Controlling a Class of Nonlinear Systems on Rectangles

Calin Belta, *Member, IEEE*, and Luc C. G. J. M. Habets

***Abstract*—In this paper, we focus on a particular class of nonlinear affine control systems of the form $\dot{x} = f(x) + Bu$, where the drift f is a multi-affine vector field (i.e., affine in each state component), the control distribution B is constant, and the control u is constrained to a convex set. For such a system, we first derive necessary and sufficient conditions for the existence of a multiaffine feedback control law keeping the system in a rectangular invariant. We then derive sufficient conditions for driving all initial states in a rectangle through a desired facet in finite time. If the control constraints are polyhedral, we show that all these conditions translate to checking the feasibility of systems of linear inequalities to be satisfied by the control at the vertices of the state rectangle. This work is motivated by the need to construct discrete abstractions for continuous and hybrid systems, in which analysis and control tasks specified in terms of reachability of sets of states can be reduced to searches on finite graphs. We show the application of our results to the problem of controlling the angular velocity of an aircraft with gas jet actuators.**

***Index Terms*—Aircraft control, convex analysis, hybrid systems, nonlinear systems.**

I. Introduction

THE central problems in *formal analysis* of systems are reachability analysis and safety verification. The goal of *reachability analysis* is to construct the set of states reached by trajectories of the system originating in a given (possibly uncountable) initial set. *Safety verification* is the problem of proving that a system does not have any trajectory from a given initial set to a given final (unsafe) set. For discrete systems with a finite number of states, these problems are *decidable*, i.e., can be solved by a computer in a finite number of steps. For continuous and hybrid (i.e., described by both continuous and discrete dynamics) systems, these problems are very difficult (in general undecidable) because of the uncountability of the state space.

One way to solve formal analysis problems for continuous and hybrid systems is to construct the set of states reached by the system, or an over-approximation of this set, by working directly in the continuous state space. Such methods are called *direct* and are not the subject of this paper. Our work is related to the group of *indirect* methods, where the main idea is to map the continuous or hybrid system to a discrete transition system through an iterative partitioning procedure producing finer and finer quotients, until the initial system and the discrete quotients become equivalent with respect to reachability properties. This procedure is called *abstraction* and the corresponding algorithm is called the *bi-simulation* algorithm. If such an iterative refinement procedure terminates, then the initial continuous or hybrid systems and their discrete quotient are called *bi-similar* and the reachability problem is called *decidable*. The bi-simulation relation was first introduced in [28], [23], formally defined for linear control systems in [27], and for nonlinear systems in an abstract categorical context in [14]. However, in [15], it has been shown that reachability is undecidable for a very simple class of hybrid systems. Several decidable classes have been identified though by restricting the continuous behavior of the hybrid system, as in the case of timed automata [3], multi-rate automata [1], [25], and rectangular automata [15], [29], or by restricting the discrete behavior, as in order-minimal hybrid systems [18], [19]. All these decidable classes are too weak to represent continuous and hybrid system models that arise in practice. Then one might be satisfied with sufficient abstractions, i.e., with a discrete quotient that can be used to over-approximate the reachable set of the initial system. But even finding the discrete quotient is not at all trivial. Related work focuses on partitioning using linear functions of the continuous variables, as in the method of predicate abstractions [2], [30], or using polynomial functions as in [30] and [10]. However, to derive the transitions of the discrete quotient, one has to be able to either integrate the vector fields of the initial system [2], or use computationally expensive decision procedures such as quantifier elimination for real closed fields and theorem proving [30], which severely limit the dimensions of the problems that can be approached.

In this paper, we focus on a particular class of nonlinear affine control systems of the form $\dot{x} = f(x) + Bu$, where the drift f is a multi-affine vector field (*i.e.,* affine in each state component), the control distribution B is constant, and the control u is constrained to a convex set. This class of continuous dynamics is rather large, and includes the celebrated Euler–Volterra [31] and Lotka-Volterra [22] equations, attitude and velocity control systems for aircraft [26] and underwater vehicles [4] (in this case the control directions capture the axes about which the control torques are applied), and models of genetic regulatory networks (where product type nonlinearities model mass action kinetics and the elements of B capture permeability of membrane) [7], [5]. For such systems, we define rectangular partitions of the state space and use the relationship between the structure of the

Manuscript received December 8, 2004; revised November 2, 2005. This work was supported in part by the National Science Foundation CAREER Award 0447721 and the National Science Foundation under Grant 0410514 at Boston University, Boston, MA.

C. Belta is with the Center for Information and Systems Engineering, the Departments of Manufacturing and Aerospace and Mechanical Engineering, Boston University, Brookline, MA 02446 USA (e-mail: cbelta@bu.edu).

L. C. G. J. M. Habets is with the Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, NL-5600 MB Eindhoven, The Netherlands, and also affiliated with the Center for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands (e-mail: l.c.g.j.m.habets@tue.nl).

Color versions of Figs. 1 and 2 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2006.884957

vector fields and the shape of the regions to solve two problems: *Problem 1: Keep the system in a rectangle for all times* and *Problem 2: drive the system through an exit facet in finite time*. In this paper, we show that if the control constraint set U is polyhedral, then the solutions to the above problems can be parameterized by polyhedral sets. The main idea in constructing solutions to Problems 1 and 2 is using a very interesting property of multiaffine functions on rectangles: *a multiaffine function is uniquely determined by its values at the vertices of a rectangle and its restriction to the rectangle is a convex combination of these values*. The solutions to Problems 1 and 2 enable one to construct computationally efficient characterizations of decidability of such systems. Indeed, a partitioned continuous system is bisimilar with the discrete quotient produced by the partition if and only if all initial states in a region either stay in the region forever or transit in finite time to just one neighbor.

This work draws inspiration from [11]–[13]. In these works, the authors study affine continuous dynamics on simplices. The starting point for their results is an observation similar to the one we use in this paper: an affine function is uniquely determined by its values at the vertices of a simplex and its restriction to the simplex is a convex combination of these values. In this paper, we extend these results to a larger class of continuous dynamics, i.e., we allow for product type nonlinearities. Moreover, we focus on a different partition geometry, which is more attractive for large dimensional problems. Although triangulations may be carried out in Euclidean spaces of any finite dimension (see e.g., [20] and [8]), rectangular grids are easier to work with, certainly in problems of higher dimension.

The rest of the paper is structured as follows. In Section II, we introduce the notation and give some basic definitions, before we formally state the problems in Section III. The interesting properties of multi-affine functions on rectangles enabling the framework of this paper are presented in Section IV. Based on this, in Section V, we present the main theorems providing solutions to the problems stated in Section III. Our approach is illustrated in Section VI by an application to the control of an aircraft with gas jet actuators. We conclude in Section VII with final remarks and directions for future work.

II. Preliminaries

Let $N \in \mathbb{N}$ and consider the N-dimensional Euclidean space $\mathbb{R}^N$. A full dimensional polytope P_N is defined as the convex hull of at least $N+1$ affinely independent points in $\mathbb{R}^N$. A *facet* of P_N is the intersection of P_N with one of its supporting hyperplanes. More generally, a *face* of P_N is the intersection of P_N with several of its supporting hyperplanes. If the dimension of the intersection is p (with $0 \le p < N$) the face is called a *p-face*. In particular, all facets of P_N are $(N-1)$-faces, and the vertices of P_N are 0-faces.

An N-dimensional rectangle in $\mathbb{R}^N$ is characterized by two vectors $a = (a_1, \ldots, a_N) \in \mathbb{R}^N$ and $b = (b_1, \ldots, b_N) \in \mathbb{R}^N$, with the property that $a_i < b_i$ for all $i \in \{1, \ldots, N\}$:

$$R_N = \{x = (x_1, \ldots, x_N) \in \mathbb{R}^N \mid \forall i \in \{1, \ldots, N\} : a_i \le x_i \le b_i\}. \quad (1)$$

The set of vertices of R_N is denoted by V_N, and may be characterized as

$$V_N = \prod_{i=1}^{N} \{a_i, b_i\}. \quad (2)$$

Let $p \in \mathbb{N}$ with $p < N$. Then every $(N-p)$-face F of the N-dimensional rectangle R_N, characterized by p equations of the form

$$\begin{array}{lcl} x_{i_1} = a_{i_1} & \text{or} & x_{i_1} = b_{i_1} \\ \vdots & & \vdots \\ x_{i_p} = a_{i_p} & \text{or} & x_{i_p} = b_{i_p} \end{array}$$

where $i_1, \ldots, i_p \in \{1, \ldots, N\}$ and $i_j \ne i_k$ for $j \ne k$, is isomorphic with an $(N-p)$-dimensional rectangle. We are particularly interested in facets. For $k = 1, \ldots, N$, let $\xi_k : \{a_k, b_k\} \longrightarrow \{0, 1\}$ denote the indicator function

$$\xi_k(a_k) = 0 \quad \xi_k(b_k) = 1 \quad k = 1, \ldots, N. \quad (3)$$

Then, R_N has $2N$ facets described by

$$F_N^{j, \xi_j(w_j)} = R_N \cap \{x \in \mathbb{R}^N \mid x_j = w_j\} \quad (4)$$

for all $w_j \in \{a_j, b_j\}$, $j = 1, \ldots, N$. The outer normal of facet $F_N^{j, \xi_j(w_j)}$ is given by

$$n_N^{j, \xi_j(w_j)} = (-1)^{\xi_j(w_j)+1} e_j \quad (5)$$

for all $w_j \in \{a_j, b_j\}$, $j = 1, \ldots, N$, where e_j, $j = 1, \ldots, N$ denote the Euclidean basis of $\mathbb{R}^N$.

We end the discussion on rectangles by noting that an arbitrary facet $F_N^{j, \xi_j(w_j)}$ has 2^{N-1} vertices $(v_1, \ldots, v_N)$, with $v_j = w_j$. Moreover, for an arbitrary vertex $(v_1, \ldots, v_N)$, the N facets containing it are given by $F_N^{j, \xi_j(v_j)}$, $j = 1, \ldots, N$.

Definition 1 (Multiaffine Function): A multiaffine function $f : \mathbb{R}^N \longrightarrow \mathbb{R}^q$ (with $N, q \in \mathbb{N}$) is a function in which each of the q components $f_1, \ldots, f_q$ is a polynomial in the indeterminates $x_1, \ldots, x_N$, with the property that the degree of f_j, $(j = 1, \ldots, q)$, in any of the indeterminates $x_1, \ldots, x_N$ is less than or equal to 1. Stated differently, f has the form

$$f(x) = f(x_1, \ldots, x_N) = \sum_{i_1, \ldots, i_N \in \{0,1\}} c_{i_1, \ldots, i_N} x_1^{i_1} \ldots x_N^{i_N} \quad (6)$$

with $c_{i_1, \ldots, i_N} \in \mathbb{R}^q$ for all $i_1, \ldots, i_N \in \{0, 1\}$ and using the convention that if $i_k = 0$, then $x_k^{i_k} = 1$.

For example, for $N = 2$ and arbitrary q, all multiaffine 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 $c_{ij} \in \mathbb{R}^q$, $i, j \in \{0, 1\}$.

Finally, note that if F is an $(N-p)$-face of R_N, then the restriction $f\,|_F$ of f to F is a multiaffine function on an $(N-p)$-dimensional rectangle.

III. Problem Formulation

With the notation and definitions introduced in the previous section, we are now ready to formulate the problems we study in this paper. As already outlined in the Introduction, we consider control systems of the form:

$$\dot{x} = f(x) + Bu, \;\; x \in R_N, \; u \in U \tag{7}$$

where the state x is restricted to a rectangular region R_N of $\mathbb{R}^N$ as defined in (1) and the input u is constrained in a convex set $U \subseteq \mathbb{R}^m$. The vector field f is assumed to be multi-affine as defined in (6) and $B \in \mathbb{R}^{N \times m}$ is a constant matrix of control directions. Note that the systems we consider are a particular class of nonlinear affine control systems [16], which have the general form $\dot{x} = f(x) + G(x)u$, where f is a "drift" vector field and $G(x)$ is a matrix spanning the control distribution. Therefore, in this paper, we consider a particular class of drift, and constant control distributions.

We first consider the problem of designing bounded feedback control laws that keep the state trajectories of the closed-loop system in the rectangle R_N.

1) Problem 1 (Rectangular Invariant): Determine a feedback control law $u = k(x) \in U$ for system (7), such that the corresponding closed-loop system is positively invariant on the rectangle R_N.

The positive invariance condition in the above problem means that, if a state trajectory $x(t)$ of the closed-loop system satisfies $x(t_0) \in R_N$, then $x(t) \in R_N$ for all $t \geq t_0$.

We then consider the problem of controlling system (7) so that in finite time the state is driven to a desired facet of R_N, without leaving R_N before this desired facet is reached.

2) Problem 2 (Control to a Facet): Determine a feedback control law $u = k(x) \in U$ for system (7) such that, independent of the initial state, all state trajectories of the closed-loop system leave R_N through a desired facet in finite time, meanwhile guaranteeing that a trajectory does not leave the rectangle through any of the the remaining facets.

To solve Problems 1 and 2, we restrict our attention to multiaffine feedback controllers $k(x)$. In this case, the feedback law is automatically continuous and bounded on R_N, and the closed-loop system $\dot{x} = f(x) + Bk(x)$ is multiaffine.

IV. Multiaffine Functions on Rectangles

In this section, we state and prove an interesting property of multiaffine functions on rectangles: *a multiaffine function (6) defined on an N-dimensional rectangle (1) is uniquely determined by its values at the vertices*. Moreover, inside the rectangle, *the function is a convex combination of its values at the vertices*. These results constitute the basis for the main theorems stated and proved in Section V, which provide solutions to Problems 1 and 2.

Lemma 1: Let R_N be an N-dimensional rectangle with V_N as vertex set. Let $f : R_N \longrightarrow \mathbb{R}^q$ be a multiaffine function, and assume that

$$\forall v \in V_N : \; f(v) = 0. \tag{8}$$

Then $f \equiv 0$.

Proof: (By induction). $N = 1$: If $f(a_1) = 0$ and $f(b_1) = 0$ and f is affine, then $f \equiv 0$.

Induction Step: There exist multiaffine functions $g_1 : \mathbb{R}^{N-1} \longrightarrow \mathbb{R}^q$ and $g_2 : \mathbb{R}^{N-1} \longrightarrow \mathbb{R}^q$ such that

$$f(x_1, \ldots, x_N) = g_1(x_1, \ldots, x_{N-1}) + x_N \cdot g_2(x_1, \ldots, x_{N-1}).$$

Then, for all vertices $(v_1, \ldots, v_{N-1}) \in V_{N-1} := \prod_{i=1}^{N-1} \{a_i, b_i\}$ of the $(N-1)$-dimensional rectangle

$$R_{N-1} := \{x = (x_1, \ldots, x_{N-1}) \in \mathbb{R}^{N-1} \mid \forall i \in \{1, \ldots, N-1\} : \, a_i \leq x_i \leq b_i\}$$

we have

$$\begin{aligned} 0 &= f(v_1, \ldots, v_{N-1}, a_N) \\ &= g_1(v_1, \ldots, v_{N-1}) + a_N \cdot g_2(v_1, \ldots, v_{N-1}) \\ 0 &= f(v_1, \ldots, v_{N-1}, b_N) \\ &= g_1(v_1, \ldots, v_{N-1}) + b_N \cdot g_2(v_1, \ldots, v_{N-1}). \end{aligned} \tag{9}$$

Subtraction of both equations yields $(a_N - b_N) \cdot g_2(v_1, \ldots, v_{N-1}) = 0$, and since $a_N - b_N \neq 0$, we obtain $g_2(v_1, \ldots, v_{N-1}) = 0$ for all $(v_1, \ldots, v_{N-1}) \in V_{N-1}$. This implies that also $g_1(v_1, \ldots, v_{N-1}) = 0$ for all $(v_1, \ldots, v_{N-1}) \in V_{N-1}$. By the induction hypothesis $g_1 \equiv 0$ and $g_2 \equiv 0$, hence $f \equiv 0$. ■

Proposition 1: Let R_N be an N-dimensional rectangle in $\mathbb{R}^N$, and let $g : V_N \longrightarrow \mathbb{R}^q$ be a map, relating every vertex of R_N to a vector in $\mathbb{R}^q$. Then there exists a unique multiaffine function $f : R_N \longrightarrow \mathbb{R}^q$ such that

$$\forall v \in V_N : \; f(v) = g(v). \tag{10}$$

Moreover, if for every $v = (v_1, \ldots, v_N) \in V_N$ the image of v under g is denoted by $g(v_1, \ldots, v_N) = y_{(v_1, \ldots, v_N)}$ and ξ_k, $(k = 1, \ldots, N)$, is given by (3), then the multi-affine map $f : R_N \longrightarrow \mathbb{R}^q$ realizing (10) is given by (11), as shown at the bottom of the page.

$$f(x_1, \ldots, x_N) = \sum_{(v_1, \ldots, v_N) \in V_N} \prod_{k=1}^{N} \left(\frac{x_k - a_k}{b_k - a_k} \right)^{\xi_k(v_k)} \left(\frac{b_k - x_k}{b_k - a_k} \right)^{1 - \xi_k(v_k)} y_{(v_1, \ldots, v_N)} \tag{11}$$

Proof: It follows from (3) that, for every $(v_1,\dots,v_N)\in V_N$, the product

$$\prod_{k=1}^{N}\left(\frac{x_k-a_k}{b_k-a_k}\right)^{\xi_k(v_k)}\left(\frac{b_k-x_k}{b_k-a_k}\right)^{1-\xi_k(v_k)}$$

contains either a factor x_k-a_k/b_k-a_k or a factor b_k-x_k/b_k-a_k; if $v_k=b_k$, then the first factor is present, and if $v_k=a_k$, then the second factor is present. This proves that f defined in (11) is multiaffine. Furthermore, for every fixed $(v_1,\dots,v_N)\in V_N$

$$\prod_{k=1}^{N}\left(\frac{x_k-a_k}{b_k-a_k}\right)^{\xi_k(v_k)}\left(\frac{b_k-x_k}{b_k-a_k}\right)^{1-\xi_k(v_k)}= \\ =\begin{cases}1 & \text{if } (x_1,\dots,x_N)=(v_1,\dots,v_N)\\ 0 & \text{if } (x_1,\dots,x_N)\in V_N\backslash\{(v_1,\dots,v_N)\}\end{cases}.$$

So, indeed $f(v_1,\dots,v_N)=y_{(v_1,\dots,v_N)}=g(v_1,\dots,v_N)$ for all $v=(v_1,\dots,v_N)\in V_N$.

If $f_2:R_N\longrightarrow\mathbb{R}^q$ is a multiaffine function satisfying (10), then $h:=f-f_2$ is multiaffine, and $h(v_1,\dots,v_N)=0$ for all $(v_1,\dots,v_N)\in V_N$. By Lemma 1, $h\equiv 0$, hence, f defined in (11) is unique. ■

Proposition 2: In every point $x\in R_N$, the value $f(x)$ of a multi-affine function $f:R_N\longrightarrow\mathbb{R}^q$ is a convex combination of the values of f at the vertices of R_N.

Proof: According to Proposition 1 we have (12), as shown at the bottom of the page, and by applying the same proposition to the identity function $h\equiv 1$, which is of course a multiaffine function from R_N to $\mathbb{R}$

$$1=\sum_{(v_1,\dots,v_N)\in V_N}\prod_{k=1}^{N}\left(\frac{x_k-a_k}{b_k-a_k}\right)^{\xi_k(v_k)}\left(\frac{b_k-x_k}{b_k-a_k}\right)^{1-\xi_k(v_k)}. \tag{13}$$

Since $a_k\le x_k\le b_k$ for all $k=1,\dots,N$, it follows that for $(v_1,\dots,v_N)\in V_N$, the product

$$\prod_{k=1}^{N}\left(\frac{x_k-a_k}{b_k-a_k}\right)^{\xi_k(v_k)}\left(\frac{b_k-x_k}{b_k-a_k}\right)^{1-\xi_k(v_k)}\in[0,1]. \tag{14}$$

Hence, (12) and (13) show that $f(x_1,\dots,x_N)$ is a convex combination of the values of f at the vertices of R_N. ■

$$f(x_1,\dots,x_N)=\sum_{(v_1,\dots,v_N)\in V_N}\prod_{k=1}^{N}\left(\frac{x_k-a_k}{b_k-a_k}\right)^{\xi_k(v_k)}\left(\frac{b_k-x_k}{b_k-a_k}\right)^{1-\xi_k(v_k)}f(v_1,\dots,v_N) \tag{12}$$

Corollary 1: Let $f:R_N\longrightarrow\mathbb{R}^q$ be a multiaffine function on the N-dimensional rectangle R_N. Let $(x_1,\dots,x_N)\in R_N$, and let F be the face of R_N of lowest dimension of which $(x_1,\dots,x_N)$ is an element. Then, $f(x_1,\dots,x_N)$ is a convex combination of the values of f at the vertices of F.

Lemma 2: Let $w\in\mathbb{R}^q$ and $d\in\mathbb{R}$. Then, $w^Tf(x)\bowtie d$ everywhere in R_N if and only if $w^Tf(v_1,\dots,v_N)\bowtie d$, for all $(v_1,\dots,v_N)\in V_N$. $\bowtie$ stands for any of $<,\le,=,\ge,>$.

Proof: The necessity follows immediately from the fact that the vertices $(v_1,\dots,v_N)$ belong to R_N. The sufficiency is also immediate from the fact that $w^Tf(x)$ is a scalar multiaffine function and, therefore, its restriction to the rectangle R_N is a convex combination of its values $w^Tf(v_1,\dots,v_N)$ at the vertices $(v_1,\dots,v_N)$. ■

It is easy to see that Lemma 2 remains valid if f is restricted to a facet F of R_N.

V. Control of Multiaffine Systems on Rectangles

The following theorem gives a complete description of the solution to Problem 1 under the assumption that the feedback controllers are restricted to multiaffine functions of the state. It basically states that there exists a multiaffine feedback controller $k(x)$ solving Problem 1 if and only if f, B, and U are such that, at each vertex $(v_1,\dots,v_N)$, we can choose a control $g_{(v_1,\dots,v_N)}\in U$ so that the velocity of the closed-loop system $f(v_1,\dots,v_N)+Bg_{(v_1,\dots,v_N)}$ at the vertex has negative projections along the outer normals of all facets containing that vertex. Formally, we have the following.

Theorem 1 (Equivalent Condition for Problem 1): There exists a multiaffine feedback control law $u=k(x)\in U$ for system (7) such that all state trajectories of the corresponding closed-loop system that start in the rectangle R_N, remain in the rectangle R_N for all times if and only if the following sets are nonempty:

$$U^I_{(v_1,\dots,v_N)} \\ =U\bigcap_{j=1}^{N}\{g\in\mathbb{R}^m|n_N^{j,\xi_j(v_j)^T}(f(v_1,\dots,v_N)+Bg)\le 0\} \tag{15}$$

for all $(v_1,\dots,v_N)\in V_N$.

Proof: For sufficiency, if all the sets $U^I_{(v_1,\dots,v_N)}$ are nonempty, than we can choose arbitrary $g_{(v_1,\dots,v_N)}\in U^I_{(v_1,\dots,v_N)}$ and let $k(x)$ be the unique multiaffine function on R_N taking the values $g_{(v_1,\dots,v_N)}$ at the vertices. Such a function can be constructed using (11). By Proposition 2, $k(x)$ is a convex combination of $g_{(v_1,\dots,v_N)}$ everywhere in R_N, and since $g_{(v_1,\dots,v_N)}\in U$ and U is convex, it follows that $k(x)\in U$, $\forall x\in R_N$.

The vector field $f(x)+Bk(x)$ of the closed-loop system is a multiaffine function on R_N with values at the vertices $f(v_1,\dots,v_N)+Bg_{(v_1,\dots,v_N)}$. The inequalities inside (15) state that, for an arbitrary vertex $(v_1,\dots,v_N)$, $f(v_1,\dots,v_N)+Bg_{(v_1,\dots,v_N)}$ has a negative projection along the outer normals of all facets containing the vertex. This is equivalent to saying that, for an arbitrary facet, the vector field of the closed-loop system is oriented inside the facet at the vertices. Formally, for any facet $F_N^{j,\xi_j(w_j)}$, we have

$$n_N^{j,\xi_j(w_j)^T}(f(v_1,\ldots,v_N)+Bg_{(v_1,\ldots,v_N)})\leq 0$$

for all $(v_1,\ldots,v_N)\in V_N$ with $v_j=w_j$. From Lemma 2, we conclude that

$$n_N^{j,\xi_j(w_j)^T}(f(x)+Bk(x))\leq 0 \tag{16}$$

for all $x\in F_N^{j,\xi_j(w_j)}$. In combination with the Lipschitz continuity of the velocity vector field $f(x)+Bk(x)$, condition (16) guarantees that the state of the closed-loop system cannot leave the rectangle through any of the facets (see, e.g., [13, App. A] for a similar proof in case of systems with affine dynamics). This proves the first part of the equivalence.

For necessity, assume there exists a multiaffine control law $u=k(x)\in U$ solving Problem 1. Then, we take $g_{(v_1,\ldots,v_N)}=k(v_1,\ldots,v_N)$ and we will prove that $g_{(v_1,\ldots,v_N)}\in U^I_{(v_1,\ldots,v_N)}$. Of course, $g_{(v_1,\ldots,v_N)}\in U$. We only need to show that

$$n_N^{j,\xi_j(v_j)^T}(f(v_1,\ldots,v_N)+Bg_{(v_1,\ldots,v_N)})\leq 0$$

for all $(v_1,\ldots,v_N)\in V_N$ and $j=1,\ldots,N$. If we assume by contradiction that there exists a vertex $(v_1,\ldots,v_N)$ and a direction $j\in\{1,\ldots,N\}$ so that the previous inequality is false (i.e., satisfied with "$>$"), then by continuity this implies that there exists a whole neighborhood of $(v_1,\ldots,v_N)$ in $\mathbb{R}^N$ in which $f(x)+Bk(x)$ has a strictly positive projection along $n_N^{j,\xi_j(v_j)}$. Then, there will exist trajectories of the system leaving the rectangle through facet $F_N^{j,\xi_j(v_j)}$. This gives a contradiction and the theorem is proved. ■

Next, we give sufficient conditions for the existence of a solution to Problem 2: If f, B, and U are such that, at each vertex $(v_1,\ldots,v_N)$, we can choose a control $g_{(v_1,\ldots,v_N)}\in U$ so that the velocity of the closed-loop system $f(v_1,\ldots,v_N)+Bg_{(v_1,\ldots,v_N)}$ at the vertex has a strictly positive projection along the outer normal of the exit facet and a negative projection along the outer normals of all facets containing that vertex different from the exit facet, then we can construct a solution $k(x)$ of Problem 2. Formally, we have the following.

Theorem 2 (Sufficient Conditions for Problem 2): There exists a multiaffine feedback control law $u=k(x)\in U$ for system (7) such that all state trajectories of the corresponding closed-loop system that start in the rectangle R_N are driven through an arbitrary facet $F_N^{j,\xi_j(w_j)}$ in finite time, without crossing other facets first, if the following sets are nonempty:

$$\begin{aligned}U^E_{(v_1,\ldots,v_N)}=U\bigcap\\ &\{g\in\mathbb{R}^m|n_N^{j,\xi_j(w_j)^T}(f(v_1,\ldots,v_N)+Bg)>0 \text{ and}\\ &n_N^{i,\xi_i(v_i)^T}(f(v_1,\ldots,v_N)+Bg)\leq 0\\ &\text{for all } i=1,\ldots,N,\ i\neq j\}\end{aligned} \tag{17}$$

for all vertices $(v_1,\ldots,v_N)\in V_N$.

Proof: Choose arbitrary $g_{(v_1,\ldots,v_N)}\in U^E_{(v_1,\ldots,v_N)}$ and let $k(x)$ be the unique multiaffine function on R_N taking the values $g_{(v_1,\ldots,v_N)}$ at the vertices as shown in (11). By Proposition 2, $k(x)$ is a convex combination of $g_{(v_1,\ldots,v_N)}$ everywhere in R_N, and since $g_{(v_1,\ldots,v_N)}\in U$, it follows that $k(x)\in U,\forall x\in R_N$.

First, using arguments similar to those in the proof of Theorem 1, we note that the state of the closed-loop system cannot leave the rectangle through any of the facets different from $F_N^{j,\xi_j(w_j)}$. Indeed, from the second line of (17), we have

$$n_N^{i,\xi_i(v_i)^T}(f(v_1,\ldots,v_N)+Bg_{(v_1,\ldots,v_N)})\leq 0$$

for all $i=1,\ldots,N,\ i\neq j$, which means that the vector field corresponding to the closed-loop system has negative projection along the outer normals of all $2N-2$ facets different from the exit facet $F_N^{j,\xi_j(w_j)}$ and the one opposite to it in the jth direction. Using the convexity property of multiaffine functions in the form of Lemma 2, and the fact that the vector field $f(x)+Bk(x)$ is Lipschitz continuous, we conclude that the state of the closed-loop system cannot leave the rectangle through any of these facets. For the facet opposite to $F_N^{j,\xi_j(w_j)}$, since its outer normal is $-n_N^{j,\xi_j(w_j)}$, the inequality is strict according to the first line of (17). Therefore, the state trajectory of the closed-loop system can only leave through $F_N^{j,\xi_j(w_j)}$.

Since

$$n_N^{j,\xi_j(w_j)^T}(f(v_1,\ldots,v_N)+Bg_{(v_1,\ldots,v_N)})>0$$

for all $(v_1,\ldots,v_N)\in V_N$, by Lemma 2, we conclude that there exists an $\varepsilon>0$ such that $n_N^{j,\xi_j(w_j)^T}(f(x)+Bk(x))>\varepsilon$ everywhere in R_N. Therefore, the state trajectories of the closed-loop system have a strictly positive speed in the direction of $n_N^{j,\xi_j(w_j)}$ and the Theorem is proved. ■

Remark 1: Under the conditions of Theorem 2, the state of the closed-loop system leaves the rectangle the very first time it hits the exit facet. On the exit facet, trajectories cannot turn back into the rectangle R_N.

Remark 2 (Necessary Conditions for Control to a Facet): The sufficient conditions in Theorem 2 are somewhat stronger than necessary ones. For example, if one additionally requires that the property described in Remark 1 has to be satisfied, one can easily prove along the same lines that the sufficient conditions become necessary if we relax the requirement that at the vertices opposed to the exit facet the projection of the closed-loop vector field along the outer normal of the exit facet is only positive as opposed to strictly positive. On the other hand, it is also possible to relax the property described in Remark 1 that all trajectories leave R_N immediately upon reaching the exit facet. Instead, one may allow that some trajectories turn back into R_N before they leave the rectangle through the required exit facet on a later occasion. In this case, it is not necessary that in all vertices of the exit facet the vector field of the closed-loop system has a positive component in the direction of $n_N^{j,\xi_j(w_j)}$.

Remark 3 (Computational Issues): The sets $U^I_{(v_1,\ldots,v_N)}$ in Theorem 1 and $U^E_{(v_1,\ldots,v_N)}$ in Theorem 2 represent allowed sets for controls at the vertices. If these sets are nonempty, any choice of control values $g_{(v_1,\ldots,v_N)}$ in these sets will lead to a perfectly valid multiaffine feedback control law $u = k(x)$ by (11). If the allowed control set U is a polyhedral subset of $\mathbb{R}^m$, then checking the nonemptiness of $U^I_{(v_1,\ldots,v_N)}$ and $U^E_{(v_1,\ldots,v_N)}$ reduces to checking the feasibility of a set of linear inequalities, for which there exist several computationally powerful algorithms and software packages (see, e.g., [9] and [17]).

Remark 4 (Constant Feedback Control): An interesting special case of Theorem 2 is when

$$\bigcap_{(v_1,\ldots,v_N)\in V_N} U^E_{(v_1,\ldots,v_N)} \neq \emptyset.$$

An element $\bar{u}$ in the aforementioned set can be used as a constant (independent of the current state) control that solves Problem 2. Note that this is consistent with (11). Indeed, if $g_{(v_1,\ldots,v_N)} = \bar{u}$ for all $(v_1,\ldots,v_N) \in V_N$, then $k(x) = \bar{u}$ due to (13). This case may be extremely useful in practical situations, where the state x is not available for feedback.

VI. Example: Angular Velocity Control

In this section, we first make the important observation that the class of systems studied in this paper includes attitude and angular velocity control systems for aircraft and underwater vehicles. We then show a numerical example for angular velocity control of an aircraft with gas-jet actuators.

A. Aircraft and Underwater Vehicles

Consider an arbitrarily shaped aircraft with a body fixed frame $\{B\}$ in motion with respect to a world frame $\{W\}$. Let G be the inertia matrix of the aircraft with respect to its body frame and m its mass. Let $\zeta_1, \zeta_2, \ldots, \zeta_p$ be the axes about which the corresponding control torques $t_1, \ldots, t_p$ are applied by means of opposing pairs of gas jets. Let ω denote the angular velocity in the body frame, v the translational velocity of the origin of the body in body coordinates, and F the total force applied to the body at the center of mass expressed in the body frame. Then, the kinematic equations of the aircraft can be written as

$$m\dot{v} = mv \times \omega + F \quad (18)$$

$$G\dot{\omega} = G\omega \times \omega + \sum_{i=1}^{p} \zeta_i t_i. \quad (19)$$

Similarly, for an underwater vehicle modeled as a neutrally buoyant rigid body submerged in an ideal fluid, if the center of gravity of the vehicle coincides with the center of buoyancy, then the equations of motion can be written as [21]

$$M\dot{v} = Mv \times \omega + F \quad (20)$$

$$G\dot{\omega} = G\omega \times \omega + Mv \times v + \sum_{i=1}^{p} \zeta_i t_i \quad (21)$$

where M is an added mass matrix which incorporates the mass of the body and the mass of the fluid replaced by the body [21] and all the remaining variables have the same meaning as before.

The position and orientation in the world frame $\{W\}$ of both systems described previously are identified with $SE(3)$, the Lie group of rigid body displacements in $\mathbb{R}^3$

$$SE(3) = \left\{ A \,|\, A = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}, R \in SO(3),\, d \in \mathbb{R}^3 \right\}. \quad (22)$$

where d denotes the displacement of the origin of the body frame $\{B\}$ in $\{W\}$ and $R \in SO(3)$ its rotation

$$SO(3) = \left\{ R \,|\, RR^T = I,\, \det(R) = 1 \right\}. \quad (23)$$

The equations relating their positions and velocities are

$$\dot{R} = R\hat{\omega} \quad (24)$$

$$\dot{d} = Rv \quad (25)$$

where $\hat{()}$ is the skew symmetric operator.

If quaternions $q = (q_1, q_2, q_3, q_4) \in S^3$ (S^3 denotes the unit sphere in $\mathbb{R}^4$) are chosen to parameterize $R \in SO(3)$, (24) can be written as

$$\dot{q} = \frac{1}{2} Q(q) \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ 0 \end{bmatrix}$$

$$Q(q) = \begin{bmatrix} q_4 & -q_3 & q_2 & -q_1 \\ q_3 & q_4 & -q_1 & -q_2 \\ -q_2 & q_1 & q_4 & -q_3 \\ -q_1 & -q_2 & -q_3 & -q_4 \end{bmatrix} \quad (26)$$

where $(\omega_1, \omega_2, \omega_3)$ are the components of the angular velocity ω.

There are situations, especially in space missions, in which one is not interested in controlling the pose (displacement and rotation) of a spacecraft or underwater vehicle in a reference frame, but rather in regulating the body velocities of translation and rotation. In this case, (18) and (19), respectively (20) and (21), can be seen as control systems with states $x = (v, \omega)$ and controls $u = (F, t_1, \ldots, t_p)$. However, there are several situations in which one is interested in controlling only the attitude of a vehicle in a given world frame, and then (19) and (26) can be seen as a control system with state $x = (q, \omega)$ and control variables $u = (t_1, \ldots, t_p)$. The main observation in this section is that all control systems mentioned before are affine control systems with multi-affine drift and constant control distribution as described in (7). The set U captures the physical control bounds. Using the results of this paper, we can approach the rigid body control problem from a totally different perspective. Our approach is somewhere in between stabilization to a point and interpolation between two end positions in the configuration space. We propose a feedback control law, that may contain some discontinuities, which allows for a “maneuvering” procedure (consisting of continuous trajectories), i.e., driving a rigid body attitude or angular velocity control system between arbitrary initial and final regions of the state–space, while satisfying bounds on inputs and state. An illustrative task that we can solve with this procedure is the following. Given an aircraft or underwater vehicle with gas jet actuators and physical bounds on the control torques, which is initially rotating at a certain angular velocity (not necessarily precisely known), we want to drive it towards a final, desired angular velocity. We also require that a priori given bounds on the velocity are satisfied during the transition. After the desired region of the state space is reached, one can use a locally stabilizing control law [6], [24], if convergence to a specific state is required. Of course we need to make sure that the local region of attraction includes the target region of our algorithm. Note that globally stabilizing controllers exist as well, but using those there is no way one can guarantee that the trajectories converging to a desired equilibrium satisfy the required bounds on inputs and state. Especially the possibility to guarantee that certain bounds on inputs and velocities are respected by the feedback controller, makes the design method proposed in this paper attractive in a large area of applications.

B. Maneuvering in the Angular Velocity Space

Consider a parallelepiped aircraft with gas-jet actuators. Assume that the frame $\{B\}$ is fixed at the center of the aircraft and aligned with its principal axis, so that $G = \operatorname{diag}\{g_1, g_2, g_3\}$. Assume that $\dim \operatorname{span}\{\zeta_1, \ldots, \zeta_p\} = 3$, *i.e.,* the system is controllable. Without loss of generality, we will take the control directions as being the Euclidean basis vectors e_i, $i = 1, 2, 3$ and the control will be reparameterized by u_i along these directions. Then, the angular control system (19) takes the form of the known controlled Euler's equations

$$\begin{aligned} \dot{\omega}_1 &= \frac{g_2 - g_3}{g_1}\omega_2\omega_3 + u_1 \\ \dot{\omega}_2 &= \frac{g_3 - g_1}{g_2}\omega_1\omega_3 + u_2 \\ \dot{\omega}_3 &= \frac{g_1 - g_2}{g_3}\omega_1\omega_2 + u_3. \end{aligned} \tag{27}$$

Assuming that the aircraft spans between θ_i and ψ_i along the direction e_i ($i = 1, 2, 3$) of the body frame $\{B\}$, we have

$$\begin{aligned} g_1 &= \frac{1}{24}m((\psi_2 - \theta_2)^2 + (\psi_3 - \theta_3)^2) \\ g_2 &= \frac{1}{24}m((\psi_3 - \theta_3)^2 + (\psi_1 - \theta_1)^2) \\ g_3 &= \frac{1}{24}m((\psi_1 - \theta_1)^2 + (\psi_2 - \theta_2)^2). \end{aligned} \tag{28}$$

Finally, the controls u_i are limited to take values in $[-1, 1]$. Control system (27) is obviously of the form (7) with $x = \omega$, the multiaffine drift

$$f(x) = \left(\frac{x_2x_3(g_2 - g_3)}{g_1}, \frac{x_1x_3(g_3 - g_1)}{g_2}, \frac{x_1x_2(g_1 - g_2)}{g_3}\right)$$

control directions $B = I_3$, and set of admissible controls $U = [-1, 1]^3$.

Consider the following control scenario. Assume that the aircraft is initially rotating around the z-axis of its body frame $\{B\}$ at speed ω_s. The goal is to control the aircraft so that it eventually rotates around its x-axis at the same speed and remains in this state for all times. Moreover, while transiting from the initial to the final state, the aircraft is forbidden to develop rotational speed ω_2 around its y-axis.

To capture the uncertainty on knowledge of the state as well as sensor noise, we allow for deviations of amplitude $\epsilon > 0$ in all directions. Under this assumption, the initial state of rotation is assumed to be the collection of all states in a small cube centered at $(0,\ 0,\ \omega_s)$ and with side 2ϵ. The amount of allowed speed of rotation around the y-axis is assumed to be ϵ and the goal is to drive and keep the system in a small cube centered at $\omega = (\omega_s,\ 0,\ 0)$, and with side 2ϵ, where $\epsilon > 0$ is a small number. Using the results of this paper, we can provide a solution to this problem in terms of a feedback control law by defining a set of rectangles in the velocity space and solving control problems of the type Problems 1 and 2.

Explicitly, according to the specifications of the task, consider a set of four pairwise adjacent rectangles as shown in Fig. 1(a). The task is accomplished if the following controllers are designed.

- Controller 1: "Drive" the system down along the ω_3-axis while keeping the absolute values of ω_1 and ω_2 less than ϵ. The solution to this problem is found by applying Theorem 2 to Rectangle 1 defined by $[-\epsilon, \epsilon] \times [-\epsilon, \epsilon] \times [\epsilon, \omega_s + \epsilon]$ with exit facet $\omega_3 = \epsilon$ [see Fig. 1(a)].
- Controller 2: "Take the turn" around origin. This control law can be derived by applying Theorem 2 to Rectangle 2 defined by $[-\epsilon, \epsilon] \times [-\epsilon, \epsilon] \times [-\epsilon, \epsilon]$ with exit facet $\omega_1 = \epsilon$ [see Fig. 1(a)].
- Controller 3: Drive the system along the ω_1-axis while keeping the absolute values of ω_2 and ω_3 less than ϵ. The solution is found by applying Theorem 2 to Rectangle 3 defined by $[\epsilon, \omega_s - \epsilon] \times [-\epsilon, \epsilon] \times [-\epsilon, \epsilon]$ with exit facet $\omega_1 = \omega_s - \epsilon$ [see Fig. 1(a)].
- Controller 4: Keep the system in a cubic box centered at $(\omega_s, 0, 0)$ and side 2ϵ. The controller is designed by applying Theorem 1 to Rectangle 4 defined by $[\omega_s - \epsilon, \omega_s + \epsilon] \times [-\epsilon, \epsilon] \times [-\epsilon, \epsilon]$ [see Fig. 1(a)].

We used the following numerical data:

$$\begin{aligned} \omega_s &= 3,\ m = 1,\ \theta_1 = -4,\ \psi_1 = 4,\ \theta_2 = -4 \\ \psi_2 &= 4,\ \theta_3 = -1,\ \psi_3 = 1,\ \epsilon = 0.1. \end{aligned}$$

A possible choice of Controllers 1–4 is given later. $g^i_{(v_1, v_2, v_3)}$, $i = 1, \ldots, 4$ represent the controls at the vertices of Rectangle i where Controller i is defined, obtained as a solution of the set of linear inequalities (17) for $i = 1, 2, 3$ and (15) for $i = 4$. u^i, $i = 1, \ldots, 4$ is the feedback control valid everywhere in the corresponding rectangle, uniquely determined by its values $g^i_{(v_1, v_2, v_3)}$ at the vertices.

1) Controller 1 (Defined in Rectangle 1):

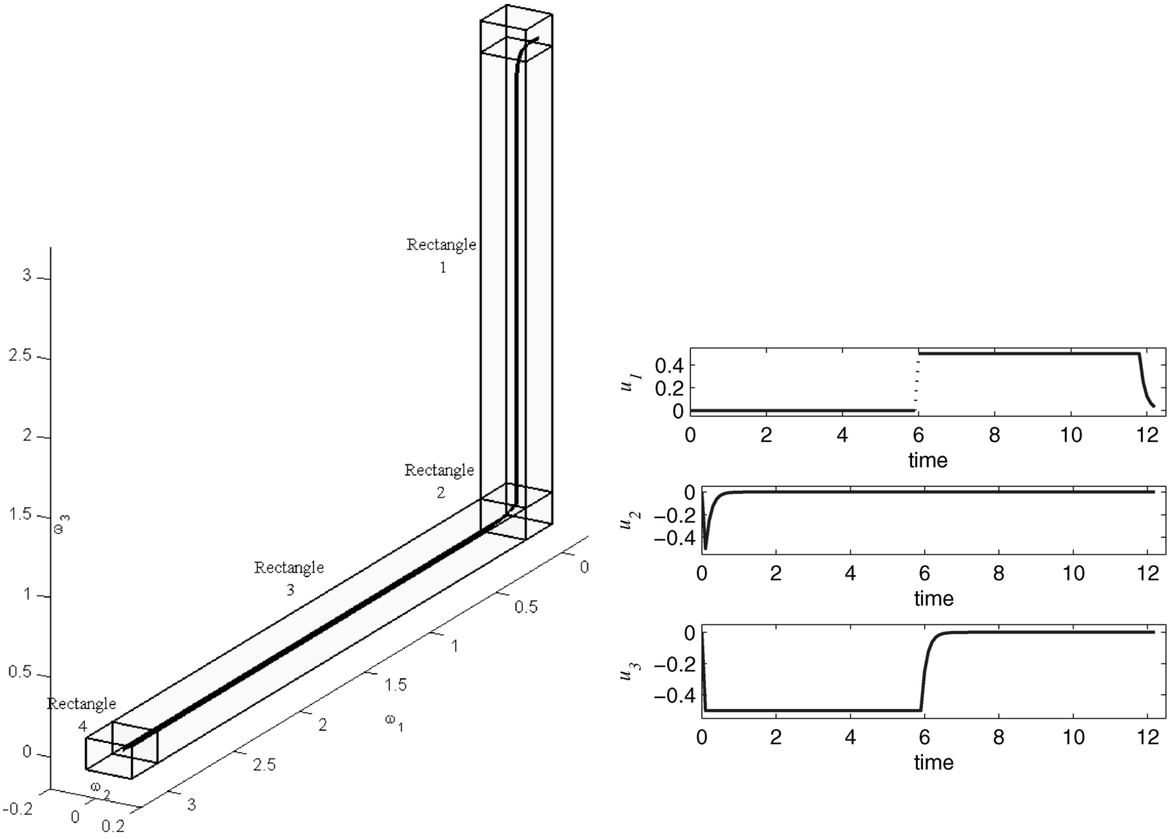
$$g^1_{(a_1,a_2,a_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \end{bmatrix} \quad g^1_{(a_1,a_2,b_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$$

$$g^1_{(a_1,b_2,a_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \end{bmatrix} \quad g^1_{(a_1,b_2,b_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$$

$$g^1_{(b_1,a_2,a_3)} = \begin{bmatrix} -0.5 \\ 0.5 \\ -0.5 \end{bmatrix} \quad g^1_{(b_1,a_2,b_3)} = \begin{bmatrix} -0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$$

$$g^1_{(b_1,b_2,a_3)} = \begin{bmatrix} -0.5 \\ -0.5 \\ -0.5 \end{bmatrix} \quad g^1_{(b_1,b_2,b_3)} = \begin{bmatrix} -0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$$

$$u^1(x) = \begin{bmatrix} -5x_1 \\ -5x_2 \\ -0.5 \end{bmatrix}.$$



(a) (b)

Fig. 1. (a) Region in the angular velocity space $(\omega_1, \omega_2, \omega_3)$ corresponding to the maneuvering task. The small rectangle on the ω_3 – axis in the upper part represents the initial state of rotation about the body z – axis. The small rectangle on the ω_1 – axis represents the final state of rotation about the body x – axis. The thick line represents a closed-loop trajectory starting at $(0, \epsilon, \omega_s)$. (b) Controls corresponding to the trajectory shown in (a).

2) Controller 2 (Defined in Rectangle 2):

$$g^2_{(a_1,a_2,a_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \quad g^2_{(a_1,a_2,b_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$$

$$g^2_{(a_1,b_2,a_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \end{bmatrix} \quad g^2_{(a_1,b_2,b_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$$

$$g^2_{(b_1,a_2,a_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \quad g^2_{(b_1,a_2,b_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$$

$$g^2_{(b_1,b_2,a_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \end{bmatrix} \quad g^2_{(b_1,b_2,b_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$$

$$u^2(x) = \begin{bmatrix} 0.5 \\ -5x_2 \\ -5x_3 \end{bmatrix}.$$

3) Controller 3 (Defined in Rectangle 3):

$$g^3_{(a_1,a_2,a_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \quad g^3_{(a_1,a_2,b_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$$

$$g^3_{(a_1,b_2,a_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \end{bmatrix} \quad g^3_{(a_1,b_2,b_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$$

$$g^3_{(b_1,a_2,a_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \quad g^3_{(b_1,a_2,b_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$$

$$g^3_{(b_1,b_2,a_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \end{bmatrix} \quad g^3_{(b_1,b_2,b_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$$

$$u^3(x) = \begin{bmatrix} 0.5 \\ -5x_2 \\ -5x_3 \end{bmatrix}.$$

4) Controller 4 (Defined in Rectangle 4):

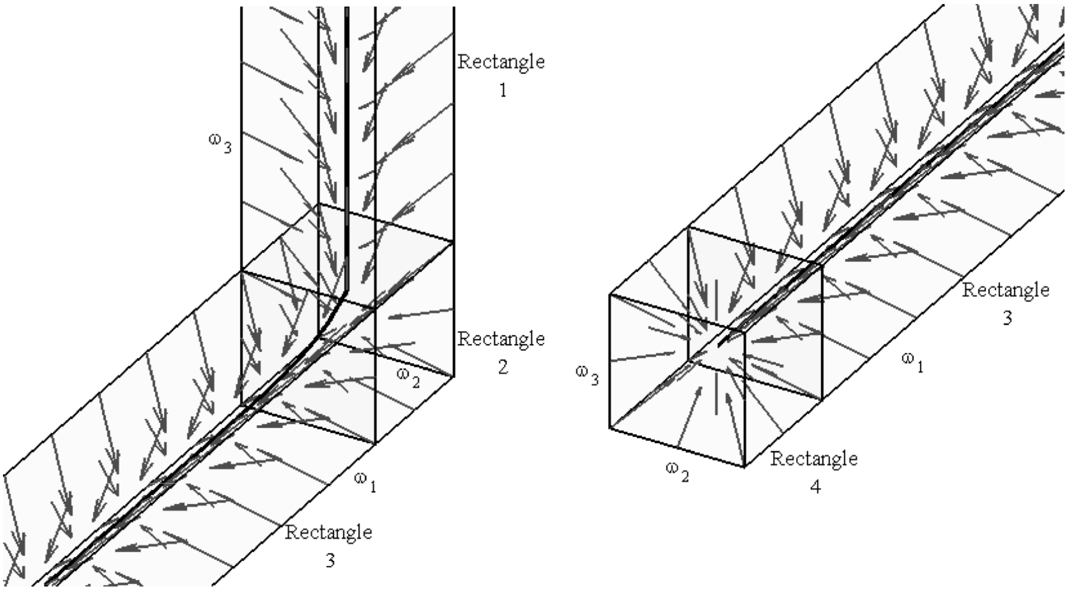
$$g^4_{(a_1,a_2,a_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \quad g^4_{(a_1,a_2,b_3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$$

$$g^4_{(a_1,b_2,a_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ 0.5 \end{bmatrix} \quad g^4_{(a_1,b_2,b_3)} = \begin{bmatrix} 0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$$

$$g^4_{(b_1,a_2,a_3)} = \begin{bmatrix} -0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \quad g^4_{(b_1,a_2,b_3)} = \begin{bmatrix} -0.5 \\ 0.5 \\ -0.5 \end{bmatrix}$$

$$g^4_{(b_1,b_2,a_3)} = \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \end{bmatrix} \quad g^4_{(b_1,b_2,b_3)} = \begin{bmatrix} -0.5 \\ -0.5 \\ -0.5 \end{bmatrix}$$

$$u^4(x) = \begin{bmatrix} 15 - 5x_1 \\ -5x_2 \\ -5x_3 \end{bmatrix}.$$



(a)

(b)

Fig. 2. Vector field of the closed-loop system is continuous everywhere, except on the boundary between Rectangles 1 and 2. In Rectangles 2 and 3, the feedback law is the same. On the common facet of Rectangles 3 and 4 again a switch to another feedback law takes place, but the vector field of the closed-loop system is continuous here because both feedback laws coincide on this common facet.

It is easily verified that on the common facet of Rectangles 2 and 3, and also on the common facet of Rectangles 3 and 4, the vector field of the closed-loop system is continuous. In Rectangles 2 and 3, the feedback laws are even the same, and no switch between different feedbacks is required, when the state trajectory crosses the common facet $\omega_1 = \epsilon$ of these two rectangles. On the common facet of Rectangles 3 and 4, i.e., the facet $\omega_1 = \omega_s - \epsilon$, the situation is slightly different. Here a switch from feedback law u^3 to feedback law u^4 occurs, but since both feedback laws coincide on the common facet, this does not lead to a discontinuity in the vector field of the closed-loop system. Note that a switch from feedback law u^3 to feedback law u^4 is required, in order to guarantee that after entering Rectangle 4, the state trajectory will never leave this rectangle anymore.

On the common facet of Rectangles 1 and 2, i.e., the facet $\omega_3 = \epsilon$, the feedback laws u^1 and u^2 do not coincide. This leads to a discontinuity in the vector field of the closed-loop system. So, in order to avoid ambiguity of the definition of the feedback law on this common facet, one has to specify it explicitly. We choose the feedback law on this common facet to be equal to u^2. In this way, it is guaranteed that the constructed feedback law solves the given reachability problem. Indeed, feedback u^1 on Rectangle 1 guarantees that every trajectory starting in Rectangle 1 reaches facet $\omega_3 = \epsilon$ in finite time, without leaving through other facets first. On the common facet $\omega_3 = \epsilon$, one switches (discontinuously) to feedback law u^2. Since the component of the closed-loop vector field in the direction of e_3 remains negative, the trajectory will cross the common facet $\omega_3 = \epsilon$, and feedback $u^2 = u^3$ guarantees that the trajectory will cross the common facet of Rectangle 2 and Rectangle 3, and reaches Rectangle 4 in finite time. After a (continuous) switch to feedback law u^4, the state trajectory will remain in Rectangle 4 forever.

Note that the feedback law is constructed in such a way that any state trajectory of the closed-loop system will only cross the common facet of two rectangles once, because on both sides of the common facet, the closed-loop vector field is pointing in the same direction w.r.t. the normal vector of this common facet.

A trajectory of the closed-loop system in the angular velocity space $(\omega_1, \omega_2, \omega_3)$ starting from $(0, \epsilon, \omega_s)$ is shown for illustration in Fig. 1(a). It can be seen that all specifications are satisfied, i.e., the trajectory travels through Rectangles 1–3 and stabilizes in Rectangle 4. The controls u_1, u_2, and u_3 producing this trajectory, which are plotted in Fig. 1(b), are bounded in $[-1, 1]$ as required. It is also interesting to note that the inputs u_2 and u_3 are continuous everywhere. This follows from the fact that on common facets the definition of the feedback laws for inputs u_2 and u_3 coincide. The only discontinuous input is u_1; as soon as at $t = 6$ Rectangle 2 is reached, it switches from 0 to 0.5. The (dis)continuity of the closed-loop vector field and the continuity of the trajectory are also illustrated in Fig. 2, where the regions around the small Rectangles 2 and 4 are zoomed in.

Remark 5: Note that the overall controller constructed in this example is a piecewise *affine* controller. This is a coincidence, caused by the particular choice of the input values at the vertices. A different choice of these input values leads to a different control law, that, in general, will be piecewise *multiaffine* instead of piecewise affine.

VII. Concluding Remarks

In this paper, we start from the important observation that a multi-affine function is uniquely determined by its values at the vertices of a full dimensional rectangle and the restriction of the function to the rectangle is a convex combination of these values. Using these properties, we derive necessary and sufficient conditions for the existence of a multiaffine feedback law keeping the state of an affine control system with multi-affine drift and constant control distribution in a rectangle. We also derive sufficient conditions for driving all state trajectories of such a system through a desired facet of a rectangle in finite time. If the control constraints are polyhedral, we show that all these conditions translate to solving sets of linear inequalities.

In the future, we will use these results to develop a framework for computationally efficient construction of discrete abstractions for continuous and hybrid systems with multiaffine dynamics. Specifically, using iterative rectangular partitions and the results presented in this paper, we want to construct discrete quotients that are either equivalent with continuous or hybrid systems with respect to reachability properties, or over-approximate their reachable sets. Even though the class of systems that we consider in this paper is rather large, including Euler–Volterra, and Lotka–Volterra equations, attitude and velocity control systems for aircraft and underwater vehicles, as well as models of biomolecular networks, in the future we will try to extend these results to more complicated dynamics, such as polynomial dynamics.

REFERENCES

[1] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho, “Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems,” in *Hybrid Systems*, R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, Eds. Berlin, Germany: Springer-Verlag, 1993, vol. 736, Lecture Notes in Computer Science, pp. 209–229.

[2] R. Alur, T. Dang, and F. Ivancić, “Reachability analysis of hybrid systems via predicate abstraction,” in *Hybrid Systems: Computation and Control*, C.J. Tomlin and M.R. Greenstreet, Eds. Berlin, Germany: Springer-Verlag, 2002, vol. 2289, Lecture Notes in Computer Science, pp. 35–48.

[3] R. Alur and D. L. Dill, “A theory of timed automata,” *Theor. Comp. Sci.*, vol. 126, pp. 183–235, 1994.

[4] C. Belta, “On controlling aircraft and underwater vehicles,” in *Proc. IEEE Int. Conf. Robotics and Automation*, New Orleans, LA, 2004, vol. 5, pp. 4905–4910.

[5] C. Belta, P. Finin, L. C. G. J. M. Habets, A. M. Halász, M. Imielinksi, R. V. Kumar, and H. Rubin, “Understanding the bacterial stringent response using reachability analysis of hybrid systems,” in *Hybrid Systems: Computation and Control* , R. Alur and G.J. Pappas, Eds. Berlin, Germany: Springer-Verlag, 2004, vol. 2993, Lecture Notes in Computer Science, pp. 111–125.

[6] P. E. Crouch, “Spacecraft attitude control and stabilization: Applications of geometric control theory to rigid body models,” *IEEE Trans. Autom. Control*, vol. AC-29, no. 4, pp. 321–331, Apr. 1984.

[7] H. de Jong, “Modeling and simulation of genetic regulatory systems,” *J. Comput. Biol.*, vol. 9, pp. 69–105, 2002.

[8] S. Fortune, “Voronoi diagrams and Delaunay triangulations,” in *Handbook of Discrete and Computational Geometry*, J. E. Goodman and J. O’Rourke, Eds. Boca Raton, FL: CRC Press, 1997, pp. 377–388.

[9] K. Fukuda, Frequently asked questions in polyhedral computation Swiss Federal Inst. Technol., Lausanne, Switzerland, 2004 [Online]. Available: www.ifor.math.ethz.ch/~fukuda/polyfaq/polyfaq.html

[10] R. Ghosh, A. Tiwari, and C. Tomlin, “Automated symbolic reachability analysis; with application to delta-notch signaling automata,” in *Hybrid Systems: Computation and Control* , O. Maler and A. Pnueli, Eds. Berlin, Germany: Springer-Verlag, 2003, vol. 2623, Lecture Notes in Computer Science, pp. 233–248.

[11] L. C. G. J. M. Habets and J. H. van Schuppen, “Control of piecewise-linear hybrid systems on simplices and rectangles,” in *Hybrid Systems: Computation and Control* , M.D. Di Benedetto and A.L. Sangiovanni-Vincentelli, Eds. Berlin, Germany: Springer-Verlag, 2001, vol. 2034, Lecture Notes in Computer Science, pp. 261–274.

[12] ——, “A controllability result for piecewise-linear hybrid systems,” in *Proc. Eur. Control Conf.*, Porto, Portugal, 2001, pp. 3870–3873.

[13] ——, “A control problem for affine dynamical systems on a full-dimensional polytope,” *Automatica*, vol. 40, pp. 21–35, 2004.

[14] E. Haghverdi, P. Tabuada, and G. Pappas, “Bisimulation relations for dynamical and control systems,” in *Electronic Notes in Theoretical Computer Science*. New York: Elsevier, 2003, vol. 69, pp. 1–17.

[15] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, “What is decidable about hybrid automata?,” *J. Comput. Syst. Sci.*, vol. 57, pp. 94–124, 1998.

[16] A. Isidori*, Nonlinear Control Systems* , 2nd ed. Berlin, Germany: Springer-Verlag, 1989.

[17] B. Jeannet, Convex polyhedra library Verimag, Grenoble, France, 1999, Tech. Rep..

[18] G. Lafferriere, G. J. Pappas, and S. Sastry, “O-minimal hybrid systems,” *Math. Control, Signals, Syst.*, vol. 13, pp. 1–21, 2000.

[19] G. Lafferriere, G. J. Pappas, and S. Yovine, “A new class of decidable hybrid systems,” in *Hybrid Systems: Computation and Control* , F. W. Varager and J. H. van Schuppen, Eds. Berlin, Germany: Springer-Verlag, 1999, vol. 1569, Lecture Notes in Computer Science, pp. 137–151.

[20] C. W. Lee, “Subdivisions and triangulations of polytopes,” in *Handbook of Discrete and Computational Geometry*, J.E. Goodman and J. O’Rourke, Eds. Boca Raton, FL: CRC Press, 1997, pp. 271–290.

[21] N. E. Leonard, “Stability of a bottom-heavy underwater vehicle,” *Automatica*, vol. 33, pp. 331–346, 1997.

[22] A. Lotka*, Elements of Physical Biology* . New York: Dover Publications, 1925.

[23] R. Milner*, Communication and Concurrency* . Englewood Cliffs, NJ: Prentice-Hall, 1989.

[24] R. E. Mortensen, “A globally stable linear attitude regulator,” *Int. J. Control*, vol. 8, pp. 297–302, 1968.

[25] X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, “An approach to the description and analysis of hybrid systems,” in *Hybrid Systems* , R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, Eds. Berlin, Germany: Springer-Verlag, 1993, vol. 736, Lecture Notes in Computer Science, pp. 149–178.

[26] H. Nijmeijer and A. J. van der Schaft*, Nonlinear Dynamical Control Systems* . New York: Springer-Verlag, 1990.

[27] G. J. Pappas, “Bisimilar linear systems,” *Automatica*, vol. 39, pp. 2035–2047, 2003.

[28] D. Park, “Concurrency and automata on infinite sequences,” in *Theoretical Computer Science* , P. Deussen, Ed. Berlin, Germany: Springer-Verlag, 1981, vol. 104, Lecture Notes in Computer Science, pp. 167–183.

[29] A. Puri and P. Varaiya, “Decidability of hybrid systems with rectangular differential inclusions,” in *Computer Aided Verification* , D. L. Dill, Ed. Berlin, Germany: Springer-Verlag, 1994, vol. 818, Lecture Notes in Computer Science, pp. 95–104.

[30] A. Tiwari and G. Khanna, “Series of abstractions for hybrid automata,” in *Hybrid Systems: Computation and Control* , C.J. Tomlin and M.R. Greenstreet, Eds. Berlin, Germany: Springer-Verlag, 2002, vol. 2289, Lecture Notes in Computer Science, pp. 465–478.

[31] V. Volterra, “Fluctuations in the abundance of a species considered mathematically,” *Nature*, vol. 118, pp. 558–560, 1926.

Calin Belta (M’02) received the B.S. and M.Sc. degrees in control and computer science from the Technical University of Iasi, Iasi, Romania, the M.Sc. degree in electrical engineering from Louisiana State University, Baton Rouge, and the M.Sc. and Ph.D. degrees in mechanical sngineering from the University of Pennsylvania, Philadelphia, in 1995, 1996, 1999, 2001, and 2003, respectively.

He is currently an Assistant Professor in the Departments of Manufacturing Engineering and Aerospace and Mechanical Engineering at Boston

University, Boston, MA. His research interests include verification and control of hybrid systems, robot planning and control, and gene and metabolic networks.

Dr. Belta received an NSF CAREER award in 2005, a Fulbright study award in 1997, and was the Valedictorian of his class in 1995. He received the Best Poster Award at the International Conference on Systems Biology in 2004 and was a finalist for the ASME Design Engineering Technical Conference Best Paper Award in 2002 and for the Anton Philips Best Student Paper Award at the IEEE International Conference on Robotics and Automation in 2001.

Luc C.G.J.M. Habets received the M.Sc. degree degree in applied mathematics and the Ph.D. degree, both from Eindhoven University of Technology, Eindhoven, The Netherands, in 1989 and 1994, respectively.

He spent three years at the Institute for Dynamical Systems at Bremen University, Germany, and returned to Eindhoven in 1997 to become a Lecturer at the Department of Mathematics and Computer Science. Since 2000, he has also been affiliated as a Researcher with the Center for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands. His main research interests include hybrid systems, time-delay systems, behavioral theory, and algebraic and computational aspects in systems and control.