Feedforward Tracking Control of Flat Recurrent Fuzzy Systems

Stefan Gering and Jürgen Adamy
Laboratory of Control Methods and Robotics at Technische Universität Darmstadt,
Landgraf-Georg-Str. 4, 64283 Darmstadt, Germany
E-mail: {sgering,adamy}@rmr.tu-darmstadt.de

Abstract. Flatness based feedforward control has proven to be a feasible solution for the problem of tracking control, which may be applied to a broad class of nonlinear systems. If a flat output of the system is known, the control is often based on a feedforward controller generating a nominal input in combination with a linear controller stabilizing the linearized error dynamics around the trajectory. We show in this paper that the very same idea may be incorporated for tracking control of MIMO recurrent fuzzy systems. Their dynamics is given by means of linguistic differential equations but may be converted into a hybrid system representation, which then serves as the basis for controller synthesis.

1. Introduction
The synthesis of controllers is usually based on a known model, which is assumed to sufficiently describe the dynamics of the process that is to be controlled. If only rough expert knowledge or measurement data is available, gray box models based on fuzzy logic are a way to derive an approximate model. Although some methods are known that attempt to control the process without any model (see, e.g., [3, 9, 14]), it is generally preferable to exploit existing rough process knowledge for controller synthesis.

In particular, Recurrent Fuzzy Systems [1, 15] represent the system dynamics approximately by means of linguistic rules, in which the conclusion part consists of the state derivative being described simply as a constant. Thus, Recurrent Fuzzy Systems may also be interpreted as zeroth-order Takagi-Sugeno fuzzy systems. In contrast to the more general Takagi-Sugeno fuzzy systems, the main benefit of Recurrent Fuzzy Systems relies on the possible linguistic interpretability of the rules obtained. Having obtained a recurrent fuzzy system, the question arises on how to control this particular system class. In [11, 12] it was shown how to derive stabilizing feedback for these dynamic fuzzy systems by means of static fuzzy control or switching polynomial controllers for known equilibria.

This paper now addresses the question of tracking control for Recurrent Fuzzy Systems. In the last two decades, the tracking control problem was answered for the broad class of (differentially) flat nonlinear systems [10] in a multitude of works (see, e.g., [4, 8, 13, 22]). See also [18–20] for applications of flatness-based control.

Since it is well known [20], that utilizing an exact linearizing feedback controller may cause severe problems in the presence of model uncertainties, a common approach is to exploit the
flatness property to obtain a nominal feedforward control from the inverse system combined with a linear controller for the error system linearized around the trajectory.

In this paper, we show that this approach is general enough to solve the tracking control problem for flat Recurrent Fuzzy Systems as well, whereas for brevity we restrict our attention to square MIMO systems. In a first step, a switched nonlinear system is obtained from the given rule base of the recurrent fuzzy system. By applying a dynamic transformation, this switched nonlinear system is then rendered input-affine, such that feedforward control can be applied locally. By linearizing the error dynamics around the trajectory, a simple PID controller then controls small perturbations around the trajectory as well as model uncertainties, which naturally arise due to the approximate nature of Recurrent Fuzzy Systems.

The benefit of the proposed method clearly is in the analytic derivation of the inverse system, which is then used for generation of the feedforward control signal. This is in contrast to the system class of Piecewise Bilinear Models, where tracking control is usually based on learning techniques for the inverse system [6, 7] and hence, there is no guarantee beforehand that the inverse system exists. On the other hand, the method proposed in this paper can similarly be applied to Piecewise Bilinear Models as well.

The remainder is organized as follows: In Sec. 2, basic definitions for Recurrent Fuzzy Systems are reviewed roughly, and a dynamic transformation for decoupling of the inputs is given. Sec. 3 then reviews well-known results from flat systems in the light of Recurrent Fuzzy Systems. The flatness property is then used in Sec. 4 for trajectory generation and synthesis of the feedforward control, whereas Sec. 5 describes the additional PID control for stabilization of the error dynamics. A numerical example is given in Sec. 6 and concluding remarks in Sec. 7.

2. Preliminaries

2.1. Recurrent Fuzzy Systems

This section briefly revisits the definition of Recurrent Fuzzy Systems (RFS) as detailed in [15] and [1]. Its block diagram is depicted in Fig. 1, with dynamic function $f(x, u)$ and output function $g(x)$ being static fuzzy systems defined in the input-state space $\mathcal{Z} = \mathcal{X} \times \mathcal{U}$, $\mathcal{X} = \mathbb{R}^n$, $\mathcal{U} \subset \mathbb{R}^m$ with $u \in [u_{\text{min}}, u_{\text{max}}]$. The output function $g(x)$ is assumed not to contain feedthrough and is thus defined over the state space only. The output values are bounded to $[y_{\text{min}}, y_{\text{max}}]$.

$$f(x, u) \quad g(x)$$

Figure 1: Recurrent Fuzzy System

Each linguistic rule of the complete rule base describing the system dynamics is of the form

If $x = L^x_j$ and $u = L^u_q$,
then $\dot{x} = L^x_{w(j,q)}$, \hspace{1cm} (1)

with $L^x_j$, $L^u_q$ denoting vectors of linguistic values in the state and input space. Similarly, $L^x_{w(j,q)}$ denotes vectors of linguistic values describing gradients quantitatively. For the output function, similar rules of the form

If $x = L^x_j$,
then $y = L^y_j$ \hspace{1cm} (2) are obtained.
In order to transfer the linguistic rules into a numerical representation, core position vectors \( s_j^i \in [x_{\text{min}}, x_{\text{max}}] \), \( s_q^i \in [u_{\text{min}}, u_{\text{max}}] \), \( s_j^q \in [y_{\text{min}}, y_{\text{max}}] \) and core position gradients \( s_j^w \in \mathbb{R}^n \) are associated with the linguistic vectors. For simplicity, the sets of core position vectors \( s_j^i \) of the dynamic function \( f(x, u) \) and those of the output function \( g(x) \) are assumed to be identical. This assumption is rather mild, since an RFS with different sets can always be converted into one with identical sets by an augmentation of the respective grids.

Clearly, the rule base defines gradients at discrete grid points \( (s_j^i, s_q^i, s_j^q) \), which is also visualized in Fig. 2. It becomes obvious, that the core position vectors induce a rectangular grid in \( \mathbb{Z} \). The interpolation region of core position vectors being adjacent to a vector \((x, u) \in \mathbb{Z}\) is termed **hypersquare**, denoted \( H_I \). Therein, \( I = (j, q) \) consist of the indices of the lower core positions. Note, that I might therefore index a single core position \( s_i \) or a hypersquare \( H_I \).

In order to obtain real-valued dynamic and output functions in the entire input-state space, membership functions \( \mu_j^x(x_i), \mu_q^u(u_p) \) are introduced, assigning to states and inputs a degree of membership to linguistic values with associated core positions \( s_j^i, s_q^i \). Having thereby fuzzified states and inputs, 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 [2]. Thus, the representation

\[
\dot{x} = \sum_{j,q} s_j^w_{(j,q)} \cdot \prod_{i=1}^n \mu_j^{x_i}(x_i) \cdot \prod_{p=1}^m \mu_q^{u_p}(u_p) \quad (3a)
\]

\[
y = \sum_j s_j^y \cdot \prod_{i=1}^n \mu_j^{x_i}(x_i) \quad (3b)
\]

is obtained for the state derivative (and output), which is a weighted summation over all core position gradients (or core position outputs, respectively). The weighting by the premise is the product of memberships in every dimension. If for membership functions \( \mu_j^x, \mu_q^u \) of states and inputs, triangular and ramp shaped functions are assumed, i.e.

\[
\mu_j^{x_i}(x_i) = \begin{cases} 
  \frac{x_i - x_{i-1}^{x_i}}{s_j^i - x_{i-1}^{x_i}}, & x_i^{min} \leq s_j^{x_i} \leq x_i < x_{i+1}^{x_i} \\
  \frac{x_{i+1}^{x_i} - x_i}{s_j^i - x_{i+1}^{x_i}}, & x_i < s_j^{x_i} \leq x_i^{max} \\
  \frac{s_{j+1}^i - x_i}{s_{j+1}^i - s_j^i}, & s_{j+1}^i \leq x_i < s_{j+1}^{x_i} \\
  1, & x_i^{min} \leq x_i \lor x_i \geq x_i^{max} \\
  0, & \text{else}
\end{cases}
\]

then the dynamics (3a) evaluates to the product of piecewise affine functions, which can be
further simplified to

$$\dot{x} = a_l + \sum_{i=1}^{n} a_{x, i} x_i + \sum_{p=1}^{m} a_{u, p} u_p + \sum_{j=2}^{n-1} \sum_{i=1}^{j-1} a_{x, i, j} x_i x_j$$

$$+ \sum_{p=1}^{m} u_p \sum_{j=2}^{N} a_{x, i, u, p} x_i + \cdots + a_{1...n_1...n_m} x_1 \cdots x_n u_1 \cdots u_m$$

(5)

for every $H_l$. By rearranging terms,

$$\dot{x} = a_l(x) + B_l(x)Z(u)$$

(6)

is obtained, with the vector of monomials in the inputs $u_p$ being denoted as $Z(u) = [u_1, u_2, u_1 u_2, \ldots, u_m, u_1 u_m, \ldots, u_1 \cdots u_m]^T \in \mathbb{R}^{2m-1}$. Obviously, the system dynamics of an RFS with $m > 1$ are not input affine, which is disadvantageous for the controller synthesis. Nevertheless, with the following dynamic transformation, an equivalent input-affine dynamics can be achieved:

First, the system input is augmented by an integrator, such that inputs $v \in \mathbb{R}^m, \dot{u} = v$ are obtained. Introducing new state variables $\mathbf{w} = Z(u)$, the differential equations of these new states then read

$$\dot{\mathbf{w}} = \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \vdots \\ \dot{u}_1 u_2 + u_1 \dot{u}_2 \\ \vdots \\ \dot{u}_1 u_2 + u_1 \dot{u}_2 \\ \vdots \\ \dot{u}_1 u_2 + u_1 \dot{u}_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_1 u_2 + u_1 v_2 \\ \vdots \\ v_1 u_2 + u_1 v_2 \\ \vdots \end{bmatrix} = \frac{\partial Z(u)}{\partial u} \cdot \mathbf{v}.$$  

(7)

Thus, the state space equations of the prolonged system read

$$\begin{bmatrix} \dot{x} \\ \dot{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} a_l(x) + B_l(x)\mathbf{w} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\partial Z(u)}{\partial u} \end{bmatrix} \cdot \mathbf{v}.$$

(8)

Therefore, the following considers without loss of generality the input-affine dynamics

$$\dot{x} = a_l(x) + \sum_{p=1}^{m} B_{l, p}(x) \cdot u_p,$$

(9a)

$$y = g_j(x)$$

(9b)

for each interpolation region $H_l$.

2.2. Further notations

Throughout the paper, we use the standard notation $\text{rank}\{M\}$ for the rank of a matrix and $I_n$ for the $n \times n$-identity matrix. By $L_a f(x) = \sum_{i=1}^{n} \frac{\partial f(x)}{\partial x_i} a_i$, a Lie derivative of a function $f$ regarding $a$ is denoted. Eigenvalues are denoted by $\lambda \in \mathbb{C}$. Usually, hypersquares $H_l$ and core position vectors $s^x_j, s^u_l$ are indexed by an integer vector $l$ corresponding to a grid position. If ambiguity is excluded, sometimes a scalar index notation $H_i, H_j$ is used for enumeration of different hypersquares and core positions.
3. Flatness Property of RFS

In the following, we briefly review the property of (differential) flatness, which is detailed in length, e.g., in [10, 13]. Nevertheless, the description is carried out from the perspective of RFS.

Definition 1 An RFS (9) is locally (differentially) flat on a hypersquare $H_1$, iff a (virtual) flat output $z \in \mathbb{R}^m$ with

$$z = \varphi_1(x, u, \hat{u}, \ldots, u^{(n)})$$

exists, such that

$$x = \psi_1^\ell(z_1, \dot{z}_1, \ldots, z_1^{(\beta_1-1)}, \ldots, z_m, \ldots, z_m^{(\beta_m-1)}),$$

$$u = \psi_1^p(z_1, \dot{z}_1, \ldots, z_1^{(\beta_1)}, \ldots, z_m, \ldots, z_m^{(\beta_m)}).$$

The RFS is globally (differentially) flat, iff it is locally flat for every $H_1 \in \mathcal{H}$.

Often, the flat outputs $z$ are incident with the measured outputs $y$. For the order of the derivatives $\beta_p$ in (11), it holds in general that $\sum_{p=1}^m \beta_p \geq n$. As noted in [21], for $\sum_{p=1}^m \beta_p = n$ it is always possible to transform the system dynamics (locally) into a linear form by means of static state feedback. Furthermore, in this case $z = \varphi_1(x)$, i.e., the flat outputs are not dependent on the input or its derivatives. Also, flatness is then directly related to observability of the system’s states via the flat outputs $z_i$ and in contrast to the case $\sum_{p=1}^m \beta_p > n$, observability is not dependent on the input signal.

For simplicity, the following therefore considers the case $\sum_{p=1}^m \beta_p = n$. Then, by local change of coordinates

$$\xi = \left[\xi_{1,1}, \xi_{1,2}, \ldots, \xi_{1,\delta_{1,1}}, \ldots, \xi_{m,1}, \ldots, \xi_{m,\delta_{1,m}}\right]^T$$

$$= \left[z_1, \dot{z}_1, \ldots, z_1^{(\delta_{1,1}-1)}, \ldots, z_m, \ldots, z_m^{(\delta_{m,m}-1)}\right]^T,$$

(6) is diffeomorphic to the controller normal form

$$\dot{\xi}_{i,j} = \xi_{i,j+1}, \quad j = 1, \ldots, \delta_{1,i} - 1$$

$$\dot{\xi}_{i,\delta_{1,i}} = \alpha_{1,i}(\xi, u).$$

(13)

Therein, $\delta_{1,p}$ denotes the local vector relative degree regarding the $p$-th output, defined by

$$L_{b_1,p} L_{a_1}^{(i)} g_{j,p}(x) = 0, \quad i = 1, \ldots, \delta_{1,p} - 2,$$

$$L_{b_1,p} L_{a_1}^{(\delta_{1,p}-1)} g_{j,p}(x) \neq 0,$$

(14a)

(14b)

which has to hold for all $x \in H_1$. The local vector relative degree $\delta_1 = [\delta_{1,1}, \ldots, \delta_{1,m}]^T$ then combines all $m$ relative degrees. By letting $\alpha_{1,j}(\xi, u) = v_i$, (6) is equivalent to a system in MIMO Brunovsky form

$$\dot{\xi}_{i,j} = \xi_{i,j+1}, \quad j = 1, \ldots, \delta_{1,i} - 1,$$

$$\dot{\xi}_{i,\delta_{1,i}} = v_i,$$

(15)

as shown in [13]. Furthermore and according to [17], if rank $\{A_1\} = m$ holds for the matrix

$$A_1 = \left[\begin{array}{ccc}
L_{b_{1,1}} L_{a_1}^{(\delta_{1,1}-1)} g_{j,1}(x) & \ldots & L_{b_{1,m}} L_{a_1}^{(\delta_{1,1}-1)} g_{j,1}(x) \\
\vdots & \ddots & \vdots \\
L_{b_{1,1}} L_{a_1}^{(\delta_{1,m}-1)} g_{j,m}(x) & \ldots & L_{b_{1,m}} L_{a_1}^{(\delta_{1,m}-1)} g_{j,m}(x)
\end{array}\right],$$

(16)

then a static state feedback may be applied to (6), such that it is input-output decoupled.
4. Inverse System and Feedforward Control

According to (11), state and input can be reconstructed from the flat outputs and a finite number of their derivatives. Under the assumption that initial conditions are consistent with the system, e.g.,

$$x_0 = \psi^u_1(z_1(0), \ldots, z_1^{(\beta_1-1)}(0), \ldots, z_m(0), \ldots, z_m^{(\beta_m-1)}(0)),$$

the nominal control $u_d(t)$ may be computed from (11b) given a sufficiently smooth trajectory $z_d(t)$.

Due to the piecewise definition of the system dynamics (9), direct computation of (11b) is not possible, since $\psi^u_l$ is implicit due to the dependence of $l$ on $x$ and $u$. To circumvent this problem, the inverse system as shown in Fig. 3 continuously reconstructs the current hypersquare $H_l$ by means of $l = \sigma(x_d, u_d)$. Within the closed-loop, the nominal input $u_d$ is then reconstructed from the nominal trajectory $z_d$ and the states $x_d$. An alternative would be to obtain necessary derivatives of $z_d$ from the trajectory generator, or calculation of the derivatives, which is always difficult in the presence of noise.

In addition, the system depicted in Fig. 3 is diffeomorphic to a linear system, thus its eigenvalues can be chosen arbitrarily stable. To see this, consider from (13) the linear dynamics in transformed coordinates

$$\begin{bmatrix}
\dot{\xi}_{1,\delta_1,1} \\
\vdots \\
\dot{\xi}_{m,\delta_1,m}
\end{bmatrix} =
\begin{bmatrix}
\alpha_{1,1}(\xi, u) \\
\vdots \\
\alpha_{1,m}(\xi, u)
\end{bmatrix} =
\begin{bmatrix}
L_{\alpha_1}^{(\delta_1,1)} g_{j_1,m}(x) \\
\vdots \\
L_{\alpha_1}^{(\delta_1,m)} g_{j_m,m}(x)
\end{bmatrix} + A_1(x) \cdot u.
$$

If for every $H_1$, (16) is invertible, then global input-output decoupling and tracking is achieved by means of the control law

$$u = A_1(x)^{-1} \left( -L_{\alpha_1}^{(\delta_1,1)} g_{j_1,m}(x) \cdots L_{\alpha_1}^{(\delta_1,m)} g_{j_m,m}(x) - K_1 \cdot \xi + v \right).$$

Therein, either $v_i = z_i^{(\delta_i,1)}$ and $K = 0$ is chosen such that in (15), $\dot{\xi}_{i,\delta_1,1} = z_i^{(\delta_i,1)}$, or $K$ is chosen such that the closed-loop linear dynamics

$$\begin{bmatrix}
\dot{\xi}_{i,1} \\
\vdots \\
\dot{\xi}_{i,\delta_1,1}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 \\
\vdots & \ddots \\
-k_{i,1} & -k_{i,2} & \cdots & k_{i,\delta_1,1}
\end{bmatrix}
\begin{bmatrix}
\xi_{i,1} \\
\vdots \\
\xi_{i,\delta_1,1}
\end{bmatrix} +
\begin{bmatrix}
0 \\
\vdots \\
v_i
\end{bmatrix}$$

Figure 3: Inverse System
are stable and \( v \) is set such that \( g_1(x_d) = z_d \). While the first choice allows in theory for the exact tracking of \( z_d \), fast changes of the nominal trajectory may cause a high amplitude of \( u_d \). In the second case, poles of the closed-loop linear system can be placed such that the transient behavior of (20) will not cause saturation effects due to the usual bounds on the control input. Obviously, the following holds:

**Proposition 1** If for each hypersquare \( H_1 \), \( \alpha_{l,i}(\xi, u) \) are such that the linear closed-loop local dynamics (20) have identical eigenvalues, the controlled RFS has a global linear input-output behavior.

It follows immediately, that if the desired trajectory \( z_d \) is sufficiently smooth, so will be the output \( z \) of the RFS. On the other hand, this smoothness property does in general not hold for the nominal input \( u_d \). To see this, consider the dynamics of the RFS (3a), which is globally Lipschitz by definition. Due to the choice of triangular membership functions (4), the piecewise defined function \( \psi^u \) and more specifically, (19) is dependent on Lie derivatives of the output function. Continuity of \( u \) then implies, e.g.,

\[
\lim_{x_i \nearrow s^{e_i}_i} \sum_{i=1}^n \frac{\partial g_i(x)}{\partial x_i} a_{l,i}(x) = \lim_{x_i \searrow s^{e_i}_i} \sum_{i=1}^n \frac{\partial g_i(x)}{\partial x_i} a_{l,i}(x),
\]

which does not hold in general. Thus, \( u_d \) may be discontinuous, even if \( z_d \) is smooth.

### 5. Extended PID control of Error Dynamics

In addition to the flatness-based feedforward control derived from the RFS model of the plant, an additional feedback controller is necessary in order to stabilize the error dynamics around the trajectory, as described in numerous works (see, e.g., [5, 13, 20, 22]). The necessity of the feedback controller is caused by minor inconsistencies of the system’s initial state with the reference trajectory as well as noise and disturbances. In the light of RFS, the main reason for the additional feedback is due to model uncertainties arising from the approximate gray box modeling.

Several synthesis concepts are proposed in the literature, which are mostly based on linearization of the error dynamics and application of linear controllers. Here, the approach of an extended PID controller as proposed in [13] is applied due to its simplicity.

Let the error state of the \( i \)-th linear subsystem be denoted

\[
e_i = \xi_i - \xi_i,d
\]

with \( \xi_i^T = [\xi_{i,1}, \ldots, \xi_{i,\delta_{l,i}}] \), \( i = 1, \ldots, m \). Extending this error state by the integrated difference between the measured output and the reference leads to

\[
e_i^T = [e_{i,0}, e_i^T],
\]

\[
e_{i,0} = \int_0^t e_{i,1}(\tau)d\tau.
\]

Thus, the nonlinear error dynamics read

\[
\dot{e}_{i,j} = e_{i,j+1}, \quad j = 0, \ldots, \delta_{l,i} - 1,
\]

\[
\dot{e}_{i,\delta_{l,i}} = \alpha_{l,i}(\xi, u) - \xi_i^{(\delta_{l,i})} = \alpha_{l,i}(\xi_d + e, u_d + \Delta u) - \xi_i^{(\delta_{l,i})}.
\]
Figure 4: Complete controlled system with flatness-based feedforward control derived from RFS, high-gain observer and extended PID control

By means of an online linearization,

\[
\Delta \dot{e}_{i,j}^{\delta_{l,i}} = \Delta e_{i,j+1} - \delta_{l,i} \cdot \Delta e
\]

\[
\Delta \dot{e}_{i,\delta_{l,i}} = \frac{\partial}{\partial e} \left( \frac{\partial e}{\partial \Delta u} \right)_{e^*,\Delta u^*} \cdot \Delta e
\]

\[
+ \frac{\partial}{\partial \Delta u} \left( \frac{\partial e}{\partial \Delta u} \right)_{e^*,\Delta u^*} \cdot \Delta u'
\]

is then obtained, for which a stabilizing controller

\[
\Delta u' = k_T \Delta e_i
\]

can be determined. Overall, the resulting control law combining the feedforward part as well as the extended PID control reads

\[
u = u_d + \Delta u.
\]

Since (22) is dependent on the full state \(\xi\), a high-gain observer [16] is utilized to reconstruct non-measurable states. The complete control structure is depicted in Fig. 4.

6. Simulation Results

In the following, the aforementioned tracking control is applied to the numerical example of the nonlinear dynamics

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{x}_3 \\
y_1 \\
y_2
\end{bmatrix} =
\begin{bmatrix}
x_2 \\
x_2 x_3 + x_3^2 + u_1 \\
x_2 (1 - x_3) - x_2^2 + u_2 \\
x_1 \\
x_3
\end{bmatrix},
\]

which will serve as ground truth reference model. Usually, when utilizing RFS, the exact plant dynamics are unknown and the fuzzy system is obtained by expert knowledge or measured data. Here, an RFS is derived from (28) by sampling the nonlinear dynamics at core positions \(s_{j_i}^{\delta_{l_i}} \in \{-2, -1, 0, 1, 2\}\) and \(s_{u_p}^{\delta_{l_p}} \in \{-10, 0, 10\}\) with associated linguistic values \(L_{j_i}^{\delta_{l_i}} \in \{\text{neg. big, neg. small, zero, pos. small, pos. big}\}\) and \(L_{u_p}^{\delta_{l_p}} \in \{\text{negative, zero, positive}\}\). The complete rule base defines a partition of the input-state space with \((5 - 1)^3 \cdot (3 - 1)^2 = 256\)
hypersquares \( H_1 \), whereas the output model is defined over a grid with \( (5-1)^3 = 64 \) hypersquares \( H_j^x \). Each local dynamics is then defined by (6). E.g., for the hypersquare \( H_1 \) bounded by \(-2 \leq x_j \leq -1, j \in \{1, 2, 3\}\), and \(-10 \leq u_q \leq 0, q \in \{1, 2\}\), the local system

\[
\dot{x} = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & 7 & -1 & 1 & 0 & 0 & 0 \\
2 & 0 & 4 & 0 & 0 & 0 & 0 & -1 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_{1x2} \\
x_{1x3} \\
x_{2x3} \\
x_{1x2x3}
\end{bmatrix}
\]

(29a)

\[
+ \begin{bmatrix}
0 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_1u_2
\end{bmatrix}
\]

\[
y = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{bmatrix}
\begin{bmatrix}
1 \\
x_1 \\
x_3 \\
x_{1x3}
\end{bmatrix}
\]

(29b)

is obtained. For each hypersquare \( H_l \), the local relative degrees

\[
\delta^T_i = [2 \ 1], \quad i = 1, \ldots, 256
\]

(30)

are then computed. Due to \( \sum_{j=1}^n \delta_{i,j} = n \), the system has no internal dynamics and can be stabilized via static state feedback. Thus, \( y = z \) is a flat output. Furthermore, for each hypersquare, the matrix \( A_1 \) in (16) is invertible, e.g., for (29) it is computed to \( A_1 = I_m \). Therefore, a local controller can be found for each hypersquare achieving global input-output decoupling.

In this example, the functions \( \alpha_{l,i} \) were chosen such that the local linear dynamics (20) have stable eigenvalues at \( \lambda_i = -10 \). Since all \( \alpha_{l,i} \) can be pre-computed offline, fast online determination of the nominal control \( u_d \) is possible by means of a look-up table and the evaluation of a function. Furthermore, the extended PID control is computed online as linear quadratic regulator. The online linearization of the error dynamics and computation of the feedback are computationally more complex than the feedforward control, and thus become the bottleneck regarding real time implementation.

In order to test the controller, the signals

\[
y_{1,d} = -1.5 \sin \frac{\pi}{8} t,
\]

\[
y_{2,d} = 0.75 \cos \frac{\pi}{6} t
\]

are considered, which are to be tracked by the flat outputs of the system. The nominal trajectories (31) are applied to the controlled ground truth (28), having an initial state of \( x_0 = [0.05, \ -0.5, \ 0.5]^T \), whereas the initial state of the inverse system as well as the high-gain observer is \( \hat{x}_0 = [0, \ 0, \ 0]^T \). As becomes apparent from the simulation results shown in Fig. 5, the flat outputs follow the nominal trajectory very well. Note that tracking is achieved even in case of initial states not being consistent with the trajectory. Furthermore, Fig. 6 depicts the development of the control input \( u = u_d + \Delta u \). Apparently, the bound on the control input \( |u_d| \leq 10 \) is not exceeded, even during the initial transient behavior.
From Fig. 5, it can also be seen that the plant output does not converge asymptotically to the nominal trajectory, since small deviations of $y$ from $y_d$ remain. The reason for this effect is mainly due to the fact that the RFS, by which the nominal control law $u_d$ is computed, is only an approximation of the ground truth model, to which the controller is applied to. In addition, the inverse system (19) does not follow the reference trajectory arbitrarily fast. By assigning faster poles to the linear subsystems (20), this effect can be milder, as shown in Fig. 7. Therein, different simulation runs are shown, each with unique set of equal eigenvalues of (20). As can be seen, neglecting the initial transient behavior, the error $e = y - y_d$ becomes smaller with faster eigenvalues. At the same time, the magnitude of $u_d$ increases with faster eigenvalues, such that especially during the initial displacement of the outputs from the reference trajectory, the control input exceeds its bound. Thus, a trade-off between fast decay of the transients and meeting the control input bounds has to be made.

7. Discussion
In this paper, an approach for flatness-based feedforward control of Recurrent Fuzzy Systems was presented. It was shown that existing techniques for tracking control of flat nonlinear systems
Figure 7: Comparison of control error and control input for different eigenvalues of the linearized dynamics: (--) $\lambda_i = -10$, (--)$ \lambda_i = -30$, and (---) $\lambda_i = -70$.

may also be applied to Recurrent Fuzzy Systems. Since the dynamics of a recurrent fuzzy system is defined by means of a linguistic rule base, it was first shown how to derive local real valued system dynamics under mild assumptions. Thus, a partitioning of the input-state space was obtained, whereas the system dynamics within each partition was described by a vector polynomial being multi-affine in the inputs and states. By means of a dynamic transformation, these local dynamics were then shown to be always equivalent to a prolonged input-affine system.

Due to space limitations, several simplifying assumptions were made throughout the paper, which do not lessen the underlying idea, but should be addressed in future work: First, only Recurrent Fuzzy Systems without internal dynamics were considered, whereas the application to those with stable internal dynamics (i.e., minimum phase Recurrent Fuzzy Systems) is straightforward. Second, only flat Recurrent Fuzzy Systems stabilizable with static state feedback were addressed. A similar discussion for the more general case of dynamic feedback remains an open topic, but may also be tackled by means of the flatness approach. In addition, only square systems with global decoupling of each input-output pair were discussed, in contrast to the more general form of Recurrent Fuzzy Systems with a number of inputs different from the number of outputs.

Finally, for feedback control of the error system, an extended PID controller was utilized, stabilizing the linearized error dynamics around the reference trajectory. Here, the PID controller was mainly chosen due to its simple implementation. Nevertheless, as mentioned in [13], any kind of linear controller can be applied. In particular, a comparison especially with robust controllers, such as $H_\infty$ controllers, would be interesting in the light of the approximate nature of the system model.
Acknowledgments
This work was supported by the Deutsche Telekom Stiftung (www.telekom-stiftung.de) and the Studienstiftung des Deutschen Volkes (www.studienstiftung.de).

References
[1] J. Adamy and A. Flemming. Equilibria of continuous-time recurrent fuzzy systems. *Fuzzy Sets and Systems*, 157(22):2913–2933, 2006.
[2] J. Adamy and R. Kempf. Regularity and chaos in recurrent fuzzy systems. *Fuzzy Sets and Systems*, 140(2):259–284, 2003.
[3] F. Allgöwer, J. Ashman, and A. Ilchmann. High-gain adaptive λ-tracking for nonlinear systems. *Automatica*, 33(5):881–888, 1997.
[4] F. Antritter. On the relations between different flatness based design methods for tracking controllers. In *American Control Conference, 2008*, pages 1740–1745. IEEE, 2008.
[5] S. Devasia, D. Chen, and B. Paden. Nonlinear inversion-based output tracking. *Automatic Control, IEEE Transactions on*, 41(7):930–942, 1996.
[6] L. Eciolaza, T. Taniguchi, M. Sugeno, D. Filev, and Y. Wang. Piecewise bilinear models for feedback error learning: On-line feedforward controller design. In *IEEE International Conference on Fuzzy Systems*, pages 1–8, 2013.
[7] L. Eciolaza, T. Taniguchi, and M. Sugeno. Brain style control scheme: Simultaneous forward and inverse model identification and controller design. In *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 2457–2464, July 2014. doi: 10.1109/FUZZ-IEEE.2014.6891893.
[8] N. Faiz, S. K. Agrawal, and R. M. Murray. Trajectory planning of differentially flat systems with dynamics and inequalities. *Journal of Guidance, Control, and Dynamics*, 24(2):219–227, 2001.
[9] M. Fliess and C. Join. Model-free control. *International Journal of Control*, 76(6):537–556, 2003.
[10] M. Fliess and E. P. Ryan. High-gain control without identification: a survey. *GAMM-Mitteilungen*, 31(1):115–125, 2008.
[11] S. Gering and J. Adamy. Synthesis of local state feedback for continuous-time recurrent fuzzy systems. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Hyderabad, India, 2013.
[12] 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*, pages 271–277, Chania, Greece, June 2013.
[13] V. Hagenmeyer and E. Delaleau. Exact feedforward linearization based on differential flatness. *International Journal of Control*, 76(6):537–556, 2003.
[14] A. Ilchmann and E. P. Ryan. High-gain control without identification: a survey. *GAMM-Mitteilungen*, 31(1):115–125, 2008.
[15] M. Fliess and J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: introductory theory and examples. *International journal of control*, 61(6):1327–1361, 1995.
[16] M. Zerar, F. Cazaurang, and A. Zolghadri. Robust tracking of nonlinear mimo uncertain flat systems. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 536–541. IEEE, 2004.