Stability Analysis for Advanced Fuzzy Systems

H. Kiendl
University of Dortmund
Faculty of Electrical Engineering
Otto-Hahn-Str. 4, D-44221 Dortmund
Phone: +49.231.755-2760
Fax: +49.231.755-2752
e-mail: kiendl@esr.e-technik.uni-dortmund.de

ABSTRACT: Strict stability analysis of dynamic systems usually requires analytical methods. However, for complicated non-linear systems such methods often fail, as they are not applicable or supply very conservative results. The method of convex decomposition allows strict, non-conservative, computer-aided stability analysis for discrete time systems with piecewise linear system equations. This method was introduced previously and is summarized, refined and extended in this paper. Furthermore it is pointed out that this method is very appropriate for the analysis of advanced fuzzy systems based on TOR-defuzzification.

KEYWORDS: stability analysis, $G_{H,N}$-stability, method of convex decomposition, TOR-defuzzification

INTRODUCTION

Non-linear controllers, especially fuzzy controllers, are often designed by first selecting a controller structure and then tuning the values of the adjustable controller parameters interactively. This may be done with the real or simulated control system or by a simulation. The result is a non-linear controller that usually supplies much better performance than does a conventional linear controller. However, the good performance is only established for those situations that were considered in the experimental or simulative tuning process. In particular, a strict proof of stability is often not possible, as existing criteria are not applicable or too conservative if the controller is highly non-linear.

A possible way to overcome this difficulty is to construct sophisticated Lyapunov functions with the aid of a computer as, for instance, in [Kiendl, Rüger 1995; Johanssen 1999]. A more direct approach, which does not necessarily require any Lyapunov function and supplies additional information on system performance, is based on the concept of $G_{H,N}$-stability and the method of convex decomposition, which were introduced in a previous paper [Kiendl 1987]. These ideas allow computer-aided stability analysis for discrete-time systems with piecewise linear system equations. Since its introduction, this method has been applied and further developed [Karweina 1989; Rumpf 1997; Kiendl 1997a; Knicker 1999; Knicker, Krause 1999; Schäfers 1999]. The main purpose of this paper is to summarize, refine and extend this method. Furthermore, it is pointed out that this method works especially well for the stability analysis of advanced fuzzy systems based on the recently introduced TOR-defuzzification [Kiendl 1996, 1997a, 1997b].

THE PROBLEM

We consider a discrete-time non-linear control system

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}(\boldsymbol{x}_k) \qquad (1)$$

with an n-dimensional state vector $\boldsymbol{x}$. Such a system is obtained, for instance, by combining a linear plant

$$\boldsymbol{x}_{k+1} = \boldsymbol{\ddot{O}}\boldsymbol{x}_k + \boldsymbol{h}u_k \qquad (2)$$

with a non-linear control law

$$u_k = g(\boldsymbol{x}_k)\,. \tag{3}$$

We assume that the control system is subjected to sporadic disturbances that displace the state vector into initial states situated in an *initial domain G* in the form of a convex polytope. For simplicity, we assume that *G* is a cuboid (Figure 1).

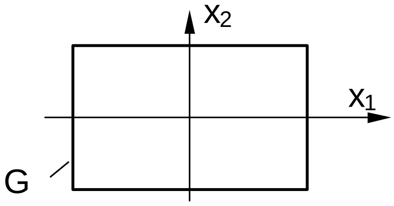

Figure 1: The initial domain *G*, a cuboid that contains all possible initial states of a control system (2).

Our problem is to investigate whether the system performance is good, or at least acceptable, for all possible initial states $\boldsymbol{x} \in G$. With the notation

$$\boldsymbol{F}(G) = \{\boldsymbol{F}(\boldsymbol{x}) | \boldsymbol{x} \in G\} \tag{4}$$

we can say that our problem is to investigate whether the *domain trajectory*

$$G \xrightarrow{\boldsymbol{F}} \boldsymbol{F}(G) \xrightarrow{\boldsymbol{F}} \boldsymbol{F}^2(G) \xrightarrow{\boldsymbol{F}} \ldots \tag{5}$$

has certain desired properties. Obviously this problem refers to the *infinite continuum* of all possible initial states. For the solution of such analysis problems referring to an infinite continuum of possible cases, *analytical* methods are preferred in principle. Think of the stability theorem that says that a *linear* system $\boldsymbol{x}_{k+1} = \boldsymbol{Ö}\boldsymbol{x}_k$ is stable if and only if for all eigenvalues m_i of the system matrix, $|m_i| < 1$ is true. This theorem covers the continuum consisting of all possible matrices $\boldsymbol{Ö}$ and all possible initial states $\boldsymbol{x}$. However, the results of *non-linear* system theory are much less far-reaching. The existing criteria due to Lyapunov and Popov, for instance, are only *sufficient* criteria and therefore supply a conservative result or are not applicable if the system equation is too complicated. On the other hand, it is obvious that the above problem cannot be solved by simple simulation of some system trajectories. That way we can only ever cover a *finite* number of initial states.

To allow a strict, computer-aided solution of the above continuum problem, two approaches have been introduced [Kiendl 1987]:

(i) The concept of $G_{H,N}$*-stability,* which is meaningful for applications and sufficiently simple for a computer-aided stability check;

(ii) The *method of convex decomposition,* which allows computer-aided stability analysis for *piecewise linear* system equations (1).

These approaches have been applied and further developed in [Karweina 1989; Rumpf 1997; Kiendl 1997a; Knicker 1999; Knicker, Krause 1999]. Here the basic ideas and concepts are summarized, refined and extended.

Remark 1: The method of convex decomposition can also be applied to the solution of other continuum problems such as the following *robustness problem*: given a matrix $\boldsymbol{A}(\boldsymbol{p})$ depending on a parameter vector $\boldsymbol{p} \in B$, where *B* is a given uncertainty box of the parameter space, with the method of convex decomposition we can investigate whether $\boldsymbol{A}(\boldsymbol{p})$ is stable for all $\boldsymbol{p} \in B$ [Kiendl 1987]. However, here we restrict our analysis to the $G_{H,N}$ -stability problem.

$G_{H,N}$-STABILITY

We consider the system (1) with a given initial domain *G*. The widely used Lyapunov concept of stability looks for *equilibrium states* $\boldsymbol{x}_E$, defined by

$$\boldsymbol{F}(\boldsymbol{x}_E) = \boldsymbol{x}_E . \tag{6}$$

Using *sufficient* Lyapunov stability criteria, it is possible to investigate whether each system trajectory that starts in *G* reaches the equilibrium *asymptoticly*. However, if the system equation is very complicated, these criteria may not be applicable.

In contrast, the concept of $G_{H,N}$*-stability* does not require equilibrium states. Instead, it considers sufficiently small *target domains H* that describe desired, accepted or tolerated states. Here the question investigated is whether each system trajectory that starts in *G* reaches the target domain within a *finite number of steps N* and stays there for all further steps (Figure 2).

Formally we state

Definition 1: The system (1) is called $G_{H,N}$-stable if and only if there exist *H*, *N* such that:

$$x_j \in H, \quad j = N, N+1,... \tag{7}$$

is valid, for each system trajectory $x_0, x_1, x_2,...$ with $\boldsymbol{x}_0 \in G$.

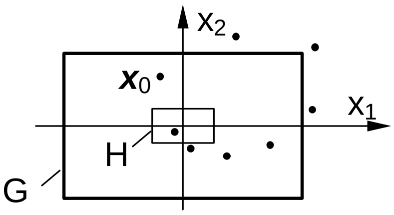

Figure 2: Definition of $G_{H,N}$-stability with initial domain *G* and target domain *H*.

Equivalently we state

Definition 2: The system (1) is called $G_{H,N}$-stable if and only if there exist *H*, *N* such that for the domain trajectory $G, \boldsymbol{F}(G), \boldsymbol{F}^2(G),...$

$$\boldsymbol{F}^j(G) \subseteq H, \quad j = N, N+1,... \tag{8}$$

is valid.

Compared with Lyapunov's asymptotic stability concept, $G_{H,N}$-stability is advantageous for applications as it considers a finite time horizon, and the target is an extended domain rather than a single point.

We establish $G_{H,N}$-stability by the following:

Stability criterion 1

Let *H* be a *Lyapunov-domain* defined by the property

$$\boldsymbol{F}(H) \subseteq H . \tag{9}$$

The system (1) is $G_{H,N}$ -stable if and only if

$$\boldsymbol{F}^{N}(G) \subseteq H \tag{10}$$

is valid.

The proof is obvious. The practical use of this criterion requires finding a domain *H* with the Lyapunov property (9), which incidentally is a natural generalization of Eq. (6). This difficulty is overcome by

Stability criterion 2

Let *K* be an auxiliary domain with the properties

$$\boldsymbol{F}^{M}(K) \subseteq K \tag{11}$$

and

$$K \subseteq H,\ \boldsymbol{F}^{m}(K) \subseteq H \quad \text{for} \quad m = 1,2,\ldots,M-1\,. \tag{12}$$

Then the system is $G_{H,N}$ -stable if one of the following two conditions holds:

(i) $$\boldsymbol{F}^{p}(G) \subseteq K \quad \text{for some} \quad p \leq N\,. \tag{13}$$

(ii) The domain G can be decomposed into subdomains $S_1, S_2, ..., S_r$ where for each subdomain

$$\boldsymbol{F}^{p}(S_i) \subseteq K \tag{14}$$

is valid for some $p \leq N$, which may depend on *i*.

The proof is obvious. With this criterion the choice of *H* is no longer restricted by the Lyapunov condition (9). However, criterion 2 is only *sufficient* in contrast to criterion 1. This difference becomes more noticeable as the *termination domain K* becomes smaller compared to *H* (Figure 3).

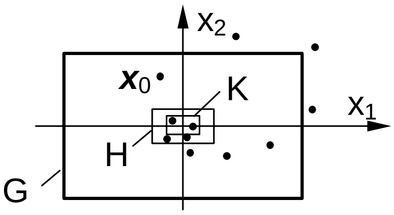

Figure 3: Check of $G_{H,N}$ -stability for the case that the target domain *H* is not a Lyapunov domain.

THE METHOD OF CONVEX DECOMPOSITION

ASSUMPTIONS

Now we assume that the function $\boldsymbol{F}(\boldsymbol{x})$ is *piecewise linear*: Let $\boldsymbol{F}(\boldsymbol{x})$ be defined by

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{cases} \boldsymbol{F}_1(\boldsymbol{x}) = \boldsymbol{\ddot{O}}_1 \boldsymbol{x} + \boldsymbol{d}_1 \ \text{ if } \ \boldsymbol{x} \in Z_1 \\ \qquad\quad \vdots \\ \boldsymbol{F}_r(\boldsymbol{x}) = \boldsymbol{\ddot{O}}_r \boldsymbol{x} + \boldsymbol{d}_r \ \text{ if } \ \boldsymbol{x} \in Z_r \end{cases} \tag{15}$$

where each linear partial function $\boldsymbol{F}_i(\boldsymbol{x})$ is valid in a corresponding zone Z_i in the form of a – not necessarily bounded – convex polytope. We include functions (15) that are *not continuous*.

A continuous piecewise linear function $\boldsymbol{F}$ is obtained, for instance, if the linear plant (2) is controlled by a non-linear controller, consisting of a linear control law and a saturation element (Figure 4). In this case the state equation consists of the three different linear functions

$$\begin{aligned} F_1(\boldsymbol{x}) &= \boldsymbol{\ddot{O}}\ \boldsymbol{x} + \boldsymbol{h} u_{\max} \\ F_2(\boldsymbol{x}) &= \left(\boldsymbol{\ddot{O}} - \boldsymbol{h}\boldsymbol{k}^T\right) \boldsymbol{x} \\ F_3(\boldsymbol{x}) &= \boldsymbol{\ddot{O}}\ \boldsymbol{x} - \boldsymbol{h} u_{\max} \end{aligned} \tag{16}$$

that are valid in the zones Z_1, Z_2 and Z_3, respectively (Figure 5). A discontinuous piecewise linear function $\boldsymbol{F}$ is obtained, for instance, if we replace the above saturation element by a non-linearity having the characteristic $u = +u_{\max}$ if $\tilde{u} \geq 0$ and $u = -u_{\max}$ otherwise.

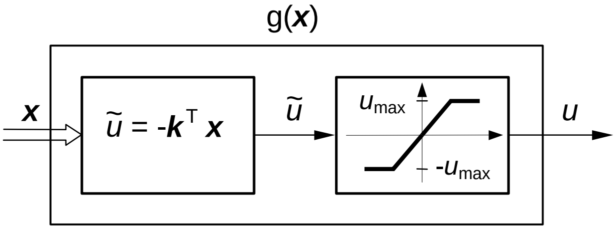

Figure 4: Non-linear controller $g(\boldsymbol{x})$ consisting of a linear state controller followed by a saturation element.

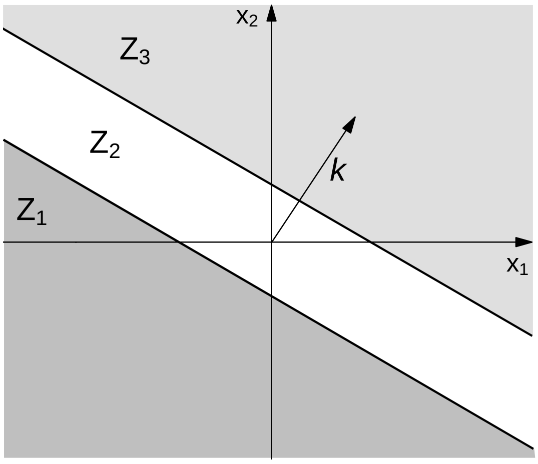

Figure 5: Decomposition of the state space into three convex zones Z_1, Z_2 and Z_3 induced by the non-linear control law of Figure 4.

For the target domain *H* and the termination domain *K* we accept any bounded convex polytope. One such choice is a cuboid

$$H = \left\{ \boldsymbol{x} \,\middle|\, \left|\boldsymbol{e}_i^T \boldsymbol{x}\right| \leq c_i,\ i = 1,2,\ldots,n \right\} \tag{17}$$

where $\boldsymbol{e}_i$ is the i-th unit vector.

Remark 2: Most of the following is also valid for arbitrary convex domains *H*. However, polytopes provide computational advantages (see below).

Remark 3: The above function ***F*** may be uniformly linear in that region of the state space where we want to specify a target domain *H*. In this case we can use known methods for finding a Lyapunov domain *H*, for instance in the form of an ellipsoid [Kiendl 1987] or a parallelepiped [Karweina 1989; Kiendl, Adamy, Stelzner 1992]. However, with the above approach we are completely free to choose the shape of the target domain *H* at will.

CHECK OF INCLUSIONS

We assume that we have specified a polytope *H* with the Lyapunov property (9). For the application of the above stability criterion 1 we must check the condition $\boldsymbol{F}^N(G) \subseteq H$. We do this as follows:

(i) *Decomposition step*

We decompose the polytope *G* to the form

$$G = G_1 \cup G_2 \cup ... \cup G_r \tag{18}$$

where the convex subpolytopes G_i are given by

$$G_i = G \cap Z_i\,, \;\; i = 1,2,...,r \tag{19}$$

(ii) *Mapping step*

For each polytope G_i

$$G_i \subseteq Z_i \tag{20}$$

holds. Therefore each $\boldsymbol{F}(G_i)$ is obtained by applying the linear partial function $\boldsymbol{F}_i$ to G_i

$$\boldsymbol{F}(G_i) = \boldsymbol{F}_i(G_i). \tag{21}$$

Consequently each $\boldsymbol{F}_i(G_i)$ is again a convex polytope.

(iii) *Check of inclusion condition*

Each polytope $\boldsymbol{F}_i(G_i)$ for which the inclusion

$$\boldsymbol{F}_i(G_i) \subseteq H \tag{22}$$

holds, stays in *H* for all further steps and therefore requires no further investigation (we may “forget” it). All other polytopes $\boldsymbol{F}_i(G_i)$ are subjected to the above three steps.

This procedure is repeated *N* times (Figure 6). If we have no remaining polytopes, $G_{H,N}$-stability is established. In an analogous way all other inclusions of criteria 1 and 2 can be checked. The essential point of the above method is that all three key operations (i), (ii) and (iii) can be performed with the aid of a computer, although they refer to infinite continua of point sets. This is due to the excellent mathematical properties of affine mappings induced by the piecewise linear system equation and of convex polytopes (see below).

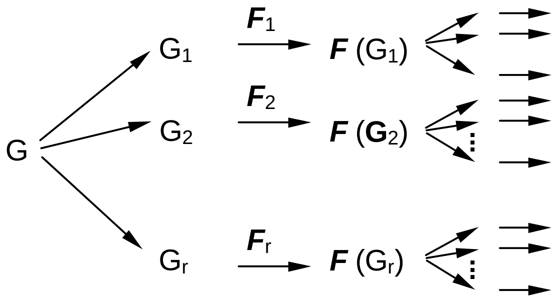

Figure 6: Check of the condition $\boldsymbol{F}^N(G) \subseteq H$ by repeated affine mapping of convex polytopes.

SPECIFICATION OF THE DOMAINS *H* AND *K*

We specify the convex polytope *H* so that it describes desired, accepted or tolerated states. In applications, system (1) often has an equilibrium state $\boldsymbol{x}_E$. In this case it makes sense to specify a sufficiently small domain *H* that contains $\boldsymbol{x}_E$. Making use of the above mapping technique, we investigate whether the Lyapunov property (9) is valid. If this is true, we work with stability criterion 1. Otherwise, we choose a convex polytope *K* contained in *H* and check whether conditions (11) and (12) can be met for some *M*. If we cannot meet these conditions, we choose a smaller domain *K*. This increases the chance that *K* has the desired property (12). This procedure is repeated until we succeed or *K* has become unreasonably small. In the latter case we break off and cannot establish $G_{H,N}$-stability.

Remark 4: Practical checks of the inclusion conditions of the above stability criteria may be executed in principle by *backward mapping.* Let *A* and *B* be arbitrary point sets, ***F*** a mapping, not necessarily bijective, and let $\boldsymbol{F}^{-j}(G)$ be defined by

$$\boldsymbol{F}^{-j}(B) = \left\{\boldsymbol{x} \,\middle|\, \boldsymbol{F}^{j}(\boldsymbol{x}) \in B\right\}. \tag{23}$$

Then the conditions

$$\boldsymbol{F}^{j}(A) \subseteq B \tag{24}$$

and

$$A \subseteq \boldsymbol{F}^{-j}(B) \tag{25}$$

are equivalent.

To prove this we apply $\boldsymbol{F}^{-j}$ to both sides of Eq. (24) and consider that

$$\boldsymbol{F}^{-1}(\boldsymbol{F}(A)) \supseteq A \tag{26}$$

is valid, whether or not ***F*** is bijective. We can therefore see that Eq. (25) follows from Eq. (24). To show conversely that Eq. (24) follows from Eq. (25) we apply $\boldsymbol{F}^{j}$ to both sides of Eq. (25) and consider that

$$\boldsymbol{F}\left(\boldsymbol{F}^{-1}(B)\right) = B \tag{27}$$

is always valid. In practice, use of backward mapping is expensive if $\boldsymbol{F}(\boldsymbol{x})$ is not bijective, as in this case $\boldsymbol{F}^{-j}(\boldsymbol{x})$ is not single-valued.

POTENTIAL OF THE METHOD

The method of convex decomposition allows computer-aided application of the above stability criteria. If we work with criterion 1, the method represents a stability check that is *sufficient* and *nearly necessary* in the sense that the restriction *nearly* is only due to the limitation of computing time.

Moreover, since the method allows us to construct the domain trajectory (5) we can determine easily the minimum and maximum values of any variable $w = c_1 x_1 + c_2 x_2 + ... + c_n x_n$ referring to the continuum of all trajectories starting in *G*. For this we need investigate only the corners of all bounded convex polytopes that represent the domain trajectory $G \to F(G) \to ... \to \boldsymbol{F}^N(G)$, or we apply linear programming. This is of particular interest in deciding whether $G_{H,N}$-stability is valid under the additional condition that no trajectory starting in *G* makes even temporary excursions outside a given domain in the form of a cuboid C of the state space. If any components of $\boldsymbol{F}^N(G)$ that are obtained in the form of convex polytopes are not contained in *H*, the system is not $G_{H,N}$-stable. In this case we map those components of $\boldsymbol{F}^N(G)$ that are contained in *H* backward *N* times: That way we obtain the subset $G' \subset G$ with the property that the system is $G'_{H,N}$-stable.

REALIZATION OF THE KEY OPERATIONS

In this section we point out how the key operations of the method of convex decomposition can be performed with the aid of a computer for different representations of the polytopes.

REPRESENTATION OF CONVEX POLYTOPES

Corner-point representation

Let $\boldsymbol{x}_{c1}, \boldsymbol{x}_{c2}, ..., \boldsymbol{x}_{cp}$ be the corner points of a *bounded* convex polytope *P*. Then *P* can be represented in the form

$$P = \text{convex hull } \{\boldsymbol{x}_{c1}, \boldsymbol{x}_{c2}, ..., \boldsymbol{x}_{cp}\}. \tag{28}$$

Conversely, each set of arbitrarily chosen points $\boldsymbol{x}_j \in R^n$ supplies a corresponding bounded convex polytope by Eq. (28). This may degenerate (have a dimension $m < n$). The corner points of *P* are given by *some* or *all* of the chosen points $\boldsymbol{x}_{cj}$ (in the first case the remaining points $\boldsymbol{x}_{cj}$ represent interior points of *P* and can therefore be omitted without affecting *P*).

Inequality representation

Let

$$T_i = \{\boldsymbol{x} | \boldsymbol{h}_i^T \boldsymbol{x} = c_i\}, \quad i = 1,2,...,q \tag{29}$$

be the tangential hyperplanes of a convex polytope *P*, not necessarily bounded. Then *P* can be represented in the form:

$$P = \{\boldsymbol{x} | \boldsymbol{h}_i^T \boldsymbol{x} \le c_i, \quad i = 1,2,...,q\}. \tag{30}$$

Conversely, each set of arbitrarily chosen inequalities $\boldsymbol{h}_i^T \boldsymbol{x} \le c_i$ defines a convex polytope (30), not necessarily bounded. This may degenerate. Assume that the chosen set of *inequalities* contains the two inequalities $\boldsymbol{h}^T \boldsymbol{x} \le c$ and $-\boldsymbol{h}^T \boldsymbol{x} \le c$; then these two inequalities can be replaced by the *equation* $\boldsymbol{h}^T \boldsymbol{x} = c$. The tangential hyperplanes of *P* are induced by *some* or *all* of the chosen inequalities. (In the first case the remaining inequalities can be omitted without affecting *P*.) This shows that a polytope (30) that degenerates can be also represented in the more compact form:

$$P=\left\{\boldsymbol{x}\left|\begin{array}{l}\boldsymbol{h}_i^T\,\boldsymbol{x}\le c_i\,,\;\; i=1,2,...,q\\ \boldsymbol{k}_j^T\,\boldsymbol{x}=c_j\,,\;\; j=1,2,...,q'\end{array}\right.\right. \tag{31}$$

Linear algebra and the techniques of linear programming supply methods to transform the corner-point representation into the inequality representation and conversely to transform the inequality representation into the corner-point representation for bounded polytopes. Furthermore, there are methods to detect superfluous points and inequalities, respectively, in the representations. These may be removed to simplify the representation, although it is not necessary to do so.

AFFINE MAPPING OF CONVEX POLYTOPES

Let *P* be a convex polytope and let

$$\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{\ddot{O}}\,\boldsymbol{x}+\boldsymbol{d} \tag{32}$$

be a linear function. We state properties and methods for computer-aided construction of the point sets $\boldsymbol{F}(P)$ and $\boldsymbol{F}^{-1}(P)$, respectively. For this, we distinguish the *regular case* where $\boldsymbol{\ddot{O}}$ is regular and consequently the mapping $\boldsymbol{F}$ is bijective and the *singular case* where $\boldsymbol{\ddot{O}}$ is singular and $\boldsymbol{F}$ is therefore not bijective.

Properties of $\boldsymbol{F}(P)$ *and* $\boldsymbol{F}^{-1}(P)$

$\boldsymbol{F}(P)$ and $\boldsymbol{F}^{-1}(P)$ are convex polytopes. If $\boldsymbol{\ddot{O}}$ is regular and *P* is bounded and does not degenerate, the same properties are valid for $\boldsymbol{F}(P)$ and $\boldsymbol{F}^{-1}(P)$. If $\boldsymbol{\ddot{O}}$ is singular and *P* does not degenerate, $\boldsymbol{F}(P)$ degenerates and $\boldsymbol{F}^{-1}(P)$ is unbounded.

Construction of $\boldsymbol{F}(P)$ *and* $\boldsymbol{F}^{-1}(P)$ *in the regular case*

(i) If *P* is bounded and given in the corner-point representation then $\boldsymbol{F}(P)$ and $\boldsymbol{F}^{-1}(P)$ are obtained in the same representation by applying the affine mapping (32) and the inverse mapping

$$\boldsymbol{F}^{-1}(\boldsymbol{x})=\boldsymbol{\ddot{O}}^{-1}(\boldsymbol{x}-\boldsymbol{d})\,, \tag{33}$$

respectively, to all corners $\boldsymbol{x}_{cj}$ of *P*.

(ii) If *P* is given in the inequality representation, $\boldsymbol{F}(P)$ and $\boldsymbol{F}^{-1}(P)$ are given in the same representation by

$$\boldsymbol{F}(P)=\left\{\boldsymbol{x}\middle|\boldsymbol{h}_i^T\,(\boldsymbol{\ddot{O}}^{-1}(\boldsymbol{x}-\boldsymbol{d}))\le c_i\,,\;\; i=1,2,...,q\right\} \tag{34}$$

and

$$\boldsymbol{F}^{-1}(P)=\left\{\boldsymbol{x}\middle|\boldsymbol{h}_i^T\left(\boldsymbol{\ddot{O}x}+d\right)\le c_i\,,\;\; i=1,2,...,q\right\}, \tag{35}$$

respectively. Both representations can be transformed into the standard form (30).

Construction of $\boldsymbol{F}(P)$ and $\boldsymbol{F}^{-1}(P)$ *in the singular case*

(iii) Let *P* be bounded and given in the corner-point representation. Then $\boldsymbol{F}(P)$ is obtained in the same way as in the regular case. However, we may obtain superfluous points $\boldsymbol{F}(\boldsymbol{x}_{cj})$ that we may (but need not) omit. As $\boldsymbol{F}^{-1}(P)$ is not necessarily bounded we do not consider the construction of $\boldsymbol{F}^{-1}(P)$ in this case.

(iv) Let *P* be given in the inequality representation. Then $\boldsymbol{F}^{-1}(P)$ is obtained in the same way as in the regular case. To construct $\boldsymbol{F}(P)$, let $m < n$ be the rank of the $n \times n$-matrix $\ddot{\boldsymbol{O}}$. Then there exist $r = n - m$ linearly independent vectors $\boldsymbol{w}_j \neq 0$ for which $\ddot{\boldsymbol{O}}\boldsymbol{w}_j = 0$ is valid. They span an *r*-dimensional subspace S_r. Let S_m be the *m*-dimensional subspace that is orthogonal with respect to S_r. Then each vector $\boldsymbol{x}$ of dimension $n = r + m$ can be represented in the form of $\boldsymbol{x} = \boldsymbol{x}_r + \boldsymbol{x}_m$ with $\boldsymbol{x}_r \in S_r$ and $\boldsymbol{x}_m \in S_m$. Obviously $\ddot{\boldsymbol{O}}\boldsymbol{x} = \ddot{\boldsymbol{O}}\boldsymbol{x}_m$ and $\ddot{\boldsymbol{O}}\boldsymbol{x}_m \in S_m$ are valid for such a vector $\boldsymbol{x}$. This shows that $\boldsymbol{F}(P)$ is situated in the *m*-dimensional linear manifold $M = \{\boldsymbol{z} \mid \boldsymbol{z} = \boldsymbol{x} + \boldsymbol{d},\ \boldsymbol{x} \in S_m\}$. This manifold is defined by the *r* equations

$$\boldsymbol{w}_j^T \boldsymbol{z} = \boldsymbol{w}_j^T \boldsymbol{d}\ . \tag{36}$$

As $\ddot{\boldsymbol{O}}\boldsymbol{w}_j = 0$ holds, each vector $\boldsymbol{w}_j$ is orthogonal to all the row vectors of $\ddot{\boldsymbol{O}}$ and consequently linearly independent of them. Therefore, the $(n + r) \times r$ -matrix

$$\ddot{\boldsymbol{O}}_E = \begin{pmatrix} \ddot{\boldsymbol{O}} \\ \boldsymbol{W}^T \end{pmatrix} \tag{37}$$

where the vectors $\boldsymbol{w}_j^T$ form the rows of $\boldsymbol{W}^T$ has full rank *n* so that the matrix $\ddot{\boldsymbol{O}}_E^T \ddot{\boldsymbol{O}}_E$ is regular. Therefore, we can transform each of the inequalities (30) into the equivalent inequality

$$\boldsymbol{h}_i^T \left(\ddot{\boldsymbol{O}}_E^T \ddot{\boldsymbol{O}}_E\right)^{-1} \left(\ddot{\boldsymbol{O}}_E^T \ddot{\boldsymbol{O}}_E\right) \boldsymbol{x} = \boldsymbol{h}_i^T \left(\ddot{\boldsymbol{O}}_E^T \ddot{\boldsymbol{O}}_E\right)^{-1} \left(\ddot{\boldsymbol{O}}^T \ddot{\boldsymbol{O}} + \boldsymbol{W}\boldsymbol{W}^T\right) \boldsymbol{x} \leq c_i\ . \tag{38}$$

The requested point set $\boldsymbol{F}(P)$ is *defined* by

$$\boldsymbol{F}(P) = \left\{ \boldsymbol{z} \,\middle|\, \boldsymbol{z} = \ddot{\boldsymbol{O}}\boldsymbol{x} + \boldsymbol{d},\ \boldsymbol{h}_i^T \boldsymbol{x} \leq c_i \right\} \tag{39}$$

However, as $\ddot{\boldsymbol{O}}\boldsymbol{x} = 0$ holds for all $\boldsymbol{x} \in S_r$ we do not omit any points of $\boldsymbol{F}(P)$ if we evaluate Eq. (39) only for points $\boldsymbol{x} \in S_m$. This simplifies Eq. (38) to

$$\boldsymbol{h}_i^T \left(\ddot{\boldsymbol{O}}_E^T \ddot{\boldsymbol{O}}_E\right)^{-1} \ddot{\boldsymbol{O}}^T \ddot{\boldsymbol{O}}\boldsymbol{x} \leq c_i \tag{40}$$

as $\boldsymbol{w}^T \boldsymbol{x} = 0$ holds for $\boldsymbol{x} \in S_m$. Replacing $\ddot{\boldsymbol{O}}\boldsymbol{x}$ by $\boldsymbol{z} - \boldsymbol{d}$ we obtain the

forward mapping inequality

$$\boldsymbol{h}_i^T \left(\ddot{\boldsymbol{O}}_E^T \ddot{\boldsymbol{O}}_E\right)^{-1} \ddot{\boldsymbol{O}}^T \boldsymbol{z} \ \leq\ c_i + \boldsymbol{h}_i^T \left(\ddot{\boldsymbol{O}}_E^T \ddot{\boldsymbol{O}}_E\right)^{-1} \ddot{\boldsymbol{O}}^T \boldsymbol{d}\ . \tag{41}$$

We conclude that $\boldsymbol{F}(P)$ is determined by the r equalities (36) and the q inequalities (41), which can be written in the form $\tilde{\boldsymbol{h}}_i^T \boldsymbol{z} \le c_i$. Some of the inequalities (41) may be superfluous or vanish if $\tilde{\boldsymbol{h}}_i = 0$ holds. Moreover, Eq. (41) shows that the vector $\tilde{\boldsymbol{h}}_i$ is situated in S_m as it has the form of $\boldsymbol{\ddot{O}} \cdot \bar{\boldsymbol{h}}$ where $\bar{\boldsymbol{h}}$ is some vector.

Remark 5: The above construction of $\boldsymbol{F}(P)$ in the singular case based on the forward mapping inequality (41) differs from the method of Karweina (1998), which was based on a special representation of convex polytopes. It is still open as to which method is more efficient for applications. However, we consider our method as the natural generalization of Eq. (34).

Remark 6: Let the piecewise linear function (15) be not continuous. In this case, the domains Z_i of definition of the linear partial functions $\boldsymbol{F}_i(\boldsymbol{x})$ do not include all border points of Z_i. This would force us in the above construction of the set $\boldsymbol{F}^N(G)$ to decide each time which partial function $\boldsymbol{F}_i(\boldsymbol{x})$ is valid for the border points. We overcome this difficulty by extending the domains of definition for each $\boldsymbol{F}_i(\boldsymbol{x})$ so that all border points of Z_i are included. Thus, $\boldsymbol{F}(\boldsymbol{x})$ becomes *many-valued* at the border points. However, this does not affect the above arguments.

CONVEX DECOMPOSITION

We assume that the convex polytopes Z_i in Eq. (19) are given in the inequality representation

$$Z_i = \left\{ \boldsymbol{x} \middle| \boldsymbol{k}_i^T \le c_i ,\ i = 1,2,\ldots,s \right\}. \tag{42}$$

For a convex polytope P given in the inequality representation (30), the convex decomposition operation

$$P_i = P \cap Z_i \tag{43}$$

is performed simply by combining the inequalities of the representations (30) and (42). If a convex polytope P is given in the corner-point representation, we consider the half-spaces

$$H_j = \left\{ \boldsymbol{x} \middle| \boldsymbol{k}_j^T \boldsymbol{x} \le c_j \right\}. \tag{44}$$

Obviously

$$P_i = (\ldots((P \cap H_1) \cap H_2) \cap \ldots \cap H_s) \tag{45}$$

is valid. This reduces the operation (43) to a repeated application of the operation

$$P'' = P' \cap H_j \tag{46}$$

where P' is a polytope in corner-point representation (Figure 7, left). To perform operation (46) we investigate which of the corner points of P' are situated in H_j. If this applies for *all* of these corner points $P'' = P'$ is valid. Otherwise $P'' \subset P'$ holds and P'' has new corner points that are not corner points of P (Figure 7, right). These new corner points are obtained as intersection points of the hyperplane $\boldsymbol{k}_j^T \boldsymbol{x} = c_i$ and the *relevant* edges of P'. These are all edges that connect two corner points of P' situated inside and outside H_j, respectively. The detection of the relevant edges of P' is simple in the case $n = 2$. In the general case $n \ge 2$ we can obtain the relevant edges by attaching to each point of the corner-point representation (28) a list of those tangential hyperplanes that intersect this point [Kiendl 1987, Karweina 1989]. However, for greater dimensions n this is very expensive.

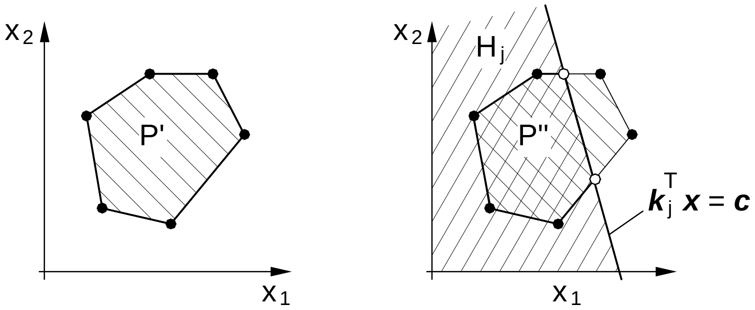

Figure 7: Convex polytopes P' (left) and P'' (right), where P'' is obtained by the intersection of P' and the half-space H_j.

CHECK OF INCLUSION CONDITION

For a convex polytope *P* given in the corner-point representation the inclusion condition

$$P \subseteq H, \tag{47}$$

where *H* is convex, is valid if and only if all corner points $\boldsymbol{x}_{cj}$ of *P* meet the condition $\boldsymbol{x}_{cj} \in H$. For convex polytopes *P* and *H* given in the inequality representation, we split condition (47) into several conditions of the form

$$P \subseteq H_j. \tag{48}$$

Here the H_j are the half spaces, the intersection of which defines *H*. Each condition (48) can be checked by linear programming.

We apply this to check conditions of the form (24) where *A* is a bounded convex polytope and $\boldsymbol{F}$ is piecewise linear. If we work with forward mapping, $\boldsymbol{F}^j(A)$ is obtained in the form of a family of bounded convex subpolytopes. Therefore, the above problem is reduced to the check of several inclusions of the form (47). If we want to check condition (25) instead of condition (24) we must take into account that $\boldsymbol{F}^{-1}(B)$ is usually not convex. However, as this set is obtained in the form of a family of convex polytopes P_j we can check condition (25) by investigating whether the union of all convex polytopes $P_j \cap A$ supplies A.

RELIEF OF BOTTLENECKS

The domain of application of the method of convex decomposition is limited by the required computing expense. This increases with the dimension *n* of the system and the number *N* of mapping steps. To extend the domain of application, several strategies have been developed to relieve the following bottlenecks:

(i) For a large number *N* of mapping steps the construction of $\boldsymbol{F}^N(G)$ in the form of a family of convex polytopes may lead to an exploding number of components (small convex polytopes).

(ii) The decomposition procedure may generate components that are more complicated (exhibit more surface elements or corner points) than the original components.

(iii) The decomposition procedure requires the determination of convex polytopes of the form $\boldsymbol{F}(P) \cap Z_i$. A straightforward way to do this is to first determine all possible sets $\boldsymbol{F}(P) \cap Z_i$ and afterwards to investigate which of these sets are empty. However, this is very time consuming if there are many zones Z_i.

STEPWISE MAPPING

To relieve bottleneck (i) we split the sequence of N mapping steps into several shorter sequences by using the following:

Stepwise mapping lemma

Let $G_0 = G, G_1, G_2, ..., G_r$ be a family of convex polytopes with the properties

$$\boldsymbol{F}^{M_i}(G_{i-1}) \subseteq G_i \cup G_{i+1} \cup ... \cup G_r, \; i = 1,2,\ldots,r \tag{49}$$

with $N = M_1 + M_2 + ... + M_r$. Then G can be decomposed into subdomains S_j in the form of convex polytopes so that for each subdomain S_j

$$\boldsymbol{F}^{p}(S_j) \subseteq G_r \tag{50}$$

is valid for some $p \leq N$. If, moreover, G_r is a Lyapunov domain then $\boldsymbol{F}^{N}(G) \subseteq G_r$ is valid.

The proof is obvious. For the practical application of this lemma we must construct the family $\{G_0, G_1, ...G_r\}$ that meets the conditions (49). To check such a condition we construct the set $\boldsymbol{F}^{M_i}(G_{i-1})$ by the above convex decomposition method in the form of a family of convex polytopes P_j. A straightforward check of condition (49) is difficult, as the set on the right hand side generally is not convex. However, a necessary condition for the validity of Eq. (49) is that each P_j is contained completely in one of the convex polytopes $G_i, G_{i+1}, ..., G_r$. This can be checked as described above. Setting $G_r = H$ and $G_r = K$, respectively, we can use this lemma to check the inclusion conditions of the above stability criteria 1 and 2.

Remark 7: In [Karweina 1989] bottleneck (i) is relieved by constructing a Lyapunov function for the system $\boldsymbol{x}_{k+1} = \boldsymbol{F}^{M}(\boldsymbol{x}_k)$. The above stepwise mapping lemma was inspired by this approach.

CONVEX ENCLOSURE

The concept of convex enclosure, introduced in [Scheel, Kiendl 1995] and further developed in [Scheel 1997, Kiendl 1997a, Knicker 1999], replaces the expensive exact determination of $\boldsymbol{F}(P)$ in the form of *many* polytopes by the simpler determination of an approximation in the form of *one single* convex polytope $\tilde{\boldsymbol{F}}(P)$ that encloses $\boldsymbol{F}(P)$:

$$\tilde{\boldsymbol{F}}(P) \supseteq \boldsymbol{F}(P). \tag{51}$$

As $\tilde{\boldsymbol{F}}(P)$ is determined by applying convex decomposition only for estimation purposes and not for mapping purposes, we can thereby eliminate the combinatorial explosion. The price for this is that the resulting stability check becomes more conservative, because $\tilde{\boldsymbol{F}}(P)$ will in general be bigger than $\boldsymbol{F}(P)$ (an undesired *blow-up effect*).

Although in this section we consider piecewise linear functions $\boldsymbol{F}(\boldsymbol{x})$ as before, we describe the method of convex enclosure more generally (because we will use it later also for other functions $\boldsymbol{F}(\boldsymbol{x})$).

Let $\boldsymbol{F}(\boldsymbol{x})$ be any function given in a bounded convex polytope P. We determine an *affine approximation*

$$\hat{\boldsymbol{F}}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} \tag{52}$$

of $\boldsymbol{F}(\boldsymbol{x})$, for instance, by minimizing the sum of the quadratic errors $\boldsymbol{E}^T(x_i)\boldsymbol{E}(\boldsymbol{x}_i)$ referring to all corner points $\boldsymbol{x}_i$ of *P*, where $\boldsymbol{E}(\boldsymbol{x})$ is the error function

$$\boldsymbol{E}(\boldsymbol{x}) = \boldsymbol{F}(\boldsymbol{x}) - \hat{\boldsymbol{F}}(\boldsymbol{x}). \tag{53}$$

We describe $\boldsymbol{F}(\boldsymbol{x})$ as *P-G-estimatable* if the error function can be estimated for all $\boldsymbol{x} \in P$ in the form of

$$\boldsymbol{E}(\boldsymbol{x}) \in Q(P), \tag{54}$$

where the *estimation polytope* $Q(P)$ is a bounded convex polytope.

Convex enclosure lemma

Let $\boldsymbol{F}(\boldsymbol{x})$ be *P*-*Q*-estimatable. Then any convex set $\tilde{\boldsymbol{F}}(P)$ containing all points

$$z = x_i + z_j \text{ (where } x_i \text{ is some corner of } \hat{\boldsymbol{F}}(P) \text{ and } z_j \text{ is some corner of } Q\text{)} \tag{55}$$

contains $\boldsymbol{F}(P)$.

The proof is straightforward. In order to obtain a *simple* set $\tilde{\boldsymbol{F}}(P)$ we construct it so that its surface elements are *parallel* to those of *P* (Figure 8).

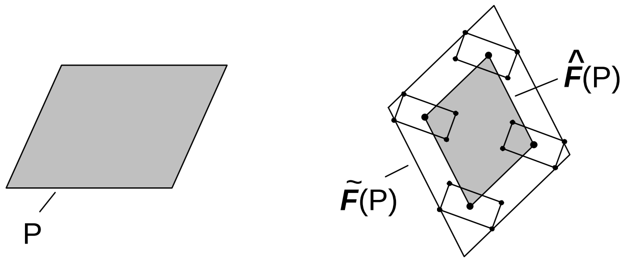

Figure 8: Construction of the convex polytope $\tilde{\boldsymbol{F}}(P)$ that encloses $\boldsymbol{F}(P)$.

If $\boldsymbol{F}(\boldsymbol{x})$ is piecewise linear according to Eq. (15), the same holds for the error function $\boldsymbol{E}(\boldsymbol{x})$. A *uniformly* linear function is known to take its extremal values in the corners of a polytope. Consequently, we obtain the extremal values of the components of $\boldsymbol{E}(\boldsymbol{x})$ by investigating the corners of all polytopes $P \cap Z_i$. These values supply the estimation polytope $Q(P)$ here in the form of an axes-parallel cuboid. If $\boldsymbol{F}(\boldsymbol{x})$ comes from a linear plant (2) combined with a piecewise linear control law $u(\boldsymbol{x})$, we obtain $Q(P)$ in the form of the one-dimensional point set

$$Q(P) = \{\boldsymbol{x} | u\,\boldsymbol{h},\ u_{\min} \le u \le u_{\max}\}. \tag{56}$$

Consequently, if $\boldsymbol{F}(\boldsymbol{x})$ is piecewise linear we may replace each determination of a set $\boldsymbol{F}(P)$ that requires convex decomposition by the determination of $\tilde{\boldsymbol{F}}(P)$ without using convex decomposition for mapping purpurses. Thus, we can oppose the tendency toward combinatorial explosion, with the price that the stability check becomes more conservative. The latter is often acceptable, especially if the system performance is very stable. Consequently, if the system equation (1) is to a good approximation given by a linear equation $\boldsymbol{x}_{k+1} = \boldsymbol{\ddot{O}}\, \boldsymbol{x}_k$ and $\boldsymbol{\ddot{O}}$ has eigenvalues m with $|m| << 1$ we will probably succeed in establishing $G_{H,N}$-stability without applying convex decomposition in any step, i.e., just by determining the polytopes $\tilde{\boldsymbol{F}}(G), \tilde{\boldsymbol{F}}^2(G), \ldots, \tilde{\boldsymbol{F}}^N(G)$.

If that approach does not succeed, we can combine the advantages of convex decomposition and the enclosure technique by applying convex decomposition only for those polytopes P for which the error $\boldsymbol{E}^T(\boldsymbol{x})\boldsymbol{E}(\boldsymbol{x})$ is large. (By doing so we can decompose *P*, making use of the piecewise structure of $\boldsymbol{F}(\boldsymbol{x})$, which leads to a small error but to more or less complicated subpolytopes. Alternatively, we can decompose the relevant sets into cuboids, which simplifies the decomposition, but leads to a greater error.) For all other polytopes we determine $\tilde{\boldsymbol{F}}(P)$.

MORE GENERAL FUNCTIONS $\boldsymbol{F}(\boldsymbol{x})$

The method of convex decomposition, including the enclosure technique, can also be used for the stability analysis of more general systems. This is valid, for instance, for systems (1) where the function $\boldsymbol{F}(\boldsymbol{x})$ is *time-varying* or *not exactly known* but is *P*-*Q*-estimatable in the above sense.

Moreover, we consider systems (1) where each component of $\boldsymbol{F}(\boldsymbol{x})$ is *multilinear*. For $n = 2$ the function

$$f(x_1, x_2) = c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1 x_2 \tag{57}$$

is multilinear. For such functions the resulting error function (53) is also multilinear in each component. Consequently, each component takes its minimum and maximum values with respect to any axes-parallel cuboid in the corners of this cuboid [Kiendl, Michalske 1992]. Therefore, we can estimate the extremal values with respect to any bounded convex polytope *P*: for this we enclose *P* by an axes-parallel polytope. The same applies if $\boldsymbol{F}(\boldsymbol{x})$ consists of *piecewise* multilinear functions $\boldsymbol{F}_j(\boldsymbol{x})$ where the domains Z_i of definition of the partial functions are axes-parallel cuboids (Figure 9). The same technique can be applied if a linear plant (2) is combined with a piecewise multilinear control law.

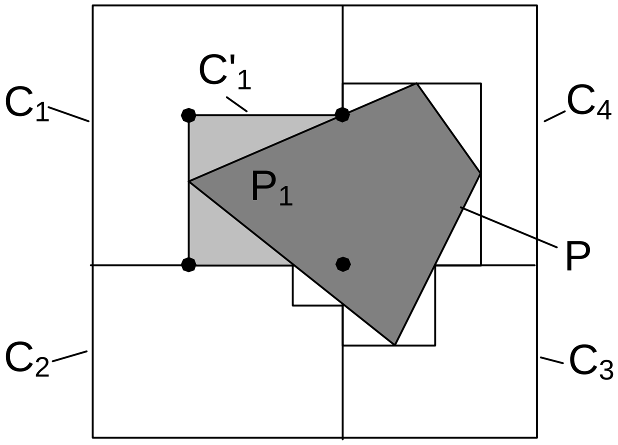

Figure 9: Estimation of a piecewise multilinear function $f(\boldsymbol{x})$ that is uniformly multilinear in each axes-parallel box C_i. The minimum and maximum value of $f(\boldsymbol{x})$ referring to the corners of C_1' supply a lower and upper bound, respectively, to estimate $f(\boldsymbol{x})$ with respect to $P_1 = P \cap C_1$, and so on.

Moreover, we can also apply convex enclosure to more general functions. Let the system equation (1), for instance, include the non-linearity $\dot{x}_2 = \sin x_1$ where the interesting domain is $-p < x_1 < +p$. We can approximate the function $\sin x_1$ by a piecewise linear function [Knicker 1999] and estimate the resulting error function. Consequently, we can investigate *strictly* whether $G_{H,N}$-stability is valid by applying convex enclosure. The same holds if we combine a linear plant with a piecewise linear or multilinear control law where such a non-linearity follows the controller.

INTELLIGENT DETERMINATION OF THE SETS $\boldsymbol{F}(P) \cap Z_i$

The main problem in the determination of the sets $\boldsymbol{F}(P) \cap Z_i$ is that most of these sets are generally empty. Therefore, criteria that allow us to decide *in advance* which of these sets are empty and consequently need not be determined are useful.

In [Karweina 1989] such criteria have been developed for the special case that the zones Z_i are generated by the intersection of several families of hyperplanes, where all hyperplanes of the same family are parallel to each other. In [Rumpf 1997] such criteria have been developed for more general structures of the zones Z_i such as occur in neural networks controllers.

ADVANCED FUZZY SYSTEMS

The method of convex decomposition presented above can be applied to the stability analysis of both conventional non-linear and fuzzy control systems. In the following, we point out that the preconditions for use of this method can be met in an especially simple way by using fuzzy systems that are based on the recently developed TOR-defuzzification. For this we consider a fuzzy controller with *n* input variables $x_1, x_2, ..., x_n$ and one output variable *u*. We provide rules R_k in the form:

$$\text{IF} < \text{premise } k > \text{THEN} < u = L_k > \tag{58}$$

where each linguistic output value L_k is modelled by a unit singleton at position u_k. We consider premises in the form of

$$(x_i = L) \wedge \ldots \wedge (x_j = L'), \quad i \neq j \tag{59}$$

consisting of $m \leq n$ terms. We model the linguistic input values such as *L* and L' by piecewise linear membership functions, and process the rules using SUM-PROD-inference. Consequently, the degree m_k of activation of each rule R_k depends in a *piecewise multilinear* fashion on the variables that enter the premise of the rule. If each rule considers *only one variable*, the dependency is *piecewise linear*. The domains where the dependency $m_k(x_1,..., x_n)$ is *uniformly* multilinear and linear, respectively, are axes-parallel cuboids (Figure 10).

If we determine the output value *u* by conventional COG-defuzzification

$$u = \frac{\sum_i m_i u_i}{\sum_i m_i}, \tag{60}$$

the characteristic surface $u = K(\boldsymbol{x})$ of the resulting fuzzy controller is piecewise multilinear and piecewise affine, respectively, on condition that we ensure that

$$\sum_i m_i = c \tag{61}$$

is always valid. This restricts our freedom in the choice of rules and membership functions considerably, especially if we design the controller stepwise by adding one rule after the other. However, if we apply the TOR-defuzzification introduced in [Kiendl 1996, 1997b] and defined by

$$u = \sum_i m_i u_i, \tag{62}$$

the characteristic surface $u = \boldsymbol{K}(\boldsymbol{x})$ is piecewise multilinear and affine, respectively, *in any case*. Consequently, if we combine this control law with a linear plant we get a system (1) that fits perfectly into the framework of the above convex decomposition and convex enclosure method to establish $G_{H,N}$-stability [Kiendl 1999].

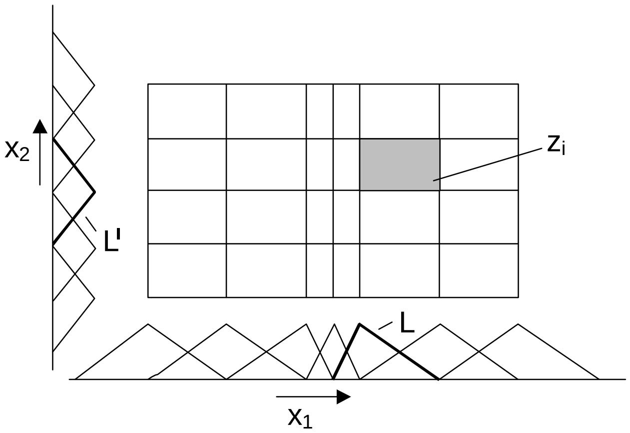

Figure 10: Piecewise linear input membership functions and resulting zones Z_i where the degree of activation of each rule R_k is a piecewise multilinear/linear function $m_k(x_1,...,x_n)$.

In the piecewise linear case we do not need the convex enclosure technique that makes the stability check more or less conservative. However, the disadvantage is that the structure of the resulting characteristic surface $u = \boldsymbol{K}(\boldsymbol{x})$ is not very flexible. In particular, it always shows an axes-parallel structure, which may be unfavourable. To overcome this problem, two approaches are proposed. Firstly, we can transform the original input variables x_i by a linear transformation

$$\tilde{\boldsymbol{x}} = \boldsymbol{T}\boldsymbol{x} \tag{63}$$

into new variables $\tilde{x}_i$ that we use as input variables for the fuzzy system. Secondly, we can provide a cascaded structure consisting of several fuzzy systems each of which supplies a piecewise linear characteristic surface (Figure 11).

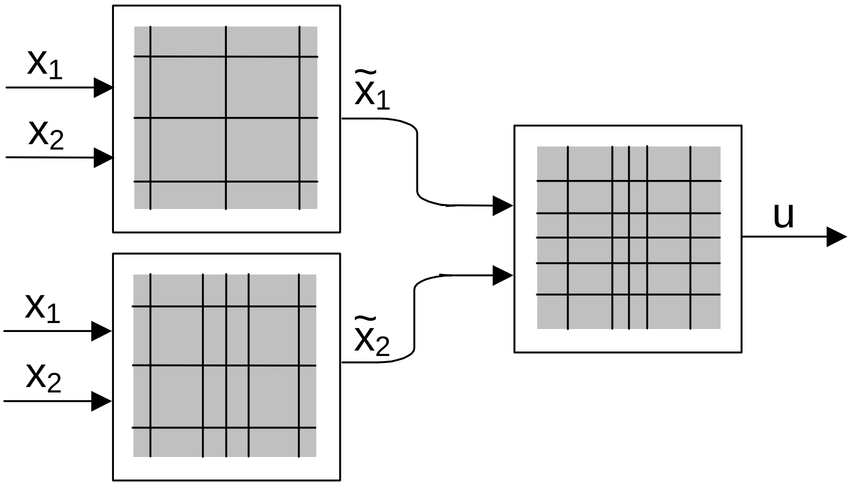

Figure 11: Cascaded fuzzy system consisting of three subsystems. Each subsystem has a piecewise linear characteristic surface that shows an axes-parallel structure.

In this case we can obtain domains Z_i – where the overall characteristic surface is uniformly affine – in the form of convex polytopes that do not have an axes-parallel orientation. Consequently, this approach gives us much more flexibility. Moreover, as in each partial fuzzy system the corresponding zones are axes-parallel, we can proceed as in [Karweina 1989] to suppress the determination of empty sets $\boldsymbol{F}(P) \cap Z_i$ and so to relieve bottleneck (iii). Moreover, this transparent structure of the domains Z_i simplifies the determination and book-keeping of the polytopes that occur.

Remark 8: An essential advantage of the TOR-defuzzification compared with COG-defuzzification is that it allows exploitation of qualitative knowledge that is based on *incomplete* information. Such knowledge should be expressed by rules that supply *incremental* or *decremental* recommendations rather than *absolute* recommendations [Kiendl 1997b, 1999]. Here we exploit the additional advantage of the TOR-defuzzification in that it simplifies the mathematical structure of the resulting characteristic surface.

EXAMPLE

For illustration we consider the well known problem of position control of a mass *m* by a force *F*: The plant is given by

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{1}{m} F \end{aligned} \tag{64}$$

where x_1 and x_2 are the position and the velocity, respectively, of the mass. We want to design a discrete time controller that produces a force $F(x_1, x_2)$ with $|F| \le 1$ so that each initial state (x_1, x_2) situated in the initial domain $G = \{\boldsymbol{x} \,|\, |\boldsymbol{x}_1| \le 6,\ |\boldsymbol{x}_1| \le 6\}$ leads to a trajectory that tends to the desired state $(x_1 = 0,\ x_2 = 0)$. We require the system to be $G_{H,N}$-stable, where *H* is a sufficiently small cuboid containing the state $(x_1 = 0,\ x_2 = 0)$. To design such a controller we provide a fuzzy system of the type shown in Figure 11, consisting of three subsystems followed by a (not depicted) saturation element. We provide rules so that each rule considers only one of the input variables x_1, x_2, $\tilde{x}_1$ and $\tilde{x}_2$, respectively. The linguistic output values are modelled by singletons. The linguistic input variables are modelled by piecewise linear membership functions of the type

$$m_j(x_i) = \begin{cases} m_j x_i & \text{if} \quad 0 \le m_j x_i \le 1 \\ 0 & \text{if} \quad m_j x_i < 0 \\ 1 & \text{if} \quad m_j x_i > 0 \end{cases} \tag{65}$$

Obviously condition (61) is not met. Nevertheless – as we apply SUM-PROD-inference and TOR-defuzzification –each of the three fuzzy subsystems has in every case a piecewise linear characteristic surface with an axes-parallel structure. We tune the parameters of the membership functions interactively by simulation of the control system. The subsystems thus obtained have uniformly linear characteristics in axes-parallel boxes as depicted in Figure 11. The overall characteristic surface of the designed system is also piecewise linear. However, we now find that the zones z_i where the characteristic surface is uniformly linear do not necessarily have an axes-parallel orientation (Figure 12). This shows the flexibility of the proposed cascaded fuzzy structure.

The control law obtained fits perfectly into the framework of the method of convex decomposition described above. Consequently, we can determine the resulting domain trajectory exactly and investigate $G_{H,N}$-stability. Figure 13 shows the square domain *G* and some of the sets $F^i(G)$, each set consisting of a family of convex polytopes (the sampling time is 0.4). We see that the method of convex decomposition allows the construction and detailed analysis of the domain trajectory. In particular, we see that $F^i(G)$ tends with increasing *i* to the centre of *G*. In [Knicker, Krause 1999] the process of establishing $G_{H,N}$-stability is described in detail for this system.

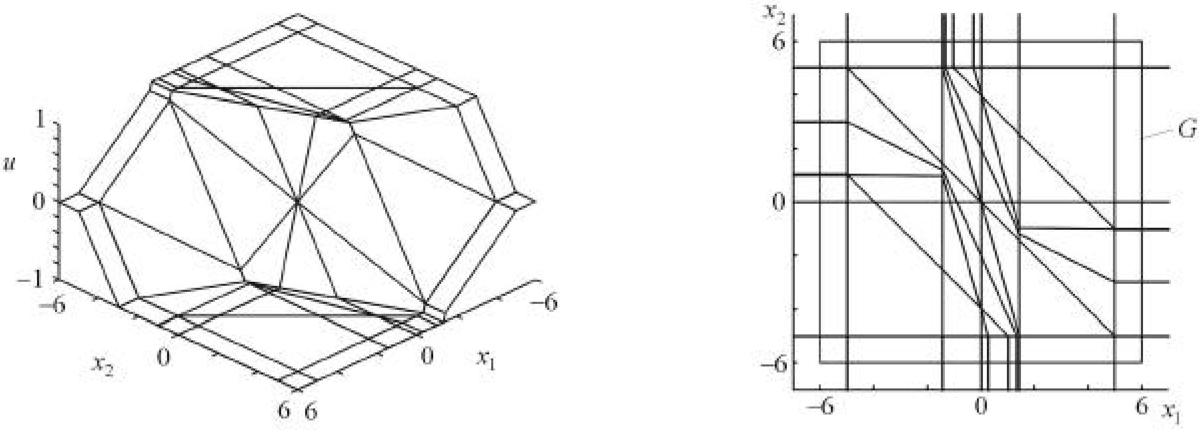

Figure 12: Characteristic surface of the fuzzy controller designed here (left). Zones where the characteristic surface is uniformly linear (right).

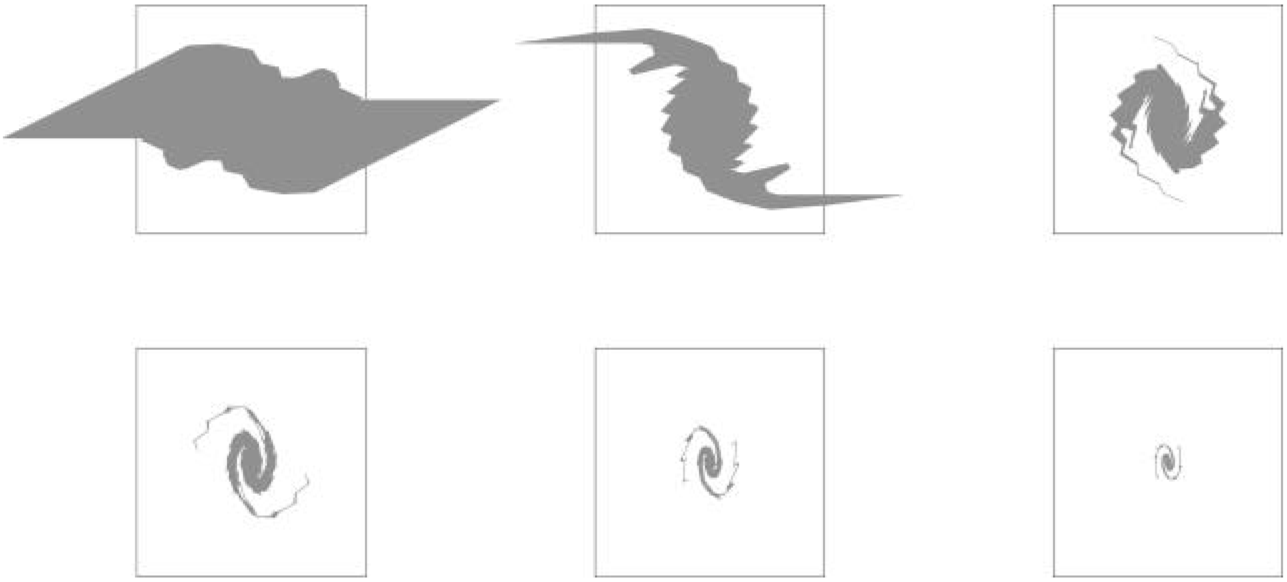

Figure 13: Domain trajectory of the control system designed here. The figure shows the square initial domain G and the sets $F^i(G)$ for $i = 5, 10, 15, 20, 25$ and 30, each consisting of a family of convex polytopes.

CONCLUSIONS

Stability analysis is an essential problem in the design of control systems. Existing analytical stability criteria often fail if the control system is highly non-linear. However, such systems are of great practical importance, either because the plant is highly non-linear or because high-quality performance often requires the use of a highly non-linear controller. The method of convex decomposition, together with the concept of $G_{H,N}$-stability, allows a strict computer-aided check of stability for a broad class of discrete-time non-linear control systems.

This paper presents several strategies to counteract the inherent tendency towards combinatorial explosion of the method. Consequently, more complex systems can now be analysed. Moreover, more general systems, such as piecewise multilinear, time variant or uncertain systems, can be investigated by using the convex enclosure technique.

Fuzzy controllers are an important class of non-linear controllers. As they are very flexible, they allow the design of highly non-linear control laws that provide excellent performance. By applying the recently proposed TOR-defuzzification, fuzzy controllers are obtained that fit perfectly into the framework of the stability analysis method presented here. Consequently – provided that we have a mathematical plant model – we can exploit the advantages of the fuzzy approach and yet establish stability strictly. This makes fuzzy controllers serious competitors in the field of applications where usually conventional controllers are preferred.

REFERENCES

Johansson, Mikael, 1999, “Piecewise linear Control Systems”, Ph. D. thesis, Department of Automatic Control, Lund, Institute of Technology, Sweden.

Karweina, Dieter, 1989, “Rechnergestützte Stabilitätsanalyse für nichtlineare zeitdiskrete Regelungssysteme, basierend auf der Methode der konvexen Zerlegung”, Fortschrittsbericht VDI Reihe 8 Nr. 181, VDI Verlag Düsseldorf, Germany.

Kiendl, Harro, 1987, “Robustheitsanalyse von Regelungssystemen mit der Methode der konvexen Zerlegung”, Automatisierungstechnik 35, Germany, pp. 192-202.

Kiendl, Harro; Michalske Andreas, 1992, “Robustness Analysis of Linear Control Systems with Uncertain Parameters by the Method of Convex Decomposition”.

Kiendl, Harro; Adamy, Jürgen; Stelzner, Peter, 1992, “Vector Norms as Lyapunov Functions for Linear Systems”, IEEE Transactions on Automatic Control, Vol. 37, pp. 839-842.

Kiendl, Harro; Rüger, Johannes-Jörg, 1995, “Stability analysis of fuzzy control systems using facet functions”, Fuzzy Sets and Systems, 70, pp. 275-285.

Kiendl, Harro, 1996, “Verfahren zu Defuzzifizierung für signalverarbeitende Fuzzy-Baueinheiten”, Patent DE 19640635.

Kiendl, Harro, 1997a, “Stability Analysis for Advanced Fuzzy Systems”, Oldenbourg Verlag, Munich, Germany.

Kiendl, Harro, 1997b, “Non-translation-invariant Defuzzification”, Sixth IEEE International Conference on Fuzzy Systems, FUZZ-IEEE`97, Barcelona, Spain, pp. 737-742.

Kiendl, Harro, 1999, “Design of Advanced Fuzzy Systems”, TOOLMET`99 Symposium - Tool Environments and Development Methods for Intelligent Systems, Oulu, Finland, pp. 57-76.

Knicker, Rainer, 1999, “Stability Analysis of Fuzzy and Other Nonlinear Systems based on the Method of Convex Decomposition”, 6th Fuzzy Days, Springer Verlag, Dortmund, Germany, pp. 391-400.

Knicker, Rainer; Krause, Peter, 1999, “The Convex Decomposition Stability Analysis Method with Gradual Mapping”, 7th EUFIT, Aachen, Germany.

Rumpf, Oliver, 1997, “Stabilitätsanalyse zeitdiskreter nichtlinearer dynamischer Systeme, auf der Basis der konvexen Zerlegung mit paralleler Implementierung”, Fortschrittsbericht VDI Reihe 8 Nr. 651, VDI Verlag Düsseldorf, Germany.

Schäfers, Elmar, 1999, “Dynamische Fuzzy-Systeme zur qualitativen Prozeßmodellierung: Eine neue Systemtheorie”, Fortschrittsbericht VDI Reihe 8 Nr. 745, VDI Verlag Düsseldorf, Germany.

Scheel, Thomas; Kiendl, Harro, 1995, “Stability Analysis of Fuzzy and other nonlinear Systems using Integral Lyapunov Functions ”, 3rd EUFIT, Aachen, Germany, pp. 765-770.

Scheel, Thomas, 1997, “Integrale Ljapunov-Funktionen zur rechnergestützten Stabilitätsanalyse nichtlinearer zeitdiskreter Regelungssysteme”, Fortschrittsbericht VDI Reihe 8 Nr. 618, VDI Verlag Düsseldorf, Germany.