) \in \phi$. As a result, $S_{\Sigma,\xi} \prec_{\mathrm{Io}(\phi)} S_{\Sigma,Q}$. □

5. Controller synthesis for LTL specifications

This section studies the controller synthesis for LTL specifications. It is well known that an LTL formula φ over a proposition set Π can be effectively converted into a Büchi automaton which accepts every infinite string over Π satisfying φ (Wolper, Vardi, and Sistla 1983). This kind of Büchi automaton is described as follows.

Definition 5.1: Given an LTL formula φ over a proposition set Π, the Büchi automaton with respect to φ, denoted as $\mathcal{B}_\varphi$, is a tuple

$$\mathcal{B}_\varphi = (B, B_0, 2^\Pi, \rightarrow_B, B_m)$$

- B, $B_0 \subseteq B$ and $B_m \subseteq B$ are finite sets of states, initial states and marked states, respectively;
- 2^Π is an input alphabet;
- $\rightarrow_B \subseteq B \times 2^\Pi \times 2^B$ is a transition relation.

Since the abstracted transition system $S_{\Sigma,\xi}$ is input–output simulated by the rectangular transition system $S_{\Sigma,Q}$, if there exists a supervisor (discrete controller) S_c for $S_{\Sigma,\xi}$ enforcing the LTL specifications, then such a supervisor also works for $S_{\Sigma,Q}$, i.e. the implementation of S_c drives the multi-affine system to fulfil the LTL specifications. Thus, we first focus on the synthesis of S_c. Here a supervisor conducts the control through restricting the behaviours of the transition system, which is captured by the following notion.

Definition 5.2: Given transition systems $S_a = (X_a, X_{a0}, U_a, \rightarrow_a, X_{ma}, Y_a, H_a)$ and $S_b = (X_b, X_{b0}, U_b, \rightarrow_b, X_{mb}, Y_b, H_b)$, the input–output parallel composition of S_a and S_b, denoted as $S_a \|_{Io} S_b$, is a transition system

$$S_a \|_{Io} S_b = (X_{ab}, X_{ab0}, U_{ab}, \rightarrow_{ab}, X_{mab}, Y_{ab}, H_{ab})$$

- $X_{ab} = \{(x_a, x_b) \in X_a \times X_b \mid H_a(x_a) = H_b(x_b)\}$;
- $X_{ab0} = (X_{a0} \times X_{b0}) \cap X_{ab}$;
- $U_{ab} = U_a \cap U_b$;
- $(x_a, x_b) \overset{u}{\rightarrow}_{ab} (x_a', x_b')$ iff $x_a \overset{u}{\rightarrow}_a x_a'$ and $x_b \overset{u}{\rightarrow}_b x_b'$;
- $X_{mab} = (X_{ma} \times X_{mb}) \cap X_{ab}$;
- $Y_{ab} = Y_a \cap Y_b$;
- $H_{ab}(x_a, x_b) = H_a(x_a) = H_b(x_b)$.

The presented input–output parallel composition is different from the usual synchronisation operator in the supervisory control literature, as besides a same control symbol $\overset{u}{\rightarrow}$ between the synchronised transitions $\overset{u}{\rightarrow}_a$ and $\overset{u}{\rightarrow}_b$, it also requires identical output values $H_a(x_a) = H_b(x_b)$ between the state pairs. Thus, the behaviours (finite/infinite language, accepted language and finite/infinite path language) of $S_a \|_{Io} S_b$ are contained in those of S_b. It follows that the supervisor S_c can restrict the behaviours of $S_{\Sigma,\xi}$ which do not satisfy the LTL specifications. This observation motivates us to construct the supervisor S_c by working with $S_{\Sigma,\xi}$ and $\mathcal{B}_\varphi$. Hence, we introduce the notion of product automaton.

Definition 5.3: Given an abstracted transition system $S_{\Sigma,\xi} = (X_\xi, X_{\xi 0}, U_\xi, \rightarrow_\xi, X_{m\xi}, Y_\xi, H_\xi)$, a Büchi automaton $\mathcal{B}_\varphi = (B, B_0, 2^\Pi, \rightarrow_B, B_m)$ and a label function L: $Y_\xi \rightarrow 2^\Pi$, the product automaton of $S_{\Sigma,\xi}$ and $\mathcal{B}_\varphi$, denoted as $S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi$, is a transition system

$$S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi = (A, A_0, U_A, \rightarrow_A, A_m, Y_A, H_A)$$

- $A = X_\xi \times B$;
- $A_0 = \{(x_\xi, b) \in X_{\xi 0} \times B \mid \exists\, b_0 \in B_0\text{: } b_0 \overset{L(H_\xi(x_\xi))}{\rightarrow}{}_B b\}$;
- $U_A = U_\xi$;
- $(x_\xi, b) \overset{u}{\rightarrow}_A (x_\xi', b')$ iff $x_\xi \overset{u}{\rightarrow}_\xi x_\xi'$ and $b \overset{L(H_\xi(x_\xi'))}{\rightarrow}{}_B b'$;
- $A_m = X_{m\xi} \times B_m$;
- $Y_A = Y_\xi$;
- $H_A(x_\xi, b) = H_\xi(x_\xi) = x_\xi$.

The result provided by de Giacomo and Vardi (2000) indicates that a string r satisfies the LTL formula φ iff $r \in L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)$. In other words, if the supervised system is an accepted language equivalent to the product automaton, then it satisfies the LTL formula φ. Let $S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi$ be the supervisor for $S_{\Sigma,\xi}$ (it also works for $S_{\Sigma,Q}$). Then, $L_A^w((S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)||_{Io} S_{\Sigma,\xi}) = L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi) \subseteq L_A^w(S_{\Sigma,\xi})$, implying the supervised system $(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)\|_{Io} S_{\Sigma,\xi}$ satisfies φ. However, there might exist some strings in the language of the supervised system that cannot be the prefixes of the accepted langauge of the product automaton, i.e. $L((S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)||_{Io} S_\xi) \neq \overline{L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)}$. It will cause blocking in the execution. To prevent the blocking, we need the following operator.

Definition 5.4: Given a transition system $S = (E, E_0, U, \rightarrow, E_m, Y, H)$, the coaccessible operator on S, denoted as $\mathrm{CoAc}(S)$, is a transition system

$$\mathrm{CoAc}(S) = (E_{\mathrm{co}}, E_{\mathrm{co}0}, U, \rightarrow_{\mathrm{co}}, E_{m\mathrm{co}}, Y_{\mathrm{co}}, H_{\mathrm{co}}),$$

where $E_{\mathrm{co}} = \{y \in E \mid \exists s \in U^* \text{ and } y' \in E_m : y \xrightarrow{s} y'\}$, $E_{\mathrm{co}0} = E_0 \cap E_{\mathrm{co}}$, $E_{m\mathrm{co}} = E_m \cap E_{\mathrm{co}}$, $\rightarrow_{\mathrm{co}} = \rightarrow|_{E_{\mathrm{co}} \times \Sigma \times E_{\mathrm{co}}}$, $Y_{\mathrm{co}} = \{H(y) \mid y \in E_{\mathrm{co}}\}$ and $H_{\mathrm{co}} = H|_{E_{\mathrm{co}}}$.

It can be seen that $L_A^w(\mathrm{CoAc}(S)) = L_A^w(S)$ and $L(\mathrm{CoAc}(S)) = \overline{L_A^w(S)}$. Thus, when $\mathrm{CoAc}(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)$ is chosen to be the supervisor S_c, it guarantees the accepted language equivalence while preventing the blocking, as stated in the following theorem.

Theorem 5.5: *Given a rectangular transition system $S_{\Sigma,Q}$ and a product automaton $S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi$, there exists a supervisor S_c for $S_{\Sigma,Q}$ such that $L_A^w(S_c||_{Io} S_{\Sigma,Q}) = L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)$ and $L(S_c||_{Io} S_{\Sigma,Q}) = \overline{L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)}$ if $L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi) \neq \emptyset$.*

Proof: Since $L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi) \neq \emptyset$, let $S_c = \mathrm{CoAc}(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)$. We use the facts: (1) $L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi) \subseteq L_A^w(S_{\Sigma,\xi})$ and $L_P^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi) \subseteq L_P^w(S_{\Sigma,\xi})$; (2) $S_{\Sigma,\xi} \prec_{\mathrm{Io}} S_{\Sigma,Q}$ implies $L_A^w(S_{\Sigma,\xi}) \subseteq L_A^w(S_{\Sigma,Q})$ and $L_P^w(S_{\Sigma,\xi}) \subseteq L_P^w(S_{\Sigma,Q})$ and (3) $L_A^w(\mathrm{CoAc} \times (S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)) = L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)$. Thus, $L_A^w(S_c||_{Io} S_{\Sigma,Q}) = L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi) \cap L_A^w(S_{\Sigma,Q}) = L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)$. Moreover, we have $L(\mathrm{CoAc}(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)) = \overline{L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)}$, $L(\mathrm{CoAc}(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)) \subseteq L(S_{\Sigma,Q})$ and $L_P(\mathrm{CoAc} \times (S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)) \subseteq L_P(S_{\Sigma,Q})$, it follows that

$$\begin{aligned} &L(S_c||_{Io} S_{\Sigma,Q}) \\ &\quad = L(\mathrm{CoAc}(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)) \cap L(S_{\Sigma,Q}) \\ &\quad = L(\mathrm{CoAc}(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)) = \overline{L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)}. \end{aligned}$$

$\square$

Remark 5.6: The proof of Theorem 5.5 is constructive as if $L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi) \neq \emptyset$, $S_c = \mathrm{CoAc}(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)$ provides a supervisor to achieve the LTL formula φ $(L_A^w(S_c||_{Io} S_{\Sigma,Q}) = L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi))$ in a non-blocking manner $(L(S_c||_{Io} S_{\Sigma,Q}) = \overline{L_A^w(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)})$. In this article, we call the supervisor obtained in Theorem 5.5 as a nonblocking supervisor.

5.1 *Implementation of discrete controllers to multi-affine systems*

We have already outlined how the nonblocking supervisor S_c, where $S_c = \mathrm{CoAc}(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi))$, enforces the satisfaction of LTL specifications with respect to $S_{\Sigma,Q}$. Then, we discuss the implementation of S_c to the multi-affine system. Since any string in $L_A^w(S_c||_{Io} S_{\Sigma,Q})$ satisfies the LTL formula φ, let $R_{k_1k_2\cdots k_n} R_{k_1'k_2'\cdots k_n'} \cdots$ be a string in $L_A^w(S_c||_{Io} S_{\Sigma,Q})$ and $R_{k_1k_2\cdots k_n} U_{R_{k_1k_2\cdots k_n}} \times R_{k_1'k_2'\cdots k_n'} U_{R_{k_1'k_2'\cdots k_n'}} \cdots$ be the corresponding infinite path. To realise $R_{k_1k_2\cdots k_n} R_{k_1'k_2'\cdots k_n'} \cdots$, we can apply the controller $U_{R_{k_1k_2\cdots k_n}}(x)$ to the multi-affine system as long as $x \in R_{k_1k_2\cdots k_n}$. When and if $x \notin R_{k_1k_2\cdots k_n}$, the string is updated to $R_{k_1'k_2'\cdots k_n'}$, then the process continues. Therefore, the implementation of S_c drives the multi-affine system to satisfy the LTL formula φ.

6. Example

Consider a path-planning example adopted from Belta and Habets (2006), where a robot with detection and positioning capabilities moves inside a rectangular region $[0, 3] \times [1, 4]$. In particular, the robot system takes the form of the following differential equation:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -6x_1 + x_2 + x_1x_2 \\ 3x_1 - 2x_2 + x_1x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 4 \end{bmatrix} u \qquad (10)$$

where x is the position of the robot and u is the control input. The rectangular region is partitioned into nine small rectangular sub-regions with respect to the coordinates (Figure 1 (left)). Let R_{23} be a dangerous sub-region and R_{13} be a goal sub-region. Thus, for each sub-region we define the label function L: $L(R_{23}) = \{Danger, \neg Goal\}$, $L(R_{13}) = \{\neg Danger, Goal\}$ and $L(R_i) = \{\neg Danger, \neg Goal\}$ $(i = 11, 12, 21, 22, 31, 32, 33)$, where *Danger* represents the dangerous sub-region and *Goal* represents the goal sub-region. In this example, the specification is to eventually go to the goal sub-region ($\diamond Goal$) while avoiding the dangerous sub-region ($\square \neg Danger$). Such an obstacle avoidance specification can be naturally expressed by the LTL formula φ: $\square \neg Danger \wedge \diamond Goal$.

To achieve the specification, we first explore the control of the robot on sub-regions. Take R_{12} as an example. If we would like to control the robot to exit from R_{12} to R_{13} through the facet $F_{R_{12}}^{2,+}$, then $U_E(1,3) = \{v \mid [0,1][-6+3+3+v, 3-6+3+4v]^\top > 0 \wedge [1,0][-6+3+3+v, 3-6+3+4v]^\top \leq 0\} = \{v > 0 \wedge v \leq 0\} = \emptyset$. Obviously, such a controller does not exist

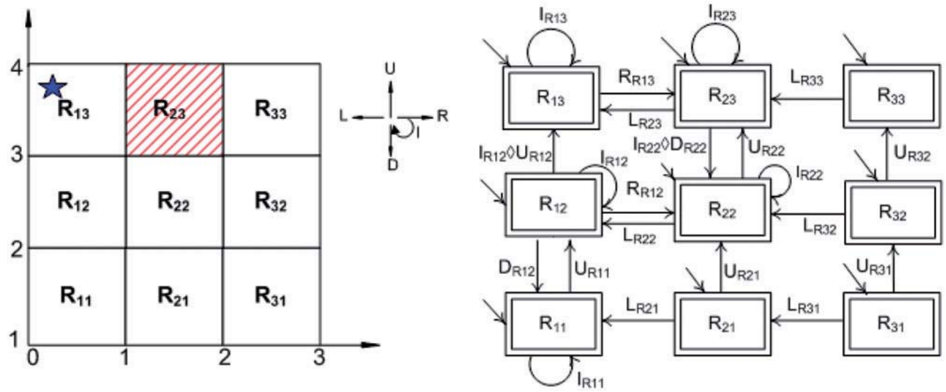

Figure 1. Rectangularly partitioned state space (left) and abstracted transition system $S_{\Sigma,\xi}$ (right).

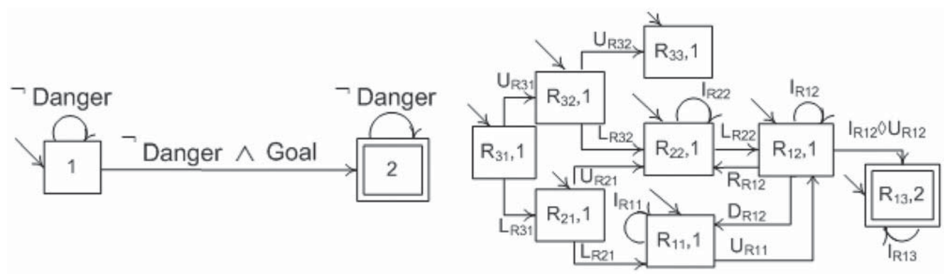

Figure 2. Büchi automaton $\mathcal{B}_\varphi$ (left) and the product automaton $S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi$ (right).

according to Lemma 2.5 (Belta and Habets 2006; Habets et al. 2006). However, by using the proposed method in this article, we can obtain a controller for the exit problem. Here we assume the accuracy limitation $\varepsilon = 10^{-4}$ and the control limitation $|u| \leq 10^7$. By Algorithm 3.10, we can design a state-based switch multi-affine controller in terms of

$$I_{R_{12}} \diamond U_{R_{12}}(x) = \begin{cases} -30x_1 - 12x_2 + 10x_1x_2 + 34 & \text{if } x \notin B_{0.01}(0.767, 2.494) \\ -11x_1 + x_1x_2 + 10 & \text{if } x \in B_{0.01}(0.767, 2.494) \end{cases}$$

to drive the robot to exit only through $F_{R_{12}}^{2,+}$. Similarly, for each sub-region $R_{mn}(m, n = 1, 2, 3)$ we can establish the controllers that steer the robot from R_{mn} to its neighbourhood sub-region (Algorithm 3.10) or to be invariant (Lemma 2.4) in R_{mn}, respectively. Thus, an abstracted transition system $S_{\Sigma,\xi}$ can be constructed (Figure 1 (right)).

On the other side, we convert the LTL formula φ to a Büchi automaton (Figure 2 (left)) and then establish the product automaton $S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi$ (Figure 2 (right)). According to Theorem 5.5, we design $\mathrm{CoAc}(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)$ (Figure 3 (left)) to be the nonblocking supervisor for $S_{\Sigma,\xi}$. After the implementation of $\mathrm{CoAc}(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)$ to the robot system, the controlled system achieves the LTL formula φ. Moreover, the simulation results of two feasible paths initialising from R_{31} and satisfying φ are shown in Figure 3 (right).

7. Conclusion

This article provided an input–output simulation approach to controlling the multi-affine system for LTL specifications in a rectangularly partitioned state space. Two novel methods were derived to control the multi-affine system on rectangles. One is based on the exit sub-region to drive all trajectories starting from a rectangle to exit only through a facet, which enlarges the classes of control systems in the context of existing literature (Belta and Habets 2006). The other provides a solution for the convergence problem by stabilising the multi-affine system towards a desired point. With the proposed control methods, a finitely abstracted transition system was constructed and it was shown to be input–output simulated by the rectangular transition system of the multi-affine system. Therefore, the controller synthesis for the multi-affine system to enforce the LTL specification can be achieved by

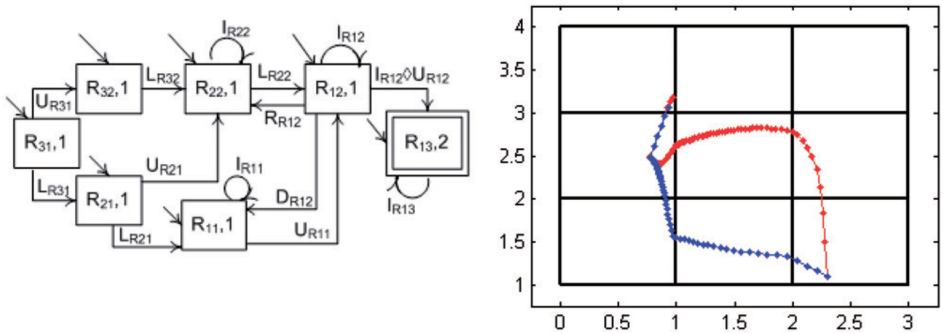

Figure 3. Nonblocking supervisor $\mathrm{CoAc}(S_{\Sigma,\xi} \times_A \mathcal{B}_\varphi)$ (left) and simulation results (right).

designing a nonblocking supervisor for the abstracted transition system and then mapped into continuous control signals. From the application point of view, this input–output simulation approach not only enables automatic and effective implementation, but also prevents blocking in the execution.

However, the result on the existence of a nonblocking supervisor enforcing LTL, i.e. Theorem 5.5, is sufficient only in the sense that if the condition of Theorem 5.5 does not satisfy, there is no conclusion on the existence of a controller for the original multi-affine system. To address this issue, our future work will investigate the necessary and sufficient condition by strengthening the input–output simulation to an input–output bisimulation. Other interesting directions are extensions of this approach to branching time logical specifications, such as computation tree logic specifications (Clarke 1997), and to more complicated dynamics, such as polynomial dynamics (Benedetto 2002).

References

Belta, C. (2004), 'On Controlling Aircraft and Underwater Vehicles', in *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 5, pp. 4905–4910.

Belta, C., Bicchi, A., Egerstedt, M., Frazzoli, E., Klavins, E., and Pappas, G. (2007), 'Symbolic Planning and Control of Robot Motion', *IEEE Robotics & Automation Magazine*, 14, 61–70.

Belta, C., and Habets, L. (2006), 'Controlling a Class of Nonlinear Systems on Rectangles', *IEEE Transactions on Automatic Control*, 51, 1749–1759.

Benedetto, R. (2002), 'Examples of Wandering Domains in p-adic Polynomial Dynamics', *Comptes Rendus Mathematique*, 335, 615–620.

Berman, S., Halász, Á., and Kumar, V. (2007), 'MARCO: A Reachability Algorithm for Multiaffine Systems with Applications to Biological Systems', in *The 10th International Conference on Hybrid Systems: Computation and Control*, pp. 76–89.

Clarke, E. (1997), 'Model Checking', in *Foundations of Software Technology and Theoretical Computer Science*, Cambridge, MA: MIT Press, pp. 54–56.

de Giacomo, G., and Vardi, M. (2000), 'Automata-theoretic Approach to Planning for Temporally Extended Goals', in *The 5th European Conference on Planning: Recent Advances in AI Planning*, pp. 226–238.

Eker, S., Knapp, M., Laderoute, K., Lincoln, P., Meseguer, J., and Sonmez, K. (2002), 'Pathway Logic: Symbolic Analysis of Biological Signaling', in *Proceedings of the Pacific Symposium on Biocomputing*, Vol. 7, pp. 400–412.

Fainekos, G., Girard, A., Kress-Gazit, H., and Pappas, G. (2009), 'Temporal Logic Motion Planning for Dynamic Robots', *Automatica*, 45, 343–352.

Fainekos, G., Kress-Gazit, H., and Pappas, G. (2005), 'Hybrid Controllers for Path Planning: A Temporal Logic Approach', in *Proceedings of IEEE Conference on Decision and Control and European Control Conference*, pp. 4885–4890.

Habets, L., Kloetzer, M., and Belta, C. (2006), 'Control of Rectangular Multi-affine Hybrid Systems', in *IEEE Conference on Decision and Control*, pp. 2619–2624.

Habets, L., and van Schuppen, J. (2004), 'A Control Problem for Affine Dynamical Systems on a Full-dimensional Polytope', *Automatica*, 40, 21–35.

Kloetzer, M., and Belta, C. (2006), 'Reachability Analysis of Multi-affine Systems', in *The 9th International Workshop on Hybrid Systems: Computation and Control*, pp. 348–362.

Kloetzer, M., and Belta, C. (2008), 'A Fully Automated Framework for Control of Linear Systems from Temporal Logic Specifications', *IEEE Transactions on Automatic Control*, 53, 287–297.

Knight, J., and Passino, K. (1990), 'Decidability for a Temporal Logic used in Discrete-event System Analysis', *International Journal of Control*, 52, 1489–1506.

Kress-Gazit, H., Fainekos, G., and Pappas, G. (2009), 'Temporal-logic-based Reactive Mission and Motion Planning', *IEEE Transactions on Robotics*, 25, 1370–1381.

Lotka-Volterra, A. (1925), *Elements of Physical Biology*, New York: Dover Publications.

Milner, R. (1989), *Communication and concurrency*, New York: Springer-Verlag.

Ogawa, K. (1993), 'Economic Development and Time Preference Schedule: The Case of Japan and East Asian NICs', *Journal of Development Economics*, 42, 175–195.

Sastry, S. (1999), *Nonlinear Systems: Analysis, Stability, and Control*, New York: Springer Verlag.

Tabuada, P., and Pappas, G. (2006), 'Linear Time Logic Control of Discrete-time Linear Systems', *IEEE Transactions on Automatic Control*, 51, 1862–1877.

Thistle, J., and Wonham, W. (1986), 'Control Problems in a Temporal Logic Framework', *International Journal of Control*, 44, 943–976.

Ulusoy, A., Smith, S., Xu, C., and Belta, C. (2012), 'Robust Multi-robot Optimal Path Planning with Temporal Logic Constraints', in *IEEE? International Conference on Robotics and Automation*, Arxiv preprint arXiv:1202.1307, http://arxiv.org/abs/1202.1307.

Volterra, V. (1926), 'Fluctuations in the Abundance of a Species Considered Mathematically', *Nature*, 118, 558–560.

Wolper, P., Vardi, M., and Sistla, A. (1983), 'Reasoning About Infinite Computation Paths', in *24th Annual Symposium on Foundations of Computer Science*, pp. 185–194.