Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Stabilization of Recurrent Fuzzy Systems via Sum of Squares-based Hybrid Control ⋆

Stefan Gering, Jürgen Adamy

Technische Universität Darmstadt, Laboratory of Control Theory and Robotics, Landgraf-Georg-Str. 4, 64283 Darmstadt, Germany Tel.: +49-6151-164989, e-mail:{sgering, jadamy}@rtr.tu-darmstadt.de.

Abstract: This paper presents an approach for stabilization of equilibria in recurrent fuzzy systems. This type of dynamic fuzzy systems being defined via linguistic rules can be interpreted as interpolation between constant gradients, and therefore as hybrid dynamical system. It is shown that the latter viewpoint allows for a precise description of the system dynamics, but on the other hand lacks transparency. In order to render a given equilibrium of the recurrent fuzzy system globally asymptotically stable, local polynomial controllers are computed via sum of squares optimization to allow only for deterministic mode transitions on a micro level. In addition, the controlled recurrent fuzzy system can then be interpreted as finite deterministic automaton, thus allowing for analysis of system properties on a more abstract macro level. Relaxations are proposed in cases where recurrent fuzzy systems may not be rendered deterministic and the method is applied to two examples.

Keywords: Recurrent Fuzzy Systems; Hybrid Systems; Automata; Polynomial Control; Sum of Squares.

1. INTRODUCTION

Dynamic fuzzy systems allow not only for an approximate modeling of system dynamics but also to a certain extent for linguistic interpretability and thus transparency. In particular, recurrent fuzzy systems are defined by means of linguistic rules with constant gradients as conclusions. Although the model being obtained via expert knowledge or measurement data (see, e.g., Schwung et al. (2011)) may capture the plant dynamics reasonably well, simplicity of the rule base comes at the cost of a complicated model structure, which has to be considered for synthesis of stabilizing controllers.

It was shown in Adamy and Kempf (2003) that recurrent fuzzy systems may exhibit an automaton-like behavior, and if operated at core position vectors only, they behave equivalently to finite automata. Due to the possibility of further insight into the system dynamics, this motivates to study recurrent fuzzy systems from the perspective of hybrid automata as discussed, e.g., in Henzinger (2000), splitting the system into discrete-time and underlying continuous-time dynamics. The feedback synthesis concept for stabilization of known equilibria, which we present here, is based on two main ideas: The first stems from supervisory control discussed in Ramadge and Wonham (1987) or Heymann and Lin (1998) for discrete event systems, and in Koutsoukos et al. (2000) for hybrid systems, which aims at control of a process such that a certain sequence of discrete states is obtained. The second idea is due to the polyhedral partitioning of the state space, motivating a local control concept as discussed in Belta and Habets (2006) or Lin and Broucke (2006). Their aim is to steer all trajectories starting within a region towards a particular facet, although their approach is limited to cases where the system is input affine and the input matrix constant. A similar idea was carried out in Girard and Martin (2012), where a simplex partition was used in combination with linearization of the system dynamics and robust local controllers. In contrast to them, we make use of the underlying hypersquare decomposition of recurrent fuzzy systems and in addition use local polynomial controllers obtained via sum of squares optimization. The idea of utilizing abstractions from the system dynamics in order to reduce the control problem to a path planning problem was also applied to discrete-time linear systems in Tazaki and Imura (2008). One benefit of our approach is the fact, that no global Lyapunov function has to be computed, as was done in Mojica-Nava et al. (2010) in order to prove stability for switched systems. In addition, we attempt to utilize local controllers such that the controlled recurrent fuzzy system becomes deterministic in the discrete mode changes, which again supports the idea of system interpretability. Thus, methods from sum of squares optimization, hybrid systems and graph theory are utilized in order to stabilize known equilibria in recurrent fuzzy systems.

The remainder is organized as follows: In Sec. 2, basic definitions of recurrent fuzzy systems are briefly reviewed and the hybrid system representation of the system class under consideration is given. Sec. 3 then introduces a finite automaton representation, allowing for a precise separation between discrete and continuous dynamics of the hybrid system. Then, a synthesis method for local

⋆ This work was supported by the Deutsche Telekom Stiftung (www.telekom-stiftung.de) and the Studienstiftung des Deutschen Volkes (www.studienstiftung.de).

controllers based on sum of squares is introduced in Sec. 4. Sec. 5 then discusses the design of the controlled discrete system, whereas it is distinguished between cases where 1) a deterministic discrete behavior may be achieved and 2) nondeterminism has to be taken into account. Simulation results of both cases applied to examples are then given in Sec. 6. Concluding remarks are given in Sec. 7.

2. PRELIMINARIES

The following briefly reviews the definition of recurrent fuzzy systems (RFS) and introduces necessary notations. Further insights on RFS can be found in Adamy and Kempf (2003) and Adamy and Flemming (2006).

RFS are defined in the input state space $\mathcal{Z} = \mathcal{X} \times \mathcal{U} \subseteq \mathbb{R}^{N+M}$ with states $\boldsymbol{x} \in [\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max}]$, and inputs $\boldsymbol{u} \in [\boldsymbol{u}_{\min}, \boldsymbol{u}_{\max}]$. The dynamics of the system are defined linguistically by a finite number of linguistic differential equations

$$\begin{aligned} &\text{If } x_1 = L^{x_1}_{j_1}, \text{ and } \dots, \text{ and } x_n = L^{x_n}_{j_n}, \\ &\text{and } u_1 = L^{u_1}_{q_1}, \text{ and } \dots, \text{ and } u_m = L^{u_m}_{q_m}, \\ &\text{then } \dot{x}_1 = L^{\dot{x}_1}_{w_1}, \text{ and } \dots, \text{ and } \dot{x}_n = L^{\dot{x}_n}_{w_n}, \end{aligned} \tag{1}$$

with the complete set of rules forming the *rule base.* Linguistic values in the i-th dimension of the state- and input variables are denoted $L^{x_i}_{j_i}$ and $L^{u_i}_{q_i}$, describing them in qualitative terms (e.g., *low* or *high*).

Similarly, linguistic values $L^{\dot{x}_i}_{w_i(\boldsymbol{j},\boldsymbol{q})}$ describe state derivatives $\dot{x}_i$. In order to quantify the system dynamics, each linguistic value $L^{x_i}_{j_i}$, $L^{u_i}_{q_i}$ is associated with a crisp *core position* value $s^{x_i}_{j_i}$, $s^{u_i}_{q_i}$. For the description of the state derivatives, linguistic values $L^{\dot{x}_i}_{w_i}$ are associated with *core position derivatives* $s^{\dot{x}_i}_{w_i}$. When using vector notation $\boldsymbol{L}^{\boldsymbol{x}}_{\boldsymbol{j}} = \left[L^{x_1}_{j_1}, \dots, L^{x_n}_{j_n}\right]^T$, $\boldsymbol{L}^{\boldsymbol{u}}_{\boldsymbol{q}} = \left[L^{u_1}_{q_1}, \dots, L^{u_m}_{q_m}\right]^T$ and $\boldsymbol{L}^{\dot{\boldsymbol{x}}}_{\boldsymbol{w}(\boldsymbol{j},\boldsymbol{q})} = \left[L^{\dot{x}_1}_{w_1(\boldsymbol{j},\boldsymbol{q})}, \dots, L^{\dot{x}_n}_{w_n(\boldsymbol{j},\boldsymbol{q})}\right]^T$ for linguistic values, and similarly $\boldsymbol{s}^{\boldsymbol{x}}_{\boldsymbol{j}}$, $\boldsymbol{s}^{\boldsymbol{u}}_{\boldsymbol{q}}$, $\boldsymbol{s}^{\dot{\boldsymbol{x}}}_{\boldsymbol{w}}$ for core position and core position derivatives, the more compact representation

$$\begin{aligned} &\text{If } \boldsymbol{x} = \boldsymbol{L}^{\boldsymbol{x}}_{\boldsymbol{j}} \text{ and } \boldsymbol{u} = \boldsymbol{L}^{\boldsymbol{u}}_{\boldsymbol{q}}, \\ &\text{then } \dot{\boldsymbol{x}} = \boldsymbol{L}^{\dot{\boldsymbol{x}}}_{\boldsymbol{w}(\boldsymbol{j},\boldsymbol{q})} \end{aligned} \tag{2}$$

is obtained. Thus, by means of the rule base, gradients are defined at discrete points $(\boldsymbol{s}^{\boldsymbol{x}}_{\boldsymbol{j}}, \boldsymbol{s}^{\boldsymbol{u}}_{\boldsymbol{q}}) \in \mathcal{Z}$. Fig. 1 depicts neighboring core positions within an RFS, showing that a rectangular grid is induced in the input state space. The convex hull of core positions that are connected with a vector $(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{Z}$ is called *elementary hypersquare*, or simply *hypersquare* and is denoted $H_{\boldsymbol{l}}$, with $\boldsymbol{l}$ consisting of the lower indices of the limiting core positions. For ease of notation, we write H_i, H_j to indicate different hypersquares if ambiguity is excluded. With a slight abuse of notation, we also write $H^{\boldsymbol{x}}_{\boldsymbol{l}} := H_{\boldsymbol{l}} \cap \mathcal{X}$ and $H^{\boldsymbol{u}}_{\boldsymbol{l}} := H_{\boldsymbol{l}} \cap \mathcal{U}$ denoting the state and input subspace of $H_{\boldsymbol{l}}$. For a given equilibrium $\boldsymbol{x}^*$, the set of hypersquares containing $\boldsymbol{x}^*$ is denoted $\mathcal{H}_0 = \{H^{\boldsymbol{x}}_{\boldsymbol{j}} : \boldsymbol{x}^* \in H^{\boldsymbol{x}}_{\boldsymbol{j}}\}$, and the complement will be denoted $\mathcal{H}_1 = \{H^{\boldsymbol{x}}_{\boldsymbol{j}}\} \backslash \mathcal{H}_0$.

By using membership functions $\mu^{x_i}_{j_i}(x_i)$, $\mu^{u_i}_{q_i}(u_i)$ for fuzzification, a degree of membership between variables x_i, u_i and core positions $s^{x_i}_{j_i}$, $s^{u_i}_{q_i}$ is determined. If the algebraic product is used for aggregation and implication, the simple sum for accumulation of the single rules, and the center of singleton method for defuzzification, the form

$$\begin{aligned} \dot{\boldsymbol{x}} &= \sum_{\boldsymbol{j},\boldsymbol{q}} \boldsymbol{s}^{\dot{\boldsymbol{x}}}_{\boldsymbol{w}(\boldsymbol{j},\boldsymbol{q})} \cdot \prod_{i=1}^{N} \mu^{x_i}_{j_i}(x_i) \cdot \prod_{p=1}^{M} \mu^{u_p}_{q_p}(u_p) \\ &= \sum_{\boldsymbol{j},\boldsymbol{q}} \boldsymbol{s}^{\dot{\boldsymbol{x}}}_{\boldsymbol{w}(\boldsymbol{j},\boldsymbol{q})} \cdot \Xi_{\boldsymbol{j},\boldsymbol{q}}(\boldsymbol{x}, \boldsymbol{u}) \end{aligned} \tag{3}$$

is obtained (see Adamy and Kempf (2003)). Thus, the derivative at any point $(\boldsymbol{x}, \boldsymbol{u})$ is given by a summation over all core position gradients weighted by a premise $\Xi_{\boldsymbol{j},\boldsymbol{q}}(\boldsymbol{x}, \boldsymbol{u})$, which is the product of memberships in every dimension.

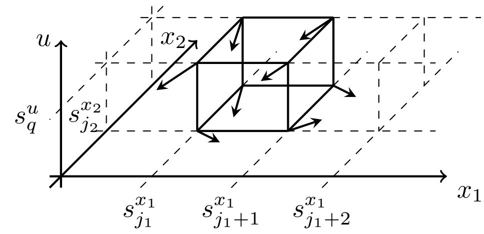

Fig. 1. Hypersquare of RFS with core position derivatives.

The transition function in (3) is depended on membership functions. Although these can be any continuous, convex functions with $\mu^{x_i}_{j_i} \mapsto [0,1]$, $\sum_{j_i} \mu^{x_i}_{j_i}(x_i) = 1$ and $\mu^{x_i}_{j_i}(s^{x_i}_{j_i}) = 1$, we consider triangular and ramp shaped membership functions

$$\mu^{x_i}_{j_i}(x_i) = \begin{cases} \frac{x_i - s^{x_i}_{j_{i-1}}}{s^{x_i}_{j_i} - s^{x_i}_{j_{i-1}}}, & x_{\mathrm{i,min}} \leq s^{x_i}_{j_{i-1}} \leq x_i < s^{x_i}_{j_i} \leq x_{\mathrm{i,max}} \\ \frac{s^{x_i}_{j_{i+1}} - x_i}{s^{x_i}_{j_{i+1}} - s^{x_i}_{j_i}}, & x_{\mathrm{i,min}} \leq s^{x_i}_{j_i} \leq x_i < s^{x_i}_{j_{i+1}} \leq x_{\mathrm{i,max}} \\ 1, & x_{\mathrm{i,min}} \geq x_i \vee x_i \geq x_{\mathrm{i,max}} \\ 0, & \text{else} \end{cases} \tag{4}$$

due to their ease of representation. By substituting (4) into (3),

$$\dot{\boldsymbol{x}} = \sum_{(\boldsymbol{j},\boldsymbol{q}) \in H_{\boldsymbol{l}}} \boldsymbol{s}^{\dot{\boldsymbol{x}}}_{\boldsymbol{w}(\boldsymbol{j},\boldsymbol{q})} \cdot \prod_{i=1}^{N} \mu^{x_i}_{j_i}(x_i) \cdot \prod_{p=1}^{M} \mu^{u_p}_{q_p}(u_p) \tag{5}$$

$$= \boldsymbol{a}_{0,\boldsymbol{l}} + \sum_{i=1}^{N} \boldsymbol{a}_{x_i,\boldsymbol{l}} x_i + \sum_{p=1}^{M} \boldsymbol{a}_{u_p,\boldsymbol{l}} u_p + \sum_{j=2}^{N} \sum_{i=1}^{j-1} \boldsymbol{a}_{x_{ij},\boldsymbol{l}} \cdot x_i x_j +$$

$$\sum_{p=1}^{M} u_p \sum_{j=2}^{N} \boldsymbol{a}_{x_i u_p,\boldsymbol{l}} x_i + \dots + \boldsymbol{a}_{x_{1\dots n} u_{1\dots M},\boldsymbol{l}} x_1 \cdots x_n u_1 \cdots u_M,$$

is obtained. Thus, for $(\boldsymbol{x}, \boldsymbol{u}) \in H_{\boldsymbol{l}}$, $\dot{\boldsymbol{x}}$ is a (linear) interpolation between connected core position derivatives and can be evaluated to a multi-affine polynomial in the state and input variables. Because of the resulting non-input affine system (5), the following assumption is made:

Assumption 1. The RFS is considered to be input affine, such that no product terms $u_i \cdot u_j$ occur in (5).

Due to the fact that many practical systems and all single input systems are input affine, Assumption 1 is not overly restrictive. Non-input affine systems can be treated similarly by appropriate linearization techniques. For brevity, this is not further elaborated on.

With Assumption 1, the local dynamics (5) can be rewritten as

$$\begin{aligned}\dot{\boldsymbol{x}} &= \overline{\boldsymbol{A}}_{\boldsymbol{l}}\overline{\boldsymbol{Z}}(\boldsymbol{x}) + \boldsymbol{B}_{\boldsymbol{l}}(\boldsymbol{x})\boldsymbol{u} \\ &= \boldsymbol{A}_{\boldsymbol{l}}\boldsymbol{Z}(\boldsymbol{x}) + \boldsymbol{a}_{0,\boldsymbol{l}} + \boldsymbol{B}_{\boldsymbol{l}}(\boldsymbol{x})\boldsymbol{u},\end{aligned} \tag{6}$$

where $\overline{\boldsymbol{Z}}(\boldsymbol{x}) = [1\ x_1\ x_2\ x_1x_2\ \dots\ x_1\cdots x_n]^T$ denotes the vector of monomials of $\boldsymbol{x}$ and $\overline{\boldsymbol{Z}}(\boldsymbol{x}) = \left[1\ \boldsymbol{Z}(\boldsymbol{x})^T\right]^T$.

2.1 Further Notation

A hypersquare facet is denoted F_{ij} with outer normal vector $\boldsymbol{n}_{ij}$ pointing from $H_i^{\boldsymbol{x}}$ to $H_j^{\boldsymbol{x}}$. The set of neighbors is denoted $\mathcal{N}(H_i^{\boldsymbol{x}}) = \{H_j^{\boldsymbol{x}} \mid H_i^{\boldsymbol{x}} \cap H_j^{\boldsymbol{x}} \neq \emptyset\}$. The distance between two hypersquares is given by $d(H_{\boldsymbol{i}}^{\boldsymbol{x}}, H_{\boldsymbol{j}}^{\boldsymbol{x}}) = \|\boldsymbol{i} - \boldsymbol{j}\|_1$. The distance between a hypersquare and a point is defined as $d(H_i^{\boldsymbol{x}}, \boldsymbol{x}) = \min_j\{d(H_i^{\boldsymbol{x}}, H_j^{\boldsymbol{x}}) \mid \boldsymbol{x} \in H_j^{\boldsymbol{x}}\}$. For symmetric matrices we use the shorthand notation $\mathrm{He}\{\boldsymbol{M}\} = \boldsymbol{M} + \boldsymbol{M}^T$, and the Hadamard product of two matrices of same dimension is written $\boldsymbol{M}_1 \circ \boldsymbol{M}_2$. Identity and matrix of all zeros are denoted $\boldsymbol{I}$ and $\boldsymbol{0}$, each being of appropriate dimensions. Eigenvalues of $\boldsymbol{M}$ are written $\lambda_i(\boldsymbol{M})$, and the imaginary unit is denoted by j. Positive and negative definiteness are denoted by $\succ$ and $\prec$. Polynomials $p(\boldsymbol{x})$ being a sum of squares (SOS) will be written $p(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}]$, meaning $p(\boldsymbol{x}) = \sum_i q_i^2(\boldsymbol{x})$. A multivariate polynomial $\boldsymbol{P}(\boldsymbol{x})$ being SOS is then denoted $\boldsymbol{P} \in \Sigma^N[\boldsymbol{x}]$. Clearly, $p(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}] \Rightarrow p(\boldsymbol{x}) \geq 0$. Strict inequality holds, if $\exists\varphi(\boldsymbol{x}) = \sum_{n=1}^N \sum_{i=1}^d \varphi_{ni}x_n^{2i}$, s.t. $p(\boldsymbol{x}) - \varphi(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}]$ (see Papachristodoulou and Prajna (2005) for details). For ease of notation, we sometimes omit $\varphi(\boldsymbol{x})$, if strict inequality is obvious.

3. FINITE AUTOMATON REPRESENTATION AND PROBLEM STATEMENT

It was shown in Sec. 2 that the dynamics of a RFS can be described by linguistic rules (2), allowing for a transparent yet approximate system model. This rule base representation on the other hand does not at first sight seem to be well suited for controller design, since it does not take the dynamics in the interpolating regions into account. The description of the system dynamics by piecewise polynomial functions (5) on the other hand uniquely defines the system dynamics in $\mathcal{Z}$, yet is no longer interpretable. To fill in the gap between these two system descriptions, we introduce in the following an intermediate system representation based on the underlying hybrid automaton representation of the system. The resulting finite automaton will then serve as basis for the synthesis of local controllers stabilizing a known equilibrium in RFS and also allows for system analysis from an abstract point of view.

From the piecewise defined dynamic function (5) it can be seen that RFS can be interpreted as *autonomous switched systems*

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{j}, \boldsymbol{q}, \boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{f}_{\boldsymbol{j},\boldsymbol{q}}(\boldsymbol{x}, \boldsymbol{u}) \tag{7}$$

depending on $H_{\boldsymbol{j}}^{\boldsymbol{x}}$ and $H_{\boldsymbol{q}}^{\boldsymbol{u}}$ being active. Thus, it may also be expressed as hybrid automaton G, which is a sixtupel

$$G = (\mathcal{H}^{\boldsymbol{x}}, \delta, \mathcal{Z}, \mathcal{F}, \mathcal{I}, \mathcal{T}). \tag{8}$$

Herein, $\mathcal{H}^{\boldsymbol{x}} = \{H_1^{\boldsymbol{x}}, \dots, H_L^{\boldsymbol{x}}\}$ represents the set of discrete hypersquares in the state space, which in terms of hybrid systems may also be referred to as *modes*. The relation $\delta \subseteq \mathcal{H}^{\boldsymbol{x}} \times \mathcal{H}^{\boldsymbol{x}}$ is the set of possible transitions between modes. The set of continuous variables is again represented by $\mathcal{Z} = \mathcal{X} \times \mathcal{U} \subseteq \mathbb{R}^{N+M}$, whereas $\mathcal{F} = \{\boldsymbol{f}_{\boldsymbol{j},\boldsymbol{q}}(\boldsymbol{x}, \boldsymbol{u})\}$ denotes the set of dynamic functions describing the state derivatives at particular modes. The initial states of the hybrid automaton are written $\mathcal{I} = \{H_0^{\boldsymbol{x}}, \boldsymbol{x}_0\}$ and the terminal states $\mathcal{T} = \{\mathcal{H}_0, \boldsymbol{x}^*\}$.

If local controllers $\boldsymbol{k}_{\boldsymbol{j}}(\boldsymbol{x})$ are applied for each mode $H_{\boldsymbol{j}}^{\boldsymbol{x}}$, the controlled RFS will be denoted

$$G_c = (\mathcal{H}^{\boldsymbol{x}}, \delta_c, \mathcal{Z}, \mathcal{F}_c, \mathcal{I}, \mathcal{T}), \tag{9}$$

with $\mathcal{F}_c = \{\boldsymbol{f}_{\boldsymbol{j},\boldsymbol{q}}(\boldsymbol{x}, \boldsymbol{k}_{\boldsymbol{j}}(\boldsymbol{x}))\}$.

Besides the transition relation δ, possible mode changes can also be expressed via the adjacency matrix $\boldsymbol{A}$, with

$$A_{ij} = \begin{cases} 1, & \text{if } (H_i^{\boldsymbol{x}}, H_j^{\boldsymbol{x}}) \in \delta, \\ 0, & \text{else.} \end{cases} \tag{10}$$

Similarly, $\boldsymbol{A}_c$ captures possible transitions of the controlled system. The main benefit of the automaton representation becomes obvious with the following proposition, in which $\boldsymbol{h} \in \{0,1\}^L$ denotes the vector of active modes, and L being the total number of hypersquares:

Proposition 1. If a RFS is controlled such that mode changes may only occur deterministically, i.e. $\sum_j A_{c,ij} \leq 1$, the sequence of modes is given by the discrete system

$$\boldsymbol{h}(k+1) = \boldsymbol{A}_c\boldsymbol{h}(k). \tag{11}$$

In this case, key properties of the RFS such as reachability, stability or existence of limit cycles can be determined by means of $\boldsymbol{A}_c$. Thus, the problem statement is twofold: First, local controllers are sought rendering G_c deterministic, and second, the controlled system is supposed to be stable with respect to a given equilibrium $\boldsymbol{x}^*$, which is assumed as $\boldsymbol{x}^* = \boldsymbol{0}$ for ease of notation but without loss of generality.

To solve this task, the following two main problems are addressed in the sequel:

Problem 1. Find a local controller $\boldsymbol{k}_i(\boldsymbol{x})$ for $H_i^{\boldsymbol{x}}$ such that a mode transition $(H_i^{\boldsymbol{x}}, H_j^{\boldsymbol{x}}) \in \delta$ is effectuated in finite time. In addition, $\forall H_i^{\boldsymbol{x}} \in \mathcal{H}_0^{\boldsymbol{x}}$, $\boldsymbol{x}(0) \in H_i^{\boldsymbol{x}}$ find a local controller $\boldsymbol{k}_i(\boldsymbol{x})$ such that the condition $\lim_{t\to\infty} \boldsymbol{x}(t) \to \boldsymbol{x}^*$ holds.

Problem 2. Assign local controllers $\boldsymbol{k}_i(\boldsymbol{x})$ for every $H_i^{\boldsymbol{x}}$ such that $\lim_{k\to\infty} \boldsymbol{h}(k) = \boldsymbol{h}^*$, $h_i = 1$, $\boldsymbol{x}^* \in H_i^{\boldsymbol{x}}$.

4. SUM OF SQUARES BASED SYNTHESIS OF LOCAL FACET CONTROLLERS

In this section, a solution to Problem 1 is presented, which is based on sum of squares optimization. The following definitions are akin to Belta and Habets (2006) and Lin and Broucke (2006):

Definition 1. (Exit and Blocked Facets). A facet $F_{ij} = H_i^{\boldsymbol{x}} \cap H_j^{\boldsymbol{x}}$ is said to be an *exit facet*, if $\boldsymbol{n}_{\mathrm{e},ij}^T \cdot \dot{\boldsymbol{x}} > 0, \forall \boldsymbol{x} \in F_{ij}$. It is said to be a *blocked facet*, if $\boldsymbol{n}_{\mathrm{b},ij}^T \cdot \dot{\boldsymbol{x}} < 0, \forall \boldsymbol{x} \in F_{ij}$.

If the dynamics of a hypersquare can be controlled such that it has only one exit facet and blocked facets else, the following definition applies:

Definition 2. (Strict Facet Controllability). A RFS is said to be *strictly facet controllable* within an elementary hypersquare $H_i^{\boldsymbol{x}}$ with respect to an exit facet F_{ij}, if for every $\boldsymbol{x}(0) \in H_i^{\boldsymbol{x}}$ there exists a trajectory $\boldsymbol{x}(t)$ such that the mode transition $(H_i^{\boldsymbol{x}}, H_j^{\boldsymbol{x}}) \in \delta$ occurs in finite time.

For now, we assume the RFS to be strictly facet controllable. In Sec. 5.2, the case is considered where this assumption is not justified.

In order for all trajectories in $H_i^{\boldsymbol{x}}$ to exit the hypersquare in finite time, it must not contain equilibria or limit cycles. Both are ensured not to exist, if $\dot{x}_n > 0$ or $\dot{x}_n < 0$ for some $n \in 1, \ldots, N$. If in addition it is assured that only F_{ij} is an exit facet, and $F_{ik}, k \neq j$ are blocked facets, the following sufficient conditions are obtained ensuring a mode transition (H_i, H_j) in finite time:

$$\boldsymbol{n}_{\mathrm{e},ij}^T \cdot \dot{\boldsymbol{x}} > 0, \quad \forall\, \boldsymbol{x} \in H_i^{\boldsymbol{x}}, \tag{12a}$$

$$\boldsymbol{n}_{\mathrm{b},ik}^T \cdot \dot{\boldsymbol{x}} < 0, \quad \forall\, \boldsymbol{x} \in F_{ij}. \tag{12b}$$

By utilizing a local polynomial feedback $\boldsymbol{u} = \boldsymbol{k}_i(\boldsymbol{x}) = \overline{\boldsymbol{K}}_i(\boldsymbol{x}) \cdot \overline{\boldsymbol{Z}}(\boldsymbol{x}) = \boldsymbol{K}_i(\boldsymbol{x}) \cdot \boldsymbol{Z}(\boldsymbol{x}) + \boldsymbol{k}_{0,i}$, the SOS conditions

$$\boldsymbol{n}_{\mathrm{e},ij}^T \cdot \left(\overline{\boldsymbol{A}}_{iq} + \boldsymbol{B}_{iq}(\boldsymbol{x})\overline{\boldsymbol{K}}_i(\boldsymbol{x})\right) \cdot \ldots \cdot \overline{\boldsymbol{Z}}(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}], \quad \forall \boldsymbol{x} \in H_i^{\boldsymbol{x}}, \tag{13a}$$

$$-\boldsymbol{n}_{\mathrm{b},ik}^T \cdot \left(\overline{\boldsymbol{A}}_{iq} + \boldsymbol{B}_{iq}(\boldsymbol{x})\overline{\boldsymbol{K}}_i(\boldsymbol{x})\right) \cdot \ldots \cdot \overline{\boldsymbol{Z}}(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}], \quad \forall \boldsymbol{x} \in F_{ik}, j \neq k, \tag{13b}$$

have to hold $\forall H_q^{\boldsymbol{u}} \in \mathcal{U}$. Applying the generalized S-procedure (see Parrilo (2000)) then yields

$$\boldsymbol{n}_{\mathrm{e},ij}^T \cdot \left(\overline{\boldsymbol{A}}_{iq} + \boldsymbol{B}_{iq}(\boldsymbol{x})\overline{\boldsymbol{K}}_i(\boldsymbol{x})\right) \overline{\boldsymbol{Z}}(\boldsymbol{x}) \ldots - t_{ijq}(\boldsymbol{x})\varepsilon_i(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}], \tag{14a}$$

$$-\boldsymbol{n}_{\mathrm{b},ik}^T \cdot \left(\overline{\boldsymbol{A}}_{iq} + \boldsymbol{B}_{iq}(\boldsymbol{x})\overline{\boldsymbol{K}}_i(\boldsymbol{x})\right) \overline{\boldsymbol{Z}}(\boldsymbol{x}) \ldots - t_{ikq}(\boldsymbol{x})\varphi_{ij}(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}],\ k \neq j, \tag{14b}$$

$$t_{ijq}(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}]. \tag{14c}$$

Therein, $\varepsilon_i(\boldsymbol{x})$ is an ellipsoidal outer approximation of $H_i^{\boldsymbol{x}}$, which is obtained by considering the bounding hyperplanes

$$\begin{bmatrix} E_{ij}^T & e_{ij} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} \geq 0, \quad j = 1, \ldots, 2N \tag{15}$$

of H_i. Using $\boldsymbol{E}_i = [\ldots\ E_{ij}\ \ldots]$ and $\boldsymbol{e}_i = [\ldots\ e_{ij}\ \ldots]$, the outer ellipse is then obtained as

$$\varepsilon_i(\boldsymbol{x}) = \left\{ \boldsymbol{x} \,\middle|\, -\begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}^T \begin{bmatrix} \boldsymbol{E}_i^T \boldsymbol{E}_i & \boldsymbol{E}_i \boldsymbol{e}_i \\ \boldsymbol{e}_i^T \boldsymbol{E}_i & \boldsymbol{e}_i^T \boldsymbol{e}_i - 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} \geq 0 \right\}. \tag{16}$$

Furthermore, facets F_{ij} in (14) are approximated via

$$\varphi_{ij}(\boldsymbol{x}) = \left\{ \boldsymbol{x} \,\middle|\, \varepsilon(\boldsymbol{x}) > 0,\ (\boldsymbol{n}_{ij}^T \cdot \boldsymbol{x}) = (\boldsymbol{n}_{ij}^T \cdot \boldsymbol{x}_{\mathrm{fp},ij}) \right\} \tag{17}$$

with $\boldsymbol{x}_{\mathrm{fp},ij}$ being an arbitrary point on F_{ij}.

Although the introduction of approximating ellipsoidal regions introduces a certain degree of conservatism since $H_i^{\boldsymbol{x}} \subset \varepsilon_i(\boldsymbol{x})$, computational complexity is reduced compared to consideration of strict polygonal regions.

In addition to constraints of the controlled dynamics in $H_i^{\boldsymbol{x}}$, constraints on the input $\boldsymbol{u}_{\min} \leq \boldsymbol{k}_i(\boldsymbol{x}) \leq \boldsymbol{u}_{\max}, \forall \boldsymbol{x} \in H_i^{\boldsymbol{x}}$ have to be taken into account as well to avoid saturation. This is ensured by means of the following equations:

$$\left(\overline{K}_{iq}^T(\boldsymbol{x}) \cdot \overline{\boldsymbol{Z}}(\boldsymbol{x}) - u_{\min,q}\right) - t_{\mathrm{sl},iq}(\boldsymbol{x})\varepsilon_i(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}], \tag{18a}$$

$$\left(u_{\max,q} - \overline{K}_{iq}^T(\boldsymbol{x}) \cdot \overline{\boldsymbol{Z}}(\boldsymbol{x})\right) - t_{\mathrm{su},iq}(\boldsymbol{x})\varepsilon_i(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}], \tag{18b}$$

$$t_{\mathrm{s\{l,u\}},iq}(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}]. \tag{18c}$$

Finally, the summarized equations read

$$\begin{aligned} &\text{Find } \boldsymbol{K}_i(\boldsymbol{x}), \boldsymbol{k}_{0,i}, \\ &\text{s.t. } \forall H_q^{\boldsymbol{u}} \in \mathcal{U},\ (14), (18). \end{aligned} \tag{19}$$

Lemma 1. The solution to (19) solves Problem 1 for $(H_i^{\boldsymbol{x}}, H_j^{\boldsymbol{x}}) \in \delta, i \neq j$.

By the same line of reasoning, a solution to Problem 1 is found for $H_i^{\boldsymbol{x}} \in \mathcal{H}_0$. In order to disable mode transitions, that is $(H_i^{\boldsymbol{x}}, H_j^{\boldsymbol{x}}) \notin \delta_c,\ i \neq j$, all facets have to be blocked, which can be ensured via (14b). Then, it remains to ensure $\boldsymbol{x}(t) \to \boldsymbol{0}$ as $t \to \infty$. Assuming a polynomial Lyapunov function $V(\boldsymbol{x}) = \boldsymbol{Z}(\boldsymbol{x})^T \boldsymbol{P} \boldsymbol{Z}(\boldsymbol{x})$, local asymptotic stability holds, if $\boldsymbol{P} \succ 0$ and

$$\dot{V}(\boldsymbol{x}) = \mathrm{He}\Big\{ \boldsymbol{Z}^T \boldsymbol{P} \boldsymbol{M}(\boldsymbol{x}) \big((\boldsymbol{A}_{iq} + \boldsymbol{B}_{iq}(\boldsymbol{x}) \boldsymbol{K}_i(\boldsymbol{x}))\, \boldsymbol{Z} + \ldots + \boldsymbol{a}_{0,iq} + \boldsymbol{k}_{0,i} \big) \Big\} < 0, \quad \forall \boldsymbol{x} \in H_i^{\boldsymbol{x}} \backslash \{\boldsymbol{0}\}, \tag{20}$$

where the abbreviation $\boldsymbol{M}(\boldsymbol{x}) := \partial \boldsymbol{Z}(\boldsymbol{x}) / \partial \boldsymbol{x}$ was used. By further introducing the shorthand notation $\tilde{\boldsymbol{A}}_{iq} = \boldsymbol{A}_{iq} + \boldsymbol{B}_{iq}(\boldsymbol{x}) \boldsymbol{K}_i(\boldsymbol{x})$, $\tilde{\boldsymbol{a}}_{0,iq} = \boldsymbol{a}_{0,iq} + \boldsymbol{k}_{0,i}$, (20) can be rewritten as

$$-\begin{bmatrix} \boldsymbol{Z} \\ 1 \end{bmatrix}^T \begin{bmatrix} \mathrm{He}\left\{ \boldsymbol{P} \boldsymbol{M} \tilde{\boldsymbol{A}}_{(\boldsymbol{j},\boldsymbol{q})} \right\} & * \\ \left(\boldsymbol{P} \boldsymbol{M} \tilde{\boldsymbol{a}}_{(\boldsymbol{j},\boldsymbol{q})} \right)^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{Z} \\ 1 \end{bmatrix} > 0,\ \forall \boldsymbol{x} \in H_i^{\boldsymbol{x}} \backslash \{\boldsymbol{0}\}, \tag{21a}$$

and by utilizing the generalized S-procedure

$$-\begin{bmatrix} \mathrm{He}\left\{ \boldsymbol{P} \boldsymbol{M} \tilde{\boldsymbol{A}}_{iq} \right\} & * \\ \left(\boldsymbol{P} \boldsymbol{M} \tilde{\boldsymbol{a}}_{iq} \right)^T & 0 \end{bmatrix} - \boldsymbol{T}_{iq}(\boldsymbol{x}) \varepsilon_i(\boldsymbol{x}) \in \Sigma^{N+1}[\boldsymbol{x}], \tag{22a}$$

$$\boldsymbol{T}_{iq}(\boldsymbol{x}) \in \Sigma^{N+1}[\boldsymbol{x}]. \tag{22b}$$

Thus, the final feasibility problem is obtained, which again takes bounds on the control input into account:

$$\text{Find } \boldsymbol{K}_i(\boldsymbol{x}), \boldsymbol{k}_{0,i}\ , \tag{23a}$$

$$\text{s.t. } \forall H_q^{\boldsymbol{u}} \in \mathcal{H}^{\boldsymbol{u}}, \quad (18), (22), \tag{23b}$$

$$\boldsymbol{P} \succ 0, \tag{23c}$$

$$t_{ijq}(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}], \tag{23d}$$

$$-\boldsymbol{n}_{\mathrm{b},ij}^T \cdot \left(\overline{\boldsymbol{A}}_{iq} + \boldsymbol{B}_{iq}(\boldsymbol{x})\overline{\boldsymbol{K}}_i\right) \overline{\boldsymbol{Z}}(\boldsymbol{x}) - t_{ijq}(\boldsymbol{x})\varphi_{ij}(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}]. \tag{23e}$$

Lemma 2. The solution to (23) solves Problem 1 for all $H_i \in \mathcal{H}_0$.

Due to bilinear terms $\boldsymbol{P}\boldsymbol{K}_i(\boldsymbol{x})$, $\boldsymbol{P}\boldsymbol{k}_{0,i}(\boldsymbol{x})$ in (22), the question arises on how to obtain a feasible solution to (23). One approach would be by a linearization approach (path-following) akin to Hassibi et al. (1999), whereas an initial solution for $\boldsymbol{k}_i(\boldsymbol{x})$ may be obtained by first neglecting (22). In some cases, setting, e.g., $\boldsymbol{P} = \boldsymbol{I}$, may already lead to a solution as well. An alternative approach for solving (23) is by appropriate variable substitution, which again leads to a feasibility problem being linear in the decision variables, as shown in Gering et al. (2013).

The advantage of this SOS-based approach is the independence of the local sub problems to be solved, since no global Lyapunov function is involved.

5. GLOBAL HYBRID CONTROL OF RFS

5.1 Hybrid Control of Strictly Facet Controllable RFS

With the previous discussion on local controllers, it is obvious to select controllers such that certain mode transitions are disabled and the set $\mathcal{H}_0$ is reached from every hypersquare in order to obtain stability. More precisely, if local controllers $\boldsymbol{k}_i(x)$ are found such that for every $H_i^{\boldsymbol{x}}$, there exists a path $(H_i^{\boldsymbol{x}}, \ldots, H_0^{\boldsymbol{x}}), H_0^{\boldsymbol{x}} = H_j^{\boldsymbol{x}} \in \mathcal{H}_0$, then the controlled RFS (9) is asymptotically stable. Thus, the search for appropriate controllers is linked to a search on the graph described by δ_c, for which efficient search algorithms exist (see, e.g., Hart et al. (1968)).

A practical approach for design of allowed mode transitions in G_c is by determining $\boldsymbol{A}_\mathrm{c}$ by means of a discrete feasibility problem, such that the conditions

$$A_{\mathrm{c},ij} = \begin{cases} 1, & (H_i, H_j) \in \delta,\ d(H_j^{\boldsymbol{x}}, \boldsymbol{x}^*) \le d(H_i^{\boldsymbol{x}}, \boldsymbol{x}^*), \\ & j = \arg\min_{k \in \mathcal{N}(i)} d(H_k^{\boldsymbol{x}}, \boldsymbol{x}^*), \\ 0, & \text{else}, \end{cases} \tag{24a}$$

$$\sum_j A_{\mathrm{c},ij} = 1, \quad (24\mathrm{b}) \qquad \boldsymbol{A}_\mathrm{c} \circ \boldsymbol{A}_\mathrm{c}^T = \boldsymbol{0}, \quad (24\mathrm{c})$$

$$\lambda_j(\boldsymbol{A}_\mathrm{c} + \boldsymbol{A}_\mathrm{c}^T) < \max_i\ \lambda_i(\boldsymbol{A}_\mathrm{c} + \boldsymbol{A}_\mathrm{c}^T), i \neq j \tag{24d}$$

hold.

Theorem 1. If for strict facet controllable RFS, local controllers are determined by (19) and (23), (24) hold, then the equilibrium $\boldsymbol{x}^*$ of the system is asymptotically stable in $\mathcal{Z}$.

Proof. With Lemma 1 it follows, that mode transitions $(H_i^{\boldsymbol{x}}, H_j^{\boldsymbol{x}})$ are effectuated in finite time if indicated by $A_{\mathrm{c},ij} = 1$. Due to (14), the modes change deterministically. Lemma 2 also ensures $\boldsymbol{x} \to \boldsymbol{x}^*$ if $\boldsymbol{x} \in \mathcal{H}_0$. Thus it remains to show that the sequence $H_i^{\boldsymbol{x}}(k)$ converges. As necessary condition, deadlocks must be excluded, that is $A_{ji} = 0$ if $A_{ij} = 1$, which is ensured via (24c). Furthermore, the graph described by $\boldsymbol{A}_\mathrm{c}$ must be weakly connected. Weak connectivity holds, if the underlying undirected graph $\boldsymbol{A}_\mathrm{c} + \boldsymbol{A}_\mathrm{c}^T$ has a maximum eigenvalue with multitude of 1 (see Gross and Yellen (2004)), which is ensured by (24d). Because the distance measure $d(H_i^{\boldsymbol{x}}, \boldsymbol{x}^*)$ induces a Lyapunov function for the macro dynamics (11), the system is stable in the sense of Lyapunov, and furthermore, due to the weak connectivity, it is ensured that no cycles occur between hypersquares $\{H_i^{\boldsymbol{x}}\}$ with $d(H_i^{\boldsymbol{x}}, \boldsymbol{x}^*) > 0$. Thus, $\mathcal{H}_0$ is reached in finite time.

Besides this sufficient condition for a stabilizing hybrid controller, the advantage of the proposed method by Theorem 1 clearly is in the resulting automaton-like behavior of the controlled RFS, which resembles to the idea of transparency of the RFS model itself.

5.2 Approximation for Hybrid Control of General RFS

The major drawback of the algorithm is its restriction to RFS with strictly facet controllable hypersquares. Loosely speaking, if this property has to hold for all facets of a hypersquare, it requires the possibility of independent control of the individual states in order to obtain a controlled RFS with deterministic hypersquare transitions. Thus, in the strict sense, the applicability is limited to a narrow subclass of RFS.

On the other hand, the idea may be carried over to general RFS by relaxing the SOS-conditions for local controllers in Sec. 4, if instead of (19), the optimization problem

$$\max_{\boldsymbol{K}_i(\boldsymbol{x}), \boldsymbol{k}_{0,i}} \sum_{j,q} \alpha_{ijq}, \tag{25a}$$

$$\text{s.t. } \forall H_q^{\boldsymbol{u}} \in \mathcal{U}, (18), \tag{25b}$$

$$\boldsymbol{n}_{\mathrm{e},ij}^T \cdot \left(\overline{\boldsymbol{A}}_{iq} + \boldsymbol{B}_{iq}(\boldsymbol{x})\overline{\boldsymbol{K}}_i(\boldsymbol{x})\right) \overline{\boldsymbol{Z}}(\boldsymbol{x}) \ldots - t_{ijq}(\boldsymbol{x})\varepsilon_i(\boldsymbol{x}) - \alpha_{ijq} \in \Sigma[\boldsymbol{x}], \tag{25c}$$

$$- \boldsymbol{n}_{\mathrm{b},ik}^T \cdot \left(\overline{\boldsymbol{A}}_{iq} + \boldsymbol{B}_{iq}(\boldsymbol{x})\overline{\boldsymbol{K}}_i(\boldsymbol{x})\right) \overline{\boldsymbol{Z}}(\boldsymbol{x}) \ldots - t_{ikq}(\boldsymbol{x})\varphi_{ij}(\boldsymbol{x}) - \alpha_{ijq} \in \Sigma[\boldsymbol{x}],\ k \neq j, \tag{25d}$$

$$t_{ijq}(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}] \tag{25e}$$

is solved for all $H_i^{\boldsymbol{x}} \in \mathcal{H}_1^{\boldsymbol{x}}$. Similarly, relaxing (23) leads to

$$\max_{\boldsymbol{K}_i(\boldsymbol{x}), \boldsymbol{k}_{0,i}} \sum_{j,q} \alpha_{ijq}, \tag{26a}$$

$$\text{s.t. } \forall H_q^{\boldsymbol{u}} \in \mathcal{U}, \quad (18), (22) \tag{26b}$$

$$\boldsymbol{P} \succ 0 \tag{26c}$$

$$t_{ijq}(\boldsymbol{x}) \in \Sigma[\boldsymbol{x}] \tag{26d}$$

$$- \boldsymbol{n}_{\mathrm{b},ij}^T \cdot \left(\overline{\boldsymbol{A}}_{iq} + \boldsymbol{B}_{iq}(\boldsymbol{x})\overline{\boldsymbol{K}}_i\right) \overline{\boldsymbol{Z}}(\boldsymbol{x}) - t_{ijq}(\boldsymbol{x})\varphi_{ij}(\boldsymbol{x}) - \alpha_{ijq} \in \Sigma[\boldsymbol{x}], \tag{26e}$$

through which a local stabilizing controller is obtained for all $H_i^{\boldsymbol{x}} \in \mathcal{H}_0^{\boldsymbol{x}}$. Note that the exact conditions (19) and (23) are recovered, if $\alpha_{ijq} \ge 0$.

The design of $\boldsymbol{A}_\mathrm{c}$ as discussed in Sec. 5.1 on the other hand remains exactly the same, which now might be referred to as *reference automaton* of the controlled RFS. Then, transparent and plausible design of stabilizing local controllers is still possible, although stability can no longer be guaranteed by means of Theorem 1 and thus has to be verified numerically instead. In addition, although $\boldsymbol{A}_\mathrm{c}$ is deterministic, this is not necessarily the case for the RFS. As a consequence of the non-deterministic mode changes, chattering effects may now occur, which have to be taken into consideration. In such a case, the effect of chattering may be lessened by hysteresis switching of the local controllers. Although the resulting vector field is in general also discontinuous if G_c has only deterministic mode changes, chattering in this case cannot occur, since after every transition $(H_i^{\boldsymbol{x}}, H_j^{\boldsymbol{x}})$, the facet F_{ji} is blocked.

6. NUMERICAL EXAMPLES

In order to illustrate the aforementioned concept, we consider two examples of RFS, for which stabilizing controllers are computed. Although the general idea of RFS is to approximately model plant dynamics based on expert knowledge or measurements in the absence of exact numeric models, they are in this case obtained from nonlinear differential equations for simplicity.

Example 1. First, we consider an omnidirectional robot as discussed in Campion et al. (1996). Since robots of this type have off-centered castor wheels or Swedish wheels rather than fixed wheels, they can move in any direction in the plane at each position. Its continuous time dynamics, which serve here as kinematic reference model, are given by

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}\cos x_3 & -\sin x_3 & 0\\ \sin x_3 & \cos x_3 & 0\\ 0 & 0 & 1\end{bmatrix} \cdot \begin{bmatrix}u_1\\ u_2\\ u_3\end{bmatrix}, \qquad (27)$$

where $(x_1, x_2) \in \mathbb{R}^2$ is the robot's absolute position in the plane, $x_3 \in [-\pi, \pi]$ is the rotational angle around the yaw axis and u_i are the control inputs.

In order to model the dynamics of the robot as RFS, first the core positions $\{s_{j_1}^{x_1}\} = \{s_{j_2}^{x_2}\} = \{-1, -0.5, 0, 0.5, 1\}$ are chosen describing the spacial variables x_1 and x_2, whereas $\{s_{j_3}^{x_3}\} = \{-\pi, -\pi/2, 0, \pi/2, \pi\}$ are selected as core positions of the rotational angle. The core position values are then associated with linguistic variables $\{L_{j_1}^{x_1}\} = \{L_{j_2}^{x_2}\} = \{L_{j_3}^{x_3}\} = \{$neg./pos. big, neg./pos. small, zero$\}$ (abbreviated nb, ns, z, ps, pb). The inputs are described by the linguistic values $L_{q_i}^{u_i} = \{$negative, zero, positive$\}$, which are associated with crisp values $\{s_{q_i}^{u_i}\} = \{-10, 0, 10\}$. The state derivatives of the CTRFS are then defined linguistically at core positions according to (27) and are given in Table 1 for the example of the rule base with fixed $L_{j_1}^{x_1} = L_{j_2}^{x_2} =$ negative and $L_{q_1}^{u_1} = L_{q_2}^{u_2} =$ positive.

Table 1. Rule base of Example 1 with $L_{j_1}^{x_1} = L_{j_2}^{x_2} =$ negative and $L_{q_1}^{u_1} = L_{q_2}^{u_2} =$ positive

$L_j^{\dot{\boldsymbol{x}}}$		$L_{j_3}^{x_3}$				
		nb	ns	z	ps	pb
$L_{q_3}^{u_3}$	n	n/n/n	p/n/n	p/p/n	n/p/n	n/n/n
	z	n/n/z	p/n/z	p/p/z	n/p/z	n/n/z
	p	n/n/p	p/n/p	p/p/p	n/p/p	n/n/p

For the equilibrium $\boldsymbol{x}^* = [\text{pos. small, pos. small, zero}]^T$, a stabilizing controller for this strictly facet controllable RFS is computed. The resulting trajectory for an initial value of $\boldsymbol{x}_0 = [-0.8,\ -0.8,\ -2]^T$ is shown in Fig. 2, where the arrows indicate the robot's orientation and the numbers inside of the hypersquare are the distances $d(H_i^{\boldsymbol{x}}, \boldsymbol{x}^*)$ to the equilibrium with regard to the x_1/x_2-plane. As can be seen from the plot, the trajectory crosses facets always towards hypersquares with smaller distance to $\boldsymbol{x}^*$. In addition, the equilibrium is reached asymptotically.

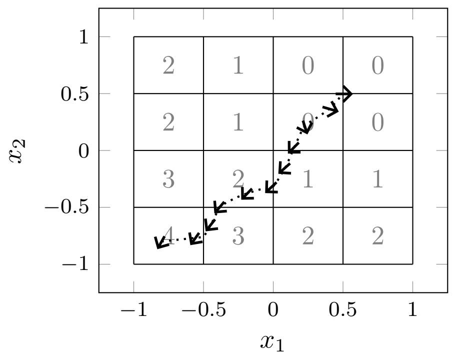

Fig. 2. Trajectory of the omnidirectional robot with initial value $\boldsymbol{x}_0 = [-0.8, -0.8, -2]^T$: Arrows indicate the robot's orientation in the x_1/x_2-plane, numbers within hypersquares are the distance to $[x_1^*, x_2^*]^T$.

Example 2. Now, stabilization is considered for an RFS, which is not strictly facet controllable for all hypersquare facets. The system under consideration is the well known Van-der-Pol oscillator augmented by an input

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}x_2\\ -x_1 + \beta\left(1 - x_1^2\right) x_2 + u\end{bmatrix}, \qquad (28)$$

which is modeled as RFS with core positions at $\{s_{j_1}^{x_1}\} = \{s_{j_2}^{x_2}\} = \{0, \pm 1, \pm 2\}$, $\{s_q^u\} = \{0, \pm 5\}$. By choice of $\beta = 1$ in (28), the open-loop system is known to have a stable limit cycle, which is shown in Fig. 3. By applying the relaxed controller synthesis procedure described in Sec. 5.2, a reference adjacency matrix $\boldsymbol{A}_c$ is obtained, which is visualized in Fig. 5b. As can be seen, mode transitions are preferably carried out towards $\mathcal{H}_0$, although the actual controlled system may behave non-deterministically.

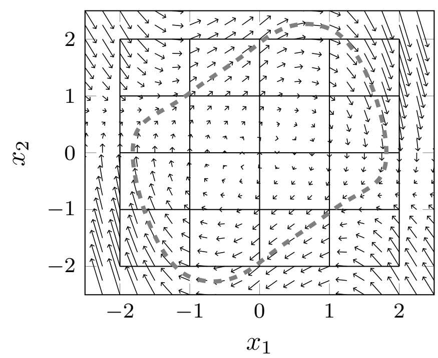

Fig. 3. Phase plot of open-loop RFS with stable limit cycle.

If on the other hand, local controllers can be found such that mode transitions occur deterministically, then stability is already guaranteed by analyzing the eigenvalues of $\boldsymbol{A}_c$, which are in this case $\lambda_{1,\ldots,12} = 0$, $\lambda_{13} = -1$, $\lambda_{14,15} = \pm j$, $\lambda_{16} = 1$. Obviously, $\boldsymbol{A}_c$ has no anti-stable eigenvalues outside the unit disk in the complex plane, but may show an oscillating behavior due to eigenvalues at $\lambda_{14,15} = \pm j$. These on the other hand correspond to an oscillation within $\mathcal{H}_0$, which does not prevent trajectories from reaching the equilibrium $\boldsymbol{x}^* = \boldsymbol{0}$ asymptotically.

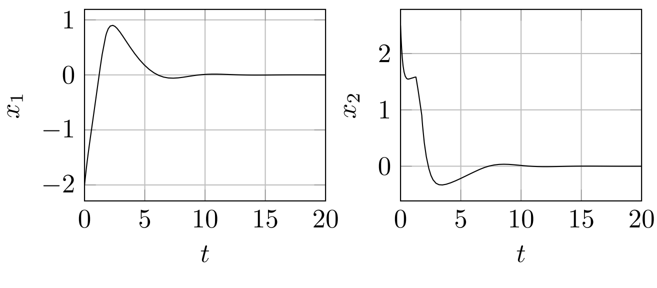

Fig. 4. Development of x_1 and x_2 in controlled RFS for initial value of $\boldsymbol{x}_0 = [-2.0,\ 2.5]^T$.

From Fig. 4, the development of the states of the controlled RFS is shown for an initial value of $\boldsymbol{x}_0 = [-2.0,\ 2.5]^T$, from which the asymptotic stability of the equilibrium can be seen. Furthermore, Fig. 5a depicts the system input during control, showing that the restriction $|u| \le 5$ holds. From the phase plot in Fig. 6, stability of the closed-loop system becomes apparent as well.

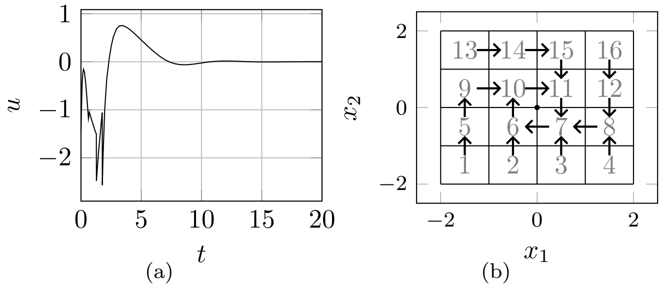

Fig. 5. (a) Input u during control and (b) controlled reference automaton with labeled hypersquares.

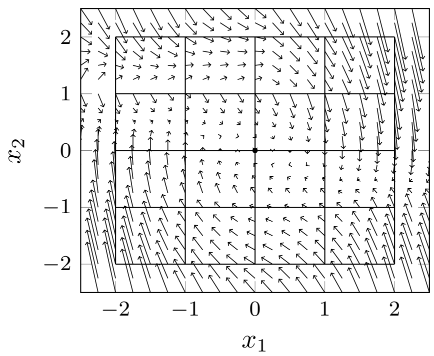

Fig. 6. Phase plot of closed-loop system.

7. CONCLUSION

A method for synthesis of hybrid controllers was discussed in order to stabilize equilibria in recurrent fuzzy systems. For strictly facet controllable recurrent fuzzy systems, this task was solved in a two-step procedure: First, local controllers are computed by means of sum of squares optimization, yielding deterministic hypersquare changes in the controlled system. Then, the resulting controlled hybrid system is solely described by means of a deterministic automaton. This allows for insight in various system properties, such as reachability, stability, occurrence of limit cycles, deadlocks, etc. Abstraction from the hybrid polynomial system offers the advantage of deeper insight into the system behavior and is therefore related to the idea of transparent system description via fuzzy logic. By relaxing the synthesis equations, the same approach may be applied to non-strictly facet controllable RFS. Although stability conditions do no longer hold in the strict sense, the concept may still be applied.

Research on this topic is far from being finished. One major future direction could be the extension to more general supervisory control allowing for set point changes. In addition, the design of nonlinear control by means of a linear system describing hypersquare transitions lends itself to existing linear controller design for discrete-time systems, which might allow for explicit restrictions on the eigenvalues of the discrete system.

REFERENCES

Adamy, J. and Flemming, A. (2006). Equilibria of continuous-time recurrent fuzzy systems. *Fuzzy Sets and Systems*, 157(22), 2913–2933.

Adamy, J. and Kempf, R. (2003). Regularity and chaos in recurrent fuzzy systems. *Fuzzy Sets and Systems*, 140(2), 259–284.

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

Campion, G., Bastin, G., and Dandrea-Novel, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Trans. on Robotics and Automation*, 12(1), 47–62.

Gering, S., Schwung, A., Gußner, T., and Adamy, J. (2013). Sum of squares approaches for control of continuous-time recurrent fuzzy systems. In *Proc. of the 21st IEEE Mediterranean Conference on Control and Automation*, 271–277. Chania, Greece.

Girard, A. and Martin, S. (2012). Synthesis for constrained nonlinear systems using hybridization and robust controllers on simplices. *IEEE Transactions on Automatic Control*, 57(4), 1046–1051.

Gross, J.L. and Yellen, J. (eds.) (2004). *Handbook of Graph Theory*. CRC Press.

Hart, P.E., Nilsson, N.J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100–107.

Hassibi, A., How, J., and Boyd, S. (1999). A path-following method for solving bmi problems in control. In *Proceedings of the American Control Conference*, volume 2, 1385 – 1389. San Diego, California, USA.

Henzinger, T.A. (2000). *The theory of hybrid automata*. Springer.

Heymann, M. and Lin, F. (1998). Discrete-event control of nondeterministic systems. *IEEE Transactions on Automatic Control*, 43(1), 3–17.

Koutsoukos, X.D., Antsaklis, P.J., Stiver, J.A., and Lemmon, M.D. (2000). Supervisory control of hybrid systems. *Proc. of the IEEE*, 88(7), 1026–1049.

Lin, Z. and Broucke, M. (2006). Resolving control to facet problems for affine hypersurface systems on simplices. In *Decision and Control, 2006 45th IEEE Conference on*, 2625–2630. doi:10.1109/CDC.2006.377067.

Mojica-Nava, E., Quijano, N., Rakoto-Ravalontsalama, N., and Gauthier, A. (2010). A polynomial approach for stability analysis of switched systems. *Systems & Control Letters*, 59(2), 98–104.

Papachristodoulou, A. and Prajna, S. (2005). A tutorial on sum of squares techniques for systems analysis. In *Proceedings of the American Control Conference*, volume 4, 2686–2700. San Diego, California, USA.

Parrilo, P.A. (2000). *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. Ph.D. thesis, California Institute of Technology Pasadena, California.

Ramadge, P.J. and Wonham, W.M. (1987). Supervisory control of a class of discrete event processes. *SIAM journal on control and optimization*, 25(1), 206–230.

Schwung, A., Gußner, T., and Adamy, J. (2011). Stability Analysis of Recurrent Fuzzy Systems: A Hybrid System and SOS Approach. *IEEE Transactions on Fuzzy Systems*, 19(3), 423–431.

Tazaki, Y. and Imura, J.i. (2008). Finite abstractions of discrete-time linear systems and its application to optimal control. In *Proceedings of the 17th IFAC world congress*, 10201–10206. Seoul, South Korea.