2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Robust Stabilization of Recurrent Fuzzy Systems via Switching Control

Stefan Gering, Wolfgang Krippner, Jürgen Adamy

***Abstract*— A method for stabilization of known equilibria in recurrent fuzzy systems is presented, which particularly accounts for model uncertainties. Since the dynamics of recurrent fuzzy systems are defined over a rectangular grid, it is first observed that for stability analysis, only gradient conditions at grid points have to be considered given that the inputs are piecewise constant. Therefore, a robust structure variable controller is proposed, switching between constant inputs. In order to prevent the system from deadlock phenomena due to the switching of the system, the structure variable control is augmented by a piecewise polynomial controller, guaranteeing asymptotic stability. The proposed method is applied to the example of an inverted pendulum.**

I. Introduction

Among the many concepts of utilizing fuzzy logic for modeling of dynamic processes, two generally different approaches can be identified: The first is driven by the need for high precision of the system model, which mostly leads to dynamic fuzzy models of high complexity. The second approach tries to incorporate the basic idea of fuzzy logic to model a system in a transparent and linguistically interpretable way. Recurrent fuzzy systems [1], piecewise bilinear systems [2] and more generally fuzzy systems with singleton consequences [3] can be subsumed in the latter class. While offering a high degree of linguistic interpretability, their drawback is the inherit model uncertainty stemming from the approximate nature of these dynamic fuzzy systems.

Nevertheless, given an approximate model, it is desirable to control the process despite model uncertainties. The question on how to stabilize equilibria in recurrent fuzzy systems by means of fuzzy controllers was already addressed (see, e.g., [4]), neglecting robustness issues. For the closely related class of piecewise bilinear systems, [5] presented an approach based on feedback linearization, taking uncertainties into account.

This paper now presents a control strategy for recurrent fuzzy systems based on structure variable control, taking model uncertainties explicitly into consideration. Because the dynamics of a recurrent fuzzy system can be interpreted as being defined piecewise over polytopes, a key observation concerning stability analysis is that for constant control inputs, dynamic properties only have to be considered at the vertices of the polytopes, which reduces the computational complexity significantly. This observation is akin to a line of research (see, e.g., [6]), which studies control to facet problems on polytopes. In contrast to [6], the dynamics of recurrent fuzzy systems studied here do not have in general a constant control vector.

Since the assumption of a piecewise constant controller is rather restrictive, the synthesis concept does not aim at determining a constant control input for each rectangle of the recurrent fuzzy systems, but instead for a finer partition obtained by a split-and-merge technique. The polytopes of this refined partition are such that dynamics defined over them is homogenous to a certain degree. The controller synthesis then takes general model uncertainties into account such that robustness is maximized.

Due to the switching control law, the dynamic function of the recurrent fuzzy system is in general discontinuous, which possibly might lead to deadlock behavior. Therefore, the robust controller only guarantees stability in the sense of Lyapunov. In order to achieve asymptotic stability, additional local polynomial controllers are computed preventing the system from deadlock.

The remainder is organized as follows: In Sec. II, recurrent fuzzy systems are revisited roughly, their general dynamic function is derived and some notation is introduced. The analysis of stability is then outlined in Sec. III, whereas the controller synthesis is then discussed in Sec. IV. In Sec. V, the proposed control strategy is applied to the inverted pendulum example and Sec. VI gives concluding remarks.

II. Preliminaries

A. Recurrent Fuzzy Systems

The definition of recurrent fuzzy systems (RFS) is briefly revisited in this section, additional insights are provided in [1] and [7]. Their dynamics are defined within the input state space $\mathcal{Z} = \mathcal{X} \times \mathcal{U} \subseteq \mathbb{R}^{n+m}$, $\mathbf{x} \in [\mathbf{x}_{\min}, \mathbf{x}_{\max}]$ and $\mathbf{u} \in [\mathbf{u}_{\min}, \mathbf{u}_{\max}]$, by means of linguistic differential equations

$$\begin{aligned} &\text{If } \mathbf{x} = \mathbf{L}^{\mathbf{x}}_{\mathbf{j}} \text{ and } \mathbf{u} = \mathbf{L}^{\mathbf{u}}_{\mathbf{q}}, \\ &\text{then } \dot{\mathbf{x}} = \mathbf{L}^{\dot{\mathbf{x}}}_{\mathbf{w}(\mathbf{j},\mathbf{q})}. \end{aligned} \tag{1}$$

$\mathbf{L}^{\mathbf{x}}_{\mathbf{j}}$, $\mathbf{L}^{\mathbf{u}}_{\mathbf{q}}$ denote vectors of linguistic values in the state and input space, whereas $\mathbf{L}^{\dot{\mathbf{x}}}_{\mathbf{w}(\mathbf{j},\mathbf{q})}$ denotes vectors of linguistic values describing gradients. In order to be able to transfer the linguistic rules into a numerical representation, core position vectors $\mathbf{s}^{\mathbf{x}}_{\mathbf{j}} \in \mathbb{R}^n$, $\mathbf{s}^{\mathbf{u}}_{\mathbf{q}} \in \mathbb{R}^m$ and core position gradients $\mathbf{s}^{\dot{\mathbf{x}}}_{\mathbf{w}} \in \mathbb{R}^n$ are associated with the linguistic vectors. By means of the rule base, gradients are defined at discrete points $(\mathbf{s}^{\mathbf{x}}_{\mathbf{j}}, \mathbf{s}^{\mathbf{u}}_{\mathbf{q}}) \in \mathcal{Z}$. This is depicted in Fig. 1, from which it can also be seen that by definition, the core position vectors induce a

The authors are with the Laboratory of Control Theory and Robotics at Technische Universität Darmstadt, Germany, { sgering, jadamy }@rtr.tu-darmstadt.de

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

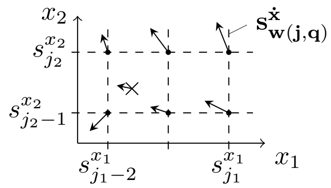

Fig. 1. Hypersquare partition of RFS with core position derivatives. Gradients within hypersquares are interpolations of core position derivatives.

rectangular grid in $\mathcal{Z}$. A *hypersquare* is the convex hull of core positions being neighbors of a vector $(\mathbf{x}, \mathbf{u}) \in \mathcal{Z}$ and is denoted $H_\mathbf{l}$, where $\mathbf{l}$ consists of the lower core position indices. With slight abuse of notation we write $H^\mathbf{x}_\mathbf{l} := H_\mathbf{l} \cap \mathcal{X}$ and $H^\mathbf{u}_\mathbf{l} := H_\mathbf{l} \cap \mathcal{U}$ for hypersquares in the state space and in the input space, respectively. If ambiguity is excluded, we also write H_i, H_j to indicate two different hypersquares. The same shorthand notation is sometimes used for core position vectors, e.g., $\mathbf{s}^\mathbf{x}_i$, $\mathbf{s}^\mathbf{x}_j$.

In order to assign a degree of membership to core positions $s^{x_i}_{j_i}$, $s^{u_p}_{q_p}$, membership functions $\mu^{x_i}_{j_i}(x_i)$, $\mu^{u_p}_{q_p}(u_p)$ are introduced for fuzzification. Furthermore, 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 [8]. This choice allows for the representation of the state derivative

$$\begin{aligned}\dot{\mathbf{x}} &= \sum_{\mathbf{j},\mathbf{q}} \mathbf{s}^{\dot{\mathbf{x}}}_{\mathbf{w}(\mathbf{j},\mathbf{q})} \cdot \prod_{i=1}^{n} \mu^{x_i}_{j_i}(x_i) \cdot \prod_{p=1}^{m} \mu^{u_p}_{q_p}(u_p) \\ &= \sum_{\mathbf{j},\mathbf{q}} \mathbf{s}^{\dot{\mathbf{x}}}_{\mathbf{w}(\mathbf{j},\mathbf{q})} \cdot \Xi_{(\mathbf{j},\mathbf{q})}(\mathbf{x}, \mathbf{u})\end{aligned} \tag{2}$$

as summation over core position gradients weighted by the premise, which is the product of memberships in every dimension.

B. Further Notation

Throughout the paper, the equilibrium of the RFS to be stabilized is denoted $\mathbf{x}^*$ and the number of hypersquares in $\mathcal{X}$ is denoted R_n. The facet between two neighboring hypersquares $H^\mathbf{x}_i, H^\mathbf{x}_j$ is denoted F_{ij}, with outer normal vector $\mathbf{n}_{ij}$ pointing from $H^\mathbf{x}_i$ to $H^\mathbf{x}_j$.

The identity matrix is denoted $\mathbf{I}$. The notation $\succ 0$, $\prec 0$ is used to indicate positive and negative definiteness of symmetric matrices. The trace of a matrix is denoted $\operatorname{tr}(\mathbf{M})$, $\operatorname{diag}(m_1, \ldots, m_n)$ is a diagonal matrix. The operator $\operatorname{vec}(\mathbf{M})$ rearranges the elements of $\mathbf{M}$ column-wise into a vector.

A multivariate normal distribution with mean $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$ is written $\mathcal{N}(\mathbf{m}, \mathbf{\Sigma})$. The signum function is written as $\sigma(\cdot)$.

For polynomials $p(\mathbf{x}) = \sum_i q_i^2(\mathbf{x})$ being a sum of squares (SOS), the notation $p(\mathbf{x}) \in \Sigma[\mathbf{x}]$ is used. Then, from $p(\mathbf{x}) \in \Sigma[\mathbf{x}]$, $p(\mathbf{x}) \geq 0$ follows. If $\exists \varphi(\mathbf{x}) = \sum_{n=1}^{N} \sum_{i=1}^{d} \varphi_{ni} x_n^{2i}$, s.t. $p(\mathbf{x}) - \varphi(\mathbf{x}) \in \Sigma[\mathbf{x}]$, then strict inequality holds (see [9] for details). For ease of notation, we omit $\varphi(\mathbf{x})$, if strict inequality is obvious.

III. Stability Analysis

The following discussion first considers the stability analysis of an autonomous RFS, which is then extended to the special case of an RFS controlled by discontinuous control. Subsequently, the equilibrium of the system is assumed to be at the origin and incident with a core position. These assumptions are not overly restrictive, since the coordinates of every RFS can be transformed such that $\mathbf{x}^* = \mathbf{0}$.

The following Theorem provides sufficient conditions for stability of the equilibrium:

Theorem 1: Consider the dynamics of the autonomous RFS

$$\dot{\mathbf{x}} = \sum_{\mathbf{j}} \mathbf{s}^{\dot{\mathbf{x}}}_{\mathbf{j}} \prod_{i=1}^{n} \mu^{x_i}_{j_i}(x_i) \tag{3}$$

with $\mathbf{x}^* = \mathbf{0}$ being incident with a core position. If there exists a $\mathbf{P} \succ 0$, such that for all $H^\mathbf{x}_k \in \mathcal{X}, \{\mathbf{s}^\mathbf{x}_i, \mathbf{s}^\mathbf{x}_j\} \in H^\mathbf{x}_k \backslash \{\mathbf{0}\}$,

$$\left(\mathbf{s}^\mathbf{x}_i\right)^T \mathbf{P} \mathbf{s}^{\dot{\mathbf{x}}}_i < 0, \tag{4a}$$

$$\frac{\left(\mathbf{s}^\mathbf{x}_i\right)^T \mathbf{P} \mathbf{s}^{\dot{\mathbf{x}}}_j + \left(\mathbf{s}^\mathbf{x}_j\right)^T \mathbf{P} \mathbf{s}^{\dot{\mathbf{x}}}_i}{2} < 0 \tag{4b}$$

and $\dot{\mathbf{x}}^* = \mathbf{0}$, then the equilibrium is asymptotically stable on $\mathcal{X}$.

Proof: By means of the well known Lyapunov theorem, the existence of $V(\mathbf{x}) = 1/2 \cdot \mathbf{x}^T \mathbf{P} \mathbf{x} > 0$, and $\dot{V}(\mathbf{x}) < 0, \forall \mathbf{x} \neq \mathbf{0}$, is a sufficient condition for asymptotic stability of (3).

With abuse of notation, let "$j \setminus \mathbf{0}$" denote all indices not corresponding to the origin. Then, with

$$\begin{aligned}\mathbf{x} &= \sum_{\mathbf{j}} \mathbf{s}^\mathbf{x}_\mathbf{j} \prod_{i=1}^{n} \mu^{x_i}_{j_i}(x_i) = \sum_{\mathbf{j}} \mathbf{s}^\mathbf{x}_\mathbf{j} \cdot \Xi_\mathbf{j}(\mathbf{x}) \\ &= \sum_{\mathbf{j} \setminus \mathbf{0}} \mathbf{s}^\mathbf{x}_\mathbf{j} \cdot \Xi_\mathbf{j}(\mathbf{x}) + \underbrace{\mathbf{x}^* \cdot \Xi_\mathbf{0}(\mathbf{x})}_{\mathbf{0}}\end{aligned} \tag{5}$$

and similar notation for $\dot{\mathbf{x}}$,

$$\begin{aligned}\dot{V} &= \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} \\ &= \left(\sum_{j \setminus \mathbf{0}} \mathbf{s}^\mathbf{x}_j \cdot \Xi_j(\mathbf{x})\right)^T \mathbf{P} \left(\sum_{i \setminus \mathbf{0}} \mathbf{s}^{\dot{\mathbf{x}}}_i \cdot \Xi_i(\mathbf{x})\right) \\ &= \sum_{j \setminus \mathbf{0}} \sum_{i \setminus \mathbf{0}} \Xi_j(\mathbf{x}) \Xi_i(\mathbf{x}) \cdot \left(\mathbf{s}^\mathbf{x}_j\right)^T \mathbf{P} \mathbf{s}^{\dot{\mathbf{x}}}_i \\ &= \sum_{i \setminus \mathbf{0}} \Xi_i^2 \left(\mathbf{s}^\mathbf{x}_i\right)^T \mathbf{P} \mathbf{s}^{\dot{\mathbf{x}}}_i + \sum_{j \setminus \mathbf{0}} \sum_{i \setminus \mathbf{0}, i \neq j} \Xi_j \Xi_i \left(\mathbf{s}^\mathbf{x}_j\right)^T \mathbf{P} \mathbf{s}^{\dot{\mathbf{x}}}_i \\ &= \sum_{i \setminus \mathbf{0}} \Xi_i^2 \cdot \underbrace{\left(\mathbf{s}^\mathbf{x}_i\right)^T \mathbf{P} \mathbf{s}^{\dot{\mathbf{x}}}_i}_{<0} + \ldots \\ &\quad + 2 \sum_{j \setminus \mathbf{0}} \sum_{i < j, i \setminus \mathbf{0}} \underbrace{\Xi_j \Xi_i}_{\geq 0} \underbrace{\frac{\left(\mathbf{s}^\mathbf{x}_j\right)^T \mathbf{P} \mathbf{s}^{\dot{\mathbf{x}}}_i + \left(\mathbf{s}^\mathbf{x}_i\right)^T \mathbf{P} \mathbf{s}^{\dot{\mathbf{x}}}_j}{2}}_{<0} < 0\end{aligned} \tag{6}$$

follows. ∎

Note that the final rearrangement of terms in (6) is akin to [10], by which slightly less conservative inequalities have to be checked.

The importance of Theorem 1 is in the establishment of a stability proof for RFS, which is reduced to a check of finitely many linear inequalities. Therefore a numerical solution can be obtained efficiently by means of interior point methods. If strict inequalities in (6) are relaxed to $\leq$, stability in the sense of Lyapunov is proven instead of asymptotic stability.

IV. Synthesis of Switching Controllers

A. Synthesis of Piecewise Constant Controllers

Consider now the RFS with inputs (2), for which a piecewise constant controller

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}_{\mathbf{j}} \tag{7}$$

is sought, such that a constant input is active for each hypersquare $H^{\mathbf{x}}_{\mathbf{j}}$.

Lemma 1: The system dynamics within a hypersquare of an RFS with constant inputs are determined solely by a convex combination of core position gradients in the state space.

Proof: The proof is obtained directly from (2). By letting $\mu^{\mathbf{u}}_{\mathbf{q}} = \prod_{p=1}^{m} \mu^{u_p}_{q_p}(u_p = \text{const.})$,

$$\dot{\mathbf{x}} = \sum_{\mathbf{j},\mathbf{q}} \mathbf{s}^{\dot{\mathbf{x}}}_{\mathbf{w}(\mathbf{j},\mathbf{q})} \cdot \prod_{i=1}^{n} \mu^{x_i}_{j_i}(x_i) \cdot \prod_{p=1}^{m} \mu^{u_p}_{q_p}(u_p) \tag{8}$$

$$= \sum_{\mathbf{j},\mathbf{q}} \left(\mu^{\mathbf{u}}_{\mathbf{q}} \cdot \mathbf{s}^{\dot{\mathbf{x}}}_{\mathbf{w}(\mathbf{j},\mathbf{q})} \right) \prod_{i=1}^{n} \mu^{x_i}_{j_i}(x_i) \tag{9}$$

$$= \sum_{\mathbf{j}} \tilde{\mathbf{s}}^{\dot{\mathbf{x}}}_{\mathbf{j}} \prod_{i=1}^{n} \mu^{x_i}_{j_i}(x_i), \tag{10}$$

from which the statement follows. ■

This result can be pictured as core position gradients of the RFS being constant for constant inputs.

As a consequence, synthesis of constant stabilizing control inputs (7) may again be carried out by solving finitely many scalar equations similar to (4). In order to obtain more precise conditions, the explicit dependency of the system dynamics (2) from the membership functions is removed:

Assumption 1: In the following, triangular and ramp shaped membership functions $\mu^{x_i}_{j_i}, \mu^{u_p}_{q_p}$ of states and inputs are assumed, i.e.

$$\mu^{x_i}_{j_i}(x_i) = \begin{cases} \frac{x_i - s^{x_i}_{j_{i-1}}}{s^{x_i}_{j_i} - s^{x_i}_{j_{i-1}}}, & x_{\text{i,min}} \leq s^{x_i}_{j_{i-1}} \leq x_i < s^{x_i}_{j_i} \leq x_{\text{i,max}} \\ \frac{s^{x_i}_{j_{i+1}} - x_i}{s^{x_i}_{j_{i+1}} - s^{x_i}_{j_i}}, & x_{\text{i,min}} \leq s^{x_i}_{j_i} \leq x_i < s^{x_i}_{j_{i+1}} \leq x_{\text{i,max}} \\ 1, & x_{\text{i,min}} \geq x_i \vee x_i \geq x_{\text{i,max}} \\ 0, & \text{else.} \end{cases} \tag{11}$$

The benefit of this assumption is threefold: Besides ease of implementation, membership functions are zero, if $\mathbf{x}$ and $\mathbf{s}^{\mathbf{x}}_{\mathbf{j}}$ are not connected, i.e., if $\mathbf{x}$ is in a hypersquare not bounded by $\mathbf{s}^{\mathbf{x}}_{\mathbf{j}}$. This locality then allows for the description of the dynamics (2) within a hypersquare $H^{\mathbf{x}}_{\mathbf{l}}$ by means of a polynomial. To see this, (11) is substituted into (2). After some math, the multiplication of local affine functions leads to the local dynamics

$$\dot{\mathbf{x}} = \mathbf{a}^{\mathbf{l}}_0 + \sum_{i=1}^{n} \mathbf{a}^{\mathbf{l}}_{x_i} x_i + \sum_{p=1}^{m} \mathbf{a}^{\mathbf{l}}_{u_p} u_p + \sum_{j=2}^{n} \sum_{i=1}^{j-1} \mathbf{a}^{\mathbf{l}}_{x_{ij}} x_i x_j$$
$$+ \sum_{p=1}^{m} u_p \sum_{j=2}^{N} \mathbf{a}^{\mathbf{l}}_{x_i u_p} x_i + \cdots + \mathbf{a}^{\mathbf{l}}_{x_{1\ldots n} u_{1\ldots m}} x_1 \ldots x_n u_1 \ldots u_m \tag{12}$$

for each hypersquare $H^{\mathbf{x}}_{\mathbf{l}}$. By rearranging terms,

$$\dot{\mathbf{x}} = \mathbf{a}_{\mathbf{l}}(\mathbf{x}) + \mathbf{B}_{\mathbf{l}}(\mathbf{x}) \mathbf{Z}(\mathbf{u}) \tag{13}$$

is obtained, with the vector of monomials in the inputs u_p being denoted as $\mathbf{Z}(\mathbf{u}) = \begin{bmatrix} u_1, & u_2, & u_1 \cdot u_2, & \ldots, & u_1 \cdots u_m \end{bmatrix}$. From (13) it becomes obvious, that except for $m = 1$, the system dynamics are per se non-input affine, which complicates the controller synthesis.

Proposition 1: The dynamics of every RFS fulfilling Assumption 1 can be transformed via dynamic extension and a change of variables into an equivalent system, whose dynamics are input-affine.

Proof: By augmenting the system input with an integrator, new inputs $\mathbf{v} \in \mathbb{R}^m$, $\dot{\mathbf{u}} = \mathbf{v}$ are obtained. By introducing new state variables $\mathbf{w} = \mathbf{Z}(\mathbf{u}) = \begin{bmatrix} u_1, & u_2, & u_1 \cdot u_2, & \ldots, & u_1 \cdots u_m \end{bmatrix}^T \in \mathbb{R}^{2^m - 1}$, the differential equations of these new states read

$$\dot{\mathbf{w}} = \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_1 u_2 + u_1 \dot{u}_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_1 u_2 + u_1 v_2 \\ \vdots \end{bmatrix}. \tag{14}$$

Thus, the state space equations of the prolonged system read

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{\mathbf{l}}(\mathbf{x}) + \mathbf{B}_{\mathbf{l}}(\mathbf{x}) \mathbf{w} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \frac{\mathbf{Z}(\mathbf{u})}{\partial \mathbf{u}} \end{bmatrix} \mathbf{v}, \tag{15}$$

which is clearly input-affine. ■

For ease of notation, we assume in the following the RFS to be input-affine, which is due to Proposition 1 without loss of generality.

Theorem 2: If for an RFS with $\mathbf{x}^* = \mathbf{0}$, there exists a $\mathbf{P} \succ 0$, and for every hypersquare $H^{\mathbf{x}}_{\mathbf{k}} \in \mathcal{X}$, there exist a $H^{\mathbf{u}}_{\mathbf{q}} \in \mathcal{U}$, such that

$$(\mathbf{s}^{\mathbf{x}}_{\mathbf{i}})^T \mathbf{P} \left(\mathbf{a}_{(\mathbf{k},\mathbf{q})}(\mathbf{s}^{\mathbf{x}}_{\mathbf{i}}) + \mathbf{B}_{(\mathbf{k},\mathbf{q})}(\mathbf{s}^{\mathbf{x}}_{\mathbf{i}}) \cdot \mathbf{u}_{\mathbf{k}} \right) < 0, \tag{16a}$$

$$\frac{1}{2} \Big((\mathbf{s}^{\mathbf{x}}_{\mathbf{i}})^T \mathbf{P} \left(\mathbf{a}_{(\mathbf{k},\mathbf{q})}(\mathbf{s}^{\mathbf{x}}_{\mathbf{j}}) + \mathbf{B}_{(\mathbf{k},\mathbf{q})}(\mathbf{s}^{\mathbf{x}}_{\mathbf{j}}) \cdot \mathbf{u}_{\mathbf{k}} \right) + \ldots$$
$$(\mathbf{s}^{\mathbf{x}}_{\mathbf{j}})^T \mathbf{P} \left(\mathbf{a}_{(\mathbf{k},\mathbf{q})}(\mathbf{s}^{\mathbf{x}}_{\mathbf{i}}) + \mathbf{B}_{(\mathbf{k},\mathbf{q})}(\mathbf{s}^{\mathbf{x}}_{\mathbf{i}}) \cdot \mathbf{u}_{\mathbf{k}} \right) \Big) < 0 \tag{16b}$$

holds for all $\{\mathbf{s}^{\mathbf{x}}_{\mathbf{i}}, \mathbf{s}^{\mathbf{x}}_{\mathbf{j}}\} \in H^{\mathbf{x}}_{\mathbf{k}} \backslash \{\mathbf{0}\}$, $\mathbf{i} \neq \mathbf{j}$ then $\mathbf{x}^*$ is asymptotically stable on $\mathcal{X}$.

Thus, by solving (16), a Lyapunov matrix $\mathbf{P}$ as well as constant inputs $\mathbf{u}_{\mathbf{k}}$ for every $H^{\mathbf{x}}_{\mathbf{k}}$ are obtained. In contrast to the stability analysis criterion in Theorem 1, the $R_n \cdot \begin{pmatrix} 2^n \\ 2 \end{pmatrix}$ inequalities (16) are no longer linear in the decision variables. Due to product terms $P_{\mu\nu} \cdot u_{k_\lambda}$ of the elements of $\mathbf{P}$ and $\mathbf{u_k}$, (16) can be rearranged as bilinear matrix inequality (BMI). In addition, this feasibility problem is only continuous in $P_{\mu\nu}, u_{k_\lambda}$, if assignments $H^{\mathbf{x}}_{\mathbf{j}} \to H^{\mathbf{u}}_{\mathbf{q}}$ are given. In the other case, a mixed integer bilinear program has to be solved, taking integer variables $\mathbf{q}$ as unknowns into account. Still, a solution may be obtained via branch-and-bound techniques [11]. Sometimes, a feasible solution may also be obtained by solving (16) independently for every hypersquare in order to obtain assignments $H^{\mathbf{x}}_{\mathbf{j}} \to H^{\mathbf{u}}_{\mathbf{q}}$, and in a second step solving (16) for all $H^{\mathbf{x}}_{\mathbf{j}}$ simultaneously with $H^{\mathbf{x}}_{\mathbf{j}} \to H^{\mathbf{u}}_{\mathbf{q}}$ fixed.

Even for known assignments $H^{\mathbf{x}}_{\mathbf{j}} \to H^{\mathbf{u}}_{\mathbf{q}}$, a bilinear matrix inequality has to be solved, which is non-convex in general. Two solution strategies are discussed briefly:

1) If a candidate Lyapunov function is known, only a linear matrix inequality has to be solved. Extending this concept to alternately fixing first the Lyapunov function and then the controller leads to the *VK-iteration algorithm* [12]. Although the guess of a Lyapunov function does not seem to be mathematically appealing, it may in practical situations already lead to good results with least implementation effort. In addition, bounds on the control input may be regarded directly as additional inequalities $\mathbf{u}_{\mathbf{k},\min} \leq \mathbf{u_k} \leq \mathbf{u}_{\mathbf{k},\max}$.
2) By means of variable substitutions

$$\mathbf{R} = \{R_{ij}\} = \text{vec}\left(P_{\mu\nu}\right) \cdot \text{vec}\left(u_{k_\lambda}\right)^T, \tag{17}$$

(16) is again rendered linear in the variables $P_{\mu\nu}$ and R_{ij}. Then, by utilizing a rank minimization approach [13],

$$\min_{P_{\mu\nu}, R_{ij}} \text{tr}\left(\mathbf{R}^{(k)}\right) - 2\left(\mathbf{r}^{(k-1)}\right)^T \mathbf{r}^{(k)}, \tag{18a}$$

$$\text{s.t. (16)}, \tag{18b}$$

$$\begin{bmatrix} \mathbf{R}^{(k)} & \mathbf{r}^{(k)} \\ \left(\mathbf{r}^{(k)}\right)^T & 1 \end{bmatrix} \prec 0 \tag{18c}$$

a solution in the transformed decision variables is obtained. If $\lim_{k\to\infty} \text{tr}\left(\mathbf{R}\right) - \mathbf{r}^T\mathbf{r} \to 0$, then the solution to (18) also solves (16). Thus, the unknown control inputs can be recovered from (17). Although by means of this approach, $\mathbf{P}$ and $\mathbf{u_k}$ may be obtained simultaneously, the drawbacks are obvious: Besides higher complexity of the implementation, (18) has to be solved iteratively, whereas beforehand there is no guarantee of convergence such that $\text{tr}\left(\mathbf{R}\right) - \mathbf{r}^T\mathbf{r} = 0$. In addition, bounds on the control input can no longer be regarded directly.

B. Optimization of Robustness

The aforementioned procedure aims at determining constant inputs by means of a feasibility problem (16). Therefore, the question arises what parameter to optimize in order to utilize the remaining degree of freedom. Although the optimization of some performance measure (e.g., the decay rate) is straightforward, it is more reasonable to account for parametric uncertainties in the system model, which always occur due to the approximate nature of RFS.

Since the system model stems either from rough expert knowledge or measurement data, it is unlikely that further information is available on how the system is dependent on the uncertainties. Therefore, they have to be regarded as general local distortion functions $\boldsymbol{\xi}_{\mathbf{j}}(\mathbf{x}) \in \mathbb{R}^n$, such that the local dynamics of every hypersquare read

$$\dot{\mathbf{x}} = \mathbf{f_l}(\mathbf{x}, \mathbf{u}) + \boldsymbol{\xi}_{\mathbf{l}}(\mathbf{x}). \tag{19}$$

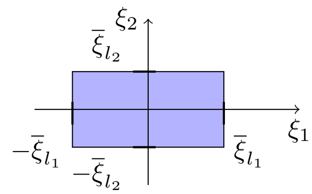

Fig. 2. Symmetric region of parametric uncertainties.

For simplicity we assume the uncertainties to be independent, symmetric and bounded by some constant $|\boldsymbol{\xi}_{\mathbf{j}}(\mathbf{x})| \leq \overline{\boldsymbol{\xi}}_{\mathbf{j}}$. Then the region of allowed parametric uncertainties is a symmetric hypersquare $H_{\xi,\mathbf{j}}$ around the origin, as shown in Fig. 2. Instead of assuming $\overline{\boldsymbol{\xi}}_{\mathbf{j}}$ as given upper bound, the volume of $H_{\xi,\mathbf{j}}$ is taken as measure of robustness, which is sought to be optimized. This procedure was carried out in [14] for linear systems with parametric uncertainties, where the problem was solved by means of a bisection algorithm. Here, the polytope volume of allowed uncertain parameters, $\sum_{\mathbf{j}} \prod_{i=1}^n \overline{\xi}_{i,\mathbf{j}}$, is utilized directly as optimization criterion. By means of the equality

$$\begin{aligned} \arg \sum_{\mathbf{j}} \max_{\overline{\boldsymbol{\xi}}_{\mathbf{j}}} \prod_{i=1}^{n} \overline{\xi}_{i,\mathbf{j}} &= \arg \sum_{\mathbf{j}} \max_{\overline{\boldsymbol{\xi}}_{\mathbf{j}}} \det \text{diag}\left(\overline{\xi}_{1,\mathbf{j}}, \ldots, \overline{\xi}_{n,\mathbf{j}}\right) \\ &= \arg \sum_{\mathbf{j}} \min_{\overline{\boldsymbol{\xi}}_{\mathbf{j}}} -\log\det \mathbf{M}_{\xi,\mathbf{j}} \end{aligned} \tag{20}$$

with $\mathbf{M}_{\xi,\mathbf{j}} = \text{diag}\left(\overline{\xi}_{1,\mathbf{j}}, \ldots, \overline{\xi}_{n,\mathbf{j}}\right)$, the optimization criterion is obviously a convex function, since $-\log\det(\cdot)$ is convex.

Thus, the complete optimization problem for synthesis of a robust switching controller reads

$$\min_{\overline{\boldsymbol{\xi}}_{\mathbf{j}}} \sum_{\mathbf{j}} -\log\det \mathbf{M}_{\xi,\mathbf{j}}, \tag{21a}$$

$$\text{s.t. } \mathbf{P} \succ 0, \tag{21b}$$

$$\left(\mathbf{s}^{\mathbf{x}}_{\mathbf{i}}\right)^T \mathbf{P} \tilde{\mathbf{s}}^{\dot{\mathbf{x}}}_{\mathbf{i}} < 0, \tag{21c}$$

$$\frac{\left(\mathbf{s}^{\mathbf{x}}_{\mathbf{i}}\right)^T \mathbf{P} \tilde{\mathbf{s}}^{\dot{\mathbf{x}}}_{\mathbf{j}} + \left(\mathbf{s}^{\mathbf{x}}_{\mathbf{j}}\right)^T \mathbf{P} \tilde{\mathbf{s}}^{\dot{\mathbf{x}}}_{\mathbf{i}}}{2} < 0, \tag{21d}$$

Therein, the core position gradients of the controlled RFS read

$$\tilde{\mathbf{s}}^{\dot{\mathbf{x}}}_{\mathbf{i}} = \mathbf{a}_{(\mathbf{j},\mathbf{q})}(\mathbf{s}^{\mathbf{x}}_{\mathbf{i}}) + \mathbf{B}_{(\mathbf{j},\mathbf{q})}(\mathbf{s}^{\mathbf{x}}_{\mathbf{i}}) \cdot \mathbf{u_j} + \left(2 \cdot \boldsymbol{\Delta}_\lambda - \mathbf{I}\right) \overline{\boldsymbol{\xi}}_{\mathbf{j}}, \tag{22}$$

and $\mathbf{\Delta}_\lambda = \text{diag}\left(b_1, \ldots, b_n\right), b \in \{0,1\}, \lambda = 1, \ldots, 2^n$. Thus, $(2 \cdot \mathbf{\Delta}_\lambda - \mathbf{I})$ permutes over all 2^n vertices of $H_{\xi,\mathbf{j}}$. As a result of (21), stabilizing control inputs $\mathbf{u_j}$ are obtained for each hypersquare $H^{\mathbf{x}}_{\mathbf{j}}$, such that upper bounds $\bar{\xi}_{\mathbf{j}}$ on the uncertainties are maximized.

C. Optimization of State Space Partition

The assumption of a piecewise constant controller $\mathbf{u}(\mathbf{x}) = \mathbf{u_j}$ has the benefit that only finitely many inequalities at core positions have to be checked. On the other hand, the assumption of piecewise constant controllers is clearly conservative, i.e., a common $\mathbf{u_j}$ for all core positions in $H^{\mathbf{x}}_{\mathbf{j}}$ might not exist. Instead, the assumption is only plausible, if the dynamics within each hypersquare are quasi-homogenous, that is, $\mathbf{s}^{\dot{\mathbf{x}}}_i \approx \mathbf{s}^{\dot{\mathbf{x}}}_j, \forall\{\mathbf{s}^{\mathbf{x}}_{\mathbf{i}}, \mathbf{s}^{\mathbf{x}}_{\mathbf{j}}\} \in H^{\mathbf{x}}_{\mathbf{k}}$. Therefore, the aforementioned controller synthesis method is extended by introduction of a rectangular state space partition $\mathcal{P}$ that is allowed to be finer than the original hypersquare partition of the RFS, and in which the dynamics of each region $P_i \in \mathcal{P}$ is quasi homogenous.

A solution to this problem can be obtained using a *split and merge* technique [15], which aims at recursively splitting each hypersquare in smaller polytopes depending on a homogeneity measure h_s, and merging neighboring partitions depending on a homogeneity measure h_m. Among the various ways of utilizing split and merge [16], the cumulative variance of gradients at corners of polytopes

$$|\mathbf{\Sigma_k}| = \frac{1}{N_\mathbf{q}} \cdot \sum_{\mathbf{q}} \det\left(\frac{1}{N_\mathbf{j}} \sum_{\mathbf{j}} \left(\mathbf{s}^{\dot{\mathbf{x}}}_{(\mathbf{j},\mathbf{q})} - \mathbf{m_q}\right) \cdot \ldots \right.$$
$$\left. \cdot \left(\mathbf{s}^{\dot{\mathbf{x}}}_{(\mathbf{j},\mathbf{q})} - \mathbf{m_q}\right)^T\right) \quad (23)$$

with $\mathbf{m_q} = \frac{1}{N_\mathbf{j}} \sum_\mathbf{j} \mathbf{s}^{\dot{\mathbf{x}}}_{(\mathbf{j},\mathbf{q})}$ is used for deciding, whether a partition should be split. Hence, a partition is split, if $|\mathbf{\Sigma_k}| > \theta_s$, where θ_s is a threshold to be selected.

The decision whether to merge two neighboring partitions P_1, P_2 is then carried out by means of a likelihood ratio test: Core position gradients of each partition are assumed to be drawn from a normal distribution

$$p(\mathbf{s}^{\dot{\mathbf{x}}}_i) = \frac{1}{\sqrt{2\pi|\mathbf{\Sigma}|}} \exp\left(-\frac{1}{2}\left(\mathbf{s}^{\dot{\mathbf{x}}}_i - \mathbf{m}\right)^T \mathbf{\Sigma}^{-1}\left(\mathbf{s}^{\dot{\mathbf{x}}}_i - \mathbf{m}\right)\right). \quad (24)$$

Then, by means of the likelihood ratio

$$L = \left(\frac{|\mathbf{\Sigma}_0|^2}{|\mathbf{\Sigma}_1| \cdot |\mathbf{\Sigma}_2|}\right)^{2^n} \quad (25)$$

it is tested, whether the gradients are drawn from the same distribution $\mathcal{N}(\mathbf{m}_0, \mathbf{\Sigma}_0)$, or from two different distributions $\mathcal{N}(\mathbf{m}_1, \mathbf{\Sigma}_1)$, $\mathcal{N}(\mathbf{m}_2, \mathbf{\Sigma}_2)$.

Again, by choosing a threshold θ_m, $L < \theta_m$ leads to a positive merge decision. For simplicity, the merging is only carried out for siblings of the same branch (see [16] for further details), such that the final partition $\mathcal{P}$ consists only of hypersquares instead of more complex polytopes.

Algorithm 1: Split and Merge

Set $\mathcal{P} = \cup_\mathbf{j} H^{\mathbf{x}}_{\mathbf{j}}$;
repeat
  foreach $P_i \in \mathcal{P}$ **do**
    Split P_i, if $|\mathbf{\Sigma}_i| > \theta_s$;
    Merge new subregions of P_i, if $L < \theta_m$.
until *no changes*;

The complete split and merge procedure is summarized in Algorithm 1.

As a result, a new partition $\mathcal{P} = \cup_i P_i = \cup_\mathbf{j} H^{\mathbf{x}}_{\mathbf{j}}$ is obtained for which the aforementioned controller synthesis method can similarly be applied to. The benefit of applying the split and merge technique is the option for an arbitrarily fine partitioning. Thereby, hypersquares with homogenous dynamics, e.g., almost static dynamic function are obtained, which increases the chance to find a common stabilizing constant control input. The drawback is in the choice of parameters θ_s, θ_m, by which the resolution of $\mathcal{P}$ is influenced. In addition, the split and merge technique with hypersquares may potentially lead to an over fitting. Therefore the maximum number of split and merge iterations should be limited in order to limit computational complexity.

D. Auxiliary Polynomial Controllers Preventing Deadlock

Due to the switching control, the dynamic function of the controlled RFS is in general discontinuous. Therefore, chattering and deadlock phenomena may occur [17]. The latter effect is particularly disadvantageous, since a solution obtained from the aforementioned synthesis method may in some cases only guarantee stability in the sense of Lyapunov. Fig. 3 depicts an exemplary situation, where deadlock may occur on the facet between two hypersquares.

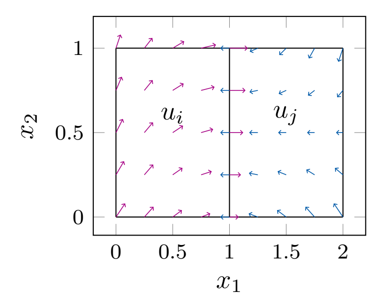

Fig. 3. Deadlock phenomenon due to switching control.

In order to avoid this undesired effect, auxiliary local controllers are determined for those partitions $P_j \in \mathcal{P}$, in which deadlock potentially may occur. The synthesis of these auxiliary controllers is a two step procedure:

1) Partitions are detected, in which deadlock may occur due to the switching control. By considering any two neighboring partitions $\{P_i, P_j\}$, deadlock occurs on the facet $F_{ij} = P_i \cap P_j$ with outer normal vector $\mathbf{n}_{ij}$, if $\exists\ \tilde{\mathbf{s}}^{\dot{\mathbf{x}}}_k \in F_{ij}$, such that $\sigma\left(\mathbf{n}^T_{ij}\tilde{\mathbf{s}}^{\dot{\mathbf{x}}}_k(\mathbf{u}_i)\right) = -\sigma\left(\mathbf{n}^T_{ij}\tilde{\mathbf{s}}^{\dot{\mathbf{x}}}_k(\mathbf{u}_j)\right)$.

2) With help of the previously determined Lyapunov function, polynomial controllers $\mathbf{u}_h = \mathbf{k}(\mathbf{x})$, $\mathbf{u} = \mathbf{u}_j + \mathbf{u}_h$, are calculated for these partitions based on sums of squares optimization. In addition to the constraint $-\dot{V}(\mathbf{x}) > 0$, the constraint $\mathbf{n}_{ij}^T \cdot \dot{\mathbf{x}} > 0, \forall \mathbf{x} \in F_{ij}$, is added to ensure that gradients point away from conflicting facets. Using sums of squares, these conditions hold, if

$$-\left(\frac{\partial V(\mathbf{x})}{\partial \mathbf{x}}\right)^T \cdot \dot{\tilde{\mathbf{x}}} \in \Sigma[\mathbf{x}],\ \forall \mathbf{x} \in P_j, \tag{26a}$$

$$\mathbf{n}_{ij}^T \cdot \dot{\tilde{\mathbf{x}}} \in \Sigma[\mathbf{x}], \forall \mathbf{x} \in F_{ij}. \tag{26b}$$

Since these constraints have to hold only locally, the generalized S-procedure [18] is utilized. Every rectangular partition P_j may be outer approximated by an ellipse $\varepsilon_j(\mathbf{x})$, which is itself in a polynomial form. The same holds for every facet F_{ij}, which is outer approximated by $\varphi_{ij}(\mathbf{x})$. Then a solution to (26) is found, if there exists a polynomial controller $\mathbf{k}_j(\mathbf{x})$ and auxiliary polynomials $t_1(\mathbf{x}), t_2(\mathbf{x})$, such that

$$-\left(\frac{\partial V(\mathbf{x})}{\partial \mathbf{x}}\right)^T \cdot \dot{\tilde{\mathbf{x}}} - t_1(\mathbf{x}) \cdot \varepsilon_j(\mathbf{x}) \in \Sigma[\mathbf{x}], \tag{27a}$$

$$\mathbf{n}_{ij}^T \cdot \dot{\tilde{\mathbf{x}}} - t_2(\mathbf{x}) \cdot \varphi_{ij} \in \Sigma[\mathbf{x}], \tag{27b}$$

$$t_1(\mathbf{x}), t_2(\mathbf{x}) \in \Sigma[\mathbf{x}]. \tag{27c}$$

Considering again model uncertainties, the equations

$$\min_{\overline{\boldsymbol{\xi}}_j} \sum_{\mathbf{j}} - \log\det \mathbf{M}_{\xi,j}, \quad \forall\ \lambda, H_q \in \mathcal{U}, \tag{28a}$$

$$\text{s.t.}\ -\frac{\partial V(\mathbf{x})}{\partial \mathbf{x}}^T \dot{\tilde{\mathbf{x}}} - t_1(\mathbf{x})\varepsilon_j(\mathbf{x}) \in \Sigma[\mathbf{x}] \tag{28b}$$

$$\mathbf{n}_{ij}^T \cdot \dot{\tilde{\mathbf{x}}} - t_2(\mathbf{x})\varphi_{ij} \in \Sigma[\mathbf{x}], \tag{28c}$$

$$t_1(\mathbf{x}), t_2(\mathbf{x}) \in \Sigma[\mathbf{x}], \tag{28d}$$

$$\mathbf{M}_{\xi,j} = \operatorname{diag}\left(\overline{\xi}_{1,j}, \dots, \overline{\xi}_{n,j}\right), \tag{28e}$$

$$\dot{\tilde{\mathbf{x}}} = \mathbf{a}_{(j,q)}(\mathbf{x}) + \mathbf{B}_{(j,q)}(\mathbf{x})\left(\mathbf{u}_j + \mathbf{k}_j(\mathbf{x})\right) + (2\boldsymbol{\Delta}_\lambda - \mathbf{I})\,\overline{\boldsymbol{\xi}}_j \tag{28f}$$

are finally obtained.

Note that the auxiliary polynomial controllers are only computed for conflicting partitions, if exist. In all other cases, only locally constant controllers are applied. Although the synthesis for polynomial controllers is also possible for every hypersquare with simultaneous computation of polynomial controllers and a common Lyapunov function (see [4]), this usually leads to a much higher computational complexity, even for small systems. Therefore, we restrict the sum of squares framework to the computation of the auxiliary controllers only.

V. Numerical Example

To illustrate the controller synthesis method, the inverted pendulum on a cart as discussed in [19] is considered, with the motion of the cart not being modeled. All computations were carried out in MATLAB, using YALMIP [20] and SEDUMI [21].

Neglecting friction, the nonlinear system dynamics read

$$\dot{x}_1 = x_2, \tag{29a}$$

$$\dot{x}_2 = -\sin(x_1) - \cos(x_1)u, \tag{29b}$$

which will serve as a ground truth model, from which a RFS is obtained. The control input u is the acceleration of the cart.

The angle x_1 of the pendulum as well as the angular velocity x_2 are considered in intervals $x_1 \in [2\pi/3,\ 4\pi/3]$ $x_2 \in [-\pi/3,\ \pi/3]$, whereas the aim will be to stabilize the upper equilibrium $\mathbf{x}^* = [\pi,\ 0]^T$. By defining core positions at $\{s_{j_1}^{x_1}\} = \{\pi \pm \pi/3,\ \pi \pm \pi/6,\ \pi \pm \pi/12,\ \pi\}$, $\{s_{j_2}^{x_2}\} = \{\pm\pi/3,\ \pm\pi/6,\ \pm\pi/12,\ 0\}$, $\{s_q^u\} = \{\pm 10, 0\}$, a RFS is derived from (29). The phase plot of the uncontrolled system is depicted in Fig. 4, in which the unstable equilibrium can clearly be seen. Due to the rectangular grid, the RFS is lin-

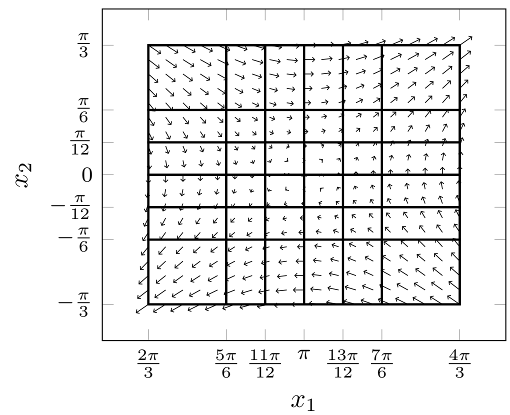

Fig. 4. Phaseplot of uncontrolled RFS.

guistically interpretable, since linguistic values $L_{j_1}^{x_1} = L_{j_2}^{x_2} = \{$neg. big/medium/small, zero, pos. small/medium/big$\}$ can be associated with the core positions $s_{j_1}^{x_1}, s_{j_2}^{x_2}$. For the input space, $L_q^u = \{$negative, zero, positive$\}$ correspond to core positions s_q^u.

In order to refine the initial partition of the RFS, Algorithm 1 is applied, choosing $\theta_s = 1$ as threshold parameter for the covariance matrix $\boldsymbol{\Sigma}_i$ of each partition P_i by which it is decided whether to split P_i. In addition, $\theta_m = 1$ is chosen as threshold for the likelihood ratio test, by which it is decided whether or not to merge two adjacent partitions P_i, P_j. The result of the split and merge procedure is shown in Fig. 5. As can be seen, by choice of the threshold parameters θ_s, θ_m, larger regions are split such that the vector field $\mathbf{f}_i(\mathbf{x}, \mathbf{u})$ is quasi homogenous within each partition P_i. This improvement comes at the cost of the loss of linguistic interpretability of $\mathcal{P}$.

Applying the synthesis concept of robust feedback described in Sec. IV-B, a positive definite matrix

$$\mathbf{P} = 10^{-6} \cdot \begin{bmatrix} 5.38 & 0 \\ 0 & 42.1 \end{bmatrix} \tag{30}$$

as well as constant inputs are obtained for each region P_i, which are listed in Table I. It also lists the resulting upper

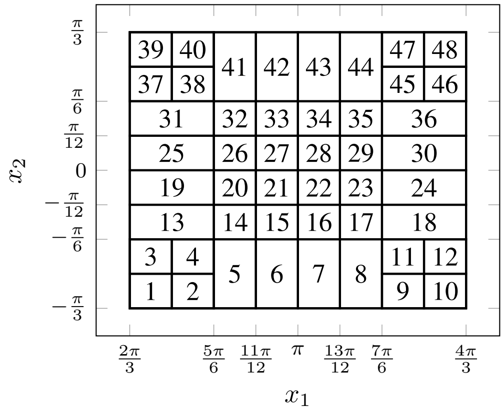

Fig. 5. Refined partition $\mathcal{P}$ of RFS with indices.

bounds $\overline{\boldsymbol{\xi}}_i$ of the allowed model uncertainties as well as the local maximum model error $\overline{\mathbf{e}}_i = \max |\dot{\mathbf{x}}_{\text{RFS}} - \dot{\mathbf{x}}_{\text{GT}}|, \mathbf{x} \in P_i$, between dynamics of the recurrent fuzzy system and the ground truth model. (Since in this example, $\overline{e}_{i,1} \approx 0$, only $\overline{e}_{i,2}$ is listed.) Note that throughout this example, model uncertainties were constraint to $\overline{\xi}_i \leq 10$ for numerical reasons. As becomes obvious from Table I, the calculated allowed bound on model uncertainties $\overline{\boldsymbol{\xi}}_i$ is always significantly larger than the actual model uncertainties $\overline{\mathbf{e}}_i$. In any case, the knowledge of $\overline{\boldsymbol{\xi}}_i$ allows to a certain extent for an evaluation of the criticality of approximation errors.

TABLE I
CONTROLLERS u_i, BOUNDS ON ADMISSIBLE PARAMETRIC UNCERTAINTIES $\overline{\boldsymbol{\xi}}_i$ FOR PARTITIONS P_i AND LOCAL DIFFERENCE $\overline{e}_{i,2}$

i	u_i	$\overline{\xi}_{i,1}$	$\overline{\xi}_{i,2}$	$\overline{e}_{i,2}$	i	u_i	$\overline{\xi}_{i,1}$	$\overline{\xi}_{i,2}$	$\overline{e}_{i,2}$
1	10	10	2.3	0.22	25	−9.84	0	5.36	0.26
2	10	10	4.77	0.22	26	−9.85	0	7.63	0.08
3	10	7.83	2	0.22	27	−9.88	0	8.25	0.08
4	10	10	4.13	0.22	28	−9.88	0	7.94	0.08
5	10	10	6.82	0.08	29	−9.87	0	6.76	0.08
6	10	10	8.73	0.08	30	−9.9	0	3.59	0.26
7	10	10	9.31	0.09	31	−10	5.87	3	0.27
8	10	10	7.95	0.08	32	−10	10	6.67	0.08
9	10	10	6.34	0.27	33	−10	10	8.67	0.09
10	10	10	4.3	0.27	34	−10	10	8.09	0.08
11	10	10	5.7	0.27	35	−10	10	5.54	0.08
12	10	10	3.45	0.27	36	−10	3.92	2	0.22
13	10	3.92	2	0.22	37	−10	10	3.45	0.27
14	10	10	5.54	0.08	38	−10	10	5.7	0.27
15	10	10	8.09	0.08	39	−10	10	4.3	0.27
16	10	10	8.67	0.09	40	−10	10	6.34	0.27
17	10	10	6.67	0.08	41	−10	10	7.95	0.08
18	10	5.87	3	0.27	42	−10	10	9.31	0.09
19	9.85	0	3.71	0.22	43	−10	10	8.73	0.08
20	9.82	0	7.14	0.08	44	−10	10	6.82	0.08
21	9.81	0	7.9	0.08	45	−10	10	4.13	0.22
22	9.83	0	8.34	0.09	46	−10	7.83	2	0.22
23	9.83	0	7.87	0.08	47	−10	10	4.77	0.22
24	9.78	0	5.08	0.27	48	−10	10	2.3	0.22

Table II shows the influence of the partition on the upper bound of allowed model uncertainties $\overline{\boldsymbol{\xi}}$. For constant $\theta_m = 1$, the threshold θ_s for the split decisions of regions is changed, resulting in different numbers of regions within the partitions. As can be seen, an increase in the number of regions and therefore a decreased size of regions may lead to a higher bound of allowed model uncertainties.

As can be seen from the phase plot Fig. 6, the closed-loop system is stable in the sense of Lyapunov, but deadlock occurs for $x_2 = 0$. On the other hand, this means the pendulum will remain at an angle of $x_1 \neq \pi$. This is only possible, if the wagon, which is not modeled here, remains at a constant speed. Thus, the entire system can be considered as unstable in this case and the necessity for auxiliary controllers guaranteeing asymptotic stability is clearly motivated.

TABLE II
DEPENDENCY OF ROBUSTNESS ON PARTITION SIZE

θ_s	regions	$\min_j \overline{\xi}_{j,1}$	$\min_j \overline{\xi}_{j,2}$
1	48	$1.03 \cdot 10^{-4}$	2
0.5	72	10	4.13
0.1	132	10	10

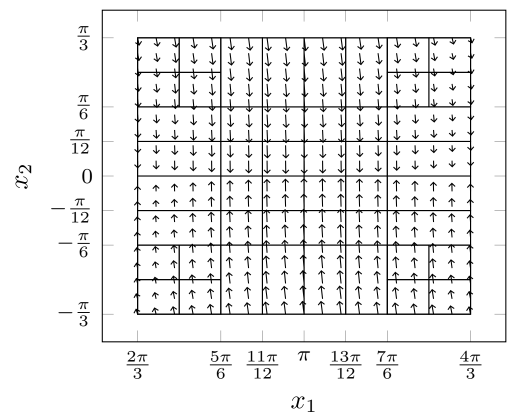

Fig. 6. Phaseplot of closed-loop RFS with piecewise constant controller.

By calculating polynomial controllers of fourth order for conflicting regions according to Sec. IV-D, a final closed-loop system is obtained as depicted in Fig. 7. For an initial value of $\mathbf{x} = [4, -1]^T$, the development of the states and the system input is shown in Fig. 8. In addition, the control law developed for the RFS is applied to the ground truth model (29) for comparison. Fig. 8 also shows that due to the inherent approximation error of the RFS, the difference $\mathbf{e} = \mathbf{x}_{\text{RFS}} - \mathbf{x}_{\text{GT}}$ between the states of the RFS and the ground truth model is nonzero in general, but converges to zero asymptotically.

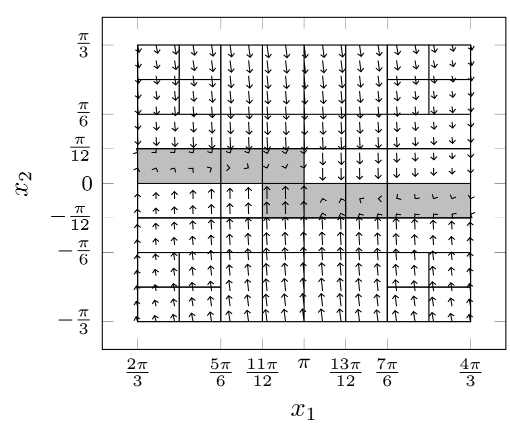

Fig. 7. Phase plot of closed-loop RFS. In shaded regions, additional polynomial controllers were computed preventing deadlock.

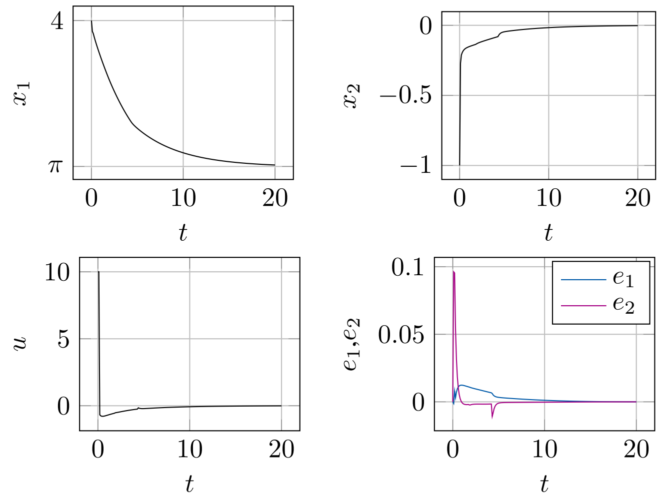

Fig. 8. Development of states x_1, x_2, input u in controlled RFS and differences $e_i = x_{i,\text{RFS}} - x_{i,\text{GT}}$ between RFS and ground truth model for initial value of $\mathbf{x}_0 = [4,\ -1]^T$.

VI. Conclusions

In this paper, a controller synthesis concept for recurrent fuzzy systems was presented, which makes use of the underlying state space partition of the system class. The key observation is that dynamical properties within each region have to be checked at corners only in case of constant inputs. This motivates the use of a constant input for each region and thus a piecewise constant controller for the entire system. To reduce conservatism, the state space partition is refined by means of split and merge techniques in order to obtain locally quasi homogenous vector fields. Then, local inputs are computed by means of bilinear inequalities such that stability of the controlled system can be guaranteed for model uncertainties up to a certain boundary, which is sought to be maximized.

Since recurrent fuzzy systems are approximate models, the strength of this concept is that model uncertainties can be taken into consideration. In addition, the switching control lends itself quite naturally to the hybrid nature of the system. Since dynamic properties have to be checked at corners of the regions, the computational complexity remains quite moderate. This efficiency would hardly be possible with more complex controllers, e.g., simultaneous computation of local polynomial controllers and a common Lyapunov function.

On the other hand, the drawback of the switching is the possibility of deadlock and chattering effects. In order to account for the first, auxiliary polynomial controllers can be incorporated in order to prevent the system from this undesired effect and to guarantee asymptotic stability. Furthermore, undesired chattering effects can easily be prevented, e.g., by hysteresis switching.

The focus of this paper was on robust control of recurrent fuzzy systems by locally maximizing a region of allowed model uncertainties. In this context, chattering effects may occur due to discontinuities in the dynamic function of the controlled system, which is here a rather undesired side effect. On the other hand, explicit sliding-mode controllers for recurrent fuzzy systems are another approach for robust control, which will be subject to further research.

References

[1] R. Kempf and J. Adamy, "Equilibria of recurrent fuzzy systems," *Fuzzy Sets and Systems*, vol. 140, no. 2, pp. 231–257, 2003.

[2] T. Taniguchi and M. Sugeno, "Stabilization of nonlinear systems with piecewise bilinear models derived from fuzzy if-then rules with singletons," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2010, pp. 1–6.

[3] M. Sugeno, "On stability of fuzzy systems expressed by fuzzy rules with singleton consequents," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 2, pp. 201–224, 1999.

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

[5] T. Taniguchi and M. Sugeno, "Robust stabilization of nonlinear systems modeled with piecewise bilinear systems based on feedback linearization," in *Advances on Computational Intelligence*. Springer, 2012, pp. 111–120.

[6] C. Belta and L. Habets, "Controlling a class of nonlinear systems on rectangles," *IEEE Transactions on Automatic Control*, vol. 51, no. 11, pp. 1749–1759, November 2006.

[7] J. Adamy and A. Flemming, "Equilibria of continuous-time recurrent fuzzy systems," *Fuzzy Sets and Systems*, vol. 157, no. 22, pp. 2913–2933, 2006.

[8] J. Adamy and R. Kempf, "Regularity and chaos in recurrent fuzzy systems," *Fuzzy Sets and Systems*, vol. 140, no. 2, pp. 259–284, 2003.

[9] A. Papachristodoulou and S. Prajna, "A tutorial on sum of squares techniques for systems analysis," in *Proceedings of the American Control Conference*, vol. 4, San Diego, California, USA, 2005, pp. 2686–2700.

[10] H. Wang, K. Tanaka, and M. Griffin, "An approach to fuzzy control of nonlinear systems: stability and design issues," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 1, pp. 14–23, 1996.

[11] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas, "Global optimization of mixed-integer nonlinear problems," *AIChE Journal*, vol. 46, no. 9, pp. 1769–1797, 2000.

[12] L. El Ghaoui and V. Balakrishnan, "Synthesis of fixed-structure controllers via numerical optimization," in *Proceedings of the 33rd Conference on Decision and Control*, vol. 3, 1994, pp. 2678–2683.

[13] S. Ibaraki and M. Tomizuka, "Rank minimization approach for solving BMI problems with random search," in *Proceedings of the American Control Conference*, vol. 3, 2001, pp. 1870 –1875.

[14] H. Horisberger and P. Belanger, "Regulators for linear, time invariant plants with uncertain parameters," *IEEE Transactions on Automatic Control*, vol. 21, no. 5, pp. 705–708, 1976.

[15] R. Jain, R. Kasturi, and B. G. Schunck, *Machine vision*. McGraw-Hill New York, 1995, vol. 5.

[16] S.-Y. Chen, W.-C. Lin, and C.-T. Chen, "Split-and-merge image segmentation based on localized feature analysis and statistical tests," *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 5, pp. 457 – 475, 1991.

[17] D. Liberzon, *Switching in Systems and Control*, T. Basar, Ed. Birkhäuser, 2003.

[18] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology Pasadena, California, 2000.

[19] G. Reissig, "Computing abstractions of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2583–2598, 2011.

[20] J. Löfberg, "YALMIP : A Toolbox for Modeling and Optimization in MATLAB," in *Proceedings of the CACSD Conference*, 2004, pp. 284–289.

[21] J. F. Sturm, "Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1–4, pp. 625–653, 1999.