Implementation of Discrete-Time Fractional-Order Controllers based on LS Approximations

Ramiro S. Barbosa[1], **J. A. Tenreiro Machado**[2]

Department of Electrotechnical Engineering
Institute of Engineering of Porto
4200-072 Porto, Portugal
[1]e-mail: rsb@isep.ipp.pt, [2]e-mail: jtm@isep.ipp.pt

Abstract: In this paper we develop rational discrete-time approximations (IIR filters) to continuous fractional-order integrators and differentiators of type $s\alpha$, $\alpha \in \Re$. For that, it is proposed the adoption of the techniques of Padé, Prony and Shanks usually applied in the signal modelling of deterministic signals. These methods yield suboptimal solutions to the problem which only requires finding the solution of a set of linear equations. The results reveal that this approach gives similar or superior approximations in comparison with other widely used methods. Their effectiveness is illustrated, both in the time and frequency domains, through several examples.

Keywords: IIR filters, rational approximations, digital differentiators, digital integrators, filter design, least-squares, discretization.

1 Introduction

The area of fractional calculus (FC) emerged at the same time as the classical differential calculus and deals with derivatives and integrals to an arbitrary order (real or even complex order) ([1], [2], [3], [4]). However, its inherent complexity postponed the application of the associated concepts. Nowadays, the FC theory is applied in almost all the areas of science and engineering being recognized its ability to yield a superior modelling and control in many dynamical systems ([1], [4], [5], [6], [7], [25]).

In the literature we can find several different definitions for the fractional integration and differentiation of arbitrary order ([1], [2], [4]). One of the most well-known definitions is given by the Grünwald-Letnikov approach ($\alpha \in \Re$):

$$ {}_{a}D_{t}^{\alpha} f(t) = \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{k=0}^{\left[\frac{t-a}{h}\right]} (-1)^{k} \binom{\alpha}{k} f(t-kh) \tag{1} $$

$$\binom{\alpha}{k} = \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)} \tag{2}$$

where $f(t)$ is the applied function, $\Gamma(x)$ is the Gamma function, h is the time increment and $[x]$ means the integer part of x. An important property revealed by equation (1) is that while integer-order operators imply finite series, the fractional-order counterparts are defined by infinite series. This means that integer operators are local operators in opposition with the fractional operators that have, implicitly, a 'memory' of all past events.

From a control and signal processing perspective, the Grünwald-Letnikov approach seems to be the most useful and intuitive, particularly for a discrete-time implementation ([4], [8]). Moreover, in the analysis and design of control systems we usually adopt the Laplace transform (L) method. The definition of the fractional-order operator (1) in the Laplace s-domain, under null initial conditions, is given by the relation ($\alpha \in \Re$):

$$L\left\{ {}_aD_t^{\alpha}\left[f(t)\right]\right\} = s^{\alpha}F(s) \tag{3}$$

where $F(s)=L\{f(t)\}$. Note that expression (3) is a direct generalization of the classical integer-order scheme with the multiplication of the signal transform $F(s)$ by the Laplace s-variable raised to a noninteger value α.

Presently, there are several fractional-order control (FOC) strategies where the fractional-order differentiator and/or integrator, s^{α} ($\alpha \in \Re$), represents its fundamental element. For example, the CRONE[1] controller ([6], [7]) and the fractional PID ($PI^{\lambda}D^{\mu}$) controller ([4], [12]) possess a superior performance comparatively with the classical PID controller, particularly when used for the control of fractional-order systems. In general, we may say that the FOC strategies are more flexible and give the possibility of adjusting more carefully the dynamical properties of a control system.

In this paper, we apply the techniques of Padé, Prony and Shanks for obtaining digital rational approximations (IIR filters) to continuous fractional-order integrators and differentiators of type s^{α} ($\alpha \in \Re$). The resulting approximations are suitable for a digital implementation of a FOC system. The determination process can be synthesized in the following steps:

1 Discretize the fractional-order operator s^{α} using a suitable generating function $H^{\alpha}(z^{-1})$;

2 Obtain the impulse response sequence $h^{\alpha}(k)$, of the fractional discrete equivalent, by performing a power series expansion (PSE) (or Taylor series) over $H^{\alpha}(z^{-1})$;

[1] French abbreviation for *Commande Robuste d'Ordre Non Entier.*

3 Apply the signal modeling techniques of Padé, Prony or Shanks to $h^{\alpha}(k)$ in order to get the desired IIR filter approximation.

The proposed method represents an alternative choice to other existing approaches, namely the widely used continued fraction expansion (CFE) method.

Bearing these ideas in mind, the paper is organized as follows. Section 2 presents some discretization schemes for continuous fractional-order integrators and differentiators, while section 3 derives their impulse response sequences. Section 4 develops the signal modeling techniques of Padé, Prony and Shanks for the design of IIR filters approximations to continuous fractional-order operators. Section 5 presents several illustrative examples showing the effectiveness of the new technique. Finally, the main conclusions are drawn.

2 Discretization of Continuous Integrators and Differentiators of Fractional-Order

In general, the discretization of the continuous fractional-order operator s^{α} ($\alpha \in \Re$) can be expressed by the so-called generating function $s = \omega(z^{-1})$ ([9], [10]). In these $s \rightarrow z$ conversion schemes (also called analog to digital open-loop design methods) the most often used are the Euler (or first backward difference), the Tustin (or bilinear) and the Simpson schemes (see [8]). Recently, new discretization formulae appeared that are weighted interpolations between the Euler-Tustin ([13], [14]) or the Tustin-Simpson ([15], [16]) schemes. For example, the interpolation of 3/4 of the Euler operator with 1/4 of the Tustin operator yields the Al-Alaoui operator (see [14]). This scheme exhibits a much better magnitude fit than the Tustin operator in high frequency range. Table 1 lists the Euler, Tustin and Al-Alaoui operators that will be used in this study.

As can be seen in Table 1, the fractional-order conversion schemes lead to non-rational z-formulae. Therefore, in order to get rational expressions we may get its power series expansion (PSE) (Taylor series) and obtain the final approximation as a truncated z-polynomial function (FIR filter) ([8], [9]). For example, using the Euler operator, $H(z^{-1}) = (1-z^{-1})/T$, and performing a PSE of $[(1-z^{-1})/T]^{\alpha}$, it yields the discretization formula corresponding to the Grünwald-Letnikov definition (1). Another possible way is to obtain a discrete transfer function in the form of rational function (i.e., as the ratio of two polynomials) (IIR filter) through the application of the continued fraction expansion (CFE) method ([9], [10], [11]).

It is well known that rational approximations frequently converge faster than polynomial approximations and have a wider domain of convergence in the complex domain. Hence, in the work that follows, we develop only IIR-type approximations of continuous fractional-order integrators and differentiators, which make them suited for *z-transform* analysis and digital implementation. For that, we propose the use of the least-squares (*LS*) based methods usually applied in the modeling of deterministic signals (see section 4).

3 Impulse Response of Discretized Integrators and Differentiators of Fractional-Order

This section derives the impulse response sequences $h^{\alpha}(k)$ of the discretization schemes listed in Table 1. It is assumed that $h^{\alpha}(k) = 0$ for $k < 0$, corresponding to a causal system.

Expanding the Euler operator $H^{\alpha}_{E}(z^{-1})$ into a power series in z^{-1}, we have:

$$\begin{aligned} H_E^{\alpha}\left(z^{-1}\right) &= \left[\frac{1}{T}\left(1-z^{-1}\right)\right]^{\alpha} = \\ &= \left(\frac{1}{T}\right)^{\alpha}\sum_{k=0}^{\infty}(-1)^k\binom{\alpha}{k}z^{-k} = \\ &= h_E^{\alpha}(0)+h_E^{\alpha}(1)z^{-1}+\ldots=\sum_{k=0}^{\infty}h_E^{\alpha}(k)z^{-k} \end{aligned} \tag{4}$$

Table 1

Discretization schemes

Method	$s \to z$ conversion
Euler Grünwald-Letnikov	$s^{\alpha} \approx \left(\frac{1-z^{-1}}{T}\right)^{\alpha}$
Tustin	$s^{\alpha} \approx \left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right)^{\alpha}$
Al-Alaoui	$s^{\alpha} \approx \left(\frac{8}{7T}\frac{1-z^{-1}}{1+z^{-1}/7}\right)^{\alpha}$

where the impulse response sequence $h^{\alpha}_{E}(k)$ is then given by:

$$h_E^{\alpha}(k)=\left(\frac{1}{T}\right)^{\alpha}(-1)^k\binom{\alpha}{k}, \quad k \geq 0 \tag{5}$$

By taking the power series expansion (PSE) of the Tustin and Al-Alaoui operators, $H^{\alpha}{}_{T}(z^{-1})$ and $H^{\alpha}{}_{A}(z^{-1})$, respectively, it yields:

$$H_T^{\alpha}\left(z^{-1}\right)=\left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right)^{\alpha}=$$

$$=\left(\frac{2}{T}\right)^{\alpha}\sum_{k=0}^{\infty}\left[\sum_{j=0}^{k}(-1)^{j}\binom{\alpha}{j}\binom{-\alpha}{k-j}\right]z^{-k}= \tag{6}$$

$$=h_T^{\alpha}(0)+h_T^{\alpha}(1)z^{-1}+\ldots=\sum_{k=0}^{\infty}h_T^{\alpha}(k)z^{-k}$$

$$H_A^{\alpha}\left(z^{-1}\right)=\left(\frac{8}{7T}\frac{1-z^{-1}}{1+z^{-1}/7}\right)^{\alpha}=$$

$$=\left(\frac{8}{7T}\right)^{\alpha}\sum_{k=0}^{\infty}\left[\sum_{j=0}^{k}(-1)^{j}\left(\frac{1}{7}\right)^{k-j}\binom{\alpha}{j}\binom{-\alpha}{k-j}\right]z^{-k}= \tag{7}$$

$$=h_A^{\alpha}(0)+h_A^{\alpha}(1)z^{-1}+\ldots=\sum_{k=0}^{\infty}h_A^{\alpha}(k)z^{-k}$$

The impulse responses sequences, $h^{\alpha}{}_{T}(k)$ and $h^{\alpha}{}_{A}(k)$, are correspondingly given as $(k \geq 0)$:

$$h_T^{\alpha}(k)=\left(\frac{2}{T}\right)^{\alpha}\sum_{j=0}^{k}(-1)^{j}\binom{\alpha}{j}\binom{-\alpha}{k-j} \tag{8}$$

$$h_A^{\alpha}(k)=\left(\frac{8}{7T}\right)^{\alpha}\sum_{j=0}^{k}(-1)^{j}\left(\frac{1}{7}\right)^{k-j}\binom{\alpha}{j}\binom{-\alpha}{k-j} \tag{9}$$

Note that the PSE method leads to impulse response sequences of infinite length. For a practically realizable form these sequences must be truncated yielding approximations in the form of FIR (finite impulse response) filters. The *s*→*z* conversion schemes just described (Euler, Tustin and Al-Alaoui) are special cases of a more general discretization formula called T-integrator (see [21]).

4 Design of IIR Filters based on LS Method

Consider that the impulse response sequence $h^{\alpha}(k)$ is specified for $k \geq 0$. The IIR filter $H(z^{-1})$, to be designed, has the form:

$$H(z)=\frac{B(z)}{A(z)}=\frac{b_0+b_1 z^{-1}+...+b_m z^{-m}}{1+a_1 z^{-1}+...+a_n z^{-n}}=\sum_{k=0}^{\infty} h(k) z^{-k} \tag{10}$$

where $h(k)$ is its impulse response and $m \leq n$.

The IIR filter approximation (10) has m+n+1 parameters, namely the coefficients a_k (k = 1, 2, …, n) and b_k (k = 0, 1,…, m), which can be selected to minimize some error criterion. Usually, we adopt the least-squares (*LS*) method in order to minimize the error $e_{LS}(k) = h^{\alpha}(k) - h(k)$, as shown in Figure 1:

$$E_{LS}=\sum_{k=0}^{N-1}\left[e_{LS}(k)\right]^2=\sum_{k=0}^{N-1}\left[h^{\alpha}(k)-h(k)\right]^2 \tag{11}$$

where N denotes the number of samples used in the summation. However, the *LS* approach leads to a nonlinear problem for the model parameters ($a_{k,}$ b_k), which requires the solution of a set of nonlinear equations and, for that reason, it is often avoided.

If we rewrite (10) as $H(z)A(z) = B(z)$, and assuming that $h^{\alpha}(k)$ is given approximately by the impulse response of $H(z)$, one can write the corresponding time-domain equation as (note that the left-hand sided corresponds to a convolution):

$$h^{\alpha}(k)+\sum_{l=1}^{n} a_l h^{\alpha}(k-l)=\begin{cases} b_k, & k=0,1,...,m \\ 0, & k>m \end{cases} \tag{12}$$

This gives a set of linear equations, which can be used in different ways to solve for the coefficients a_k and b_k ([17], [18], [19], [20], [21]). Our objective is to use simple (indirect) methods that can handle more easily the determination of the IIR filter parameters. In this perspective, this study considers the application of three linear suboptimal solutions: the Padé approximation, the Prony's method and the Shanks' method [18].

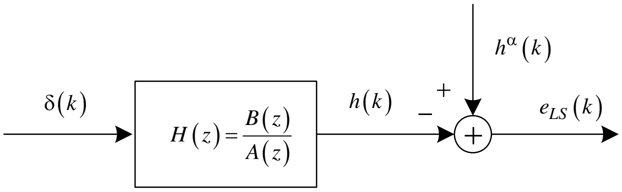


Figure 1
Least-squares method

4.1 Padé Approximation

The Padé approximation yields an IIR filter that fits exactly $h^{\alpha}(k)$ during the first m+n+1 values of k. Then, equation (12) becomes:

$$h^{\alpha}(k)+\sum_{l=1}^{n} a_l h^{\alpha}(k-l)=\begin{cases} b_k, & k=0,1,...,m \\ 0, & k=m+1,...,m+n \end{cases} \tag{13}$$

where $h^{\alpha}(k) = 0$ for $k < 0$.

A two-step approach is used to solve for the coefficients a_k and b_k. In the first step, solving for the coefficients a_k, we use the last n equations of system (13), which after simple manipulations, may be written in matrix form as:

$$\mathbf{H}_2\mathbf{a} = -\mathbf{h}_{21} \tag{14}$$

with

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \in \Re^n, \quad \mathbf{h}_{21} = \begin{bmatrix} h^{\alpha}(m+1) \\ h^{\alpha}(m+2) \\ \vdots \\ h^{\alpha}(m+n) \end{bmatrix} \in \Re^n$$

$$\mathbf{H}_2 = \begin{bmatrix} h^{\alpha}(m) & h^{\alpha}(m-1) & \cdots & h^{\alpha}(m-n+1) \\ h^{\alpha}(m+1) & h^{\alpha}(m) & \cdots & h^{\alpha}(m-n+2) \\ \vdots & \vdots & \ddots & \vdots \\ h^{\alpha}(m+n-1) & h^{\alpha}(m+n-2) & \cdots & h^{\alpha}(m) \end{bmatrix}$$

where $\mathbf{H}_2\in\Re^{n\times n}$ is an n×n nonsymmetric Toeplitz matrix (see [22]). If $\mathbf{H}_2$ is nonsingular, the coefficients a_k (k = 1, 2, ..., n) are uniquely determined by:

$$\mathbf{a} = -\mathbf{H}_2^{-1}\mathbf{h}_{21} \tag{15}$$

In the second step, the coefficients b_k are found using the first m+1 equations of system (13), which may be written in matrix form as:

$$\mathbf{H}_1\bar{\mathbf{a}} = \mathbf{b} \tag{16}$$

where

$$\bar{\mathbf{a}} = \begin{bmatrix} 1 \\ \mathbf{a} \end{bmatrix} \in \Re^{n+1}, \quad \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix} \in \Re^{m+1}$$

$$\mathbf{H}_1 = \begin{bmatrix} h^{\alpha}(0) & 0 & \cdots & 0 \\ h^{\alpha}(1) & h^{\alpha}(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h^{\alpha}(m) & h^{\alpha}(m-1) & \cdots & h^{\alpha}(m-n) \end{bmatrix} \in \Re^{(m+1)\times(n+1)}$$

In this way, we obtain a perfect match between $h(k)$ and the desired impulse response sequence $h^{\alpha}(k)$ for the first $m+n+1$ values of k. The success of this method depends strongly on the number of selected model coefficients. Since the design method matches $h^{\alpha}(k)$ only up to the number of model parameters, the more complex the model, the better is the approximation to $h^{\alpha}(k)$ for $0 \le k \le m+n$. However, in practical applications, this introduces a major limitation of the Padé method because the resulting approximation must contain a large number of poles and zeros (see [23]).

It can be shown that the approximations obtained by the CFE method are identical to those resulting by application of the Padé approximation to power series expansion ($m = n$) (see [24]). Nevertheless, the CFE approach is computationally less expensive than the Padé technique.

4.2 Prony's Method

Prony's method differs from the Padé approximation in the scheme of finding the denominator coefficients a_k (k = 1, 2, ..., n) (see Figure 2). These coefficients are determined by LS minimization of the error $e_P(k) = a_k * h^{\alpha}(k) - b_k$ (where the symbol * denotes convolution):

$$e_P(k) = h^{\alpha}(k) + \sum_{l=1}^{n} a_l h^{\alpha}(k-l), \quad k = m+1, ..., N-1 \tag{17}$$

Setting the error $e_P(k) = 0$ in system (17) and writing these equations in matrix form:

$$\mathbf{H}_2\mathbf{a} = -\mathbf{h}_{21} \tag{18}$$

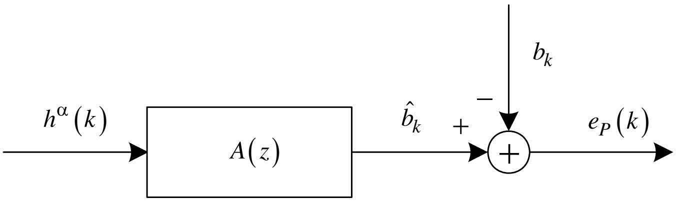


Figure 2
Prony's method

where

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \in \Re^n, \quad \mathbf{h}_{21} = \begin{bmatrix} h^\alpha(m+1) \\ h^\alpha(m+2) \\ \vdots \\ h^\alpha(N-1) \end{bmatrix} \in \Re^{N-m-1}$$

$$\mathbf{H}_2 = \begin{bmatrix} h^\alpha(m) & h^\alpha(m-1) & \cdots & h^\alpha(m-n+1) \\ h^\alpha(m+1) & h^\alpha(m) & \cdots & h^\alpha(m-n+2) \\ \vdots & \vdots & \ddots & \vdots \\ h^\alpha(N-2) & h^\alpha(N-3) & \cdots & h^\alpha(N-n-1) \end{bmatrix} \in \Re^{(N-m-1)\times n}$$

It is obvious that in this case system (18) cannot be solved exactly. Therefore, we find the LS solution by solving the set of normal equations:

$$\left(\mathbf{H}_2^T \mathbf{H}_2\right)\mathbf{a} = -\mathbf{H}_2^T \mathbf{h}_{21} \tag{19}$$

If $(\mathbf{H}_2^T\mathbf{H}_2) \in \Re^{n\times n}$ is nonsingular, a unique solution of (19) exists and the coefficients a_k are determined by:

$$\mathbf{a} = -\left(\mathbf{H}_2^T \mathbf{H}_2\right)^{-1} \mathbf{H}_2^T \mathbf{h}_{21} = -\mathbf{H}_2^+ \mathbf{h}_{21} \tag{20}$$

where $\mathbf{H}_2^+ = \left(\mathbf{H}_2^T \mathbf{H}_2\right)^{-1} \mathbf{H}_2^T$ is the pseudoinverse of $\mathbf{H}_2$.

With a_k given, the coefficients b_k are found using the Padé method of forcing $h(k) = h^\alpha(k)$ for $k = 0, 1, \ldots, m$ (see previous subsection 4.1):

$$b(k) = h^\alpha(k) + \sum_{l=1}^{n} a_l h^\alpha(k-l), \quad k = 0, 1, ..., m \tag{21}$$

4.3 Shanks' Method

Shanks' method provides an alternative to Prony's method of finding the numerator coefficients b_k ($k = 0, 1, \ldots, m$) (see Figure 3). Thus, instead of forcing an exact fit for the first m+1 values of the impulse response sequence, it performs a *LS* minimization of the error $e_S(k) = h^\alpha(k) - \hat{h}(k)$ over the interval [0, N–1]:

$$e_S(k) = h^\alpha(k) - \sum_{l=0}^{m} b_l g(k-l), \quad k = 0, 1, ..., N-1 \tag{22}$$

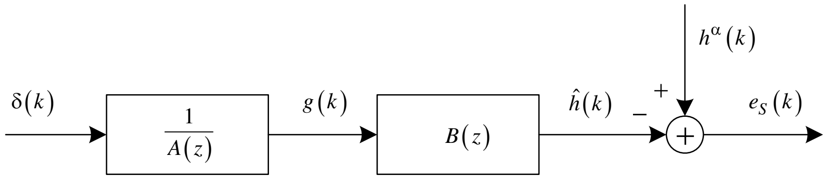


Figure 3
Shanks' method

Firstly, the coefficients a_k are determined in the same way as in Prony's method, that is, by a LS fit over the interval $[m+1, N-1]$ (see previous subsection 4.2). With a_k given, the coefficients b_k are determined following the sequence illustrated in Figure 3:

1 Compute the impulse response sequence $g(k)$ of the filter $1/A(z)$ using, for example, the recursion:

$$g(k) = \delta(k) - \sum_{l=1}^{n} a_l g(k-l), \quad k = 0, 1, ..., N-1 \tag{23}$$

with $g(k) = 0$ for $k < 0$.

2 Solve for the coefficients b_k by setting the error $e_S(k) = 0$ in system (22) and writing these equations in matrix form:

$$\mathbf{Gb} = \mathbf{h} \tag{24}$$

where

$$\mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix} \in \Re^{m+1}, \quad \mathbf{h} = \begin{bmatrix} h^{\alpha}(0) \\ h^{\alpha}(1) \\ \vdots \\ h^{\alpha}(N-1) \end{bmatrix} \in \Re^{N}$$

$$\mathbf{G} = \begin{bmatrix} g(0) & 0 & \cdots & 0 \\ g(1) & g(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g(N-1) & g(N-2) & \cdots & g(N-m-1) \end{bmatrix} \in \Re^{N\times(m+1)}$$

The *LS* solution is found by solving the linear equations:

$$\left(\mathbf{G}^T\mathbf{G}\right)\mathbf{b} = \mathbf{G}^T\mathbf{h} \tag{25}$$

If the matrix $(\mathbf{G}^{\mathrm{T}}\mathbf{G}) \in \Re^{(m+1)\times(m+1)}$ is nonsingular, the coefficients a_k can be uniquely determined by:

$$\mathbf{b} = \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{h} = \mathbf{G}^{+}\mathbf{h} \tag{26}$$

where $\mathbf{G}^+ = (\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T$ is the pseudoinverse of $\mathbf{G}$.

Note that the Prony and the Shanks methods should yield superior approximations than those obtained by the Padé technique, since $h(k)$ approximates $h^{\alpha}(k)$, in a least-squares sense, for values of $k > m+n$. Therefore, it will be expected a good match even outside the interval $[0, m+n]$. Also notice that the algorithms just described can be easily evaluated using one of the *LS* solvers available.

5 Illustrative Examples

In this section we use the techniques proposed previously to develop IIR filters approximations of half-differentiators ($s^{1/2}$) and half-integrators ($s^{-1/2}$), sampled at $T = 0.01$ *s*. For comparison purposes, in the figures that follows, we also plot the IIR filter approximation obtained by the Padé (or the CFE) method for $m = n = 5$.

Figures 4 and 5 depict the Bode diagrams of Prony's approximations to the Tustin and Al-Alaoui operators for $s^{-1/2}$, with $m = n = 5$ and different impulse response lengths of $N = \{11; 100; 200; 500; 1000\}$, respectively. Figures 6 and 7 show the same approximations considering now distinct orders of the IIR filters, namely of $m = n = 1; 3; \ldots; 9$ with $N = 1000$. Clearly, the higher the order $m = n$ (or the impulse response length N), of the IIR filter, the better the fitting, in a least-squares sense, to the ideal half-integrator $s^{-1/2}$ (dashed-dotted lines). Note that the Al-Alaoui scheme improves the high frequency magnitude response comparatively to the Tustin scheme. We also verify that the LS approach increases the performance in the low frequency range (corresponding to the steady-state time response) by increasing the order (or the number of impulse values used), resulting in better approximations than those given by the Padé (or the CFE) method. In Figures 8 and 9 are presented the impulse response sequences of the approximations revealing, again, its effectiveness in fitting the discretization schemes of the Tustin and Al-Alaoui operators.

Figures 10 and 11 depict the distribution of poles and zeros of Prony's approximations to the Tustin and Al-Alaoui operators for $s^{-1/2}$, $N = 1000$ and $m = n = \{1; 5; 7; 9\}$. It can be observed that the approximations fulfill the two desired properties: (i) all the poles and zeros lie inside the unit circle, and (ii) the poles and zeros are interlaced along the segment of the real axis corresponding to $z \in (-1, 1)$. Therefore, the resulting approximations are causal, stable and minimum phase, suitable for a digital implementation.

To further illustrate the effectiveness of the proposed techniques, the IIR filters approximations are used to calculate the half-integral and/or the half-derivative of the square and sawtooth functions, as shown in Figures 12 and 13, respectively. Once more, they highlight the effectiveness of the approximations in fitting the ideal curves (dashed-dotted lines).

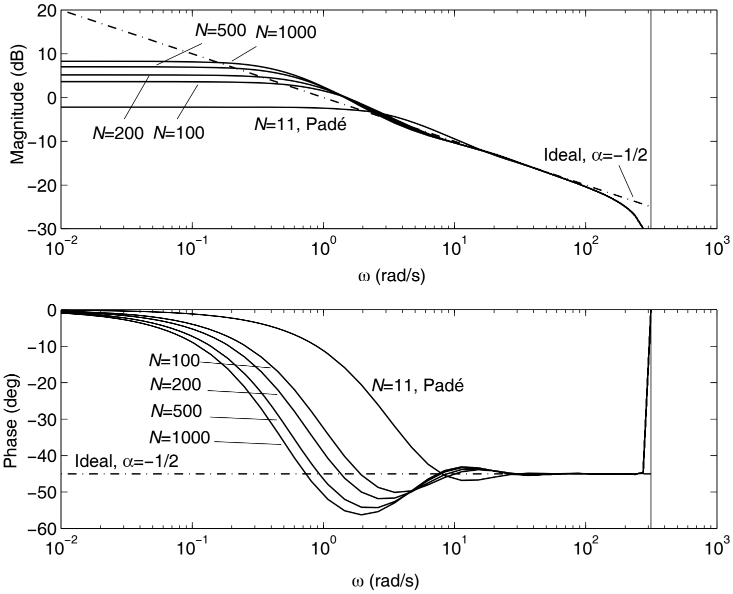


Figure 4

Bode diagrams of Prony's approximations to Tustin operator of $s^{-1/2}$, $m = n = 5$, $T = 0.01$ s and $N = \{11; 100; 200; 500; 1000\}$

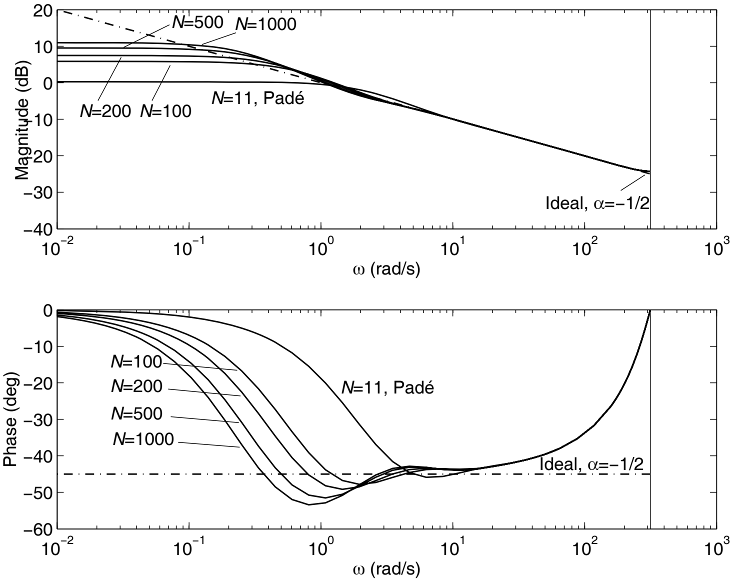


Figure 5

Bode diagrams of Prony's approximations to Al-Alaoui operator of $s^{-1/2}$, $m = n = 5$, $T = 0.01$ s and $N = \{11; 100; 200; 500; 1000\}$

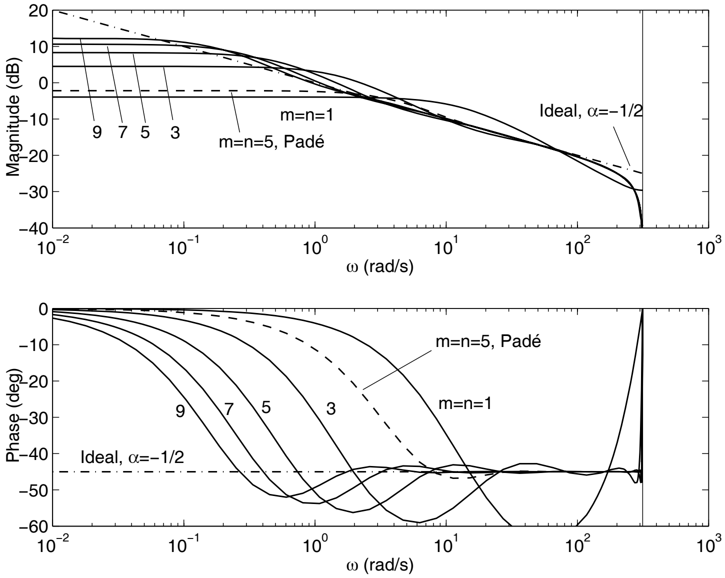


Figure 6

Bode diagrams of Prony's approximations to Tustin operator of $s^{-1/2}$, $N = 1000$, $T = 0.01$ *s* and $m = n = 1; 3; \ldots; 9$

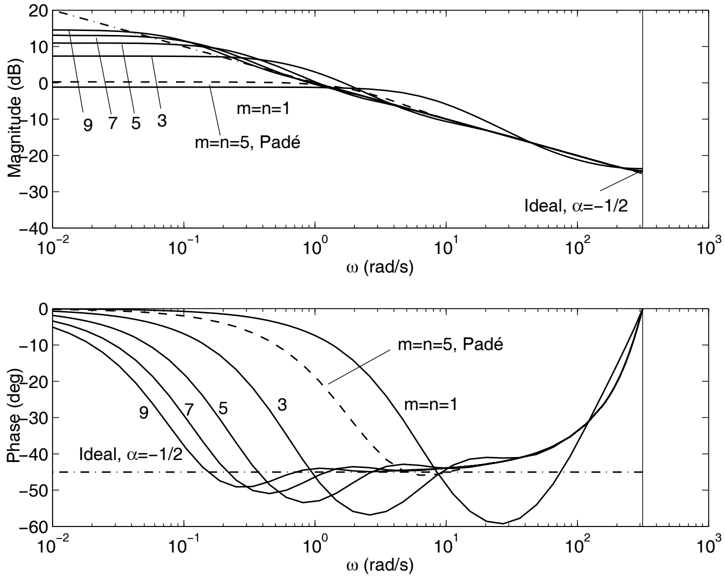


Figure 7

Bode diagrams of Prony's approximations to Al-Alaoui operator of $s^{-1/2}$, $N = 1000$, $T = 0.01$ *s* and $m = n = 1; 3; \ldots; 9$

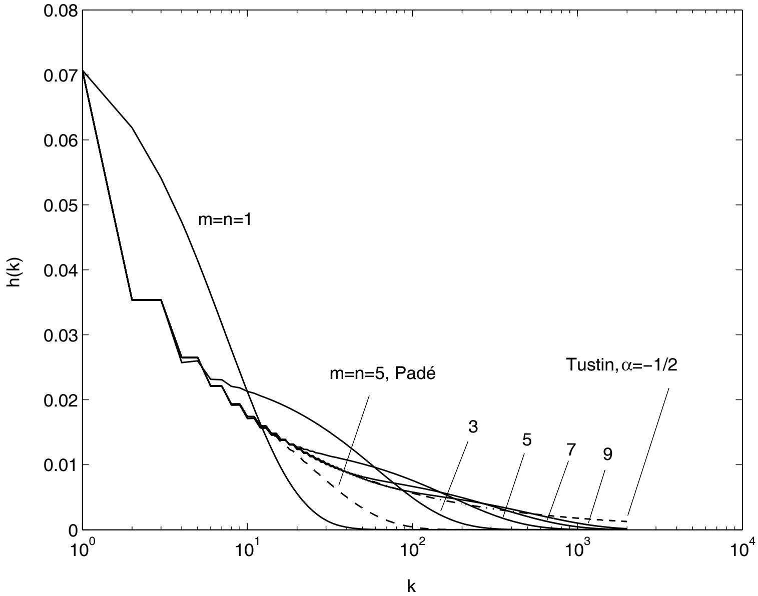


Figure 8

Impulse response sequences of Prony's approximations to Tustin operator of $s^{-1/2}$, N = 1000, T = 0.01 s and m = n = 1; 3; …; 9

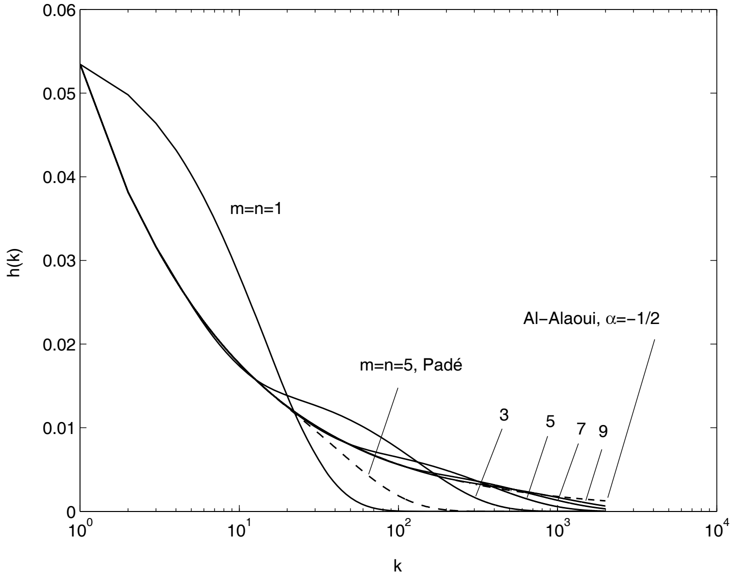


Figure 9

Impulse response sequences of Prony's approximations to Al-Alaoui operator of $s^{-1/2}$, N = 1000, T = 0.01 s and m = n = 1; 3; …; 9

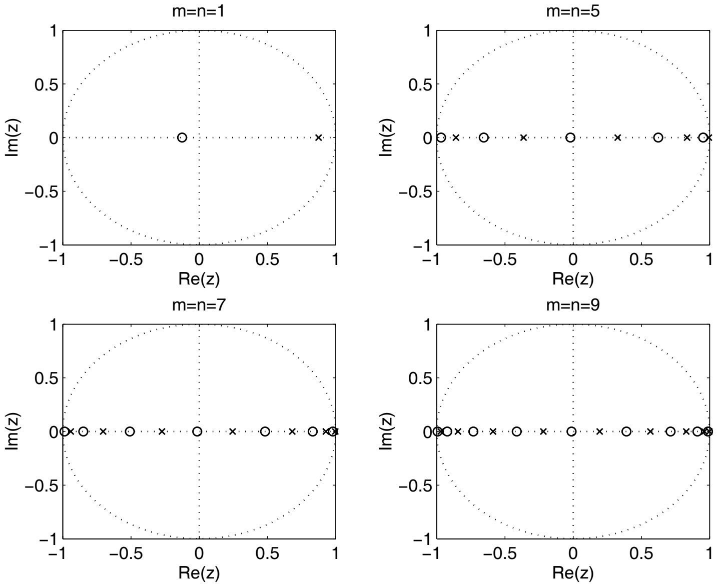


Figure 10

Pole-zero maps of Prony's approximations to Tustin operator of $s^{-1/2}$, $N = 1000$, $T = 0.01$ s and $m = n = \{1; 5; 7; 9\}$

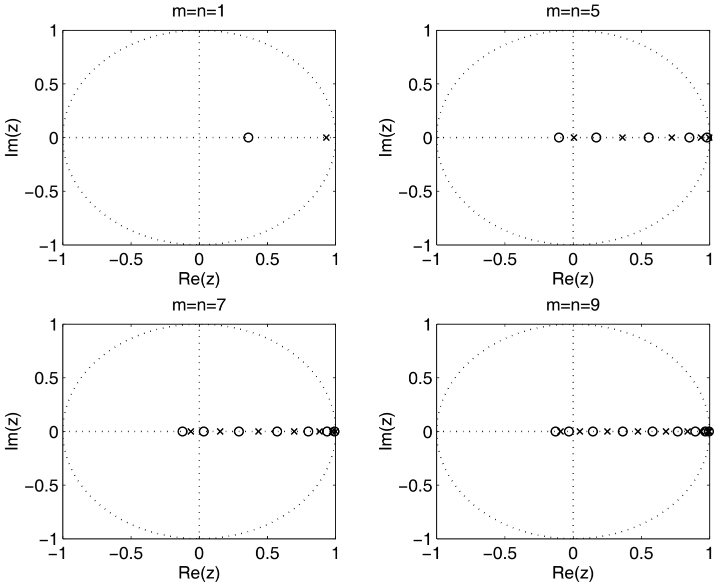


Figure 11

Pole-zero maps of Prony's approximations to Al-Alaoui operator of $s^{-1/2}$, $N = 1000$, $T = 0.01$ s and $m = n = \{1; 5; 7; 9\}$

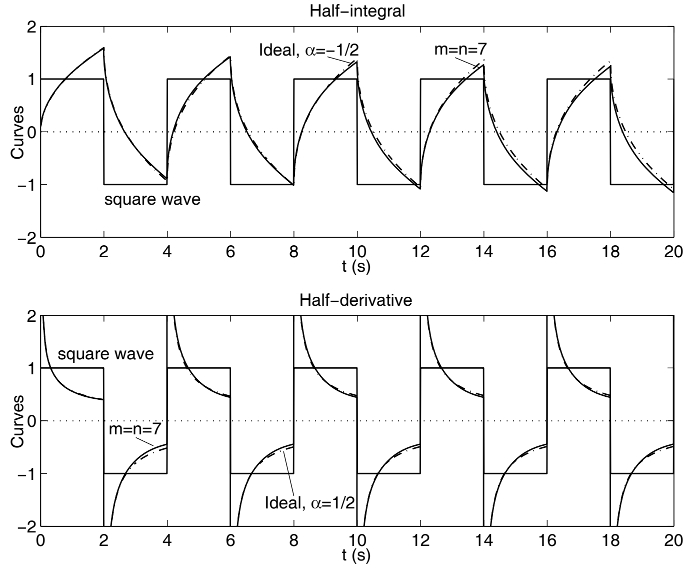


Figure 12

Half-integral/derivative of the square wave function with Shanks' approximation to Euler operator for $N = 1000$, $T = 0.01$ s and $m = n = 7$

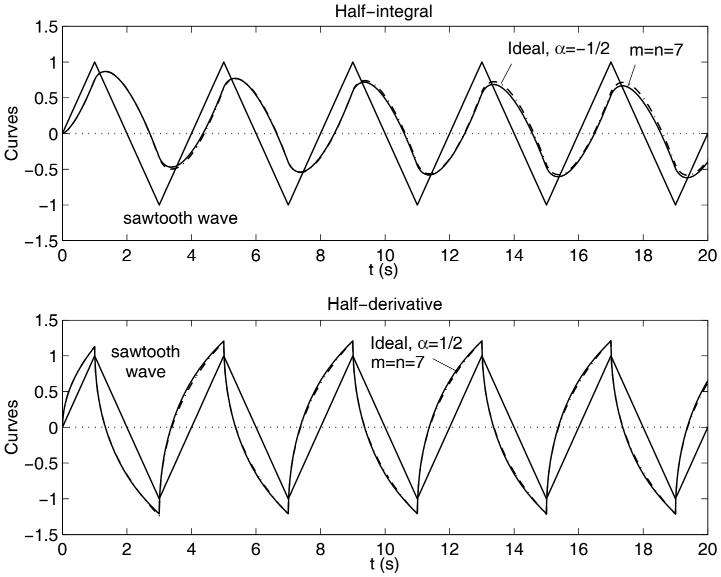


Figure 13

Half-integral/derivative of the sawtooth wave function with Prony's approximation to Euler operator for $N = 1000$, $T = 0.01$ s and $m = n = 7$

Conclusions

In this paper we have described the application of the techniques of Padé, Prony, and Shanks for the design of IIR filters approximations of continuous fractional-order integrators and differentiators of type s^{α}, $\alpha \in \Re$. The illustrated techniques only require finding the solution of a set of linear equations, yielding good approximations both in the time and the frequency domains. Moreover, it can produce superior approximations than other existent methods, namely the widely used CFE method. Also, the obtained approximations are causal, stable and minimum-phase, suitable for a digital implementation. Some examples are given that demonstrate the effectiveness of the proposed techniques. The results indicate that the least-squares based methods are adequate techniques for obtaining digital approximations of continuous fractional-order operators and, consequently, for the practical realization of fractional-order controllers.

References

[1] K. B. Oldham, J. Spanier: The Fractional Calculus, Academic Press, New York, 1974

[2] K. S. Miller, B. Ross: An Introduction to the Fractional Calculus and Fractional Differential Equations, Wiley & Sons, New York, 1993

[3] S. G. Samko, A. A. Kilbas, O. I. Marichev: Fractional Integrals and Derivatives: Theory and Applications, Gordon and Breach Science, Amsterdan, 1993

[4] I. Podlubny: Fractional Differential Equations, Academic Press, San Diego, 1999

[5] R. Hilfer: Applications of Fractional Calculus in Physics, World Scientific, Singapore, 2000

[6] A. Oustaloup: La Commande CRONE: Commande Robuste d'Ordre Non Entier, Editions Hermès, Paris, 1991

[7] A. Oustaloup: La Dérivation Non Entière: Théorie, Synthèse et Applications, Editions Hermès, Paris, 1995

[8] J. A. Tenreiro Machado: Analysis and Design of Fractional-Order Digital Control Systems, SAMS Journal of Systems Analysis, Modelling and Simulation 27(1997), 107-122

[9] B. M. Vinagre, I. Podlubny, A. Hernández, V. Feliu: Some Approximations of Fractional Order Operators Used in Control Theory and Applications, FCAA Fractional Calculus and Applied Analysis 3(2000), 231-248

[10] Y. Q. Chen, K. L. Moore: Discretization Schemes for Fractional-Order Differentiators and Integrators, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications 49(2002), 363-367

[11] Y. Q. Chen, B. M. Vinagre, I. Podlubny: Continued Fraction Expansion Approaches to Discretizing Fractional Order Derivatives-An Expository Review, Nonlinear Dynamics 38(2004), 155-170

[12] I. Podlubny: Fractional-order Systems and PI D -controllers, IEEE Transactions on Automatic Control 44(1999), 208-214

[13] M. A. Al-Alaoui: Novel Digital Integrator and Differentiator, Electronics Letters 29(1993), 376-378

[14] M. A. Al-Alaoui: Filling the Gap Between the Bilinear and the Backward-Difference Transforms: An Interactive Design Approach, International Journal of Electrical Engineering Education 34(1997), 331-337

[15] M. A. Al-Alaoui: A Class of Second-Order Integrators and Low-Pass Differentiators, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications 42(1995), 220-223

[16] M. A. Al-Alaoui: Novel Stable Higher Order s-to-z Transforms, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications 48(2001), 1326-1329

[17] T. H. Parks, C. S. Burrus: Digital Filter Design, Wiley-Interscience, New York, 1987

[18] Monson H. Hayes: Statistical Digital Signal Processing and Modeling, Wiley & Sons, New York, 1996

[19] Ramiro S. Barbosa, J. A. Tenreiro Machado, Isabel M. Ferreira: Least-Squares Design of Digital Fractional-Order Operators, Proceedings of the First IFAC Workshop on Fractional Differentiation and its Applications (FDA'04), Bordeaux, France, July 19-21 (2004), 434-439

[20] Ramiro S. Barbosa, J. A. Tenreiro Machado, Isabel M. Ferreira: Pole-Zero Approximations of Digital Fractional-Order Integrators and Differentiators Using Signal Modeling Techniques, Proceedings of the 16th IFAC World Congress, Prague, Czech Republic, July 4-8 (2005), CD-ROM

[21] Ramiro S. Barbosa, J. A. Tenreiro Machado, Manuel F. Silva: Time Domain Design of Fractional Differintegrators using Least-Squares, Signal Processing, Elsevier, To appear in 2006

[22] A. D. Poularikas: The Handbook of Formulas and Tables for Signal Processing, CRC Press, Boca Raton, 1999

[23] J. G. Proakis, D. G. Manolakis: Digital Signal Processing: Principles, Algorithms and Applications, Prentice-Hall, 3rd edition, Upper Saddle River, 1996

[24] L. Lorentzen, H. Waadeland: Continued Fractions with Applications, Addison-Wesley, North-Holland, Amsterdam, 1992

[25] Imre J. Rudas, Jozsef K. Tar, Béla Pátkai, Compensation of Dynamic Friction by a Fractional Order Robust Controller, pp. 15-20, ICCC 2006 - 2006 IEEE International Conference on Computational Cybernetics, pp. 9-14, Tallinn, Estonia, August 20-22, 2006, ISBN 1-4244-0072-4, IEEE Catalog Number 06EX1270C

Tensor Product Model Transformation-based Sliding Surface Design

Péter Korondi

Department of Automation and Applied Informatics
Budapest University of Technology and Economics
PO. Box 91, H-1521 Budapest, Hungary
korondi@elektro.get.bme.hu

Abstract: Motion control has been a fruitful ground for applying Variable Structure Systems (VSS) theory. This paper provides an assessment of the state of the art of the relevant theoretical results for sliding mode control. The design of a sliding-mode controller consists of three main steps. First step is the design of the sliding surface, the second step is the design the control law which holds the system trajectory on the sliding surface, and the third and key step is the chattering-free implementation. The main contribution of that paper is a new method for sliding surface sector design based on tensor product (TP) model transformation to reduce the chattering.

Keywords: Sliding mode control, sliding sector design, Tensor product, friction compensation

1 Introduction

Sliding mode has been introduced in the late 1970's [1, 2] for highly coupled nonlinear dynamics, with unknown system parameters and disturbances. In the early 1980's, sliding mode was further introduced for the control of induction motor drives [3]. Its utility in this hybrid discipline, consisting of power electronics and motion control, is to provide direct switching strategy [4] to the power electronics devices such that, in spite of the nonlinear dynamics of the induction motor, the control design is decomposed into a nonlinear control synthesis problem, and a linear control design problem of reduced order. These early applications of sliding mode indicated the versatility of the underlying control principles in the design of feedback control systems for motion control, regardless of the origin or the nature of the particular system performance specifications and design goals.

This initial works were followed by a large number of research papers in robotic manipulator control and in motor drive control. References can be found in [5]. In some of these works, experimental results were published [6, 7, 8, 9]. However, despite of the theoretical predictions of superb closed loop system performance of sliding mode, some of the experimental works indicated that sliding mode in practice has limitations due to the need of high sampling frequency to reduce the high frequency oscillation phenomenon about the sliding mode manifold -- collectively referred to as ‘chattering’. Among these experimental works, a few succeeded to show closed loop system behaviour, which were predicted by theory. Those who failed to manage the experimental designs successfully concluded that chattering is a major problem in realizing sliding mode control in practice. The usual sources of chattering are the limited switching frequency and the un-modeled dynamics, which are ignored in the theoretical design steps [10]. A detailed simulation of the whole system including controller and discrete semiconductor switches can be an important middle step in the chattering free implementation of sliding mode. Another promising method for reducing chattering is the sliding sector design [20], which is in the focus of that paper. A tensor product model transformation is proposed for design a sliding surface.

The tensor product (TP) model form is a dynamic model representation whereupon Linear Matrix Inequality (LMI) based control design techniques [11]–[13] can immediately be executed. It describes a class of Linear Parameter Varying (LPV) models by the convex combination of linear time invariant (LTI) models, where the convex combination is defined by the weighting functions of each parameter separately. An important advantage of the TP model forms is that the convex hull of the given dynamic LPV model can be determined and analysed by one variable weighting functions. Furthermore, the feasibility of the LMIs can be considerably relaxed in this representation via modifying the convex hull of the LPV model.

The TP model transformation is a recently proposed numerical method to transform LPV models into TP model form [14], [15]. It is capable of transforming different LPV model representations (such as physical model given by analytic equations, fuzzy, neural network, genetic algorithm based models) into TP model form in a uniform way. In this sense it replaces the analytical derivations and affine decompositions (that could be a very complex or even an unsolvable task), and automatically results in the TP model form. Execution of the TP model transformation takes a few minutes by a regular Personal Computer. The TP model transformation minimizes the number of the LTI components of the resulting TP model. Furthermore, the TP model transformation is capable of resulting different convex hulls of the given LPV model.

The rest of the paper is organized as follows: Section 2 describes the main steps of sliding mode control design including basis of the proposed tensor product model transformation. Section 3 presents an application example including simulation results. Finally, Section 4 concludes the results.

2 Theoretical Background of the VSS

The design of a sliding-mode controller consists of three main steps. First is the design of the sliding surface, the second step is the design the control law which holds the system trajectory on the sliding surface, and the third and key step is the chattering-free implementation.

2.1 Design of the Sliding Manifold

The following linear time invariant (LTI) system is considered; first the reference signal is supposed to be constant and zero. The system (which is assumed to be controllable) is transformed to a regular form [16].

$$\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_2 \end{bmatrix} \mathbf{u} \qquad \begin{matrix} \mathbf{x}_1 \in \Re^{n-m} \\ \mathbf{x}_2 \in \Re^{m} \\ \mathbf{u} \in \Re^{m} \end{matrix} \tag{1}$$

The switching surfaces, $\boldsymbol{\sigma}$ of the sliding mode, where the control vector components have discontinuities, can be written in the following form [17], where $\boldsymbol{\Lambda}$ is the 'surface matrix'.

$$\boldsymbol{\sigma} = \mathbf{x}_2 + \boldsymbol{\Lambda}\mathbf{x}_1 = 0 \quad \boldsymbol{\sigma} \in \Re^m \quad \text{and} \quad \boldsymbol{\Lambda} \in \Re^{mx(n-m)} \tag{2}$$

When sliding mode occurs (when $\boldsymbol{\sigma}$=0 and $\mathbf{x}_2 = -\boldsymbol{\Lambda}\mathbf{x}_1$), the design problem of the sliding surfaces can be regarded as a linear state feedback control design for the following subsystem:

$$\dot{\mathbf{x}}_1 = \mathbf{A}_{11}\mathbf{x}_1 + \mathbf{A}_{12}\mathbf{x}_2 \tag{3}$$

In (3), $\mathbf{x}_2$ can be considered as the input of the subsystem. A state feedback controller $\mathbf{x}_2 = -\boldsymbol{\Lambda}\mathbf{x}_1$ for this subsystem gives the switching surface of the whole VSS controller. In sliding mode

$$\dot{\mathbf{x}}_1 = (\mathbf{A}_{11} - \mathbf{A}_{12}\boldsymbol{\Lambda})\mathbf{x}_1 \tag{4}$$

In nineties, various linear control design methods based on state feedback are proposed for (3) to the design a stable switching surfaces in a form (4) (survey in [17]). **The main problem is that this method cannot be applied for a non-linear system which is the main challenge. The solution can be the Tensor Product model transformation.**

2.2 Sliding Surface Design based on Tensor Product Model Transformation

This section is intended to discuss the fundamentals of TP model transformation. Consider a parametrically varying dynamical system

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{p}(z))\mathbf{x}(t) + \mathbf{B}(\mathbf{p}(z))\mathbf{u}(t) \tag{5}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$$

with input $\mathbf{u}(t)$, output $\mathbf{y}(t)$ and state vector $\mathbf{x}(t)$. The system matrix is a parameter-varying object, where $\mathbf{p}(z) \in \mathbf{\Omega}$ is time varying *N*-dimensional parameter vector, and is an element of the closed hypercube $\mathbf{\Omega} = [a_1, b_1]\mathrm{x}[a_2, b_2]\mathrm{x}\cdots\mathrm{x}[a_N, b_N] \in \Re^N$. The parameter $\mathbf{p}(z)$ can also include some elements of $\mathbf{x}(t)$.

The TP model transformation starts with the given LPV model (5) and results in the TP model representation

$$\dot{\mathbf{x}}(t) = \sum_{r=1}^{R} w^r(\mathbf{p}(z))\mathbf{A}^r\mathbf{x}(t) + \sum_{r=1}^{R} w^r(\mathbf{p}(z))\mathbf{B}^r\mathbf{u}(t) \tag{6}$$

where $w^r(\mathbf{p}(z)) \in [0,1]$ are weighting coefficients. For further details about TP model transformation, refer to [14], [15]. According to (2), a sliding surface is designed for each system $\mathbf{A}^r\ \mathbf{B}^r$, which are assummed to be controllable.

$$\boldsymbol{\sigma}^r = \mathbf{x}_2 + \mathbf{\Lambda}^r\mathbf{x}_1 = 0 \quad \boldsymbol{\sigma}^r \in \Re^m \tag{7}$$

2.3 Control Law

There are two main approaches of design of a control law for the sliding mode on the surface. In the first 'global' case, to ensure that the system remains in the sliding mode ($\boldsymbol{\sigma} = 0 \quad \boldsymbol{\sigma} \in \Re^m$) the condition

$$\dot{\boldsymbol{\sigma}}\boldsymbol{\sigma}^T < 0 \tag{8}$$

should be hold. In the second 'local' approach, sliding mode exists only in the intersection of the switching surfaces. In this case, the condition for the existence of a sliding mode is

$$\sigma_i\dot{\sigma}_i < 0, \tag{9}$$

where the subscription *i* refers to the *i*th element of the corresponding vector. The simplest control law which can lead to 'local type' sliding mode is the relay:

$$u_i = M_i \cdot sign(\sigma_i) \tag{10}$$

This is easy to realize by power electronic circuits. The relay type of controller can directly control the semiconductor switching elements, but it does not ensure the existence of sliding mode for the whole state space, and relatively big values of M_i are necessary which might cause a severe chattering phenomenon. This control law is preferable if the controller's sample frequency is nearly equal to the maximum switching frequency of semiconductor switching elements.

If sliding mode exists then there is continuous control, so-called ‘equivalent’ control, $\mathbf{u_{eq}}$, which can hold the system on the sliding manifold. In the practice, there is never perfect knowledge of the whole system and its parameters. Only $\hat{\mathbf{u}}_{\mathbf{eq}}$, the estimation of $\mathbf{u_{eq}}$, can be calculated. Since $\mathbf{u_{eq}}$ does not guarantee the convergence to the switching manifold in general, a discontinuous term is usually added to $\hat{\mathbf{u}}_{\mathbf{eq}}$.

$$u_i = \hat{u}_{eq,i} + M_i \cdot sign(\sigma_i) \tag{11}$$

The control laws (11) do not control the semiconductor switching elements directly; additional PWM is needed. Usually, this is no problem since the switching frequency of the semiconductor elements can be much higher then the sampling frequency of the fastest digital controller. The general concept of TP model based control strategies is that the control signal is the weighed sum of the control signal of the component systems

$$\mathbf{u} = \sum_{r=1}^{R} w^r(\mathbf{p}(z))\mathbf{u}^r \tag{12}$$

2.4 Chattering Free Implementation

The chattering is essential in the basic VSC due to the requirement that the system state must stick to the switching surface. Obviously this requirement is too restrict when only finite switching rate is available. Replacing the switching surface to the sliding sector may enable the system state to move continuously. From now on single input and two sliding surfaces are assumed, the whole state space is divided into three regions

$$\begin{aligned} &R_1 \in \{\mathbf{x} \mid \sigma^1(\mathbf{x}) < 0 \quad \text{and} \quad \sigma^2(\mathbf{x}) < 0\} \\ &R_2 \in \{\mathbf{x} \mid \sigma^2(\mathbf{x}) > 0 \quad \text{and} \quad \sigma^2(\mathbf{x}) > 0\} \\ &R_3 \in \{\mathbf{x} \mid \sigma^1(\mathbf{x})\sigma^2(\mathbf{x}) \le 0\} \end{aligned} \tag{13}$$

Here the region R_3 is a sliding sector introduced in [18]. The control is composed of two terms

$$u = u_c + u_d \tag{14}$$

where u_c is a feedforward compensation term based on the estimation of the 'equivalent' control and the given uncertainty bounds, u_d is a switching term to suppress the system parameter variations and disturbances. According to [18], in conventional (non TP) case

$$u_c = \hat{u}_{eq} \text{ and } u_d = \begin{cases} M\text{sign}\left(\dfrac{\sigma^1+\sigma^2}{2}\right) & \text{if} \quad \mathbf{x} \in R_1 \cup R_2 \\ M\left(\dfrac{\sigma^1+\sigma^2}{\left|\sigma^1\right|+\left|\sigma^2\right|}\right) & \text{if} \quad \mathbf{x} \in R_3 \end{cases} \tag{15}$$

The sliding sector control inherits the robustness of the classical sliding mode control in certain cases (see details in [18]). The sliding sector concept can be extended to the TP model based sliding mode control as well.

$$u_c = \sum_{r=1}^{2} w^r((p(z))\hat{u}^r_{eq} \text{ and } u_d = \begin{cases} M\text{sign}\left(\dfrac{\sigma^1+\sigma^2}{2}\right) & \text{if} \quad \mathbf{x} \in R_1 \cup R_2 \\ M\left(\dfrac{\sum_{r=1}^{2} w^r((p(z))\sigma^r}{\sum_{r=1}^{2} w^r((p(z))\left|\sigma^r\right|}\right) & \text{if} \quad \mathbf{x} \in R_3 \end{cases} \tag{16}$$

3 Application

The experimental system consists of a conventional DC servo gear motor with encoder feedback and variable inertia load coupled by a relatively rigid shaft, as shown in Fig 1. The controller is implemented using a DSP as the computation engine.

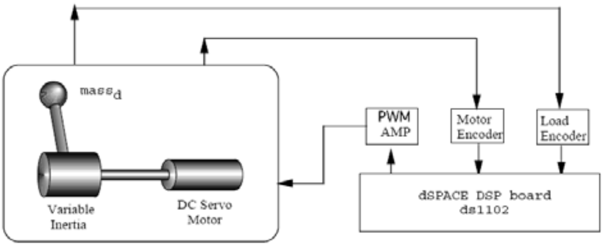


Figure 1
The experimental system

3.1 System Equations

In the course of control design, the flexibility of the shaft is ignored. The state variables are the shaft position, *θ*, the shaft angular velocity, *ω*, and the input current, *i*, the control signal *u* is the motor voltage.

$$\begin{pmatrix} \dot{\theta} \\ \dot{\omega} \\ \dot{i} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{K_t}{J} \\ 0 & -\frac{K_\omega}{L_a} & -\frac{R_a}{L_a} \end{pmatrix} \begin{pmatrix} \theta \\ \omega \\ i \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{pmatrix} u \tag{17}$$

where J is the inertia of the motion control system, K_t and K_ω are the torque constant and the back-EMF constant R_a and L_a are the resistance inductance of the armature. The effect of $mass_d$ is considered as a disturbance. The model calculated from the nominal parameters of the system is as follows:

$$\begin{pmatrix} \dot{\theta} \\ \dot{\omega} \\ \dot{i} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 42 \\ 0 & -4600 & -2450 \end{pmatrix} \begin{pmatrix} \theta \\ \omega \\ i \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1100 \end{pmatrix} u \tag{18}$$

The viscous, Coulomb and Stribeck frictions are modelled by Hess and Soom [19] in the following way, where the second two terms are nonlinear:

$$\dot{\omega} = -\underbrace{\frac{F_v}{J}}_{\text{viscous term}}\omega - \underbrace{\left(\frac{2F_c}{J(1+e^{-500\omega})} - \frac{F_c}{J}\right)}_{\text{Coulomb term}} - \underbrace{\left(\frac{\frac{2F_s}{1+e^{-500\omega}} - F_s}{J(1+(v/v_s)^2)}\right)}_{\text{Stribeck term}} + \frac{K_t}{J}i \tag{19}$$

where F_v, F_c and F_s are constants for the viscous, Coulomb and Stribeck frictions, v_s is the characteristic velocity of the Stribeck curve. F_v was given in data sheet of the servo motor, F_c, F_s and v_s are selected after some tests. Fig. 2 shows the simulated Stribeck curve.

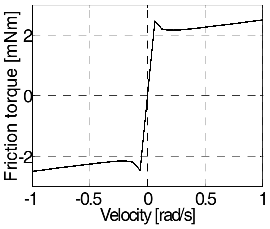


Figure 2
Stribeck curve by simulation

Applying the tensor product transformation, the above nonlinear system can be modelled by the combination of the following two linear systems.

$$\mathbf{A}^1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -20.2 & 42 \\ 0 & -4600 & -2450 \end{pmatrix} \mathbf{B}^1 = \begin{pmatrix} 0 \\ 0 \\ 1100 \end{pmatrix} \tag{20}$$

$$\mathbf{A}^2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -6239 & 42 \\ 0 & -4600 & -2450 \end{pmatrix} \mathbf{B}^2 = \begin{pmatrix} 0 \\ 0 \\ 1100 \end{pmatrix} \tag{21}$$

The two weighing coefficients as functions of the speed are shown in Fig. 3.

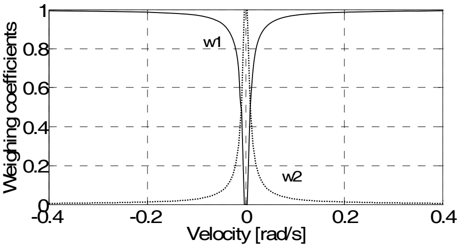


Figure 3
The weighing coefficients as a function of the velocity

It is easy to explain. The nonlinear friction terms are modelled by varying viscous coefficient, which is represented by the a_{22} element in the system matrix. $\mathbf{A}^1$ with small viscous coefficient dominates at the high speed range, where the Coulomb friction is relatively small and the $\mathbf{A}^2$ with very big viscous coefficient dominates at the low speed range, where the Coulomb friction is relatively big.

3.2 Sliding Surface Design

According to (7) and (17), the surfaces have the following form:

$$\sigma^r = i + \lambda_\omega^r \omega + \lambda_\theta^r \theta = 0 \qquad \text{where} \qquad r=1,2 \tag{22}$$

The poles for the reduce order systems of

$$\mathbf{A}_r^1 = \begin{pmatrix} 0 & 1 \\ 0 & -20.2 \end{pmatrix} \ \mathbf{B}_r^1 = \begin{pmatrix} 0 \\ 42 \end{pmatrix} \tag{23}$$

$$\mathbf{A}_r^1 = \begin{pmatrix} 0 & 1 \\ 0 & -6239 \end{pmatrix} \ \mathbf{B}_r^2 = \begin{pmatrix} 0 \\ 42 \end{pmatrix} \tag{24}$$

are selected as

$$P=[-17\ -35]. \tag{25}$$

Applying the Matlab pole placement function:

$$\lambda_\theta^1 = \lambda_\theta^2 = 14.1667 \quad \lambda_\omega^1 = 0.7571 \quad \lambda_\omega^2 = -147.3 \tag{26}$$

3.3 Simulation Results

As model verification, the real and simulated velocities (ω_r, ω_S) are compared in Fig. 4, where the input voltage of the motor is a shifted sinusoidal function with amplitude of 12 V (open loop responses). The value of the input voltage is divided by 5 to plot the velocity and input voltage in the same figure. One kind of nonlinearity of the system is borne from the huge friction of the harmonic gear. It can be seen in the Fig. 4, if the motor is in standstill, at least 2 V should be switched across the motor to start it. On the other hand, the motor sticks, if the input voltage is under 1.2 V. According to Fig. 4, the simulated model is acceptable from engineering point of view. The power electronic PWM unit is saturated at 22V. It is also a kind of nonlinearity which can be handled by TP model but this paper concentrates on the friction that is why only the nonlinearity of the friction is simulated by TP model.

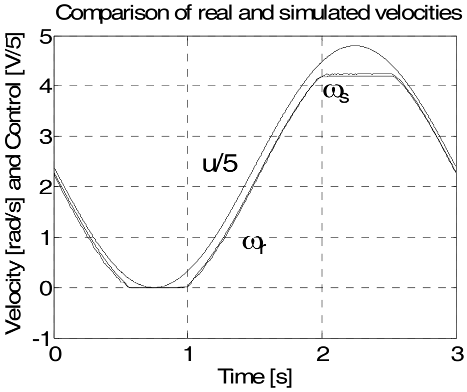


Figure 4

The open loop responses for sinusoidal input voltage

Performances of two controllers are compared. In both cases, the system starts from the following initial state

$$\theta=1 \text{ rad}, \ \omega=0 \text{ rad/sec and } i=0 \text{ A} \tag{27}$$

The aim of the controller is to move all state variables to 0. The sampling frequency of both controllers is relatively small, 100 Hz.

CONTROLLER C-SMC

It is a classical sliding mode controller (C-SMC), where the sliding surface σ^1 and (11) type control law are selected. The equivalent control is calculated from ($\mathbf{A}^1$ $\mathbf{B}^1$) system matrixes.

CONTROLLER TP-SMC

It is a TP model based sector sliding mode controller(TP-SMC), where the two sliding surfaces are selected by (26) and (16) type control law is applied with two linear system components (20),(21) and the weighting coefficients of Fig. 3. Two equivalent controls are calculated from ($\mathbf{A}^1$ $\mathbf{B}^1$) and ($\mathbf{A}^2$ $\mathbf{B}^2$) system matrixes.

There is a small difference between two position responses in Fig. 5 since the conventional (static) sliding surface cannot be identical to the sliding sector. The main difference appears in the control activities and in the velocity responses (see in Figs. 6 and 7). The conventional sliding mode is very robust but it needs intensive control action (see in Fig. 6), which might cause significant audio noise as well. The chattering of the velocity could be reduced by increasing the sampling frequency but this paper demonstrates that the reduction of chattering and the intensity of the control action (the audio noise) is significant at the same sampling rate, if the TP based sector sliding mode is applied instead of the traditional sliding mode control.

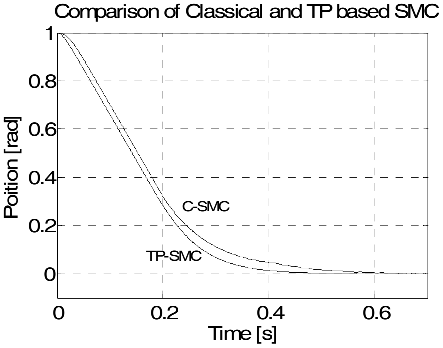


Figure 5

Comparison of the position responses

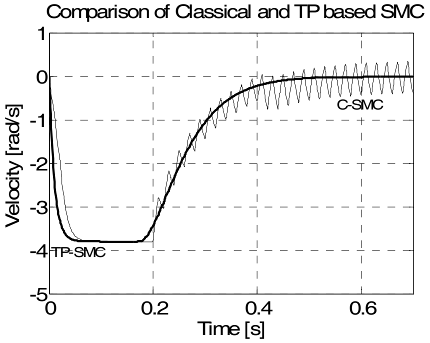


Figure 6

Comparison of the velocity responses

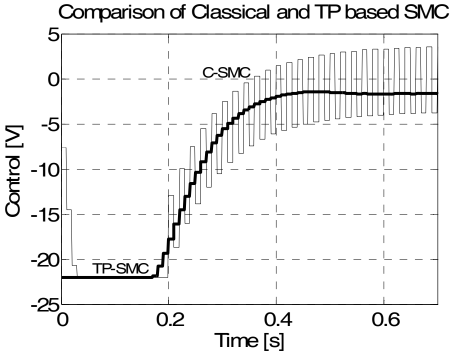


Figure 7

Comparison of the control signals

Conclusions

In this paper, a modified variable structure control strategy with continuous switching control has been developed in detail for the nonlinear system with uncertainty. The control strategy can be regarded as the extension of conventional VSS based sliding mode control method through expanding the switching surface to the sliding sector. The sliding sector is designed by a tensor product model transformation. The major advantage of the proposed control scheme is the introduction of the continuous switching control which successfully achieves smooth control response and retains the robustness of VSC simultaneously. Both theoretical analysis and simulations demonstrate the attractiveness and the asymptotic stability of the sliding sector with the use of the proposed switching control which is essentially an interpolated control.

Acknowledgement

The authors wish to thank the National Science Research Fund (OTKA K62836), Control Research Group and János Bolyai Research Scholarship of Hungarian Academy of Science for their financial support and the support stemming from the Intergovernmental S & T Cooperation Program.

References

[1] Utkin,V.: Variable Structure Systems with Sliding Mode, *IEEE Trans. Vol. AC-22*, No. 2, pp. 212-222, 1977

[2] Young, K-K. D.: Controller Design for Manipulator Using Theory of Variable Structure Systems, *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. SMC-8, February 1978, pp. 101-109

[3] Sabanovic A., D. Izosimov: Application of Sliding Modes to Induction Motor Control, *IEEE Trans. Industrial Appl.*, Vol. IA-17, No. 1, pp. 41-49, 1981

[4] Harashima F., H. Hashimoto, S. Kondo: MOSFET Converter-fed Position Servo System with Sliding Mode Control, *IEEE Trans. Industrial Electronics*, Vol. IE-32, No. 3,pp. 238-244, 1986

[5] Young, K. D. (editor): *Variable Structure Control for Robotics and Aerospace Applications*, pp. 157-172, Elsevier Science Publishers, 1993

[6] Hashimoto H., K. Maruyama, F. Harashima: A Microprocessor-based Robot Manipulator Control with Sliding Mode, *IEEE Transactions on Industrial Electronics*, Vol. IE-34, No. 1, pp. 11-18, 1987

[7] Hashimoto H., Y.Ishikawa: Robust Digital Sliding Mode Applied to Motion Control, Japan/USA Symposium on Flexible Automation ASME 1992, Vol. 1. pp. 25-30

[8] Andreescu GD, Popa A, Spilca A: Two Sliding Mode-based Observers for Sensorless Control of PMSM, ELECTR POW COMPO SYS 30 (2): 121-133 Feb 2002

[9] Vittek, J., Dodds, S. J.: Forced Dynamics Control of Electric Drives, EDIS – Publishing Centre of Zilina University, June 2003, ISBN 80-8070-087-7

[10] P. Korondi, H. Hashimoto, V. Utkin: Direct Torsion Control of Flexible Shaft based on an Observer Based Discrete-time Sliding Mode, *IEEE Trans. on Industrial Electronics* Vol. IE, No. 2. pp. 291-296, 1998

[11] S. Boyd, L. E. Ghaoui, E. Feron, V. Balakrishnan: Linear Matrix Inequalities in System and Control Theory, *Philadelphia PA:SIAM, ISBN 0-89871-334-X*, 1994

[12] P. Gahinet, A. Nemirovski, A. J. Laub, M. Chilali: *LMI Control Toolbox*. The MathWorks, Inc., 1995

[13] C. W. Scherer, S. Weiland: *Linear Matrix Iequalities in Control*, ser. DISC course lecture notes, DOWNLOAD: http://www.cs.ele.tue.nl/SWeiland/lmid.pdf, 2000.

[14] P. Baranyi: TP Model Transformation as a Way to LMI-based Controller Design, *IEEE Transaction on Industrial Electronics*, Vol. 51, No. 2, pp. 387-400, April 2004

[15] Zoltán Petres, Barna Reskó, Péter Baranyi: TP Model Transformation-based Control of the TORA System, *Production Systems and Information Engineering*, 2:159-175, 2004

[16] Luk'uyanov A. G., Utkin V. I.: Method of Reducing Equations of Dynamic Systems to Regular Form, Automation and Remote Control, 42, (P.1), pp. 413-420, 1981

[17] P. Korondi, H. Hashimoto: Sliding Mode Design for Motion Control (12 pages) Studies in Applied Electromagnetics and Mechanics Vol. 16. ISBN 90 5199 487 7, IOS Press 2000

[18] Jian-Xin Xu, Tong-Heng Lee, Mao Wang, X. H. Yu: On the Design of Variable Structure Controllers with Continuous Switching Control, International Journal of Control Vol. 65, No. 5, pp 409-431, 1996

[19] Hess, D. P., Soom, A.: Friction at a Lubricated Line Contact Operating at Oscillating Sliding Velocities, *J. of Tribology* **112**(1), 1990, 147-152

[20] Furuta. K., Y. Pan: Variable Structure Control with Sliding Sector, Automatica, Vol. 36, pp. 211-228, (2000)

Agents from Functional-Computational Perspective

Jozef Kelemen

Institute of Computer Science, Silesian University, Opava, Czech Republic
VSM College of Management, Bratislava, Slovak Republic
kelemen@fpf.slu.cz

Abstract: The contribution sketches a functional-computational typological scale of agents starting form the reactive ones, and puts the family of (at least minimally) conscious agents into the proposed typology. Then it discusses the traditional computational properties of agents according their types, and sketches a way of a rather non-traditional computational characterization of conscious agents using the concept of hyper-computation. The contribution ends with relating the sketched formal approach to agents with agents embodiment, and relates embodiment of agents with their emergence of hyper-computational power.

Keywords: Agent, functional specification, computational specification, emergence, grammar systems, hyper-computation, consciousness

1 Introduction

Agents might be considered from several points of view. The most usual is their constructivistic specification through their architectural modules and their mutual interrelatedness. Another usual point of view is derived from analyzing agents functional properties. The agents’ functional properties are often very closely related with computational properties implemented in their architectural modules or emerging from the interactions of them. So, the functional-computational characterization of agents may contribute to our better understanding of agents. This article is devoted to such kind of characterization.

We give a stratification of agents starting from the reactive ones up to the simple emotional agents, and a way how to extend the standard typology of agents by the family of (at least minimally) conscious agents. The particular strata will be specified by most characteristic functions of agents performed in their environments or in their internal states (if any). These functions are related with the question of their effective computability, and it is hypothesized that in order to achieve (at least minimally) conscious agents, the traditionally used Turing-type computable functions must by replaced by the so called hyper-computable functions.

By an *agent* we mean – similarly as it is almost generally accepted in present day computer science, cognitive science, and artificial intelligence; cf. e.g. (Tecuci, 1998) – any autonomously active entity with certain possibilities to sense its environment, and act in it in order to achieve certain states of this environment in which certain previously specified goals are achieved.

Agents we will propose to specify according their *functions* performed during their interactions with their outer environment on dependence of their internal states as well as these performed on their internal constituents (states, if any). These functions we will supposed (but we will not define them in this level of rigor!) as well-formalized in the sense that their precise computational analysis is possible. The most interesting property of functions supposed by us will be their *computability* in the sense of theoretical computer science. We will suppose that all the functions which specify the classes in our model (expect perhaps those specifying the conscious agents) are computable in the sense of traditional Turing-computability.

The principal common property of all of artificially designed, and more or less autonomous agents is that their parts (also in many of the cases when these agents are fully reactive) are programmed, so that they have the form of computer programs. In the theoretical level it means that these programs are equivalent with computable functions in the sense of the Turing computability. Accepting the traditional so called *Church-Turing Thesis*[1] this means that agents are not able to do more that the universal Turing machine can compute. In this article we will deal with different types of agents and we will argue that the computational power of certain agents may go behind the above-sketched boundary given by the Church-Turing thesis.

After a short functional stratification of agents we will concentrate to the functional specification of agents with (at least certain low level of) consciousness and we will discuss from a specific computational point of view a possibility of emergence of this kind of consciousness in agents. Our specific view will be based on considering consciousness as an emergent phenomenon appearing in agents thanks to their inner multi-agent like functional structure, and their massive interactions with outer environments. This view is in certain extent inspired by the Fodor's concept of the so called *modularity of mind* (Fodor, 1983) and by the concept of mind as a society of communicating agents in the Minsky's *society theory of mind* (Minsky, 1986).

[1] For more details on the history and formulations of the Church-Turing thesis by W. Sieg see in (Wilson, Keil, 1999, pp. 116-117).

We will formulate and analyze the question how realistic is the proposed functional specification of (conscious) agents from the computational point of view, and we will discusses in certain details a possibility of overcoming the limitations of traditional Turing-computability by a computationally perhaps exotic – the so called hyper-computational – power of (at least minimally) conscious agents.

2 The Traditional Functional-Computational Stratifiction of Agents

In this section we will give a stratification of agents based on their relatively traditional functional specification, in other words according their abilities to couple sensing with actions in a computationally realistic way, and in certain rational ways with respect the goals which agents trying to achieve during their interactions with their environments. In majority of cases mentioned below the reality of computer implementation of functions performed by the agents proves – if we accept the so called Church-Turing hypothesis – their computability in traditional sense. The agents' rationality ranges – thanks to performed computationally sound functioning of them – form very low level rationality of reactive and hysteretic agents – cf. (Kelemen, 1996) for more details about that level of rationality – up to the higher in the cases of deliberative and emotional ones.

2.1 Reactive Agents

The overall framework for the following stratification supposes an agent situated in its real environment (the outer – usually the physical – world with respect the agent's perspective) W, and having certain possibilities to sense this environment. Functionally, we suppose a mapping

$$\textit{sensing}\colon \mathrm{W} \rightarrow \mathrm{S},$$

where S states for the set of all data observable from the agent point of view in the environment W[2]. This is a very important function of agents, because connects of the real environment with the aspects of it which are accessible for the agents thanks to their sensors. Often the distinction between W and S is ignored, but in our stratification it plays some roles, so we will deal with it in the following.

[2] From the formalistic point of view W is only a symbol denoting the all possible states of agents real outer environments. Because of that the definition of the function sensing is perhaps problematic. This function is in fact defined for any agent by its technical sensors and their capacities and quality.

Another important property of any agent is the ability to act in its environment. In order to choice what to do in what situations in order to achieve the followed goal the data sensed by an agent might be associated directly (in a hard-wired manner, for instance) with certain set A of acts performable by the agent. This association might be expressed functionally as follows:

$$\textit{acting}: S \times A \rightarrow W$$

Agents of the resulting functional structure

$$RAG = (W, S, A, \textit{sensing}, \textit{acting}),$$

where both of the functions *sensing* and *acting* are supposed to be computable in the sense of Turing-computability.

Such agents are usually, e.g. in (Arkin, 1998), called *reactive agents*. Some simple examples of such robotic agents may be found in the first part of (Brooks, 1999), for instance.

2.2 Hysteretic Agents

An important property of more complicated agents as the purely reactive ones is their ability to act in its environment not only on the base of sensed data, but on the base of their inner states, too. We denote the set of inner states of such agents by I in the following. All other notations remain unchanged.

Perhaps the simplest case of the use of inner states in agents functional architectures is the definition of a set of explicit goals the agents follow. Act selection then consists in associating of the just sensed data and the explicit (just followed) goal belonging to certain set of goals considered as a subset of the set I of the agent, with activity with certain element of the set A of all possible acts of this agent which it is able to perform in its environment. This coupling of sensed data and the internal state of the agents with acts performed by agents in their environment we express formally as a (computable in the traditional sense) function:

$$\textit{act-selection}: S \times I \rightarrow A$$

The execution of the selected act by an agent environment means certain change of this environment caused by the agent activity in the just actual state of it. So, the acting of such type of agents will be expressed functionally as:

$$\textit{acting}: S \times A \rightarrow W$$

Note that for a given particular $s \in S$ and $i \in I$ the *act-selection* function defines an act $a \in A$. The *acting* is in this particular case applied, of course, to the selected given $s \in S$, and to the before selected act a. So, the function act may be defined also – and perhaps in a more precise, but from other point of view in a less readable form – also as follows:

$$\textit{acting}(s, \textit{act-selection}(s, i)) = w$$

where $w \in W$.

However, we will omit the just demonstrated way of expression of how the function appearing in agents specifications are interrelated, and we believe that it will cause no serious problems with respect of comprehensibility of the text.

Of course, the inner states of the agent may change after executing an act in the environment (by achieving a given goal another goal might become actual, for instance). This change of the agent inner state will be defined by the mapping:

$$\textit{inner-state}: I \times A \rightarrow I$$

Agents of the resulting functional structure

$$HAG = (W, S, A, I, \textit{sensing}, \textit{act-selection}, \textit{acting}, \textit{inner-state})$$

are usually, e.g. in (Genesereth, Nilsson, 1987), called *hysteretic agents*. We mention that all the functions appearing in the specification are supposed to be computable in the traditional sense.

Good examples of this type of agents are those developed in the frame of the so called new NAI (new artificial intelligence) movement of 80ties and 90ties of the past century, esp. under the intellectual influence of papers collected in (Brooks, 1999).

2.3 Deliberative Agents

Deliberative agents are able to select the appropriate sequences of acts for achieving their goals from their starting states as a result of certain level of their 'reasoning', inferring some new data on the base of sensed data and stored knowledge represented in their memories in suitable data and knowledge representation structures.

The inference process – executed usually by some *inference engine* of agents– is based on agents inner states, the sensed (and well-represented knowledge about) states of outer environment of agents, and on goal states (belonging to S, and to I, res.) of agents. This selection process is usually called *planning*, and as a part of I use well-represented *knowledge-bases* of agents. However, at least from the functional point of view, the most important functional properties of those type of agents can be expressed formally by the mapping

$$\textit{planning}: I \times S \times I \rightarrow A^{+}$$

where A^+ states for the set of all nonempty sequences of the elements of A. So, *deliberative agents* might be specified as follows:

$$\text{DAG} = (\text{W, S, A, I}, \textit{sensing}, \textit{act-selection}, \textit{acting}, \textit{inner-state}, \textit{planning})$$

Good examples of deliberative agents are those developed during the research in the field of the so called GOFAI (good old-fashioned artificial intelligence) in 60ties and 70ties of the past century as presented e.g. in (Winston, 1992). Particularly the Shekey/STRIPS system (Fikes, Nilsson, 1071) is usually mentioned as a typical project of this developmental period of AI. The existence of particular deliberative agents proves again the possibility of defining the above-mentioned functions in computationally traditional ways as Turing-computable functions.

3 Emotional and Attentive Agents

During the 90ties of the past century, the problem of robot emotions attracted the interest of the specialists of advanced robotics, artificial intelligence, and cognitive science. M. Minsky (1986) discussed emotions in some details in the context of his society theory of mind, for instance.

Thank to the increasing professional interest, the problem has been shifted from the contemplative level to the level of laboratory design and experimentation with emotion machines. C. Breazeal developed perhaps the first robotic head – the Kismet – with the ability to achieve (and to express externally) its own inner emotional states (Breazeal, 2002).

Analyzing both of the above mentioned approaches to machine emotions form functional perspective we recognize that emotional states of the agents result from the just sensed environment, the just actual internal state of the agents, and of the (possible contradictory) goals with which agents are just confronted. C. Breazeal (2002, p. 110), introducing her robotic head, writes: *The emotions are triggered by various events that are evaluated as being of significance to the "well-being" of the robot*. So, using our formalization, we may write:

$$\textit{emotion}: \text{I} \times \text{S} \times \text{I} \times \text{E} \rightarrow \text{E}$$

where E denotes the set of all emotional states of an agent[3].

[3] It is possible, of course, to suppose also the very realistic situation, that the agent's present emotional state determine in certain extent its internal and/or emotional states in the future, but we do not formalize this situation. However – because we are inspired by Breazeal's

Note two things concerning the above function: First, that we suppose two (not necessary different!) internal states, the actual state of the agent, and its goal state, and, second, that the computed emotional state of an agent depends also on the just actual emotional state of it.

Emotions play important roles in generating and optimizing plans of behaviors of real living agents (human beings) with respect of the necessity of deciding between perhaps contradictory sub-goals generated during the inference processes (and pushing agents into the states of ‘well-being’), when the necessity of conflict resolution appears during the inference processes, or during the processes of bounded rational decision-making. No matter how neutral and rational a goal may seem, it will eventually conflict with other goals if it persists for long enough. No long-term project can be carried out without some defense against competing interest, and this is likely to produce what we call emotional reactions to the conflicts that come about among our most insistent goals, writes M. Minsky (1986, p. 163). So, the question is not whether intelligent machines can have any emotions, but whether machines can be intelligent without any emotions, he continues.

The presence of emotions in the agent functional structure influence the planning process of emotional agents. This fact will be, at certain level of simplification, expressed in our formal model as follows:

$$\textit{emotional-planning}\text{: I} \times \text{S} \times \text{I} \ \times \text{E} \to \text{A}^{+}$$

Therefore, emotion agents may be characterized in the presented typology as:

$$\text{EMAG} = (\text{W, S, A, I, E, } \textit{sensing}\text{, } \textit{act-selection}\text{, } \textit{acting}\text{, } \textit{inner-state}\text{, } \textit{emotion}\text{,}$$
$$\textit{emotional-p planning}\text{)}$$

The so called externally manifested attention – perhaps the best known example of external attention is eye movement and foveation – we will understood as a necessary restriction of the capacity of agents sensory inputs with respect their actual sensed state, and perhaps also their inner states and their emotional states. It may help agents e.g. in the process for selecting their possible actions, and may change their inner and emotional states. More formally, attention as a mapping relates to a tuple consisting of agent’s just actual observed state, its inner state, and its just actual emotional state a tuple of an observable state, an inner state, an action, and an emotional state. Formally:

$$\textit{attention}\text{: S} \times \text{I} \times \text{E} \to \text{S} \times \text{I} \times \text{A} \times \text{E}$$

The *attentive agents* are, according the above understanding of attention, described as:

Kismet as an implemented emotional agent – form the computational perspective it seems for us realistic enough to suppose the function expressing the just mentioned situation to be computationally tractable in the traditional sense.

ATAG = (W, S, A, I, E, *sensing*, *act-selection*, *acting*,
inner-state, *emotion*, *emotional-planning, attention*)

We note that the focusing of agents attention using the above defined function may makes a change only in what the agent senses (this is the main result of focusing the attention) and on its internal and (if the agent is emotional one) also on its emotional state. The result of the focusing of attention by *attention* is then used by function participating in the process of action selection of the agent.

We suppose that the bounded rationality – in the sense of (Simon) – of some agents behaviors may be influenced (or be the result of) focusing the agents attention in the process of their inferences to the 'important: part of their environments, an may reduce the necessary search during the inferences.

However, in present days embodied autonomous agents, e.g. in the robotic upper-torso Cog (Brooks et al., 1999), attention is usually implemented in its simplest form, only, e.g. as a mapping *attention*: S × I → S × I × A. (But for our following purposes the more general form will be more useful.) For instance, the Cog's quite rudimental visual attention works as follows (Brooks et al., 1999): Given a visual stimulus (by waving an object in front of Cog's cameras) it saccades to foveate on that object, and then reach out its arm toward the target. So, the result of attention focusing is the change of the robots sensed state, and it causes also some action(s) performed by the robots arm. The right arm motion is the subject of a learning process, so it s quality depends not only on the observed state of the environment, but also on the robots inner state.

4 Towards Conscious Agents

The main problem with the study and the attempts to model, simulate or implement consciousness in artificially created systems consist in the fact that no for scholars and for engineers have a generallz accepted meaning of the word *consciousness*. Zeman, approaching the study of consciousness from positions of a neurophysiologist with a professional recognized philosophical background, writes on three basic meenings of consciousness, all related with knowledge (Zeman, 2003): *Being awake, our first sense of consciousness*, he writes, *is a pre-condition for acquiring knowledge of all kinds. Once awake, we usually come* – he continues – *by knowledge through ex-perience, the second sense of cons-ciousness. The knowledge we gain is then 'conscious' in the third sense we distinguished*, he completes his an-alysis (Zeman, 2002, p. 36).

From another position – from the position typical for the fields of artificial intelligence and cognitive science – is the subject of consciousness treated by P. O. Haikonen (2003). As a crutial for forming the consciousness he recognize the phenomenon of perception of the self. *What we actually see is only the projected image on the retina, what we actually hear is the vibration of the eardrums, so why don't we just perceive these as such, percepts originating at the senses or originating at the related sensory nerve endings? How can we possibly perceive the situation being anything else?* he asks (Haikonen, 2003, p. 71).

Such questions lead to the questions on the 'internal' (mental) representation of the 'external' (physical) stimuli sensed by machines, and in consequences to the familiar mind/body problem of the philosophy of mind. His position is explained by an example (pp. 248-249): *The operation of the signals in the cognitive machine can be compared to radio transmission where a carrier signal is modulated to carry the actual radio signal. The carrier wave is what is received, yet what is detected is the modulation, the actual sound signal that is in causal connection to the original physical sound via a microphone. We do not hear the carried signal even through without it there would be no music. Thus it is possible to perceive carried informa-tion without the perception of the material basis of the carrier.*

Intuitively we also feel that for an agent to be conscious necessarily requires attentiveness and emotionality. This opinion is expressed clearly e.g. in the Aleksander and Dunmall (2003) attempt to a formal axiomatic definition of consciousness, an approach by which we will inspired in this Section. According the just mentioned authors, being a *conscious agent* means – intuitively and roughly speaking – to have some kind of agent's *private sense* of an outer world, of a self in this world, of self's contemplative planning of when, what and why to act, of self's own inner emotional states Moreover it means also the conscious agents ability to include the self's *private sense* into all of its above mentioned functional capabilities.

How to incorporate the perhaps mysterious *private sense* into the above presented spectra of functional models of different types of agents? Such an effort requires minimally an enlargement of the class of all conscious agents – say the CONAGs – by all possible variants of ATAGs, and to include the CONAG itself (at least in the form of an ATAG) into this class. In this way, some self-reference will be included into the model which will be not remains without some hypotheses concerning some interesting consequences. In simplest types of agents software design, e.g. in certain HAGs, DAGs and ATAGs a very simple variant of this idea is present, however, e.g. in the form of the record of an agent's own position in its inner representation of its outer environment, e.g. in the case of robots like the MetaToto (Stein, 1994), the Shekey/STRIPS (Fikes, Nilsson, 1971), and also in the case of the case of the Cog (Brooks et al., 1999).

There exists also an opinion according which …*consciousness will not need to be programmed in. They will emerge*, states one among the leading personalities of the present days artificial intelligence and advanced robotics in (Brooks, 1999, p. 185). Our main goal consists in the rest of this paper in arguing form the theoretical computer science point of view for the *possibility* that consciousness *may* emerge in such a way under some circumstances (but not for *necessity* that it *will* emerge).

First of all, how to define functionally the *private sense* and how to enlarge the general functional structure of CONAGs by a suitable function? In present days mathematics it is technically extremely unusual to define functions appearing as members of their own domain of variables and domains of values. (The closest – but not identical! – concept is perhaps the more computationally oriented – so-called *recursive* – definitions of functions.)

Perhaps the simplest way how to include the simplest form of the private sense of a given particular ATAG, say B, consists in defining the mapping

$$\textit{private-sense}_B\colon S_B \times \{B\} \rightarrow I_B$$

and include this mapping into the functional description of this agent B which we will call a *minimally conscious agent*. So, we have a new type of agents, the type we will call MICONAG, functionally characterized as:

$$\text{MICONAG} = (W, S_{\text{MICONAG}}, A, I_{\text{MICONAG}}, E, \textit{sensing}, \textit{act-selection}, \textit{acting}, \textit{inner-state}, \textit{emotion}, \textit{emotional-planning}, \textit{attention}, \textit{private-sense}_{\text{MICONAG}})$$

Note that the MICONAG type of agents definition refers to their '*itselfs*' in some similar manner as it was in the case of the above mentioned recursive definition of the functions. However, the situation is much more complicated as in the case of simple recursive definitions because of the structure of any agent belonging into the class of the MICONAG consists not only in one mapping, but contains much more, and the 'cross-references' of functions which specifies the agent functionally, are very huge. Particularly, the (physical) environments of agents are usually noisy, dynamic, full of unpredictable changes, etc. and all these influence the behavior of agents situated in them. Because of that we may suppose that the not only the behavior of agents, but also the states of their consciousness do not be computable in the traditional sense of Turing-computability.

Nevertheless, in the following sections we show a possibility how agents constructed from simple parts, e.g. according the simple architectural principle of subsuming, may produce – under some not unrealistic conditions – more than the traditional Turing machine can compute.

5 Computational Analysis of the *Privat Sense*

The above mentioned experimental robots have from our perspective one important common feature: At least in certain extent, their behavioral, representational, and decision making capacities are fundamentally based on the abilities of the present day computers to execute more or less complicated computations. In the theoretical level, these computations might be reduced into the form of the theoretical abstraction of computational processes known as Turing-computation. The Turing machine programs, with respect of their computational power in the machines over-simplified environments of symbols on a tape, and a head going one step left or right and rewriting the symbols according simple instructions sequences, differs in some senses very significantly from the real agents (e.g. embodied robots) situated in dynamically changing (data or physical) environments, and interacting with these environments very massively in many different ways. However, there exists a largely accepted hypothesis in theoretical computer science – the already mentioned Church-Turing hypothesis – according which all what is intuitively in certain sense computable (so, transformable from certain inputs into certain outputs according precisely defined and exactly executed sequences of rules – according computer programs) is computable by a Turing machine.

Especially interactions are very appealing for re-consideration of the form of 'computation' performed by such agents, and for drawing perhaps new boundaries between what we consider as computable and what as non-computable[4]. In present day theoretical computer science there are efforts to demonstrate that the notion of computation might be enlarged beyond the traditional boundaries of the Turing-computability[5]. In (Burgin, Klinger, 2004) is proposed to call algorithms and automata that are more powerful than Turing machines as *super-recursive*, and computations that cannot be realized or simulated by Turing machines as *hyper-computations*.

To have the *private sense* as sketched in the previous section means – metaphorically speaking – to have an ability of a given agent AG to consider itself as another agent identical with AG, and to consider this type of 'schizophrenia' in the work of other functions which characterize the agent AG. This type of recursion is too complicated for expressing it in the frame of the traditional paradigm of one-processor computation. It requires at least some suitable framework for dealing with behaviors that appear thanks to interrelations between individually autonomous entities (agents).

The appearing situation insinuates the framework of considering a conscious agent as a system consisting in more then one agent, so in a form of a multi-agent systems. Might be that the phenomenon of consciousness *emerges*[6] from

[4] Cf. (Wegner, 1997).

[5] For more details on the effort, see e.g. (Eberbach, Wegner, 2003) or the monothematic issue of the *Theoretical Computer Science* **317**, No. 1-3 (2004) 1-269.

[6] Whether the nature of consciousness is emergent or not, it is an actual question not of the present day cognitive science only, but also of the philosophy of mind. Freeman (2001) and Holland (2003) provides the overview of different opinions present in both of the fields.

interactions of such virtual agents representing of ourselves and from representations of other agents in our individual minds. The function private sense mentioned in the previous section is perhaps also an emerging product of interactions of several other functions. Inspired by Fodor's view (Fodor, 1983) we may suppose a conscious agent as structured into certain simpler 'modules' or – inspired by views of M. Minsky (Minsky, 1986) – formed as a society of simpler, unconscious and massively interacting 'agents'. Perhaps the conscious behavior of such an agent might be then described as a phenomenon, which emerges – in the sense proposed in (Holland, 1998)[7] – from interactions of traditionally computable behaviors of simpler constituting parts of it, and has the form of a hyper-computation. Let us try to demonstrate in the next section how it is possible to proceed in this way.

In the following Section we will try to connect the above mentioned computational problems of conscious agents with one particular formal – in certain extent rule-based – model of hyper-computations, and illustrate that from simple, massively interacting components may emerge the hyper-computational power if relatively slightly imaginable influences of real physical environments of agents are in some way taken into consideration.

6 Increasing Computational Power

In this Section we will illustrate that there exists at least one rigorous formal model of agents with components producing a rule-governed Turing-computable behaviors each, but producing – considered as a whole – a behavior which does not be generated traditionally by any Turing-equivalent generative device, so which requires the generative power of hyper-computation. We will consider in this role the so-called eco-grammar systems. First, we introduce in a few words this model, presented originally in (Csuhaj-Varjú et al., 1997).

An *eco-grammar system* (or an EG system for short) consists of several abstract, formally specified autonomous entities called components. *Components* are described by strings of symbols (with the capability to developing according precisely defined so called Lindenmayer-type, or L- rules) acting on their *inner environment*, by pure rewriting of symbols in this string-like *outer environment* using precisely defined rules applied sequentially. The outer environment

[7] In (Holland, 1998, pp. 121-122) emergence is explained as '... about all a product of coupled, context-dependent interactions. Technically these interactions, and the resulting system, are nonlinear: The behavior of the overall system cannot be obtained by summing the behaviors of its constituent parts... However, we can reduce the behavior of the whole to the lawful behavior of its parts, if we take nonlinear interactions into account' (emphasized by J. H. Holland).

described by a string of symbols may also develop according to a set of L-rules. The rules used by each component for development depend on the state of its inner environment, and on the state of its outer environment. The rules used for acting in the environment depend on the state of the components.

More formally – according (Csuhaj-Varjú et al., 1997) – an *eco-grammar system* Σ consists, roughly speaking, of

- a finite alphabet V,
- a fixed number (say n) of components evolving according sets of rules $P_1, P_2, ..., P_n$ applied in a parallel way as it is usual in L-systems (Rozenberg, Salomaa, 1980), and of
- an environment of the form of a finite string over V (the states of the environment are described by strings of symbols w_E, the initial one by w_0).
- the functions φ and ψ which define the influence of the environment and the influence of other components, respectively, to the components (these functions will be supposed in the following as playing no roles, and will not be considered in the model of eco-grammar systems as treated in this article).

The rules of components depend, in general, on the state (on the just existing form of the string) of the environment. The particular components act in the commonly shared environment by sets of sequential rewriting rules $R_1, R_2, ..., R_n$. The environment itself evolves according a set P_E of rewriting rules applied in parallel as in L systems.[8]

The evolution rules of the environment are independent on components' states and of the state of the environment itself. The components' actions have priority over the evolution rules of the environment. In a given time unit, exactly those symbols of the environment that are not affected by the action of any agent are rewritten.

In the EG-systems we assume the existence of the so-called *universal clock* that marks time units, the same for all components and for the environment, and according to which the evolution of the components and of the environment is considered.

In (Csuhaj-Varjú, Kelemenová, 1998) a special variant of EG-systems have been proposed in which components are grouped into subsets of the set of all components – into the so-called *teams* – with fixed number of members. The idea was to express in the model the embodiment (the technical side) of components in certain computationally tractable form. Especially in this case through limitation of activities of components by some physical constrains, for instance.

In (Wätjen, 2003) a variant of EG-systems without internal states of components is studied. The fixed number of members proposed in (Csuhaj-Varjú,

[8] So, the triplet (V, P_E, w_E) is (and works as) a Lindenmayer-system.

Kelemenová, 1998) is, however, replaced by a dynamically changing number of components in teams.

As the mechanism of reconfiguration, a function, say *f*, is defined on the set *N* of integers with values in the set *{0, 1, 2, ...,n}* (where *n* is the number of components in the corresponding EG-system) in order to define the number of components in teams. For the *i*-th step of the work of the given EG-system, the function *f* relates a number $f(i) \in \{0, 1, 2, \ldots n\}$. The subset of the set of all components of thus EG-system of the cardinality *f(i)* is then selected for executing the next derivation step of the EG system working with Wätjen-type teams.

Wätjen (2003) proved, roughly speaking, that there exist EG-systems such that if *f* is (in the traditional sense) non-recursive function, then the corresponding EG-system generates a non-recursive (in fact a super-recursive) language.

Whether or not has the computational power of EG systems received in the above described way really emerges from the recursive computations of suitable configurations of its components we may test using the test of emergence proposed in (Ronald et al., 1999). The tries offers an operant definition of emergence (esp. for phenomena appearing in the experiments of Artificial Life, but we consider its validity also for the study of agents and multi-agent systems). The requirements putted onto systems in which the emergence of some phenomenon appears are the following (Roland et al., 1999):

Design. The designer designs the systems by describing *local* interactions between components in a language L_1.

Observations. The observer describes global behaviors of the running system using a language L_2.

Surprise. The language of design L_1 and the language of observation L_2 are distinct, and the causal link between the elementary interactions programmed in L_1 and the observations observed in L_2 is non-obvious.

The emergent nature of the behavior (language) generated by the above described EG system is clear, because of the components of the given EG system generate recursive languages each, the local interactions of the components are given only and, surprisingly the whole system generates a non-recursive language (behavior).

From the technical point of view some surprising properties of components – e.g. their consciousness – may emerge from their simple parts and from interactions of these parts instead of being implemented in certain clever way into the systems. This idea may be supported not only by some theoretical considerations (or speculations) but also by the feeling of those who construct more or less intelligent robots. To illustrate that we quote R. Brooks again: *Thought and consciousness will not need to be programmed in. They will emerge* (Brooks, 1999, p. 185).

7 Embodiment and Hyper-Computation

The above-sketched Wätjen's results might be interpreted in the context proposed in this contribution for characterization of agents as a proof of the following statement:

An agent considered as a multi-agent system in the meaning as used in the theory of eco-grammar systems where the changing number of sub-agents which in a given moment actively participate in the generation of the behavior of the whole system

- might be described as consisting of rule-governed parts (modules or simpler agents) with abilities to perform traditional Turing-type computations, and
- is able to produce a behavior beyond the limits of traditional computability which seems to be necessary for appearing of the phenomenon of the agent's consciousness.

One among the most important achievement during the development of Artificial Intelligence (AI) was the discovery of the methodologically new possibility how to test our hypotheses on how (some of) the intellectual processes run. The history of AI is full of different hypotheses on how to 'automate' processes like general problem solving, theorem proving, natural language understanding and communication, diagnostics, image processing and recognition, scene analysis, etc. in order to obtain working computer-based systems performing these tasks at the similar (or at better) qualitative level as (specially trained) human beings perform them. In all these cases

(1) a working hypothesis is produced first – in the majority of the cases it is based on author's own introspection, then

(2) the formulated hypothesis is implemented (often using a suitable programming language that might be developed for such purposes), and

(3) the developed system of programs (the implemented version of the hypothesis) is then tested on real (or more or less similar to the real ones) data.

To proceed according such methodological guidance seems to us as something natural. It might be because intellectually we feel prepared for contemplations about our own intellectual capacities.

The situation is completely different in the cases when the agents (intended to be intelligent in certain sense, e.g. cognitive robots) are situated and execute tasks in real physical environments. In such a case the systems are faced with physically grounded ontologies of objects with real physical properties that exist and act in real time scales. Very hard problems appearing in such situations in the traditional AI were pointed out first from very different positions and with very different conclusions by M. Minsky, cf. e.g. (1986), and R. Brooks, cf. e.g. (1999).

Brooks in his concept of the novel AI emphasizes the principal role of systems reactivity, which is necessary for their low-level rationality, while Minsky emphasizes the principle of decentralization and organization of simplest units (agents) into more complex ones (agencies) and presupposes that an agency may play the role of an agent in a more complex agency. Both of these positions might be – according to our conviction – combined into one unified approach. The main idea consists in two basic steps:

(1) to emphasize the role of as direct as possible interaction of the cognitive systems with their environments at least at the lowest level of sensing and acting, and

(2) to exploit the power of organization and of the emergence in highest levels in order to receive more complex behaviors.

Both of the above mentioned steps lead us to realize the principal difference between *implementation* of our ideas on how cognitive processes run in natural systems and how they may run in artificial ones, into more or less traditional but in certain sense rigid computers usually equipped with suitable input-output devices which isolate them form their environments by providing data from it for them, and between *embodiment* of our ideas into artificially created agents.

Realize now that our physically embodied and working in real physical environments agents are constructed form physically engineered and constructed (it means functionally not completely reliable) parts like sensors providing signals for them, units for processing signals and perhaps compute the decisions, actuators for making changes in their environments, and situated and working continuously in real, dynamic, and noisy environments. Realizing that the example and the theoretical result proved on the mathematically sound model of agents shown in the previous Section illustrate the possible influence of different physical (so, related with the 'body' of agents) states of components of agents into their computational properties.

Supposing a technically relatively acceptable situation – very well known for all those who do experiments with real embodied robots, for instance – that the components of robots are unreliable in certain level, we may interpret the Wätjen's model of EG systems as an acceptable model of agents with very simple architecture (reflected in very simple communication between parts forming a whole agent) and having a hyper-computational power. This computational power is perhaps the necessary force, which pushes agents from the traditional Turing-computable behaviors toward their consciousness.

Conclusions

Concluding the above notes and positions we state the following: It seems to be realistic that he deep philosophical and ethical question 'To have conscious agents or not?' is reducible to the much more technical question 'To have much more complicated robots as we have now, or not?' We have demonstrated in a rather formal way that the consciousness of agents *may* emerge as R. Brooks has supposed that.

Acknowledgement

The present work was supported bz the grant MSM 4781305903. The author's research on the topic is partially supported by the Czech Science Foundation Grant No. 201/04/0528, and by Gratex International, Inc., Bratislava, Slovakia.

References

[1] Aleksander, I., Dunmall, B.: Axioms and Tests for the Presence of Minimal Consciousness in Agents. In: *Machine Consciousness* (O. Holland, Ed.). Imprint Academic, Exeter, 2003, pp. 7-18

[2] Arkin, R. C.: *Behavior-Based Robotics*. The MIT Press, Cambridge, Mass., 1998

[3] Breazeal, C.: *Designing Sociable Robots*. The MIT Press, Cambridge, Mass., 2002

[4] Brooks, R. A.: *Cambrian Intelligence*. The MIT Press, Cambridge, Mass., 1999

[5] Brooks, R. A., Breazeal, C., Marjanovic, M., Scassellati, B., Williamson, M. M.: The Cog Project – Building a Humanoid Robot. In: *Computation for Metaphors, Analogy, and Agents* (C. Nehaniv, Ed.). Springer, Berlin, 1999, pp. 52-87

[7] Burgin, M., Klinger, A.: Three Aspects of Super-Recursive Algorithms and Hyper-Computation or Finding Black Swans. *Theoretical Computer Science* **317** (2004) 1-11

[8] Csuhaj-Varjú, E., Dassow, J., Kelemen, J., Paun, Gh.: *Grammar Systems*. Gordon and Breach, Yverdon, 1994

[9] Csuhaj-Varjú, E., Kelemen, J., Kelemenová, A., Paun, Gh.: Eco-Grammar Systems – a Grammatical Framework for Lifelike Interactions. *Artificial Life* **3** (1997) 1-28

[10] Csuhaj-Varjú, E., Kelemenová, A.: Team Behaviour in Eco-Grammar Systems. *Theoretical Computer Science* **209** (1998) 213-224

[11] Eberbach, E., Wegner, P.: Beyond Turing Machines. *Bulletin of the EATCS* **81** (2003) 279-304

[12] Fikes, R. E., Nilsson, N. J.: STRIPS – a New Approach to the Application of Theorem Proving in Problem Solving. *Artificial Intelligence* **2** (1971) 189-208

[13] Fodor, J.: *The Modularity of Mind.* The MIT Press, Cambridge, Mass., 1983

[14] Freeman, A. (Ed.): *The Emergence of Consciousness*. Imprint Academic. Thorverton, 2001

[15] Genesereth, M. R., Nilsson, N. J.: *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, Los Altos, Cal., 1987

[16] Haikonen, P. O.: The Cognitive Approach to Conscious Machines. Imprint Academic, Exeter, 2003

[17] Holland, J. H.: *Emergence*. Addison-Wesley, Reading, Mass., 1998

[18] Holland, O. (Ed.): *Machine Consciousness*. Imprint Academic. Exeter, 2003

[19] Kelemen, J.: A Note on Achieving Low-Level Rationality from Pure Reactivity. *Journal of Experimental and Theoretical Artificial Intelligence* **8** (1996) 121-127

[20] Kelemen, J.: May Embodiment Cause Hyper-Computation? In: *Advances in Artificial Life* (eds. M. S. Capcarrére et al.) Springer, Berlin, 2005, pp. 31-36

[21] Minsky, M.: *The Society of Mind*. Simon and Schuster, New York, 1986

[22] Ronald, E. M. A., Sipper, M., Capcarrére, M. S.: Design, Observation, Surprise! A test of Emergence. *Artificial Life* **5** (1999) 225-239

[23] Rozenberg, G., Salomaa, A.: *Theory of L-Systems*. Academic Press, New York, 1980

[24] Simon, H. A.: *The Science of Artificial, 2nd Edition*. The MIT Press, Cambridge, Mass., 1982

[25] Stein, L. A.: Imagination and Situated Cognition. *Journal of Experimental and Theoretical Artificial Intelligence* **6** (1994) 393-407

[26] Tecuci, Gh.: *Building Intelligent Agents*. Academic Press, San Diego, Cal., 1998

[27] Wätjen, D.: Function-Dependent Teams in Eco-Grammar Systems. *Theoretical Computer Science* **306** (2003) 39-53

[28] Wegner, P.: Why Interaction is more Powerful than Algorithms. *Communications of the ACM* **40** (1997) No. 5, 81-91

[29] Wilson, R. A., Keil, F. C. (eds.): *The MIT Encyclopedia of the Cognitive Sciences.* The MIT Press, Cambridge, Mass., 1999

[30] Winston, P. H.: *Artificial Intelligence, 3rd Edition*. Addison-Wesley, Reading, Mass., 1992

[31] Zeman, A.: *Consciousness – A User's Guide*. Yale University Press, New Haven, 2002

Tuning of Fractional Controllers Minimising H_2 and H_∞ Norms

Duarte Valério, José Sá da Costa

Department of Mechanical Engineering, Instituto Superior Técnico, Technical University of Lisbon, GCAR, Av. Rovisco Pais, 1049-001 Lisboa, Portugal, e-mail: {dvalerio, sadacosta}@dem.ist.utl.pt

Abstract: H_2 and H_∞ controllers minimise the H_2 or the H_∞ norm of a suitable loop transfer function involving the plant to control and some weighing transfer functions chosen to fulfil performance specifications. In this paper this type of controllers is developed for the case when the plant and / or the weighing transfer functions are of fractional (commensurate) order. Since no analytical results similar to those existing for the integer case have been found, a genetic algorithm is used to minimise the desired norm. An application to temperature control is used to illustrate the method.

Keywords: H_2 and H_∞ controllers, fractional plants, genetic algorithms, temperature control

1 Introduction

Over the last decades, a technique for devising controllers has been developed that consists essentially in minimising the H_2 or the H_∞ norm of the control loop. These two norms have simple interpretations: to put it plainly, the H_2 norm reflects how much a dynamic system amplifies – or attenuates – its input over all frequencies, and the H_∞ norm reflects how much a dynamic system amplifies – or attenuates – its input at the frequency at which the amplification is maximal. This control technique may be applied to both SISO (single-input, single-output) and MIMO (multiple-input, multiple-output) plants, and its results achieve a remarkable robustness (Lublin *et al.*, 1996; Doyle *et al.*, 1989).

The usual method for developing this sort of controllers is based upon a state-space representation of the plant. It cannot, unfortunately, be applied to plants of fractional order – that is to say, to plants that have a dynamic behaviour corresponding to differential equations involving fractional derivatives. State-space representations for such plants do exist (Malti *et al.*, 2003), but, since they involve fractional derivatives of the states, the algorithms developed for integer-order plants (those with a dynamic behaviour corresponding to differential equations involving usual, integer-order derivatives only) cannot be directly transposed to fractional-order plants.

In this paper this problem is addressed by using a numerical minimisation method to perform the minimisation of the H_2 or the H_∞ norm when the plant is of fractional order. The following sections cover the following issues. Section 2 briefly summarizes a few results from the theory of fractional calculus. Section 3 summarizes algorithms for reckoning the H_2 norm of a plant. Section 4 summarizes algorithms for reckoning the H_∞ norm of a plant. Section 5 describes the control loop usually employed together with H_2 and H_∞ controllers. Section 6 describes algorithms to minimise an appropriate norm of that control loop. Section 7 documents an example of application of this control strategy to a fractional order plant. The paper closes with some conclusions.

2 Fractional Order Systems

Fractional calculus is a generalisation of ordinary calculus. Its main idea is to develop a functional operator *D*, associated to an order ν (not restricted to integer numbers), that generalises the usual notions of derivatives (for a positive ν) and integrals (for a negative ν). There are several alternative definitions of operator *D*; the one addressed here is due to the works of Grünwald and Letnikoff. It generalises the usual definition of derivative:

$$Df\left(x\right) \overset{\text{def}}{=} \lim_{h\to 0} \frac{f\left(x\right)-f\left(x-h\right)}{h} \tag{1}$$

It is easily proved by mathematical induction that higher order derivatives are given by

$$D^n f\left(x\right) = \lim_{h\to 0} \frac{\sum_{k=0}^{n}\left(-1\right)^k \binom{n}{k} f\left(x-kh\right)}{h^n}, \quad n\in\mathbb{N} \tag{2}$$

The gamma function, defined as

$$\Gamma\left(x\right) \overset{\text{def}}{=} \begin{cases} \int_0^{+\infty} e^{-y} y^{x-1} dy, \text{ if } \ x\in\mathbb{R}^+ \\ \dfrac{\Gamma\left(x+n\right)}{\prod_{i=0}^{n-1}\left(x+i\right)}, \ n=-\lfloor x\rfloor, \text{ if } \ x\in\mathbb{R}^-\backslash\mathbb{Z} \end{cases} \tag{3}$$

has the convenient property of generalising the factorial to all real numbers (with the exception of negative integers, for which it is not defined), since

$$\Gamma(n+1) = n!,\ n \in \mathbb{N} \tag{4}$$

and this property may be used to generalise combinations for non-natural numbers as follows:

$$\binom{a}{b} \overset{\text{def}}{=} \frac{\Gamma(a+1)}{\Gamma(b+1)\Gamma(a-b+1)} \tag{5}$$

As a consequence, (2) gives the same result as

$$D^n f(x) = \lim_{h\to 0} \frac{\sum_{k=0}^{m} (-1)^k \binom{n}{k} f(x-kh)}{h^n}, \quad m,n \in \mathbb{N} \wedge m > n \tag{6}$$

since

$$\binom{n}{k} = \frac{\Gamma(n+1)}{\Gamma(k+1)\Gamma(n-k+1)} = \frac{n!}{k!\infty} = 0 \tag{7}$$

And expression (6) may be generalised for non-natural differentiation orders as follows:

$$D^\nu f(x) = \lim_{\substack{h\to 0\\ m\to+\infty}} \frac{\sum_{k=0}^{m} (-1)^k \binom{\nu}{k} f(x-kh)}{h^\nu} \tag{8}$$

Two things should be taken into account. First, the upper limit of the summation *m* has to be taken up to infinity because, since ν need not be integer, terms will not be zero from some value of *k* on. Second, when ν is a negative integer, the result should be an integration, and it would be good to know which integration indexes result from using this definition. This question is easy to answer for $\nu = -1$; we will have

$$D^{-1} f(x) = \lim_{\substack{h\to 0\\ m\to+\infty}} \frac{\sum_{k=0}^{m} (-1)^k \frac{(-1)^k \Gamma(k+1)}{\Gamma(k+1)\Gamma(1)} f(x-kh)}{h^{-1}} = \lim_{\substack{h\to 0\\ m\to+\infty}} \sum_{k=0}^{m} f(x-kh)h \tag{9}$$

Now this is the Riemann definition of integral ${}_cD_x^{-1} f(x)$ if $h = (x-c)/m$. So, for orders other than –1, we will have

$${}_cD_x^\nu f(x) \overset{\text{def}}{=} \lim_{\substack{h\to 0\\ mh=x-c}} \frac{\sum_{k=0}^{m} (-1)^k \binom{\nu}{k} f(x-kh)}{h^\nu} \tag{10}$$

This is the Grünwald-Letnikoff definition of fractional derivative. Notice that when ν is a non-integer positive number operator *D* still needs integration limits; in other words, *D* is a local operator only when ν is a natural number (the case of usual derivatives). Thorough expositions of the theory of fractional calculus may be found in (Miller and Ross, 1993; Podlubny, 1999; Samko *et al.*, 1993).

The Laplace transform of *D* follows rules similar to those for integer derivatives and integrals:

$$\mathcal{L}\left[{}_0D_x^\nu f(x)\right] = s^\nu F(s) \tag{11}$$

Zero initial conditions being assumed, systems with a dynamic behaviour described by differential equations involving fractional derivatives give rise to transfer functions with fractional powers of *s*.

«Fractional» calculus and «fractional» order systems are the usual names though ν may assume irrational values as well – definition (10) handles both cases without distinction. In practice all orders are known with limited precision and all orders may indeed be assumed rational. Fractional transfer functions *G*(*s*) dealt with become far more manageable if there is a *Q* verifying

$$G(s) = \frac{\sum_{k=1}^{n} a_k s^{k/Q}}{\sum_{k=1}^{m} b_k s^{k/Q}}, \quad Q \in \mathbb{N}\,. \tag{12}$$

Such fractional transfer functions are called commensurate. The rational commensurate order is 1/*Q*. All transfer functions addressed hereafter are assumed to be like (12). Such transfer functions may be used to model several physical systems in different areas, such as heat transfer, diffusion, behaviour of viscoelastic materials, electrical circuits, and many more (Podlubny, 1999, pp. 243-308).

3 The H$_2$ Norm

The H$_2$ norm of a transfer function matrix *G* is defined as

$$\|G\|_2 = \sqrt{\frac{1}{2\pi}\int_{-\infty}^{+\infty} \mathrm{tr}\left[G(j\omega)\overline{G(j\omega)^T}\right] d\omega}\ . \tag{13}$$

Now let **A** be a matrix with *m* lines and *n* columns. Then the product $\overline{\mathbf{A}\mathbf{A}^T}$ is a square matrix with *m* lines and *m* columns. Its elements are given by

$$\left[\mathbf{A}\overline{\mathbf{A}^T}\right]_{ij} = \sum_{k=1}^{n} \mathbf{A}_{ik} \overline{\left[\mathbf{A}^T\right]_{kj}} = \sum_{k=1}^{n} \mathbf{A}_{ik} \overline{\mathbf{A}_{jk}} \tag{14}$$

Elements in the main diagonal are given by

$$\begin{aligned} &\left[\mathbf{A}\overline{\mathbf{A}^T}\right]_{ii} = \\ &= \sum_{k=1}^{n} \mathbf{A}_{ik} \overline{\mathbf{A}_{ik}} = \\ &= \sum_{k=1}^{n} \left(\mathrm{Re}\left[\mathbf{A}_{ik}\right] + j\,\mathrm{Im}\left[\mathbf{A}_{ik}\right]\right)\left(\mathrm{Re}\left[\mathbf{A}_{ik}\right] - j\,\mathrm{Im}\left[\mathbf{A}_{ik}\right]\right) = \\ &= \sum_{k=1}^{n} \mathrm{Re}^2\left[\mathbf{A}_{ik}\right] + \mathrm{Im}^2\left[\mathbf{A}_{ik}\right] = \sum_{k=1}^{n} \left|\mathbf{A}_{ik}\right|^2 \end{aligned} \tag{15}$$

From the definition of trace, we get

$$\mathrm{tr}\left[\mathbf{A}\overline{\mathbf{A}^T}\right] = \sum_{i=1}^{m}\sum_{j=1}^{n} \left|\mathbf{A}_{ij}\right|^2 \tag{16}$$

The result above means that, for a MIMO system with *n* lines and *m* columns,

$$\left\|G\right\|_2 = \sqrt{\sum_{j=1}^{n}\sum_{i=1}^{m} \left\|G_{ij}\right\|_2^2}\,. \tag{17}$$

So the problem of finding the H_2 norm is reduced to SISO systems.

3.1 Integer Plants

Let *G* be an integer order system given by the state-space representation

$$\begin{aligned} \dot{x} &= \mathbf{A}x + \mathbf{B}u \\ y &= \mathbf{C}x + \mathbf{D}u \end{aligned} \tag{18}$$

Then its H_2 norm may be found solving one of the equations (Lublin *et al.*, 1996, p. 636)

$$\mathbf{A}\mathbf{L}_C + \mathbf{L}_C\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = \mathbf{0} \tag{19}$$

$$\mathbf{A}^T\mathbf{L}_O + \mathbf{L}_O\mathbf{A} + \mathbf{C}^T\mathbf{C} = \mathbf{0} \tag{20}$$

for $\mathbf{L}_C$ (the controllability gramian matrix) or $\mathbf{L}_O$ (the observability gramian matrix). (These two linear matrix equations belong to a class of equations known as Riccatti type equations.) Then

$$\left\|\mathbf{G}\right\|_2 = \sqrt{\mathrm{tr}\left(\mathbf{C}\mathbf{L}_C\mathbf{C}^T\right)} \tag{21}$$

$$\left\|\mathbf{G}\right\|_2 = \sqrt{\operatorname{tr}\left(\mathbf{B}^T \mathbf{L}_C \mathbf{B}\right)} \tag{22}$$

3.2 Fractional Plants

No results similar to (21) or (22) have been found for fractional systems, but the norm may be found as follows (Malti *et al.*, 2003). Let 1 / *Q* be the commensurate order of the system, and

$$\left.\begin{matrix} q = \mathbf{C}\ (Q) \\ p = Q - q \end{matrix}\right\} \Rightarrow p + q = Q \tag{23}$$

$$x = \omega^{1/Q} \Leftrightarrow \omega = x^Q \Rightarrow d\omega = Qx^{Q-1}dx \tag{24}$$

Since the complex conjugate may be obtained changing the sign of the imaginary part, we will have

$$\left\|G\right\|_2^2 = \frac{1}{\pi}\int_0^{+\infty} Qx^{Q-1} G(jx) G(-jx)\, dx \tag{25}$$

Let *A* and *B* be the polynomials in the numerator and denominator of product

$$G(jx)G(-jx) = \frac{A(x)}{B(x)} \tag{26}$$

Notice that the imaginary unit *j* has been considered as part of polynomials *A* and *B*. If we now let

$$\frac{x^q A(x)}{B(x)} = \frac{R(x)}{B(x)} + \sum_{k=0}^{q+\deg(A)-\deg(B)} a_k x^k \tag{27}$$

(where deg(*P*) represents the degree of polynomial *P*), then (25) becomes

$$\left\|G\right\|_2^2 = \frac{Q}{\pi}\int_0^{+\infty} x^{p-1} x^q \frac{A(x)}{B(x)} dx = \frac{Q}{\pi}\int_0^{+\infty} x^{p-1}\left(\frac{R(x)}{B(x)} + \sum_{k=0}^{q+\deg(A)-\deg(B)} a_k x^k\right) dx \tag{28}$$

Three cases are to be distinguished when reckoning (28).

3.2.1 Case $q + \deg(A) - \deg(B) > 0$

In this case the summation in (28) is not identically zero. Its integral will be

$$\int_0^{+\infty} \sum_{k=0}^{q+\deg(A)-\deg(B)} a_k x^{k+p-1} dx = \sum_{k=0}^{q+\deg(A)-\deg(B)} \left[\frac{a_k}{k+p} x^{k+p}\right]_0^{+\infty} \tag{29}$$

and since k is 1 or higher and p is positive,

$$x^{k+p}\Big|_{x=+\infty}=+\infty\Rightarrow\|G\|_2=\infty \tag{30}$$

3.2.2 Case $q+\deg(A)-\deg(B)\le 0\wedge p\ne 0$

In this case let $B(x)$ have b different poles, $s_1,s_2,\ldots s_b$, and let m_k be the multiplicity of pole s_k. Then we may perform a partial fraction expansion of

$$\frac{x^qA(x)}{B(x)}=\sum_{k=1}^{b}\sum_{n=1}^{m_k}\frac{a_{k,n}}{\left(x+s_k\right)^n} \tag{31}$$

(Recall that a pole of multiplicity m will appear m times in the expansion.) Then (28) becomes

$$\|G\|_2^2=\frac{Q}{\pi}\int_0^{+\infty}\sum_{k=1}^{b}\sum_{n=1}^{m_k}\frac{a_{k,n}x^{p-1}}{\left(x+s_k\right)^n}\,dx=\frac{Q}{\pi}\sum_{k=1}^{b}\sum_{n=1}^{m_k}\frac{a_{k,n}}{s_k^n}\int_0^{+\infty}\frac{x^{p-1}}{\left(\frac{x}{s_k}+1\right)^n}dx \tag{32}$$

This becomes (Gradshteyn and Ryzhik, 1980, p. 285)

$$\begin{aligned}&\|G\|_2^2=\\&=\frac{Q}{\pi}\sum_{k=1}^{b}\sum_{n=1}^{m_k}\frac{a_k}{s_k^n}(-1)^{n-1}\frac{\pi}{s_k^{-p}}\binom{p-1}{n-1}\operatorname{cosec}(p\pi)=\\&=\sum_{k=1}^{b}\sum_{n=1}^{m_k}\frac{(-1)^{n-1}Qa_ks_k^{p-n}\binom{p-1}{n-1}}{\sin(p\pi)}\end{aligned} \tag{33}$$

3.2.3 Case $q+\deg(A)-\deg(B)\le 0\wedge p=0$

In this case the integration rule quoted above cannot be applied, so an alternative expansion is carried out. Let s_1 be one of the poles of (31), arbitrarily chosen. (Actually it would be better to choose the one minimising numerical errors, but it is difficult to know beforehand which one does.) Then we may write

$$\frac{x^{q-1}A(x)}{B(x)}=\sum_{k=2}^{b}\frac{c_k}{\left(x+s_1\right)\left(x+s_k\right)}+\sum_{k=1}^{b}\sum_{n=2}^{m_k}\frac{d_{k,n}}{\left(x+s_k\right)^n} \tag{34}$$

Notice that poles with multiplicity 1 do not appear in the second summation. After some straightforward calculus, expression (28) becomes

$$\begin{aligned}
\|G\|_2^2 &= \frac{Q}{\pi}\left(\sum_{k=2}^{b}\int_0^{+\infty}\frac{c_k}{\left(x+s_1\right)\left(x+s_k\right)}dx+\sum_{k=1}^{b}\sum_{n=2}^{m_k}\int_0^{+\infty}\frac{d_{k,n}}{\left(x+s_k\right)^n}dx\right)= \\
&=\sum_{k=2}^{b}\left[\frac{Q}{\pi}\int_0^{+\infty}\left(\frac{1}{x+s_1}+\frac{-1}{x+s_k}\right)\frac{c_k}{s_k-s_1}dx\right]+\sum_{k=1}^{b}\sum_{n=2}^{m_k}\left[\frac{Q}{\pi}\int_0^{+\infty}\frac{d_{k,n}}{\left(x+s_k\right)^n}dx\right]= \\
&=\sum_{k=2}^{b}\left\{\frac{Qc_k}{\pi\left(s_k-s_1\right)}\left[\ln\frac{x+s_1}{x+s_k}\right]_0^{+\infty}\right\}+\sum_{k=1}^{b}\sum_{n=2}^{m_k}\left\{\frac{Qd_{k,n}}{\pi\left(-n+1\right)}\left[\left(x+s_k\right)^{-n+1}\right]_0^{+\infty}\right\}= \\
&=\sum_{k=2}^{b}\left[\frac{Qc_k}{\pi\left(s_k-s_1\right)}\ln\frac{s_k}{s_1}\right]+\sum_{k=1}^{b}\sum_{n=2}^{m_k}\frac{Qd_{k,n}s_k^{1-n}}{\pi\left(-n+1\right)}
\end{aligned}\tag{35}$$

3.3 Summing up

The algorithm for finding the H_2 norm of G may be summed up as follows:

- If G is of integer order ($Q = 1$), then apply (19) and (21) or (20) and (22).
- If G is of fractional order ($Q \neq 1$), then:
  - If G is MIMO, find the H_2 norm of its components and then apply (17).
  - If G is SISO, then:
    - If $q+\deg\left(A\right)-\deg\left(B\right)>0$, the norm is ∞.
    - If $q+\deg\left(A\right)-\deg\left(B\right)\leq 0$, then:
      - If $p \neq 0$, apply (33).
      - If $p = 0$, apply (35).

It should be noticed that, even though several different formulas are to be applied depending on the value of Q, the norm is a continuous function thereof.

4 The H$_\infty$ Norm

The H$_\infty$ norm of a transfer function matrix G is defined as

$$\|G\|_\infty = \sup_{\omega}\max\sigma\left[G\left(j\omega\right)\right]\tag{36}$$

where $\sigma\left(\mathbf{A}\right)$ represents the set of singular values of matrix **A**. (This set has a finite number of values, and thus has a maximum; on the other hand, the set resulting of sweeping all frequencies may have no maximum, but only a supreme value.) If G is SISO, (36) becomes simply

$$\|G\|_\infty = \sup_\omega \max \left| G(j\omega) \right| . \tag{37}$$

This norm may be found by direct evaluation at several frequencies. Frequencies clearly above or below all the frequencies of poles and zeros need not be searched. The result is, of course, equal to or below the exact result – it can never be above.

5 The Control Paradigm

Roughly speaking, the idea of H_2 and H_∞ controllers is to minimise (at least over a certain range of frequencies we are interested in) one of those norms, ensuring that the input is never amplified to such an extent that instability will arise. It is usual to include judiciously chosen shaping transfer functions in the control loop so that control efforts be exerted at those frequencies desired by the control designer. Should the weights be adequately chosen, it is possible to find, by minimising one of the two norms, a control-loop that is stable and robust to plant variations. These are expected to cause a worse performance but not instability. (The above is of course an oversimplified description; see for instance Lublin *et al.* (1996) or Doyle *et al.* (1989) for details.)

H_2 and H_∞ controllers make use of the control structures of the block diagrams in Figure 1, where **K** is the controller, **A**, **B**, **C** and **D** are the matrixes of the state space representation of **G**, and **L** models how noise affects the states (or the inputs, should **L** = **B**). Vector w collects all inputs (references, noise, disturbances…) save the control actions u. Vector z collects all variables showing the performance of the control system, namely outputs and control actions (whose magnitude may have to be limited). Weights W_1 to W_4 may be transfer functions, and usually are. They let the control designer shape the result by telling the loop in what frequencies control actions, outputs, etc., have to be large or small.

The above interconnections give rise to this transfer function

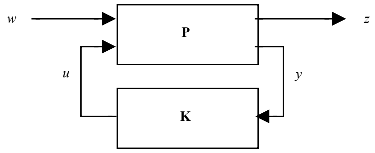
$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} W_3\mathbf{S}\mathbf{G}_1W_1 & W_3\mathbf{S}\mathbf{G}_2\mathbf{K}W_2 \\ W_4\mathbf{K}\mathbf{S}\mathbf{G}_1W_1 & W_4\mathbf{K}\mathbf{S}W_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{38}$$

$$\mathbf{G}_1 = \mathbf{C}\left(s\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{L} \tag{39}$$

$$\mathbf{G}_2 = \mathbf{C}\left(s\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{B} \tag{40}$$

$$\mathbf{S} = \left(\mathbf{I} - \mathbf{G}_2\mathbf{K}\right)^{-1} \tag{41}$$

It is a norm of this matrix transfer function that we want to minimise.



where P is given by

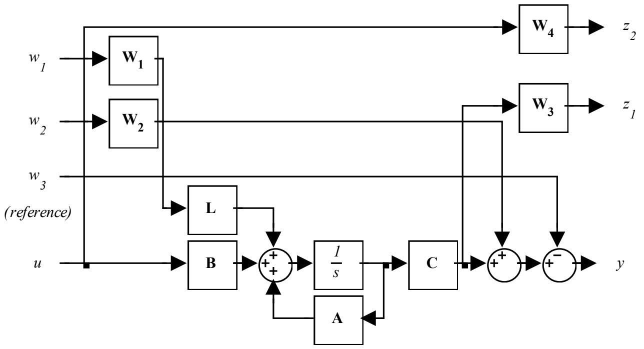


Figure 1
Block diagrams for H_2 and H_∞ controllers

6 Finding Controllers

For integer order plants it is possible to find a controller minimising either the H_2 or the H_∞ norm of a suitable loop transfer function by analytical means (Lublin *et al.*, 1996; Doyle *et al.*, 1989). Unfortunately no such relations have yet been found for fractional-order plants; those for integer-order plants are derived from mathematical formulations for the norms very different from the formulations available for the fractional case. But it is possible to use numerical methods to minimise such norms. (Petras and Hypiusova (2002) also suggest a numerical multicriteria optimisation method for the H_∞ case.) Among all possible optimisation algorithms, genetic algorithms have been used in this paper.

Genetic algorithms are an optimisation method that emulates the evolutionary principle of the survival of the fittest. This is because several possible solutions for our minimisation problem are handled simultaneously, as though each were an individual of a population; each iteration attempts to discard the poorest solutions and to improve the best already found. See for instance Jang (1997) for more details; the description of the particular genetic algorithm used, given below in section 7.1, also helps to make out how the optimisation is performed.

Genetic algorithms were chosen for they ensure reasonable results for nearly all minimisation problems. Other methods may perform better in particular cases, but knowing in advance which ones might is guesswork. On the other hand, genetic algorithms have fairly good results in almost all cases (Silva *et al.*, 2005).

7 Example

Let

$$G(s)=\frac{1}{39.69s^{1.26}+0.598}\,. \tag{42}$$

This transfer function describes a thermal system heated by an electrical radiator (the input being a voltage) with the temperature measured by a pyrometer (the output being a voltage too) (Vinagre *et al.*, 2001).

The parameters of (42) have been identified by numerically fitting its step response to experimental values. So there are no reasons why the much more tractable commensurate (Q = 4) transfer function

$$G(s)=\frac{1}{39.69s^{5/4}+0.598} \tag{43}$$

should not be used instead, its step and frequency responses being indistinguishable from those of (42). Suppose that we model (white, 0.01 V^2 intensity) noise as affecting both input (**L** = **B** in Fig. 1) and output. We want the output to remain unchanged in spite of noise, with the transfer function from w_1 to z_1 smaller than -6 dB over all frequencies and the (not weighted) transfer function from w_2 to z_1 decaying significantly (say, at -40 dB/decade at least) for high frequencies (say, above 1 rad/s, given the nature of the plant). The transfer functions mentioned are those without weights, SG_1 and SG_2K.

After some trial and error, the following weights have been selected:

$$\begin{aligned}
&W_1(s)=\frac{1}{0.01}\frac{0.1s^{1/4}+10}{s^{1/4}+1.25}\\
&W_2(s)=\frac{1}{0.01}\frac{1429s^{1/4}+5000}{s^{1/4}+1000}\,.\\
&W_3=W_4=1
\end{aligned} \tag{44}$$

These weights are fractional-order transfer functions, but integer-order weights might have been used instead. For this particular problem, fractional-order weights allowed attaining the control objectives more easily, but this is not always necessarily so. Integer-order weights have the additional advantage of having frequency responses easier to obtain. Furthermore, even though in this particular case a single set of weights sufficed for both the H_2 and the H_∞ controllers, this is not always necessarily so: different weights might have been necessary.

7.1 Algorithm

The algorithm to find a controller was as follows:

▪ A population with fifty individuals is created. Each individual is a transfer function matrix with a dimension compatible with the dimensions of the plant (in this case, controllers are SISO). The orders of the numerators and the denominators are those of the plant or the ones immediately above or below. Parameters are stored as real numbers.

▪ The H_2 or H_∞ norm of the matrix transfer function in (38) is evaluated for all individuals (in this case, this is a 2×2 matrix). The smaller the norm, the fitter the individual is.

▪ A new population is created with 90% of the size of the original one. Individuals are selected for this group according to their fitness. These will be the parents in the next step.

▪ The parents are recombined and replaced by their offspring. In other words, parents are matched in pairs; each pair is replaced by two new individuals, called the offspring; the parameters of each of the offspring are randomly chosen from those of its parents.

▪ The offspring undergo a mutation. In other words, some of their parameters, randomly chosen, are changed by addition of random values. The mutation probability is such that the average number of mutated parameters per individual is 0.5.

▪ These mutated descendents replace the less fit individuals in the original population. The resulting population is used for a new iteration, beginning with the evaluation of the norms, as explained in the second step above.

▪ Iterations stop after a certain maximum number of iterations (500 in this case) or after a certain maximum number of iterations without improvement in the results (in this case 50 for the H_2 norm and 30 for the H_∞ norm; this last value was smaller because calculations were slower).

The Genetic Algorithm Toolbox for Matlab (Chipperfield *et al.*, 1994) was used to implement this algorithm.

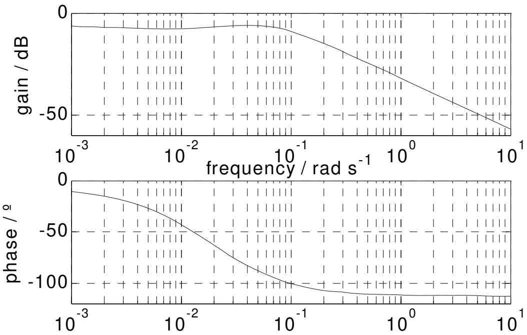


Figure 2

Bode diagram of SG_1 for H_2 controller (45)

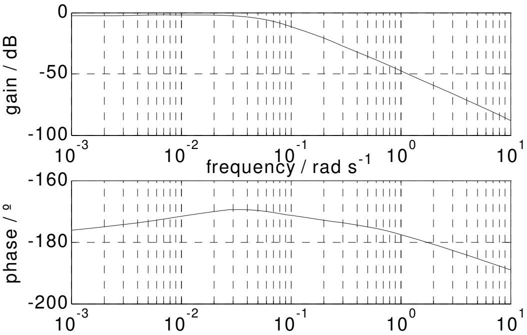


Figure 3

Bode diagram of SG_2K for H_2 controller (45)

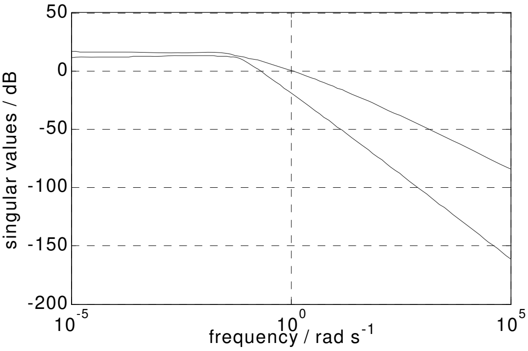


Figure 4

Singular values of loop transfer function (38) for H_2 controller (45)

7.2 Results

The following H_2 controller was found, with a norm of (38) equal to 1.7905:

$$K(s) = \frac{5.704}{s^{5/4} + 10s + 10s^{3/4} + 9.999^{1/2} + 9.422s^{1/4} - 5.399} \tag{45}$$

The relevant Bode and singular value plots are found in Figure 2, Figure 3 and Figure 4.

The following H_∞ controller was found, resulting in a norm of (38) equal to 6.7115:

$$K(s) = \frac{7.448}{s^{5/4} + 9.383s + 8.642s^{3/4} - 2.316s^{1/2} + 9.227s^{1/4} - 4.736} \tag{46}$$

The relevant Bode and singular value plots are found in Figure 5, Figure 6 and Figure 7.

7.3 Discussion

For both controllers, the magnitude of SG_1 is always below -6 dB (actually its maximum is -6.05 dB for the H_2 controller and -7.13 dB for the H_∞ controller). SG_2K decays with -45 dB/decade with both controllers. This means the objectives were attained in both cases. This is also shown by simulations of the resulting control loops. Changes in plant parameters are also handled by the resulting controllers.

Other values for weights W_1 to W_4 allow obtaining different results and shaping the loop in other ways. Thus it would be possible to cope with different performance specifications.

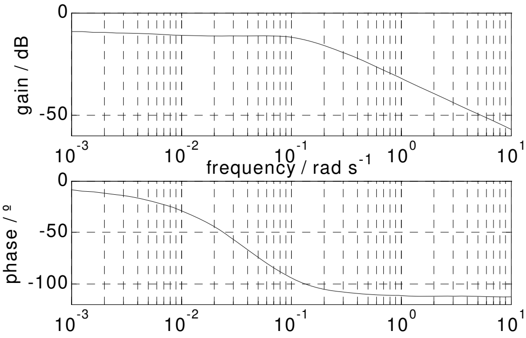


Figure 5
Bode diagram of SG_1 for H_∞ controller (46)

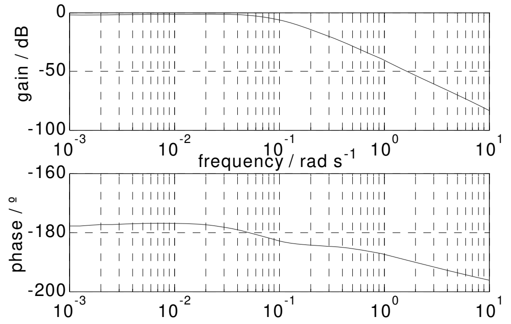


Figure 6

Bode diagram of SG_2K for H_∞ controller (46)

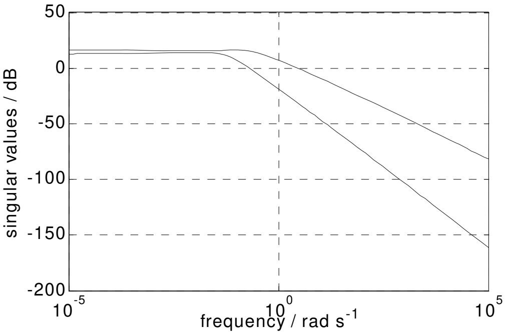


Figure 7

Singular values of loop transfer function (38) for H_∞ controller (46)

Two important questions arise from the use of a numerical minimisation method. The first is the time needed to reach a solution. For the H_2 case of the problem above, this means about 1 minute and 38 seconds of computation per iteration are needed (in a Pentium IV @ 2.53 GHz); for the H_∞ case, this may mean up to about 2 minutes and 50 seconds per iteration (in the same machine), but depends on how close is the mesh of frequencies used to estimate the norm. These are bearable values, though faster results would, of course, be desirable.

Still concerning this point, it is worth noticing that an increase in the dimension of (38) reflects very heavily on the computational effort needed. Dimensions above 4 often prevent the numerical algorithm from reaching a solution.

The second question concerns how far the optimisation went when it stops. The best validation possible is to use the algorithm for integer plants, for which there is an analytical solution available, and then compare both results. This was tried for several cases, and the numerical method always got very close to the analytical result. Assuming the same to happen with the fractional case, it seems that further possible reductions in norms are not very relevant.

Conclusion

In this paper the task of finding H_2 and H_∞ controllers for fractional plants was successfully addressed. This involves finding the norms of transfer function matrixes. A numerical optimisation method (a genetic algorithm) was used, given the absence of known analytical methods.

The drawbacks of this way of dealing with the problem are long simulation times (with the magnitude of hours) and the impossibility of knowing for sure how far the optimisation went. Thus, and beyond checking the validity of these results, the future work in this area consists in looking for analytic methods of developing these controllers.

References

[1] A. Chipperfield, P. Fleming, H. Pohlheim, C. Fonseca: Genetic Algorithm Toolbox for Use with Matlab, 1994, Available: http://www.shef.ac.uk/~gaipp/ ga-toolbox/

[2] J. Doyle, K. Glover, P. Khargonekar, B. Francis: State-Space Solutions to Standard H_2 and H_∞ Control Problems, *IEEE Transactions on Automatic Control*, Vol. 34, No. 8, pp. 831-847, Aug. 1989

[3] I. Gradshteyn, I. Ryzhik: *Table of Integrals, Series and Products*. Boston, Academic Press, 1980

[4] J.-S. R. Jang: Derivative-free optimization, in *Neuro-fuzzy and Soft Computing*. Upper Saddle River: Prentice Hall, 1997, pp. 173-196

[5] L. Lublin, S. Grocott, M. Athans: H_2 (LQG) and H_∞ control, in *The control handbook*. Boca Raton: CRC Press, 1996, pp. 635-650

[6] R. Malti, M. Aoun, O. Cois, A. Oustaloup, F. Levron: H_2 Norm of Fractional Differential Systems, in *ASME Symposium on Fractional Derivatives and Their Applications*, Chicago, 2003

[7] K. Miller, B. Ross: *An Introduction to the Fractional Calculus and Fractional Differential Equations*. New York: John Wiley and Sons, 1993

[8] I. Petras, M. Hypiusova: Design of Fractional-Order Controllers via H_∞ Norm Minimisation, in *Selected Topics in Modelling and Control*, Slovak University of Technology Press, Bratislava, 2002, pp. 50-54

[9] I. Podlubny: *Fractional Differential Equations*. San Diego: Academic Press, 1999

[10] S. Samko, A. Kilbas, O. Marichev: *Fractional Integrals and Derivatives: Theory and Applications*. Yverdon: Gordon and Breach Science Publishers, 1993

[11] C. A. Silva, J. M. Sousa, T. Runkler, J. Sá da Costa: A Logistic Process Scheduling Problem: Genetic Algorithms or Ant Colony Optimization?, in *IFAC 2005*, to be published

[12] B. M. Vinagre, I. Petras, P. Merchan, L. Dorcak: Two Digital Realizations of Fractional Controllers: Application to Temperature Control of a Solid, in *European Control Conference*, Porto, 2001, pp. 1764-1767

Research of Competitiveness Factors of SME

György Kadocsa

Keleti Károly Faculty of Economics, Budapest Tech
Népszínház u. 8, H-1081 Budapest, Hungary
E-mail: kadocsa.gyorgy@kgk.bmf.hu

Abstract: An SME is able to cope with the global challenge if it realizes reliable, balanced and high-standard operation in its business. There are some management and organizational methods increasing the competitiveness of SME. Controlling as a management tool and management function as well as a factor affecting competitiveness came more into focus. Outsourcing of activities not belonging to the main profile of the enterprise seems natural for most SME-s. Family business as the driving motor of the business can bear bigger loads than SME of similar sizes but not organized around a family.

Keywords: competitiveness factors, management methods, family business speciality

1 Introduction

The challenge of the 21st century for SME-s is global competitiveness. This means that customers need to be provided with constant and reliable products and services of a recognized quality, while the market environment is characterized by global competition.

An SME is able to cope with the global challenge if it realizes reliable, balanced and high-standard operation in its business.

The Institute for Organisation and Management of the Budapest Tech where an SME research and development team is operating – in co-operation with other research institutes in Hungary and abroad – puts great emphasis on investigating the role of controlling in increasing the competitiveness of businesses.

One of the surveys of BT SME research group, currently in progress, analyses competitiveness, focusing on EU membership and a new competitive environment created by globalisation.

The operational and organisational conditions of competitiveness are manifold and involve each specialist area as well as management and direction itself.

SME research was started in 1994 at the Budapest Tech (BT). In 1993 at a summer university forum held at the University of Economics (WU) in Vienna and at a consequent international conference attention of the participants from reform countries was drawn to the increasing importance of the SME-s in developed European countries. At the same time an international research program called MER ('Management, Entwicklung, Razvoj') was launched at the Maribor University in Slovenia, namely under the guidance of Prof. Dr. Janko Belak and with the effective cooperation of Prof. Dr. Josef Mugler from Vienna. In addition to Austria, Switzerland, and Germany almost all of the reform countries were represented in the program. On behalf of the Bánki Donát Polytechnic, the legal predecessor of BT, a still existing Organization and Management Institute (SZVI) was the one to take part from the beginning in the researches and the MER cooperation.

At the beginning of the 90's our experiences gained in the neighbouring countries and domestically confirmed our view that great emphasis needs to be placed on SME research and development. Therefore SZVI included enterprises, and within this SME research in its program of education development and research [3].

The SZVI research program features the following activities and results [4]:

Establishment and introduction into education of a new subject

Literature review

SME counselling

Surveys done about SME-s

In the framework of the SME research conducted at Budapest Tech we launched a survey of SME in 2002-2003, focusing on the main characteristics of family businesses (FB), as well as their position and prospects in Hungary. Based on the theory and technical literature available in 2002, in 2003 we analysed 200, and in 2004 further 200 businesses in practice, the majority of which were family businesses. This report summarises the research results of the last three years.

The former theoretical and research projects contained in publications confirmed also, by the analysis of approximately 800 family businesses, our intention to continue researching the position of the family businesses analysing their development options.

The survey described below did not target at agriculture, or family farms operating in agriculture; it covered the other sectors of the economy.

2 Research Method

There were several methods applied during the SME researches and developments. Our selection from among the possible methods was greatly influenced by the fact that we had no or only rather scarce external financial resources available to us. The test sample was determined and picked also accordingly. In this respect, however, there was a favourable condition serving us: both the number of students and their circle of acquaintances cover the whole country. Accordingly, the enterprises selected and inspected by them cover – at least in a geographical sense – the whole country.

We have used our resources so far to apply the following research methods [6]:

Interviews and studies by students

Personal experiences gained by counsellors

Quick test disclosing the application of organisation and management methods

There were also several surveys were conducted in the subject of SME-s. While education was running, students were included in the program by receiving interim practical tasks in connection with the theme of SME-s. Such were for instance:

Launch of SME-s

Business planning for SME-s

Investigation of SME life lines

Investigation of family businesses

Since almost 400 students were studying ‘Business Set-Up’ in the past years, thus the number of SME case studies were available to us to carry out research thereon.

Approximately 100 SME-s have completed the questionnaire (quick test) sufficiently for the research too.

We involved students in their last year, participating in the enterprise management course, to visit SME-s, acquaintance or local community relations or at random selection.

The students conducting the survey came from almost all parts of the country, from small and large settlements. Thus, in our opinion, the enterprises found and analysed by them represent the Hungarian average well among the SME-s.

A case study was prepared on each business on the basis the criteria indicated above. The institute’s research group later analysed this case study. The analysis was made with the method of individual interpretation and evaluation, with occasional supplementary information, and not with a mathematical statistical method. Thus the most important factor in evaluating phenomena and tendencies behind figures was finding the stress, and highlighting the essential information.

SME counselling began in 1994 when the German RKW (Rationalisierungskuratorium der Deutschen Wirtschaft) came to Hungary. Three members of the present research group of our Institute took part jointly in RKW's programs, and we carried out SME counselling all over the country for years. Considerable experiences were gained with such counselling and related at different domestic and international forums.

This report provides a synthesis of the results obtained with various research methods to date. Of the factors of competitiveness, we would like to highlight controlling, as we analyse its implementation opportunities in the SME sector, primarily among small enterprises.

3 Analised Factors Influencing the Competitiveness

The factors influencing competitiveness of SME-s can be divided into two groups, into external and internal factors.

External factors:	Internal factors:
Employment	Marketing
Productivity	Innovation
Capital supply opportunities	Productivity
Globalisation	Knowledge-based development
EU	Capital supply
Business relations	Management, organisation, structure
Alliances	Cost-efficiency
Networks	Compliance

Table 1

Influencing factors of competitiveness

Measuring competitiveness is the most difficult task. A few values, indicators, or characteristic features that can be quantified and accessible, or are not quantifiable or accessible at all, or are difficult to quantify or access, have to be identified at corporate level. These measuring points are the following:

Revenues, Export, Profit, Market share, Productivity,

Technical standard, Corporate value, good-will, Image,

Customer satisfaction, 'Value' of the product, service

The structure, aspects and factors of the analysis of globalisation and competitiveness outlined in this study represent an initial phase of a longer research programme, focusing clearly of SME-s and specifically small enterprises.

4 Competitiveness of SME in Hungary

On the basis of the surveys, SME-s compete in the following areas:

Retail trade,

Repair, assembly and personal services for the population,

Real estate brokerage, real estate trade,

Restaurants, catering,

Intellectual property (consultation, software development),

Travel arrangements, holidays,

Agricultural farmer activities (plant cultivation, animal breeding).

The composition of enterprises participating in the surveys more or less reflects the composition according to activity.

According to Hungarian statistics, most of the SME-s are operating in the commercial and service sectors.

It is an important factor that currently we do not analyse agricultural SME-s.

4.1 Experiences of Application of Certain Management and Organisation Methods

Without trying to be fully comprehensive, we analysed the following areas:

Planning,

Marketing,

Quality assurance,

Organisational methods,

Controlling.

Planning was analysed at strategic and business (annual) planning levels. We concluded that of the surveyed SME-s 37% had a written strategic plan and 46% had written business plan.

In terms of the strategic plan, this proportion is acceptable, but the written form of the business plan shows a very low figure. Among the analysed SME-s, strategic thinking was not missing, but they did not consider that it was important to put it into a written form. A bigger insufficiency is the lack of registration of short-term business planning. It cannot always be explained with the uncertainty of planning, much more with the lack of professional business planning skills and underestimating of the importance of planning.

The presence of planning in some separately analysed enterprises confirmed our belief that a written strategy and business plan are both very useful for an enterprise, irrespective of which SME segment it operates.

In terms of marketing, 40% of the enterprises have an individual or organisation who and which fulfil the marketing function. Of the remaining 60%, 18% plan to create and operate a marketing group or organisation in the near future.

This proportion also indicates that SME-s consider marketing, i.e. management of market activities, very important. However, in terms of their future plans, the 18% enhancement indicates rather reserved development ideas.

The focus on marketing activities is explained with the direct use of market activities, aimed primarily at advertising activities and sales incentives.

In the area of quality management, very few enterprises, only 15%, have an ISO certified quality assurance system.

On the other hand, the future image is promising: of the other 85%, more than one-third, i.e. in total 30% plan to introduce an ISO quality assurance system in the future. The introduction of a quality management system is mainly motivated by their intentions to become a supplier or act as a supplier, it is more rarely considered part of a separate market image.

In terms of the application of various organisation methods, we have limited information. The analyses covered only a few special methods selected at random.

Although the BPR method is not completely unknown among SME-s, only a small proportion (approximately 10%) have applied it, or are willing to reorganise their business processes.

Regular and continuous development of operation is an ordinary and right method for the majority of enterprises (approximately two-thirds). The Kaizen method with similar contents as a definition is almost completely unknown though.

Outsourcing of activities not belonging to the main profile of the enterprise seems natural for most SME-s. However, this does not mean that outsourcing is a fully comprehensive deliberate activity. It rather means that entrepreneurs manage their external resource requirements on the basis of the principles of needs and there is no plan for how to outsource their activities. The controlling function is present in 20% of the analysed enterprises, but it does not mean that 20% of the enterprises have created a controller's function, or a controlling organisation. Of the remaining 80%, only 11% plan to establish the controlling function in their enterprise in the near future.

4.2 Outsourcing

Outsourcing of activities not belonging to the core activities is the result of natural distribution of work among SME-s. It is a general practice that an SME does not conduct the following activities in its own organisation but uses them as an external service:

Legal representation, Legal consultation,

Accounting, Tax consultation, Business and financial consultation,

Market research, survey, Training, further training,

Compilation of loan applications and other applications.

Outsourcing of these activities is not a deliberate activity, but a kind of feature of SME-s. They cannot economically create and operate capacities that would be specialised for such activities within their organisation.

Naturally, various resources are also outsourced in the material processes, which represent ‘real’ outsourcing in this context, but for the purpose of this survey, these relations are not very important.

The quality and reliability of activities outsourced for competitiveness are very important. Let us just imagine new environmental conditions involving global competition, such as

A tender for raising funds,

International law, accounting and tax regulations,

New capital investment and loan opportunities,

Joining supplier systems.

The outsourced activities are also different from each other that they contain some regular and irregular occasional activities. For example, accounting is typically an outsourced continuous activity, while taking loans or legal representation take place occasionally.

A key factor of the competitiveness of the Hungarian SME sector is cost-efficiency and productivity. The controlling system has already proved to be useful in the development of cost-efficiency in the entire world, and it has become a widely used useful and indispensable tool of the management of large companies.

There is also a theoretical possibility to outsource the controlling function, but according to our experiences to date, SME-s do not use this opportunity very often. Regular consultation, relating, for example, to accounting and business consultation services, would be ideal to encourage the manager of an enterprise to use controlling tools. Another possible solution is the adoption of a complex enterprise system, containing also the right management methods and tools. Such a system is a franchise system, for example. Below, we shall review the controlling and franchise situation and their mutual opportunities in terms of improving the competitiveness of SME-s.

4.3 Controlling in SME-s

One of the factors of competitiveness is cost-efficiency, which is required for real processes, as well as planning and controlling of costs.

Controlling as management tools fulfils the following main functions [2]:

Planning,

Measurement,

Comparison,

Evaluation.

These functions form a regularly repeated closed system, considered a sub-system of management, and discussed as a management function on its own in the technical literature.

In the controlling system, cost controlling is very important, which is suitable to increase cost-efficiency and thus, competitiveness.

Empirical studies and theoretical and literary researches have proved that controlling contains planning, coordination and control functions. Empirical signs indicate that it also plays a role in management. Development from an economic operation model to the management model indicates that controlling, an integrating management function, develops towards a new model.

On the basis of the results of empirical researches, the theoretical approach can be assumed. Thus controlling is:

strictly and mainly practice-based,

dominated by profit-orientation concerning its objectives,

has an economic operation approach,

its theoretical basis originates from economic operation, too,

and its main functions are coordination and an information system.

As it was indicated above, the controlling function is present in 20% of the analysed enterprises. However, it does not mean that these enterprises have created a separate controller's position, or controlling organisation.

A closer study of individual cases has indicated that the manager of an enterprise requires regular generation and availability of information relating to the controlling area. Most often the information is generated by the accountant or business consultant of the enterprise, and the controlling report, i.e. the inclusion of the information in a system and its interpretation – is also prepared with external assistance. It happens very rarely that the entrepreneur's own administration prepares the information and the controlling report is prepared by a competent expert – controller – or the manager himself.

Controlling is a professionally demanding function, involving an expenditure, the yield of which is not always seen directly by the entrepreneur. This explains that of the 80% of SME-s without a controlling function, only 11% plans to implement a controlling function in their enterprise within the near future.

At the same time, the personal interviews have also revealed that the managers of such enterprises consider themselves rather cost-sensitive and economic persons, who do everything to increase cost-efficiency.

Such a low level of the presence of the controlling function among the SME-s, can be explained best with the fact that the qualifications of the managers of the analysed enterprises (a sample of 400 companies) show a rather varying picture. 30% of the managers have diploma, 42% have a secondary qualification.

Assuming that each SME manager with a diploma knows exactly what controlling is and intends to apply it too, the proportion of existing and planned controlling functions is close to 30%. Although attempts for cost-efficiency can be expected from managers with secondary or lower qualifications and it can actually be observed, but the solution, i.e. application of a controlling system or function does not even occur to them without the relevant knowledge or information.

Thus, the presence of a need does not point to controlling, but to an economic system that somehow contains both the controlling function and system.

4.4 Appearance of Franchise and its Potential Role in Increasing Competitiveness

Franchise is widely used in the entire world, and it has been introduced to Hungary recently, too. Franchise is a type of enterprise based on close cooperation, in the framework of which the franchiser, i.e. the owner of the system provides a complex system, established on professional and trading considerations, and successfully tested in the market environment, with a full scope of training, use of a brand name and continuous control and assistance to the franchisee, who operates the system in his own enterprise at his own profit and risk, according to the requirements of the franchiser, on the agreed area and for the agreed period, in exchange for a fee [7].

However, there are several definitions of the franchise concept. In America, the business aspects are highlighted most, but there are other ideas, too, that underline support and assistance in terms of franchise. The German literature looks at franchise mainly in terms of organisation.

A business franchise can be characterised best by saying that in this framework the franchiser does not only provide the use of a product service or trade market to the franchisee, but also makes available the strategic, marketing, quality control, accounting and other business information required for the successful operation of the whole enterprise to the franchisee.

The majority of the franchise system operating in Hungary are in the trade and service sectors, which is the sector containing most SME-s. Disclosed elements of the franchise contracts also indicate that in many cases the low amount of start-up capital clearly encourages SME-s to use already tested franchise ideas in their enterprise instead of implementing their own individual ideas.

In order to increase competitiveness, cost efficiency and be successful, the franchise contract should be extended to the management functions as widely as possible. This means the expansion of rights and obligations of the franchiser in this context.

Primarily, planning, accounting, control and feedback functions require a lot of attention, as these are the weak points of the Hungarian SME-s.

A controlling system required and practiced by a franchiser could also help franchisees, having little information in the area of controlling to use a modern and effective tool for such purposes. This solution could also mean the extension of controlling, i.e. controlling outsourcing.

5. Competitiveness Speciality of FB in Hungary

In accordance with the Hungarian technical literature, the FB criteria are the following [1]:

At least two people of the family participate in the business

either in the owner's position, and/or

with an involvement in the management of the business, and/or

with an involvement in the daily operation of the business.

It was possible to evaluate in total 378 SME-s, of which 322 SME-s satisfied the criteria of FB. This ratio corresponds to 85.2%.

The quantified results of the analysis are contained in Table 2 [5].

The figures indicate the following phenomena:

Most FB-s were founded in the first half of the 1990s, and these days they mostly operate as unlimited partnerships or limited liability companies (83%), in almost equal proportions. Almost all enterprises were established for an indefinite period

(98%) and, since with a few exceptions they are still active, it indicates more thorough preparations. We have found FB-s among the businesses that operated well, and have survived several generations too.

Points of views		FB	%
Date of FB establishment	before 1990	43	13
	1990-95	139	43
	1996-2001	122	38
	2002-	18	6
Scope of activities	service	147	45
	trade	108	34
	industrial prod. and sale	67	21
Origin of the FB property	own	108	34
	family	207	64
	external resources	7	2
Owners of FB	family	570	92
	external	50	8
Managers of FB	family	634	92
	external	57	8
Employees	family	370	21
	external	1411	79
FB viability	stable	193	60
	uses credit to develop	48	15
	unstable	81	25
Technical level	out of date	212	66
	adequate	80	25
	up-to-date	30	9
Owner’s motivations	committed, purposeful	162	50
	wishes to change	160	50
Owner’s qualification	degree	80	25
	secondary qualification	180	55
Imagined future for the family	civil life (middle layer)	100	30
	welfare, prosperity	222	70

Table 2
Results of the analysis of FB

The origin of assets illustrates well the role of family business and economic organisations consisting of two or more members in the market. Nearly 70% of the analysed FB-s had assets originating from the family and several family members. Business were started from family capital, establishing business relations between brothers and sisters, parents and more distant relatives, in this order of frequency. Having analysed the scope of activity, we concluded that most

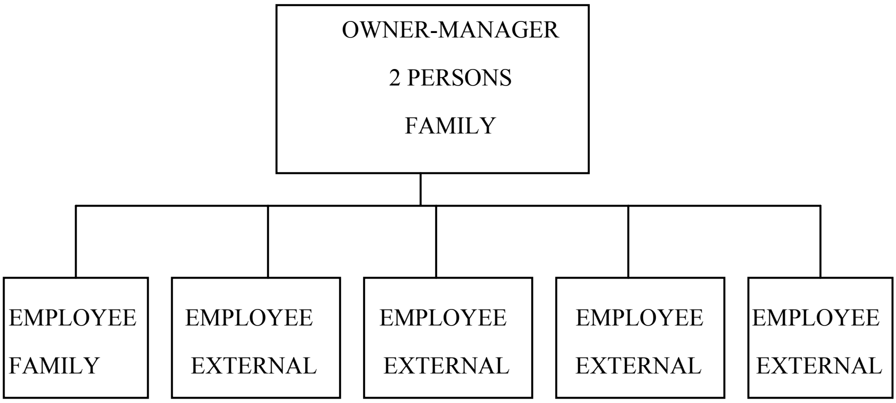
FB-s operated in services, followed by trade and the industrial sector. Almost all FB-s are managed by a family member or family members who is/are owners in the business.

There are very few external individuals among the owners, representing only 8%. The composition of management is similar.

There is not a lot of different in the number of employees of the FB-s between sectors. The fewest employees work in FB-s engaged in services. Each FB has economic staff management and employs the minimum number of staff required for their activities. According to the origin of employees, 21% have family relations and 79% are external employees.

Looking at the number of FB-s and number of managers and employees, the average FB structure can be seen on Picture 1.

Approximately 25% of owner-managers have a degree, 55% have secondary qualifications, and experts with a lot of practice are present only in FB-s established long time ago as managers.



Picture 1
Structure of average Hungarian FB

The majority of FB participants (approximately 80%) are aged between 30-55 years. In these businesses the issue of succession occurs less frequently than in other FB-s established a long time ago, in which the owner is over 60 years of age or more. In this category, the owners would prefer to maintain the business within the company. In fact, family businesses where succession has occurred as an issue succession within the company would be preferred.

Tendering is more frequently applied for selecting the management in recently founded FB-s, influenced by external owners, too. This step may also change the future of FB-s compared to FB-s where management remains in the hands of the family.

Approximately two thirds of the analysed businesses consider their economic position stable and outlooks positive. At the same time, only 50% of the owners are fully committed to their own business, the other half would like to make fundamental changes and intend to change the current operation and directions of the FB.

In approximately two thirds of the FB-s, the technical standards, the applied instruments and procedures are out of date and obsolete, they are adequate in approximately 25%, and only 8-10% of FB-s could be considered up-to-date.

Concerning economic stability and development, only 15% of FB-s use loans for development, 60% seem stable and 25% consider the position of their business unstable. 50% of the businesses cover their investment needs from their own families.

The future vision of the family and business also involves a link to primarily financial welfare and wealth. This is the specific objective for 70%, and only 30% said that their intention was to provide a good life for their families. At the same time, it is perhaps not surprising that the interviewees were reluctant to talk about the financial position of their businesses or families.

Concerning experiences, we should note that we have studied a few businesses that have covered the same path as known in the experience of Western countries as an FB tuning into a multinational enterprise sooner or later, or already operating as an international business. Studies of clearly successful exemplary businesses are analysed further, and we intend to use the research results in future research activities as well as education.

Conclusions

Certain management and organisation methods are successfully applied at SME-s.

The presence of planning in some separately analysed enterprises confirmed our belief that a written strategy and business plan are both very useful for an enterprise, irrespective of which SME segment it operates. The focus on marketing activities is explained with the direct use of market activities, aimed primarily at advertising activities and sales incentives. The introduction of a quality management system is mainly motivated by their intentions to become a supplier or act as a supplier, it is more rarely considered part of a separate market image.

Outsourcing of activities not belonging to the main profile of the enterprise seems natural for most SME-s. However, this does not mean that outsourcing is a fully comprehensive deliberate activity. It rather means that entrepreneurs manage their external resource requirements on the basis of the principles of needs and there is no plan for how to outsource their activities.

A closer study of individual cases has indicated that the manager of an enterprise requires regular generation and availability of information relating to the controlling area. Most often the information is generated by the accountant or business consultant of the enterprise, and the controlling report, i.e. the inclusion of the information in a system and its interpretation, is also prepared with external assistance.

Family business are typically small and medium enterprises. Our statements relating to certain features of family businesses could be summarized as follows.

The scope of activities of family businesses, in accordance with the SME sector's nature, includes mainly services, trade, and to a smaller degree production and sales activities.

Their capital adequacy, terms of and possibilities for receiving credit are very unfortunate, their technical level is lower than the average, and is not modern. Capital restrictions present serious difficulties to the development of family business and the increase of their compatibility, especially under the current economic conditions - considering stagnation, or a slow growth. With respect of the system of regulations and available supports, only agricultural family businesses enjoy slightly better conditions.

It can be seen from the survey of human resources of family businesses that despite any unfortunate external conditions, family relationships and personal resources have rendered recently many family businesses into successful enterprises. This shows that the family as the driving motor of the business can bear bigger loads than SME of similar sizes but not organized around a family. This is the most important lesson of our researches so far.

If our further studies confirm this phenomena then distinguished legal and financial treatment of family businesses may rightfully demanded of both the society and the economic management.

References

[1] Baranyai, Cs., Gere, I., Kiss, M., Kuczi, T., Laczkó, Zs.: Családi vállalkozások Magyarországon. Budapest, SEED, 1997, pp. 53-104

[2] Francsovics, A.: Controlling. Budapest, BMF -Ligatura, 2005, pp. 8-22

[3] Kadocsa, Gy.: Vállalkozások Szervezése. Budapest, AMICUS, 2001, pp 156-173

[4] Kadocsa, Gy.: Forschung, Förderung, Unterricht der Fmilienunternehmen in Ungarn. Internat. Conference MEB. Budapest, BMF, 2003, pp. 117-124

[5] Kadocsa, Gy., Sütő, L.: Wettbewerbsfähigkeit der Familienunternehmen in Ungarn. Internat. Conference MEB. Budapest, BMF, 2004, pp. 59-68

[6] Kadocsa, Gy.: Research and Development of SME-s at Budapest Tech. International Jubilee Conference. Budapest, BMF, 2004, pp. 307-315

[7] Poór J., Zentai, K.: Franchise, út a sikeres vállalkozáshoz. Budapest, NMK, 1991, pp. 19-43

Nonlinear System Control Using Neural Networks

Jaroslava Žilková, Jaroslav Timko, Peter Girovský

Department of Electrical Drives and Mechatronics
Technical University of Košice
Letná 9, 042 00 Košice, Slovak Republic
jaroslava.zilkova@tuke.sk, jaroslav.timko@tuke.sk, peter.girovský@tuke.sk

Abstract: The paper is focused especially on presenting possibilities of applying off-line trained artificial neural networks at creating the system inverse models that are used at designing control algorithm for non-linear dynamic system. The ability of cascade feedforward neural networks to model arbitrary non-linear functions and their inverses is exploited. This paper presents a quasi-inverse neural model, which works as a speed controller of an induction motor. The neural speed controller consists of two cascade feedforward neural networks subsystems. The first subsystem provides desired stator current components for control algorithm and the second subsystem provides corresponding voltage components for PWM converter. The availability of the proposed controller is verified through the MATLAB simulation. The effectiveness of the controller is demonstrated for different operating conditions of the drive system.

Keywords: Artificial neural network, control methods, non-linear dynamic system, induction motor, speed control

1 Introduction

In recent years artificial neural networks (ANNs) have gained a wide attention in control applications. It is the ability of the artificial neural networks to model non-linear systems that can be the most readily exploited in the synthesis of non-linear controllers. Artificial neural networks have been used to formulate a variety of control strategies [4], [5].

There are two basic design approaches:

- Direct inverse control – it uses a neural inverse model of the process as a controller.
- Indirect design – the controller uses a neural network to predict the process output.

Different structures of neural controllers for control of non-linear plants, especially induction motor drives have been presented [1]-[3], [7]-[9].

In this work we are concerned with design neural controller for induction motor control on principle of system inverse model. Part one of the paper is focused on explaining the method, while the following one demonstrates the design of the neural networks for purposes of control and the use in simulation studies for a squirrel-cage induction motor.

2 The Inverse Model of a Dynamic System

Although the system inverse model plays an important part in the theory of control, the attainment of its analytical form is pretty strenuous. Anticipating that a dynamic system can be described by the differential equation

$$y(k+1)=f\left[y(k),\dots,y(k-n+1),u(k),\dots,u(k-m+1)\right] \tag{1}$$

where the system output *y(k+1)* depends on the preceding *n*-output and *m*-input values, the system inverse model can be generally presented in the following form

$$u(k)=f^{-1}\left[r(k+1),y(k),\dots,y(k-n+1),u(k),u(k-m+1)\right] \tag{2}$$

Here *y(k+1)* is an unknown value, and hence can be substituted by the output quantity desired value *r(k+1)*. The simplest way to arrive at a system inverse neural model is it to train the neural network to approximate the system inverse model.

In real life, the most frequently used are two concepts of inverse neural model architecture: the 'general training' architecture (Fig. 1), and the 'specialized training' architecture (Fig. 2), respectively.

If in the general training architecture, signal *u* is applied to the system input, signal *y* is obtained at the system output. The different between the incoming signal *u* and the neural model output u_N is the error e_N which can be utilized for network learning.

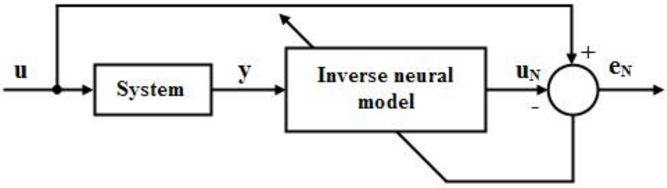


Figure 1
The 'general training' architecture

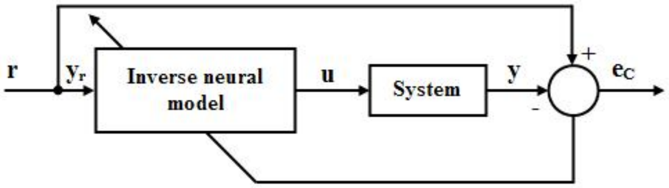


Figure 2
The ‘specialized training’ architecture

Contrary to the previous approach, it is error e_c that is in the ‘specialized training’ architecture utilized for neural network training. Error e_c is here obtained as the difference between the desired input signal r and the signal y that represents the actual system output.

The former of the two methods brings several substantial disadvantages:

The selection of varying output signal y values, specified for training of the network, cannot guarantee that the trained system output will fall exactly into those regions that are important for its successful utilization in controlling.

If the controlled system happens to be multidimensional, the attained model that is represented by the inverse neural model may be incapable of imitating a real system.

The above outlined drawbacks can be circumvented by the letter method – the ‘specialized training’ architecture-that yields some advantages when compared with the former one:

The method (Fig. 2) is intended directly for controlling, whereas the training signal is formed in dependence on the difference between the system desired and real outputs.

In the case of multidimensional dynamic system, the real inverse model can closely simulate a real system.

Multi-layer neural networks can be utilized when creating a system inverse neural model. The use of the MLP type static neural networks presents the simplest solution, however the representation of the system dynamic remains problematic with this neural model. The application of a MLP type neural network with time delaying of the input layer signals can present the solution for introducing the process dynamics into MLP type static neural network. The solution falls among the simplest ones, and the advantage of utilizing this network type rests with the opportunity of its training by traditional backpropagation algorithm for multi-layer networks.

3 Design of the Neural Controller

The main requirement we have specified is maintaining the desired speed of the induction motor. Considered for the neural controller output were the voltage components that would present an action intervention for PWM modulation, which would eventually produce the stator voltage desired values from the mains voltage (rectified via using an uncontrolled rectifier).

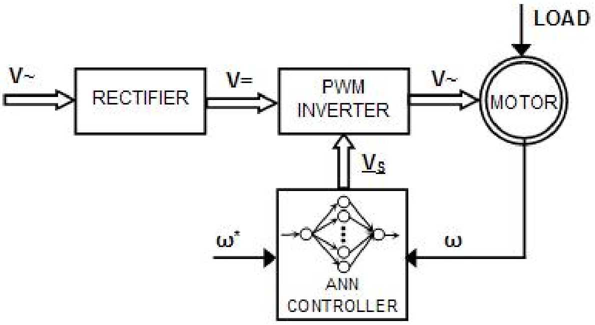


Figure 3
The scheme of the neural controller

Since the neural controller output in such a structure is not directly equal to voltage fed into the motor we have abandoned the idea to establish an accurate inverse model; considered for input quantity of the quasi-inverse neural model were rather the desired and at a time also real (measured) motor speed (Fig. 3). The design of the neural controller is based rightly on known values of these speeds.

A typical technique for control synthesis purposes is based on using a description of the induction motor in rotating reference frames (*x, y*). The use of such rotating reference frames has the benefit of simplifying the model of the motor from the point of view of controller design.

In this section design of the neural controller will be presented. The speed controller consists of two neural networks subsystems with backpropagation learning algorithm. The first subsystem (Fig. 4) of the controller serves for desired current components reconstruction and the second subsystem serves for corresponding voltage components reconstruction for PWM converter. These voltage components present action intervention for PWM modulation that would make up the desired stator voltage values from the mains voltage (rectified using an uncontrolled rectifier). The overall control structure is shown in Fig. 3.

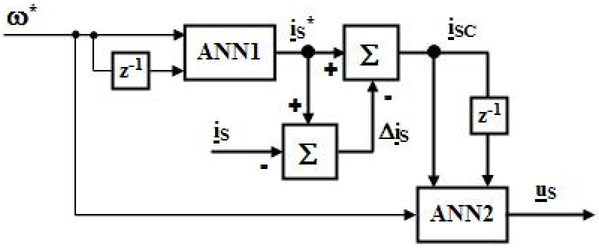


Figure 4

Concept of the neural controller

3.1 Identification of Stator Currents

The first subsystem ANN1 consists of two neural networks ANN1.1 and ANN1.2, according to equations (3). The networks are trained to approximate the time-varying function of f_1 and f_2 to give estimated one-step-ahead predicted stator current components:

$$\begin{aligned} i_{sx}^*(k) &= f1\left[\omega^*(k), \omega^*(k-1), \underline{w}\right] \\ i_{sy}^*(k) &= f2\left[\omega^*(k), \omega^*(k-1), \underline{w}\right] \end{aligned} \tag{3}$$

The inputs of the first neural subsystem ANN1 are values of desired angular speed, in *k*-th and (*k*-1)th step. The three layers feedforward neural network (FFNN) with one hidden layer has been elected, in first step, to approximate the non-linear function $\underline{f}$.

Further working showed that cascade feedforward neural networks (CFNN), which outputs $a_1(k)$ from 1st and $a_2(k)$ from 2nd layer of a CFNN are given by:

$$\begin{aligned} \underline{a}_1(k) &= tan\,sig\left(\underline{I}\,\underline{w}_i + bias_1\right) \\ \underline{a}_2(k) &= purelin\left(\underline{I}\,\underline{w}_j + \underline{a}_1(k)\underline{w}_k + bias_2\right) \end{aligned} \tag{4}$$

where *I* is input vector and *w* represents weights of the network, with one hidden layer reached the better approximation properties then FFNNs.

The structure of the cascade network ANN1.1 is shown in Fig. 5. The structure of ANN1.2 is identical.

The hidden layer of the cascade network contains twenty neurons with tansig nonlinear activation function. The neurons of the subsystem ANN1 outputs have linear activation function.

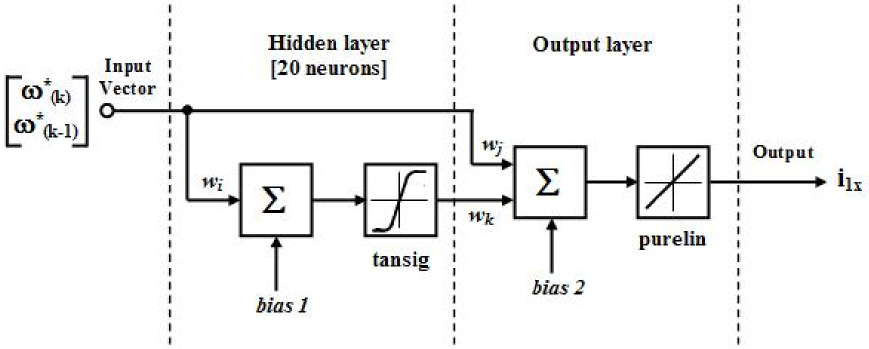


Figure 5
The structure of the ANN1.1

3.2 Rotor Speed Control

Measured actual stator current (3) corrects desired value of stator current:

$$\Delta \underline{i}_s(k) = \underline{i}_s^*(k) - \underline{i}_s(k) \tag{5}$$

Resulting signal of the correction:

$$\underline{i}_{sc}(k) = \underline{i}_s^*(k) + \Delta \underline{i}_s(k) \tag{6}$$

in *k*-th and *(k-1)*th steps and the desired speed value present inputs to the second ANN2, which generates appropriate voltage values for PWM converter:

$$\underline{u}_s(k+1) = g\left[\underline{i}_{sc}(k), \underline{i}_{sc}(k-1), \omega^*, \underline{w}\right] \tag{7}$$

The cascade feedforward neural network is used for *g* approximation, too. The relation (7) determines the number of inputs of ANN2. Twenty hidden neurons in one hidden layer of the neural subsystem employ the hyperbolic tangent function. The structure of the ANN2 is shown in Fig. 6.

The ANN's may be trained on-line or off-line. In the case of the off-line training, one requires the input-output characteristics of the system. Training patterns for speed controller were obtained by numerical simulations of the induction motor model with help of MATLAB-Simulink. In simulations the nominal data of a 3kW induction motor were used. All the networks are trained off-line in order to minimise the control performance. The backpropagation training algorithm with Levenberg-Marquardt´s modification was used for the training modes.

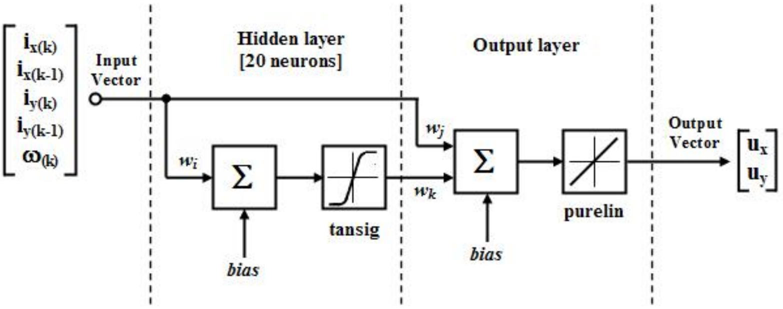


Figure 6
The structure of the ANN2

4 Simulation Results

Presented in this section will be the results obtained in MATLAB environment for given connection of the control diagram shown in Fig. 3, where the designed neural controller was implemented. The testing of the neural controller was performed on the induction motor with the following parameters:

$U = 220$ V/50 Hz, $I_N = 6.9$A, $P_N = 3$ kW, $n_N = 1420$ RPM,

$R_1 = 1.81\ \Omega$, $R_2 = 1.91\ \Omega$, $L_{1\sigma} = L_{2\sigma} = 0.00885$ H,

$L_h = 0.184$H, $p_p = 2$, $T_N = 20.17$ Nm, $J_N = 0.1$ kgm^2.

The neural speed controller was trained in the wide range of speed and load torque changes. Then the trained controller was tested for speed reference signal different than used in the training procedures. These testing signals together with results of simulations are presented in Figs. 7 and 8. Fig. 7 shows speed response waveforms in a case of speed variation. The command speed is 120, -30, 20 and 50 rad/sec, respectively. Fig. 8 shows speed response waveforms in a case of load variation. The command speed is 100 rad/sec, that is increased from zero speed and the 100% rated load is applied at 1sec and 150% rated torque is applied at 2 sec.

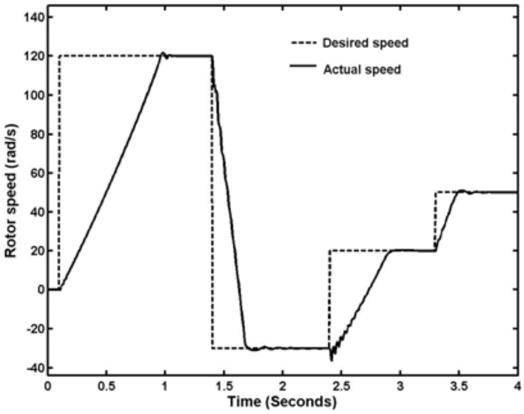


Figure 7
Drive system response under a variable speed reference and actual motor speed

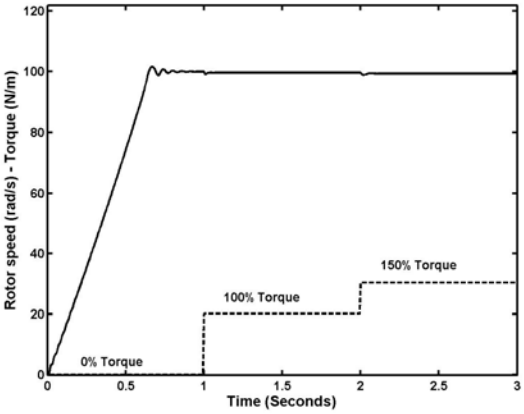


Figure 8
Speed response under load changes

Conclusion

In this paper, an off-line neural network controller for induction motor drives was presented. The design of neural controller is based on sensor information pertaining to angular speed and stator current of the induction motor. The neural controller consists of two cascade feedforward neural networks subsystems.

First subsystem of the neural controller serves for desired current components reconstruction and the second subsystem serves for corresponding voltage components reconstruction for PWM converter.

Cascade feedforward neural networks are used for all functions approximations. Used with these networks was learning by use of Levenberg-Marquardt algorithm. Training samples for the speed controller were attained via simulation of an induction motor model in MATLAB-Simulink environment. Simulation results using MATLAB verify the effectiveness of proposed controller.

Acknowledgements

Authors gratefully acknowledge support of VEGA 1/2177/05 at this work.

References

[1] P. Branštetter, M. Skotnica: Application of artificial neural network for speed control of asynchronous motor with vector control, Proceedings of International Conference of Košice, EPE-PEMC 2000, pp. 6-157-6-159 Sept. 2000

[2] M. A. Brdys, G. J. Kulawski: Dynamic Neural Controllers for Induction Motor, IEEE Transactions on Neural Networks, Vol. 10, No. 2, pp. 340-355, 1999

[3] B. Burton, R. G. Harley, G. Diana, J. L. Rodgerson: Implementation of a Neural Network to Adaptively Identify and Control VSI-Fed Induction Motor Stator Currents, IEEE Transaction on Industry Applications, Vol. 34, No. 3, May/June 1998

[4] A. U. Levin, k. s Narendra: Control of Nonlinear Dynamical Systems Using Neural Networks: Controllability and Stabilization. IEEE Transactions on Neural Networks, 1993, Vol. 4, pp.192-206

[5] A. U. Levin, k. s Narendra: Control of Nonlinear Dynamical Systems Using Neural Networks- Part II: Observability, Identification and Control. IEEE Transactions on Neural Networks, 1996, Vol. 7, pp. 30-42

[6] J. Timko, J. Žilková, D. Balara: Applications of Artificial Neural Networks in Electrical Drives. (in Slovak), Calypso s.r.o., Košice, 2002, p. 239

[7] P. Vas: Artificial-Intelligence-based Electrical Machines and Drives. Oxford University Press, Oxford, 1999

[8] M. Vittek, J. Rapšík, Buday: Shaft Sensorless Direct Vector Control of Induction Motor with Forced Dynamic. Proc. EPE-PEMC'02, Cavtat & Dubrovnik, Croatia, CD-ROM, Sept. 2002

[9] M. T. Wishart, R. G. Harley: Identification and Control of Induction Machines Using Artificial Neural Networks. IEEE Transaction on Industry Applications, Vol. 31, No. 3, May/June 1995

List of Symbols and Abbreviations

P_N - nominal power

T_N - nominal load torque

J_N - moment of inertia

p_p - pairs of poles

ω - angular speed

n_N - motor speed

* - desired value

R_1, R_2 - stator and rotor resistances

L_1, L_2 - stator and rotor inductance

L_h - mutual inductance

$[i_{1x}\ i_{1y}]$ - x , y components of stator current

$[u_x\ u_y]$ - x , y components of stator voltage

I – input vector

w – synaptic weight

ANN - artificial neural networks

FFNN - feedforward neural network

CFNN - cascade feedforward neural network

Tensor Product Model Transformation-based Controller Design for Gantry Crane Control System – An Application Approach

Fetah Kolonic, Alen Poljugan, Ivan Petrovic

Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, HR-10000 Zagreb, Croatia
fetah.kolonic@fer.hr, alen.poljugan@fer.hr, ivan.petrovic@fer.hr

Abstract: The Tensor Product (TP) model transformation is a recently proposed technique for transforming given Linear Parameter Varying (LPV) state-space models into polytopic model form, namely, to parameter varying convex combination of Linear Time Invariant (LTI) systems. The main advantage of the TP model transformation is that it is executable in a few minutes and the Linear Matrix Inequality (LMI)-based control design frameworks can immediately be applied to the resulting polytopc models to yield controllers with tractable and guaranteed performance. Various applications of the TP model transformation-based design were studied via academic complex and benchmark problems, but no real experimental environment-based study was published. Thus, the main objective of this paper is to study how the TP model transformation performs in a real world problem and control setup. The laboratory concept for TP model-based controller design, simulation and real time running on an electromechanical system is presented. Development system for TP model-based controller with one hardware/software platform and target system with real-time hardware/ software support are connected in the unique system. Proposed system is based on microprocessor of personal computer (PC) for simulation and software development as well as for real-time control. Control algorithm, designed and simulated in MATLAB/SIMULINK environment, use graphically oriented software interface for real-time code generation. Some specific conflicting industrial tasks in real industrial crane application, such as fast load positioning control and load swing angle minimization, are considered and compared with other controller types.

Keywords: Parallel Distributed Compensation, Linear matrix inequalities, TP model transformation, Single Pendulum Gantry (SPG), position control, swing angle control

1 Introduction

The main contribution of this paper is that it investigates the performance of the TP model transformation-based control design in a real experimental setup, and evaluates and compares the results. The study is conducted through the example of a translational electromechanical system, the Single Pendulum Gantry (SPG), an educational testbed of University of Zagreb. We derive three different controllers. One is based on a classical linearization and pole placement technique. It approximates the given model by one LTI system and derives one feedback gain. The second and the third ones are based on the TP model, that is the nonlinear combination of LTI models. The second design is very similar to the first one, but determines feedback gains by pole-placement to all LTI component of the model. Finally the control value is given by nonlinear combination of the feedback gains. The third one also generates one feedback gains to each LTI system, but the gain are oprimised by linear matrix inequalities instead of pole-placement. The performance of the three controllers are compared on the reail system.

The TP model representation belongs to the class of polytopic models. The TP model represents the Linear Parameter Varying state-space models by the parameter varying combination of Linear Time Invariant (LPV) models. The TP model transformation was proposed as a uniform and automatic way to transform given LPV models to TP model form [4, 5]. The TP model transformation was soon introduced as the Higher Order Singular Value Decomposition (HOSVD) of Linear Parameter Varying (LPV) state-space models, and the result of the TP model transformation was defined as the HOSVD-based canonical form of LPV models [20, 21]. Further, the TP model transformation offers options to satisfy various convexity constrains on the type of the resulting parameter varying combination. For instance, the Linear Matrix Inequality-based control designs [1, 2, 3], under the Parallel Distributed Compensation framework [6], can immediately be executed on the resulting polytopic model if the parameter varying combination defines a convex combination. Furthermore, if it is, for instance, define tight convex hull then the feasibility of the LMI-based design is significantly relaxed. The TP model transformation is capable of generating various types of convex parameter combinations for the resulting polytopic model automatically [10, 13, 14, 16].

The TP model transformation was applied in non-linear complex and benchmark problems for controller and observer design [8, 9, 16, 18]. The approximation properties of the TP model form were examined in papers [11, 12]. Tradeoff property of the TP model transformation was studied in [11]. Further computational improvement of the TP model transfromation is presented in [19, 20, 21, 22].

The practical advantage of the TP model transformation-based control design framework is that it can be uniformly and automatically executed on a regular computer without human interaction. Recently, the TP transformation is applied for sliding surface sector design of a variable structure system to reduce the chattering, which is the main problem of sliding mode control [15].

The paper is organized as follows: Section II discuses the theoretical background of TP model transformation-based control design. Section III introduces the laboratory development system. Section IV describes mathematical model of the experimental set up. Section V explains the basic steps of the controller design. Section VI presents the experimental results and Section VII concludes this paper.

2 Tensor Product Model Transformation-based Control Design Methology

2.1 Definition of the TP Model Form of LPV Models

Consider the following linear parameter-varying state-space model:

$$\begin{pmatrix} \dot{x}(t) \\ y(t) \end{pmatrix} = S(p(t)) \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \tag{1}$$

with input **u**(t), output **y**(t) and state vector **x**(t). The system matrix **S**(**p**(t)) $\in$ R$^{\mathrm{OxI}}$ is a parameter-varying object, where **p**(t) $\in$ Ω is time varying N-dimensional parameter vector, and is an element of the closed hypercube $\Omega = [a_1;b_1]\times[a_2;b_2]\times\ldots\times[a_N;b_N] \subset \mathrm{R}^N$. **p**($t$) can also include the elements of the state-vector **x**(t), therefore (1) is considered in the class of nonlinear dynamic state-space models.

Definition 1 ***Finite element TP model:*** *The* **S**(**p**(t)) *of (1) is given for any parameter* **p**(t) *as the convex combination of LTI system matrices* **S** *also called* vertex systems:

$$\begin{pmatrix} \dot{x}(t) \\ y(t) \end{pmatrix} = S \underset{n=1}{\overset{N}{\otimes}} w_n(p_n(t)) \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \tag{2}$$

where row vector $w_n(p_n) \in \mathrm{R}^{In}$ $n = 1,\ldots,N$ *contains one variable weighting functions* $w_{n,i_n}(p_n),(i_n = 1..I_n)$. *Function* $w_{n,i_n}(p(t)) \in [0,1]$ *is the* i_n*-th weighting function defined on the n-th dimension of* Ω, *and* $p_n(t)$ *is the n-th element of vector* **p**(t). $I_n < \infty$ *denotes the number of the weighting functions used in the n-th dimension of* Ω. *Note that the dimensions of* Ω *are respectively assigned to the elements of the parameter vector* **p**(t). *The* (N+2)*-dimensional coefficient tensor* $S \in R^{I_1 \times I_2 \times \cdots \times I_N \times O \times I}$ *is constructed from LTI vertex systems* $\mathbf{S}_{i_1 i_2 \ldots i_N} \in R^{O \times I}$.

For tensor notation we refer to [23]

Definition 2 ***Convex finite element TP model:*** *Assume that we have the explicit function (2) of a TP model. If the weighting functions satisfy that:*

$$\forall n, p_n : \sum_{i=1}^{I_n} w_{n,i}(p_n) = 1 \tag{3}$$

then (2) becomes a convex combination, namely, the LTI systems $\mathbf{S}_{i_1,\dots,i_N}$ *form a convex hull of the given LPV model.*

2.2 The Main Steps of the TP Model Transformation

The TP model transformation starts with (1) and results in the convex finite element TP model form (2) with (3). Main steps of the Tensor-Product Model Transformation is shown in Fig. 1. First the transformation space is defined by Ω which the parameter vector $\mathbf{p} \in \Omega$ varies in. Then the parameter varying system matrix is discretised in Ω. This means the computation of system matrix **S**(**g**) over the grid points **g** of a hyper rectangular grid net defined in Ω. The second step extracts the singular value-based orthonorm structure of the system, namely, this step determines the minimal number the LTI systems in orthonorm position according to the ordering of the singular values and defines the orthonorm discretised weighting functions of the searched polytopic model. The second part of this step is capable of modifying the LTI systems and the discretised weighting functions, in order to satisfy further conditions for the weighting functions. For instance, this step can ensure the convexity of the weighting functions (3). The third step determines the continuous weighting functions from the discretised ones. If the given LPV model has no TP structure, then the resulting TP model is an approximation of the given LPV model. The approximation accuracy can be controlled by the TP model transformation.

2.3 TP Model Transformation-based Control

The structure of the control design is shown in Fig. 1. The LMI-based control design theorems under the PDC framework can immediately be executed on the finite element convex TP model. The multi-objective control performance can be expressed in terms of LMIs. For instance, various LMI theorems are proposed in [6] for different conditions. We also can find a number of further and relaxed theorems in the related literature. LMI theorems are also proposed for obsrever design as well.

In conclusion, we can substitute the LTI systems of the resulting TP model into LMIs selected according to the desired control performances. The solution of these LMIs determines one LTI feedback gain **F** to each LTI system. Computing the feedback gains over the same weighting functions by tensor product gives the control value as:

$$\mathbf{u}(t) = -\mathcal{F} \bigotimes_{n=1}^{N} \mathbf{w}_n(p_n(t))\mathbf{x}(t) \tag{4}$$

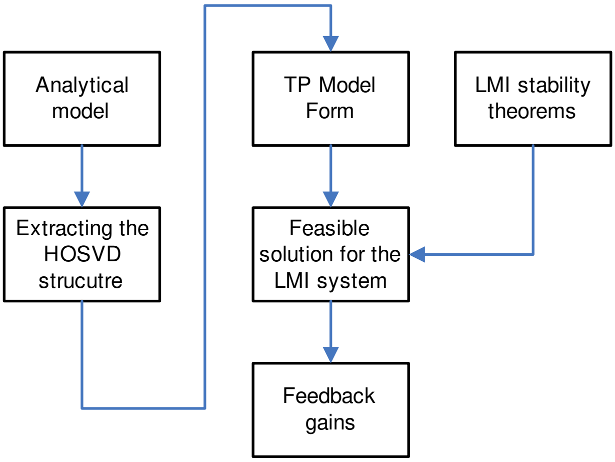

Analytical model
Extracting the HOSVD strucutre
TP Model Form
Feasible solution for the LMI system
Feedback gains
LMI stability theorems

Figure 1
Main steps of Tensor Product Model Transformation-based Control Design

This can always be given in the typical polytopic form where the same feedback gains are applied, but with linear indexing and the tensor product of the one variable weighting functions are given as multi variable weighting functions with the same indexing:

$$u(\mathrm{t}) = -\left(\sum_{r=1}^{R} w_{\mathrm{r}}(p(\mathrm{t}))F_{\mathrm{r}}\right)x(\mathrm{t}) \tag{5}$$

3 Concept of the Development System

The development system is realized in the frame of mechatronics laboratory (Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia). It is based on the different electromechanical modules, which can be controlled, analyzed and optimized from a personal computer, Fig. 2.

Instead of a separate target microcontroller for the controlling task (each electromechanical module), proposed solution is based on the microprocessor of the personal computer and advanced software tools. The testbed consists of specific electromechanical plant (mechanical subsystem) controlled by PC. The user application is modeled, simulated, programmed and run on the PC. The communication with electromechanical plant is provided by a data acquisition card (DAC) mounted in PCI slot of a personal computer and terminal board. The terminal board covers a broad range of input and output signals allowing interfacing to a variety of devices via analogue and digital signals as well as quadrature encoders. Communication between the computer and the electromechanical plant is fast

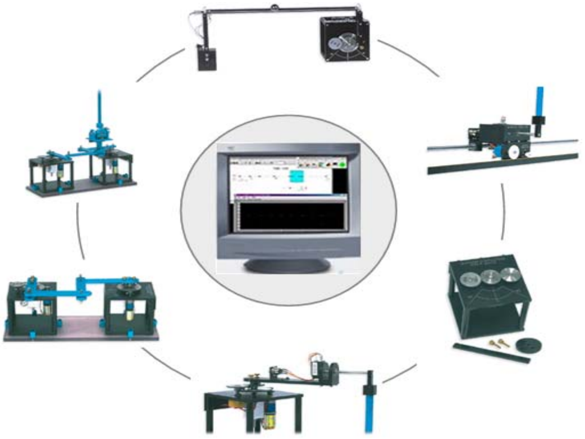


Figure 2
Development system based on different PC controlled electromechanical models

enough to ensure real time controlling of the system. This solution is based on the Windows operating system which is not real-time environment and because of that, specific and optimized software tools have to be used.

Systems ‘hardware chain’ consists of a personal computer (PC), data acquisition board (DAC), terminal board, and power supply with amplifier unit (UPM) and different electromechanical plants, Fig. 3. There are no strong demands on PC, it should be Pentium class processor or better (the faster the better), 16 MB RAM minimum, with Windows 95/98/Me/NT/2000/XP. Terminal unit is connected to DAC board supplied with 16 differential 14 bit analogue inputs, 4 analogue 12 bit outputs, 6 optical encoder inputs, 48 programmable digital inputs. Universal power module (UPM) with +/-15V, 3A has amplifier for electromechanical plant's actuators (DC motors). Electromechanical plants are modular in construction, each one has module with rotational or translational output, [24, 25]. This is according to the possibility of industrial translational and rotational crane models investigations. Rotational module is equipped with DC motor with planetary gearbox, incremental encoder as a speed feedback, load antibacklash gearbox and additional mass for experiments with variable inertia load. In rotational experiments with pendulum, incremental encoder for pendulum angle measurement is added.

Planar translational module (Single Pendulum Gantry, SPG) consists from a cart moving on the horizontal track and suspended pendulum. There is DC motor on the cart (the same as for rotational module) with planetary gearbox and two incremental encoders for cart position feedback and pendulum angle measurement. These two modules are core of practically all mechatronic experiments, aimed for TP model transformation-based controller developing, as well as for other controller types investigation.

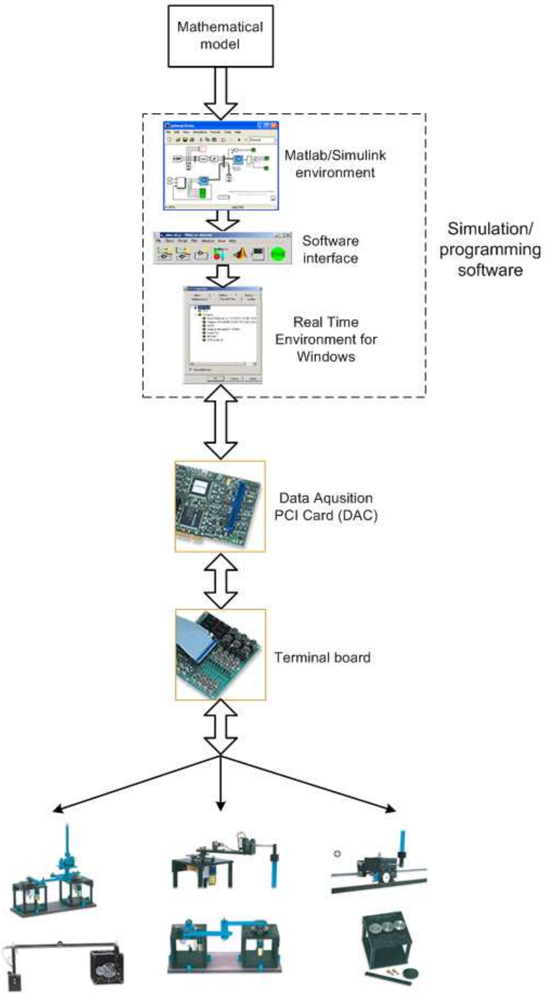
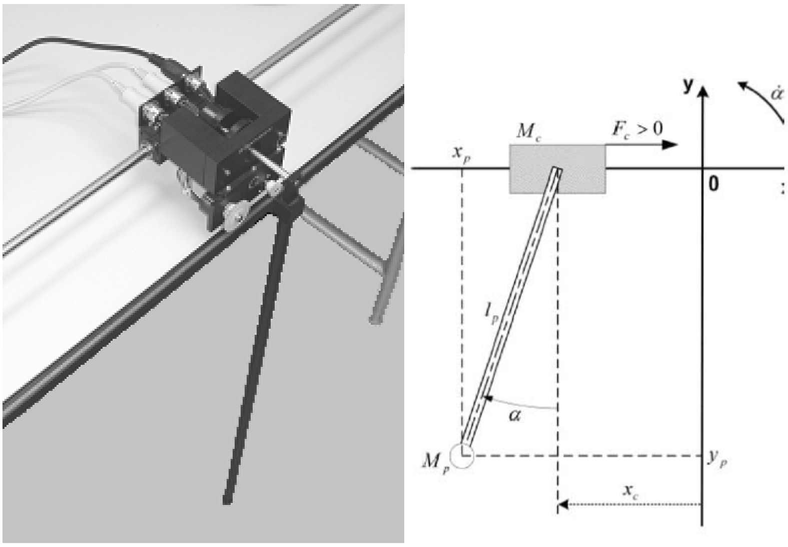

Mathematical model
Matlab/Simulink environment
Software interface
Real Time Environment for Windows
Simulation/ programming software
Data Aqusition PCI Card (DAC)
Terminal board

Figure 3
Structure block diagram of development system with different electromechanical models

Other modules are coupled with basic rotational and translational accessories (pendulums, arms, gears, etc.) forming different type of experiments. It is possible to run close to twenty different experiments with different levels of difficulty. Some of them are:

- Position and speed control with rotational and translational electromechanical plants
- Ball and beam experiment with balancing the ball on the beam
- Antipendulum control in rotational and translational moving (SISO and MIMO experiments)
- MIMO experiments with 2D gantry and 2D robot inverted pendulum
- Self erected inverted pendulum in rotational and translational moving (only SISO experiments)

The system software core is WinCon, real-time Windows 2000/XP application, [26]. It allows running code generated from a Simulink diagram in real-time on the same PC (also known as local PC) or on a remote PC. There is no need to write code by hand. Before a Simulink model may be run in real-time, it is needed first to generate the real-time code in Real-Time Workshop (RTW). Changes are as easy as modifying the Simulink diagram. Data from the real-time running code may be plotted on-line in WinCon scopes and model parameters may be changed on the fly through WinCon control panels as well as Simulink. The automatically



a) b)

Figure 4
SPG photo in mechatronics laboratory a), and schematics of the model b)

generated real-time code constitutes a stand-alone controller (i.e. independent from Simulink) and can be saved in WinCon projects together with its corresponding user-configured scopes and control panels.

4 TP Model-based Controller Design to the Single Pendulum Gantry

The Single Pendulum Gantry system, shortly described in prior section, is used for experiment verification of the TP model transformation-based position and load swing angle controller, Fig. 4a. It is also used for education and research purposes in Laboratory of Mechatronics at University of Zagreb. It is an experimental test-bed, and the goal is to design, compare and evaluate several controller approaches, [17].

4.1 Equation of Motion of the Single Pendulum Gantry

Let us consider the stabilization problem as shown in Figure 4b. Only a brief discussion is presented here, for detailed description, please, refer to [17]. Letting

$\mathbf{x} = \left(\mathrm{x}_1 \quad \mathrm{x}_2 \quad \mathrm{x}_3 \quad \mathrm{x}_4\right)^{\mathrm{T}} = \left(\mathrm{x}_{\mathrm{c}} \quad \dot{\mathrm{x}}_{\mathrm{c}} \quad \alpha \quad \dot{\alpha}\right)^{\mathrm{T}}$, the equations of motion in linear parameter-varying state-space form is:

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}(\mathbf{x})u \tag{6}$$

where

$$\mathbf{A}(\mathbf{x}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & a_1/a_x & a_2/a_x & a_3/a_x \\ 0 & 0 & 0 & 1 \\ 0 & a_4/a_x & a_5/a_x & a_6/a_x \end{pmatrix} \qquad \mathbf{B}(\mathbf{x}) = \begin{pmatrix} 0 \\ b_1/a_x \\ 0 \\ a_2/b_x \end{pmatrix}$$

$$a_1 = -\left(I_p + M_p l_p^2\right)\left(\frac{\eta_g K_g^2 \eta_m K_t K_m}{R_m r_{mp}^2} + B_{eq}\right)$$

$$a_2 = \frac{M_p^2 l_p^2 g \cos(x_3)\sin(x_3)}{x_3}$$

$$a_3 = \left(M_p^2 l_p^3 + l_p M_p l_p\right)\sin\left(x_3\right)x_4 + M_p l_p B_p \cos\left(x_3\right)$$

$$a_4 = M_p l_p \cos\left(x_3\right)\left(B_{eq} - \frac{\eta_g K_g^2 \eta_m K_t K_m}{R_m r_{mp}^2}\right)$$

$$a_5 = \frac{-\left(M_c + M_p\right)M_p l_p \sin\left(x_3\right)}{x_3}$$

$$a_6 = -\left(M_c + M_p\right)B_p - M_p^2 l_p^2 \cos\left(x_3\right)\sin\left(x_3\right)x_4$$

$$a_x = \left(M_c + M_p\right)I_p + M_c M_p l_p^2 + M_p^2 l_p^2 \sin^2\left(x_3\right)$$

$$b_1 = -\left(I_p M_p l_p\right)^2 \frac{\eta_g K_g \eta_m K_t}{R_m r_{mp}}$$

$$b_2 = -M_p l_p \cos\left(x_3\right)\frac{\eta_g K_g \eta_g K_t}{R_m r_{mp}}$$

The parameters of the experimental system are given in Table I.

TABLE I
PARAMETERS OF THE SPG SYSTEM

Description	Parameter	Value	Units
Equivalent viscoust damping coefficien	B_{eq}	5.4	N ms/rad
Viscous damping coefficient	B_p	0.0024	N ms/rad
Planetary gearbox efficiency	η_g	1	—
Motor efficiency	η_m	1	—
Gravitational constant of earth	g	9.81	m/s^2
Pendulum moment of inertia	I_p	0.0078838	kg m^2
Rotor moment of inertia	J_m	3.9001e-007	kg m^2
Planetary gearbox gear ratio	K_g	3.71	—
Back electro-motive force constant	K_m	0.0076776	Vs
Motor torque constant	K_t	0.007683	Nm/A
Pendulum length from pivot to COG	l_p	0.3302	m
Lumped mass of the cart system	M_c	1.0731	kg
Pendulum mass	M_p	0.23	kg
Motor armature resistance	R_m	2.6	Ω
Motor pinion radius	$r_m p$	0.00635	m

4.2 TP Model Representations of the Single Pendulum Gantry

Observe that the nonlinearity is caused by state values $x_3(t)$ and $x_4(t)$. The operation range of the pendulum's tip is limited to ±25deg for safety reasons, and the angular acceleration for the motor is maximum 0.7 rad/s. For the TP model transformation we define the transformation space as $\mathrm{W} = \left[\frac{-27}{180}\pi, \frac{27}{180}\pi\right] \times [-0.8, 0.8]$ (note that these intervals can be arbitrarily defined). Let the density of the sampling grid be 137×137. The sampling results in $\mathbf{A}^s_{i,j}$ and $\mathbf{B}^s_{i,j}$, where *i, j* = 1…137. Then we construct the matrix $\mathbf{S}^s_{i,j} = \left(\mathbf{A}^s_{i,j} \quad \mathbf{B}^s_{i,j}\right)$, and after that the tensor $S^s \in R^{137\times137\times4\times5}$ from $\mathbf{S}^s_{i,j}$. If we execute HOSVD on the first two dimensions of S^s then we find that the rank of S^s on the first two dimensions are 7 and 2 respectively. The singular values are as follows in the dimension x_3: $S_{1,1}$ = 1609.4, $s_{1,2}$ = 206.72, $s_{1,3}$ = 12.604, $s_{1,4}$ = 10.719, $s_{1,5}$ = 2.3109, $s_{1,6}$ = 0.14075, $s_{1,7}$ = 0.001854, and in the dimension x_4: $s_{2,1}$ = 1622.7, $s_{2,2}$ = 10.965. This means that the SPG system can be exactly given as convex combination of $7 \times 2 = 14$ linear vertex models (the L_2 numerical error of the TP model transformation for exact model is less than 10^{-12}). The TP model transformation describes SPG system as:

$$\mathbf{S}(p) = \sum_{=1}^{14} w_r(x_3, x_4)(\mathbf{A}_r x + \mathbf{B}_r u) \tag{7}$$

As in most cases it is to expensive in computational sense to work with 14 LTI models, and in real world situations the actuators accuracy is much worth than the modeling accuracy, it is possible to reduce the model. If we only keep the four biggest singular values in dimension x_3 and keep the two singular values in dimension x_4, the system can be reduced to 8 LTI models. The theoretical maximum L_2 approximation error is the sum of the discarded singular values $s_{1,5} + s_{1,6} + s_{1,7} = 2.4535$. However by checking the actual L_2 error for 10000 test points, an average maximal error of 0.080307 is received. Thus, the system can be reduced to a system of half the complexity while it is still accurate enough for real world experiments. The resulting basis functions are depicted in Figure 5.

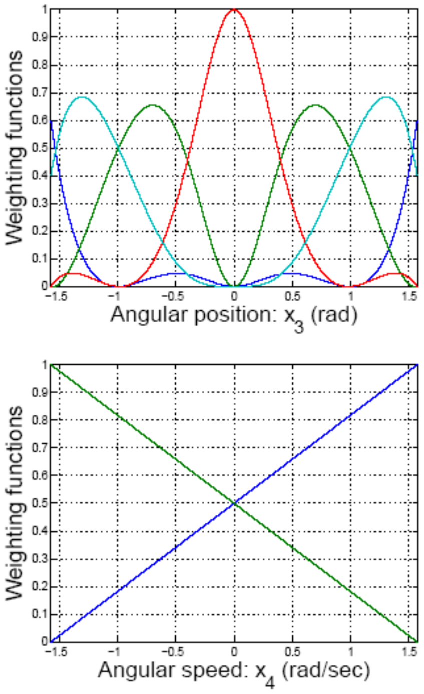


Figure 5

Weighting functions of the TP model

The LTI system matrices of the TP model are:

$$\mathbf{A}_1 = \begin{pmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & -11.2630 & 1.2457 & -0.0192 \\ 0 & 0 & 0 & 1.0000 \\ 0 & 22.8870 & -24.2374 & -0.0311 \end{pmatrix} \qquad \mathbf{B}_1 = \begin{pmatrix} 0 \\ 1.4794 \\ 0 \\ -3.0061 \end{pmatrix}$$

$$\mathbf{A}_2 = \begin{pmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & -11.2906 & 1.2657 & 0.0270 \\ 0 & 0 & 0 & 1.0000 \\ 0 & 23.1794 & -24.3744 & -0.1306 \end{pmatrix} \qquad \mathbf{B}_2 = \begin{pmatrix} 0 \\ 1.4830 \\ 0 \\ -3.0455 \end{pmatrix}$$

$$\mathbf{A}_3 = \begin{pmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & -11.8223 & 1.6427 & 0.0052 \\ 0 & 0 & 0 & 1.0000 \\ 0 & 28.5811 & -26.9299 & -0.0852 \end{pmatrix} \qquad \mathbf{B}_3 = \begin{pmatrix} 0 \\ 1.5528 \\ 0 \\ -3.7540 \end{pmatrix}$$

$$\mathbf{A}_4 = \begin{pmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & -12.4388 & 2.1008 & 0.0066 \\ 0 & 0 & 0 & 1.0000 \\ 0 & 35.3681 & -30.0863 & -0.0901 \end{pmatrix} \qquad \mathbf{B}_4 = \begin{pmatrix} 0 \\ 1.6338 \\ 0 \\ -4.6455 \end{pmatrix}$$

$$\mathbf{A}_5 = \begin{pmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & -11.2630 & 1.2457 & 0.0275 \\ 0 & 0 & 0 & 1.0000 \\ 0 & 22.8870 & -24.2374 & -0.1316 \end{pmatrix} \qquad \mathbf{B}_5 = \begin{pmatrix} 0 \\ 1.4794 \\ 0 \\ -3.0061 \end{pmatrix}$$

$$\mathbf{A}_6 = \begin{pmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & -11.2906 & 1.2657 & -0.0185 \\ 0 & 0 & 0 & 1.0000 \\ 0 & 23.1794 & -24.3744 & -0.0324 \end{pmatrix} \qquad \mathbf{B}_6 = \begin{pmatrix} 0 \\ 1.4830 \\ 0 \\ -3.0455 \end{pmatrix}$$

$$\mathbf{A}_7 = \begin{pmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & -11.8223 & 1.6427 & 0.0053 \\ 0 & 0 & 0 & 1.0000 \\ 0 & 28.5811 & -26.9299 & -0.0855 \end{pmatrix} \qquad \mathbf{B}_7 = \begin{pmatrix} 0 \\ 1.5528 \\ 0 \\ -3.7540 \end{pmatrix}$$

$$\mathbf{A}_8 = \begin{pmatrix} 0 & 1.0000 & 0 & 0 \\ 0 & -12.4388 & 2.1008 & 0.0063 \\ 0 & 0 & 0 & 1.0000 \\ 0 & 35.3681 & -30.0863 & -0.0894 \end{pmatrix} \qquad \mathbf{B}_8 = \begin{pmatrix} 0 \\ 1.6338 \\ 0 \\ -4.6544 \end{pmatrix}$$

5 Controller Design

We compare the control performances to various different alternative solutions.

5.1 Conventional Controller based on Pole Placement

CONTROLLER 1: A linearized model is selected for the conventional state feedback control design as

$$\mathbf{A}_{lin} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -11.651 & 1.521 & 0.0049 \\ 0 & 0 & 0 & 1 \\ 0 & 26.845 & -26.109 & -0.0841 \end{pmatrix} \qquad \mathbf{B}_{lin} = \begin{pmatrix} 0 \\ 1.530 \\ 0 \\ -3.526 \end{pmatrix} \tag{8}$$

The poles of the closed loop linearized system (8) with state feedback are selected in the following way

$$\mathbf{Poles} = \begin{pmatrix} -1.8182 + 1.9067i \\ -20 \\ -1.8182 - 1.9067i \\ -40 \end{pmatrix} \tag{9}$$

The state feedback control is

$$u = -\mathbf{F}x, \qquad \mathbf{F} = \begin{pmatrix} 160 & 88 & -210 & 23 \end{pmatrix}. \tag{10}$$

5.2 Derivation of TP-based Controllers

In the present case the controller (5) has the following form:

$$u = -\left(\sum_{r=1}^{8} w_r(x_3, x_4)\mathbf{F}_r \right) x\,. \tag{11}$$

Two methods are presented to define the feedback gains **F***r* for the eight systems.

CONTROLLER 2: The feedback gains **F***r* are selected separately for the all systems to place closed loop system poles to (9).

CONTROLLER 3: We design here a controller capable of asymptotically stabilize the SPG and satisfy the given constraints. We apply the following LMIs. The derivations and the proofs of these theorems are fully detailed in [6].

Theorem 1 (Asymptotic stability) *convex finite element TP model (2) with control value (5) is asymptotically stable if there exist* $\mathbf{X} > 0$ *and* **M***r* *satisfying equations*

$$-\mathbf{X}\mathbf{A}_r^T - \mathbf{A}_r\mathbf{X} + \mathbf{M}_r^T\mathbf{B}_r^T + \mathbf{B}_r\mathbf{M}_r > 0 \tag{12}$$

for all r and

$$-\mathbf{X}\mathbf{A}_r^T - \mathbf{A}_r\mathbf{X} - \mathbf{X}\mathbf{A}_s^T - \mathbf{A}_s\mathbf{X} + \mathbf{M}_s^T\mathbf{B}_r^T + \mathbf{B}_r\mathbf{M}_s + \mathbf{M}_r^T\mathbf{B}_s^T + \mathbf{B}_s\mathbf{M}_r \geq 0 \tag{13}$$

for r < s · R, except the pairs (*r; s*) *such that wr*(**p**(*t*))*ws*(**p**(*t*)) = 0,*8***p**(*t*), *and where the feedback gains are determined form the solutions* **X** *and* **M****r *as*

$$\mathbf{F}_r = \mathbf{M}_r\mathbf{X}^{-1} \tag{14}$$

In order to satisfy the constraints defined earlier, the following LMIs are added to the previous ones.

Theorem 2 (Constraint on the control value) *Assume that k***x**(0)*k* · f, *where* **x**(0) *is unknown, but the upper bound* f *is known. The constraint k***u**(*t*)*k*2 · *μ is enforced at all times t* ¸ 0 *if the LMIs*

$$\varphi^2\mathbf{I} \leq \mathbf{X}$$

$$\begin{pmatrix} \mathbf{X} & \mathbf{M}_i^T \\ \mathbf{M}_i & \mu^2\mathbf{I} \end{pmatrix} \geq 0$$

Theorem 3 (Constraint on the output) *Assume that k***x**(0)*k* · f, *where* **x**(0) *is unknown, but the upper bound* f *is known. The constraint k***y**(*t*)*k*2 · l *is enforced at all times t* ¸ 0 *if the LMIs hold.*

$$\varphi^2\mathbf{I} \leq \mathbf{X}$$

$$\begin{pmatrix} \mathbf{X} & \mathbf{X}\mathbf{C}_i^T \\ \mathbf{C}_i\mathbf{X} & \lambda^2\mathbf{I} \end{pmatrix} \geq 0$$

The bounds of the control value and the output is guaranteed by Theorem 2 and 3. Thus we solve these LMIs for the constrains together with the LMIs of Theorem 1 to guarantee asymptotic stability. By using the LMI solver of MATLAB Robust Control Toolbox, the following feasible solution and feedback gains are obtained for the controller:

$$\mathbf{F}_1 = \begin{pmatrix} 118.3947 & 51.3126 & -45.6237 & 16.4703 \end{pmatrix}$$

$$\mathbf{F}_2 = \begin{pmatrix} 118.0638 & 51.3291 & -46.1783 & 16.4069 \end{pmatrix}$$

$$\mathbf{F}_3 = \begin{pmatrix} 117.5669 & 52.2320 & -50.2620 & 15.9900 \end{pmatrix}$$

$$\mathbf{F}_4 = \begin{pmatrix} 141.1224 & 63.2115 & -56.1795 & 19.1934 \end{pmatrix}$$

$$\mathbf{F}_5 = \begin{pmatrix} 118.2570 & 51.2608 & -45.6394 & 16.4747 \end{pmatrix}$$

$$\mathbf{F}_6 = \begin{pmatrix} 118.2075 & 51.3926 & -46.1995 & 16.3999 \end{pmatrix}$$

$$\mathbf{F}_7 = \begin{pmatrix}117.5665 & 52.2318 & -50.2620 & 15.9900\end{pmatrix}$$

$$\mathbf{F}_8 = \begin{pmatrix}141.1182 & 63.2101 & -56.1805 & 19.1918\end{pmatrix}$$

6 Experimental Results

The basic structure of TP model transformation-based controller used for simulation and for real time running too, is realized in Simulink environment and presented in Fig. 6.

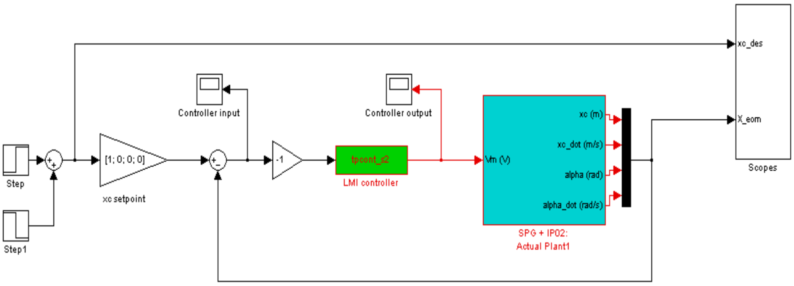


Figure 6

TP model transformation-based position and swing angle controller for SPG electromechanical system realized in Simulink environment

The experimental results with the three controllers, mentioned in chapter V, are presented in Figs. 7-9. The reference was a pulse train. In the first set of plots (Fig. 7), the time functions of the reference and the load position is shown. In the second set of plots (Fig. 8), the time functions of the angle of the load are shown. As it was expected, the performances of **CONTROLLER 1** and **CONTROLLER 2** are quite similar since they are set to have the same poles. The **CONTROLLER 3** seems to be faster but there are no significant differences among the three responses. The main difference appears in the control activity. According to Fig. 9, the **CONTROLLER 3** has the smoothest time functions.

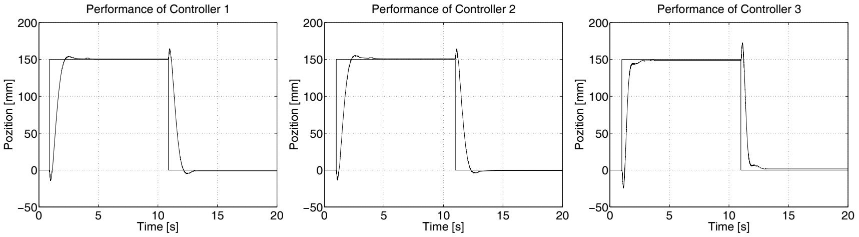


Figure 7

The position of the load (*Mp*), comparison of the performances of three controllers

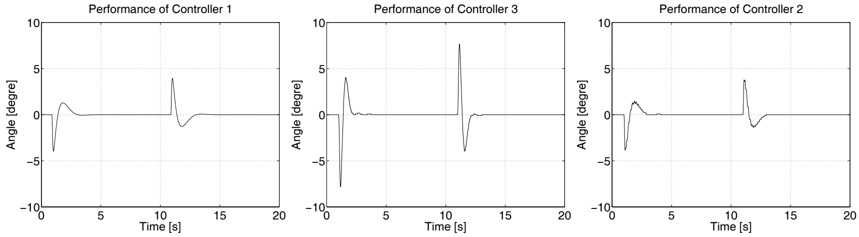


Figure 8
The angle of the load (*Mp*), comparison of the performances of three controllers

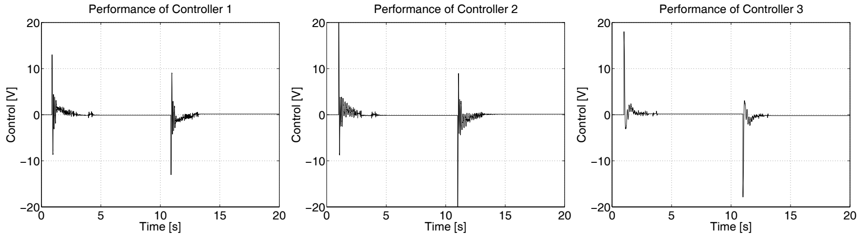


Figure 9
The control signal, comparison of the performances of three controllers

Conclusion

This paper presented a method by which a TP-based controller can be automatically designed for a non linear system using commercial Matlab functions. For investigation of such controller in real-time environment, development system with different electromechanical models with complete development tools is realized. Development system is based on the microprocessor of the personal computer and advanced software, enabling modeling, simulation, programming and real-time running of different electromechanical systems on the PC. For adequate case study, single pendulum gantry electromechanical system has been chosen, in order to mimic the real industrial task-load position and swing angle control in gantry crane load (e.g. container) handling application. Keeping the four biggest singular values for angular position and two for angular speed, 14 LTI models are reduced to 8, resulting for real-world application acceptable trade-off between modeling accuracy and computational time. Owing to this key step, the bridge between theoretical background to practical applications of TP model transformation technique is built. Experimental results realized on SPG electromechanical system, confirmed that TP model transformation-based controller can handle real time application in a good manner.

Acknoledgment

The authors wish to thank for their financial support stemming from the Hungarian-Croatian Intergovernmental Science and Technology Cooperation Program.

References

[1] S. Boyd, L. E. Ghaoui, E. Feron, V. Balakrishnan: Linear Matrix Inequalities in System and Control Theory, *Philadelphia PA:SIAM, ISBN 0-89871-334-X*, 1994

[2] P. Gahinet, A. Nemirovski, A. J. Laub, M. Chilali: *LMI Control Toolbox*. The MathWorks, Inc., 1995

[3] C. W. Scherer, S. Weiland: *Linear Matrix Iequalities in Control*, ser. DISC course lecture notes, DOWNLOAD: http://www.cs.ele.tue.nl/SWeiland/lmid.pdf, 2000.

[4] P. Baranyi, D. Tikk, Y. Yam, R. J. Patton: From Differential Equations to PDC Controller Design via Numerical Transformation, *Computers in Industry, Elsevier Science*, Vol. 51, pp. 281-297, 2003

[5] P. Baranyi: TP Model Transformation as a Way to LMI-based Controller Design, *IEEE Transaction on Industrial Electronics*, Vol. 51, No. 2, pp. 387-400, April 2004

[6] K. Tanaka, H. O. Wang: *Fuzzy Control Systems Design and Analysis — A Linear Matrix Inequality Approach*. John Wiley and Sons, Inc., 2001

[7] P. Baranyi: Tensor-Product Model-Based Control of Two-Dimensional Aeroelastic System, *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 2, March-April, 2006, pp. 391-400

[8] P. Baranyi, Z. Petres, P. Várlaki, P. Michelberger: Observer and Control law Design to the TORA System via TPDC Framework, *WSEAS Transactions on Systems*, Issue 1, Vol. 5, January 2006, pp. 156-163 (ISSN 1109-2777)

[9] G. Hancke, Á. Szeghegyi: Application Study of the TP Model Transformation in the Control of an Inverted Pendulum, in *International Conference on Computational Cybernetics (ICCC)*, Siófok, Hungary, 2003

[10] J. Bokor, P. Baranyi, P. Michelbereger, P. Várlaki: TP Model Transformation in Non-Linear System Control, in *3rd IEEE International Conference on Computational Cybernetics (ICCC)*, Mauritius, April 13-16, 2005, pp. 111-119, (plenary lecture) ISBN: 0-7803-9474-7

[11] P. Baranyi, P. Korondi, R. J. Patton, H. Hashimoto: Trade-off between Approximation Accuracy and Complexity for TS fuzzy Models, *Asian Journal of Control*, Vol. 6, No. 1, Issue 21, March 2004, pp. 21-33

[12] D. Tikk, P. Baranyi, R. J. Patton, J. Tar: Approximation Capability of TP model forms, *Australian Journal of Intelligent Information Processing Systems*, Vol. 8, No. 3, 2004, pp. 155-163

[13] Z. Petres, P. L. Várkonyi, P. Baranyi, P. Korondi: Different Affine Decomposition of the TORA System by TP Model Transformation, in *IEEE 9th*

International Conference on Intelligent Engineering Systems, Athens, Greece, September 16-19, 2005, pp. 93-98

[14] P. Baranyi, P. L. Várkonyi, P. Korondi: Different Affine Decomposition of the Model of the Prototypical Aeroelastic Wing Section by TP Model Transformation, in *IEEE 9th International Conference on Intelligent Engineering Systems*, Athens, Greece, September 16-19, 2005, pp. 105-110

[15] P. Korondi: Sliding Sector Design based on Tensor Product Model Transformation, *Transaction on Automatic Control and Computer Science*, Vol. 51, No. 1, pp. 101-108, 2006

[16] P. Baranyi: Output Feedback Control of 2-D Aeroelastic System, *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 3, May-June 2006, pp. 762-767 (ISSN 0731-5090)

[17] F. Kolonic, A. Poljugan, A. Slutej: Modern Laboratory Concept for Mechatronic Education, in *XXVII International Convention MIPRO 2004*, Opatija, Croatia, May 2004, pp. 143-146

[18] P.Baranyi: Case Study of the TP Model Transformation in the Control Design of a Complex Dynamic Model with Structural Non-Linearity, *IEEE Transaction on Industrial Electronics,* Vol. 53, No. 3, July 2006. pp. 895-904

[19] Z. Petres, P. Baranyi, H. Hashimoto: Decrease of the Computational Load of TP Model Transformation, Proc. of 3rd International Conference on Mechatronics (ICM 2006), Budapest, Hungary, July 3-5, 2006, pp. 655-659

[20] P. Baranyi, P. Várlaki, L. Szeidl: Numerical Reconstruction of the HOSVD-based Canonical Form of Polytopic Dynamic Models, in proc. 10th Int. Conf. on Intelligent Engineering Systems (INES 2006), London, United Kingdom, June 26-28, 2006, pp. 196-201 (ISBN 1-4244-9708-8)

[21] P. Baranyi, L. Szeidl, P. Várlaki, Y. Yam: Definition of the HOSVD-based Canonical Form of Polytopic Dynamic Models, Proc. of 3rd International Conference on Mechatronics (ICM 2006), Budapest, Hungary, July 3-5, 2006, pp. 660-665

[22] P. Baranyi, A. R. Várkonyi-Kóczy, Y. Yam, R. J. Patton: Adaption of TS Fuzzy Models without Complexity Expansion: HOSVD-based approach, *IEEE Transaction on Instrumentation and Measurement*, Vol. 54, No. 1, February, 2005, pp. 52-60

[23] L. D. Lathauwer, B. D. Moor, J. Vandewalle: A Multi Linear Singular Value Decomposition, Journal on Matrix Analysis and Applications, Vol. 21, No. 4, 2000, pp. 1253-1278

[24] SRV02, Instruction Manual, Quanser

[25] IP02, Instruction Manual, Quanser

[26] WinCon, Instruction Manual, Quanser

Conditions for Inference Invariant Rule Reduction in FRBS by combining rules with identical consequents

Jayaram Balasubramaniam

Department of Mathematics and Computer Sciences,
Sri Sathya Sai Institute of Higher Learning
Prasanthi Nilayam, A.P. -515134, India
jbala@ieee.org

Abstract: Following the wide spread usage of Fuzzy Systems, Rule Reduction has emerged as one of the most important areas of research in the field of Fuzzy Control. Many rule reduction methods have been proposed in the literature and can be broadly classified into Lossless *or* Lossy *with respect to the inference, based on whether the outputs of the original and the reduced rule bases are identical or not. In a typical Multi-Input-Single-Output fuzzy system the number of rules far exceeds the number of fuzzy sets defined on the output domain. This suggests that the rule base can be partitioned into sets of rules, each set being mapped to a single consequent fuzzy set. In this paper, we investigate the conditions on the inference operators employed in a fuzzy system that enable "lossless" merging of rules with identical consequents.*

After briefly surveying the many techniques that have been proposed towards reducing the number of rules, we propose a general framework for Inference in Fuzzy Systems and then propose some sufficiency conditions on this general framework that give us a class of Fuzzy Systems that allow lossless rule reduction of the type mentioned above. We then explore these conditions in the setting of Fuzzy Logic. We find that R- and S-implications play a very critical role. We give examples from the above class of Fuzzy Systems. In this study we apply the above technique only on rules whose antecedents and consequents are fuzzy sets.

Keywords: Fuzzy Systems, Rule Reduction, Residuated Implications, Strong Implications, Fuzzy Inference.

1 Introduction

Following the wide spread usage of Fuzzy Systems, Rule Reduction has emerged as one of the most important areas of research in the field of Fuzzy Control. It is well known that an increase in the number of input variables and/or the number of membership functions in the input domains quickly lead to a combinatorial explosion in the number of rules. On the other hand the number of output / consequent fuzzy sets remains a constant and is usually far less than the number of rules. This suggests that the rule base can be partitioned into sets of rules, each set being mapped to a single consequent fuzzy set. Thus the rules, though with different antecedents, but with identical consequents can be merged into a single rule. But such merger of rules, though reduces the number of rules may not be *lossless*, i.e., the inference obtained from the original rule base and the reduced rule base for a given input may not be identical. In this paper, we investigate the conditions on the inference operators employed in a fuzzy system that enable ”lossless” merging of rules with identical consequents. This provides us with a class of Fuzzy Systems in which the antecedents of the rules with identical consequent can be combined to reduce the number of rules in an inference invariant manner.

In section 2 we give a brief survey of the various rule reduction techniques proposed in the literature. In section 3 we propose a general framework for Inference in Fuzzy Systems and in section 4 we give sufficiency conditions on the inference framework that ensure lossless rule reduction of the type mentioned above. In section 5 we explore each of these conditions in the setting of Fuzzy Logic. In section 6 we give a few examples from the above class of SISO Fuzzy Systems that satisfy the above sufficiency conditions.

2 Rule Reduction as an Issue

2.1 Rule Reduction Techniques in the Literature

For an n-input Multi-Input Single-Output (MISO) fuzzy system, with n_i membership functions defined on each of the input domains $X_i (i = 1, 2, \ldots, n)$, we have $m = n_1 \times n_2 \times \ldots \times n_n = \prod_{i=1}^{n} n_i$ rules. Thus an increase in the number of input variables and/or the number of membership functions in the input domains quickly lead to a combinatorial explosion in the number of rules.

The several approaches taken towards Rule Reduction in Fuzzy Systems can be classified into the following categories:

- Selection of important rules that contribute significantly to the inference.
- Elimination of redundant rules based on some criteria.
- Merger of rules that share some common property.

2.1.1 Rule Reduction while Building a Fuzzy Rule Base

While trying to build a minimal fuzzy system, the authors in [52, 63] have employed Genetic Algorithm (GA) or GA-type optimisation to eliminate redundant rules and/or identify important or significant rules.

In [44] the authors have converted a linear fuzzy system in which the growth of the parameters with respect to inputs is exponential to an equivalent non linear fuzzy system in which their growth is linear. Works have also appeared that reduce the number of rules by reducing the number of input variables through Mathematical Fusion or through Symbolic Fusion, which involves the use of multi-dimensional fuzzy sets. In [70] a fuzzy binary box tree data structure has been proposed. In [43] the authors have designed a Fuzzy Logic Controller (FLC) based on Variable Structures techniques to be assured of Stability. They have reduced the number of rules from m^n to mn, where there are n input domains and m fuzzy sets on each domain.

2.1.2 Rule Reduction in an existing Fuzzy Rule Base

Towards reducing the number of rules in an existing fuzzy rule base, L.T. Koczy and Hirota [51] reduced a dense rule base to sparse rule base, containing the essential information in the original rule base, and all other rules were replaced by the Interpolation Algorithm that can recover them to a certain accuracy prescribed before reduction.

Following the Selection of significant rules or elimination of redundant rules, Rule Reduction has been addressed in [46,48,52] using GA and Evolutionary Algorithms, in [61,79,82] using Orthogonal Transformations, in [13] using Singular Value Decomposition, in [72] using Linear Matrix Inversion. [66] employs a Similarity Measure to prune the rules.

In [67] the authors use a similarity measure to merge rules with fuzzy antecedents and/or consequents that are similar to each other above a specified threshold. Their main stated intention is the reduction in number of fuzzy sets used in the model.

In cases where coupling effects between different inputs are small, the design of an MISO fuzzy system has been reduced to that of designing a set of SISO fuzzy systems, in a decentralised fashion, each SISO fuzzy system being designed for a pair of input-output variables. Many approaches based on the approximation or decomposition of multi-dimensional fuzzy relations into two-dimensional ones have been studied [19,47]. In [41] the conditions for reducing multi-dimensional fuzzy relations into two-dimensional ones are studied for systems using max-min composition operator. However, such approximation may lead to unsatisfactory results if some peculiarities of the process are neglected.

In hierarchical fuzzy controllers introduced in [62] the number of rules increase linearly with the number of system inputs, but the decision of where the different variables are to be put in the hierarchy is often a difficult process.

2.2 Need for Lossless Rule Reduction Techniques

Many of the rule reduction methods in the literature give rise to an approximation error, i.e., the inference obtained from the original rule base and that obtained from the reduced rule base may not be the same.

In [14] Baranyi et al, discuss the trade off between Approximation Accuracy and Complexity. See also [50] for a discussion on the trade off between computation time and precision. Thus the approximation accuracy achieved should not be sacrificed in the process of complexity reduction. All these necessitate a study on rule reduction techniques that are lossless with respect to inference.

2.2.1 Lossless Rule Reduction Techniques in the literature

A few of the rule reduction techniques that are lossless are listed below. We define ”lossless” in the sense that, the inference obtained from the original rule base and that obtained from the reduced rule base is identical.

In [64] an enhanced two-level Boolean Synthesis methodology is employed, where in, a given fuzzy rule with fuzzy connectives is mapped to a corresponding expression with boolean connectives, with each input fuzzy set being given a label. The method seeks to reduce the number of connectives employed in the antecedents of the rule. In [45] the authors in order to apply Karnaugh maps for rule reduction represent the linguistic values on a domain as 0 or 1. Though the reduced rule base can infer ”sensibly” even if the original rule base were incomplete, if the output is identical in rules where one or more antecedents are different the method does not merge these rules and thus the rule reduction is incomplete.

In [22] the authors represent a Fuzzy System as a Fuzzy Inference Graph and try to minimise the number of nodes - rules - by a two step process. Again the rule reduction is incomplete since non-interacting antecedents are not combined even though their outputs are identical and also it is lossless only for the min implication operator.

In [23] the authors have proposed a novel, though much debated [24, 25, 30, 59], rule configuration called the Union Rule Configuration, wherein the growth in number of rules is only linear instead of exponential, but the proposed method is applicable only if there is monotonicity or ordering among inputs and membership functions and a one-one correspondence between input and output membership functions.

In [49] the author follows a similar approach as ours, that of merging rules with identical consequents by proposing new fuzzy operations where certain properties of regular fuzzy operations have been either relaxed or not imposed.

In [16] Baranyi et al., discuss both exact and non-exact reduction methods using Singular Value Decomposition methods, where by removing only the zero-Singular values one obtains lossless rule reduction and in the case when all Singular values below a threshold are discarded, the error bounds for some special types of fuzzy systems are also given in [11, 12, 80, 81]. Also [14, 15, 17, 18] discuss complexity reduction in Fuzzy Rule Bases using SVD. [13, 69, 82] give an excellent review of rule reduction techniques based on Orthogonal Transformations and discuss their goodness.

2.3 Our Approach towards Lossless Rule Reduction

The approach we take towards Lossless Rule Reduction is to merge rules with identical consequents even with different antecedents. We do not propose any new fuzzy operations to this end, but obtain some conditions that the different operators employed in a fuzzy inference system should satisfy. Also the final reduced rule base, obtained by employing our method, will contain only as many rules in the rule base, as there are output membership functions that featured in the original rule base. If there are n input domains and m input fuzzy sets in each domain the total number of rules that give a complete rule base is m^n . The best theoretical limit, so far, of a reduced rule base is mn [43]. With our method it reduces to k , where k is the number of output fuzzy sets that featured in the original rule base, and typically $k \ll m$.

3 A General Framework for Inferencing in Fuzzy Systems

First we give some preliminaries on Fuzzy Logic Operators that will be required in the rest of this work. As usual we will denote by I the unit interval $[0, 1]$.

3.1 Fuzzy Logic Operators

Definition 1 ([37], Definition 1.1, Pg 3)**.** *A Negation N is a function from I to I such that:*

- $N(0) = 1;\ N(1) = 0$*;*
- *N is non-increasing.*

A negation N is called strict if in addition N is strictly decreasing and continuous. A strong negation N is a strict negation N that is also involutive, i.e., $N(N(x)) = x,\ \forall x \in I$.

Definition 2 ([34] Definition 2.1 Pg 6)**.** *A t-norm T is a function from I^2 to I such that $\forall a,\ b,\ c \in I$,*

- $T(a, 1) = a$*,*
- $T(a, b) = T(b, a)$*,*
- $T(a, T(b, c)) = T(T(a, b), c)$*,*
- $T(a, b) \leq T(a, c)$ *whenever* $b \leq c$*.*

Definition 3 ([34] Definition 3.1 Pg 10)**.** *A t-conorm S is a function from I^2 to I such that $\forall a,\ b,\ c\ \in I$,*

- $S(a, 0) = a$*,*
- $S(a, b) = S(b, a)$*,*
- $S(a, S(b, c)) = S(S(a, b), c)$*,*
- $S(a, b) \leq S(a, c)$ *whenever* $b \leq c$*.*

Definition 4 ([34] Definitions 6.1 Pg 17 & 6.11 Pg 18)**.** *A t-norm T is said to be*

- ***Continuous*** *if it is continuous in both the arguments;*
- ***Archimedean*** *if for each $(x, y) \in (0, 1]^2$ there is an $n \in \mathbb{N}$ with $x_T^{(n)} < y$, where $x_T^{(n)} = T(\underbrace{x, \cdots, x}_{n\ times})$;*
- ***Strict*** *if T is continuous and stricly monotone, i.e., $T(x, y) < T(x, z)$ whenever $x > 0$ and $y < z$;*

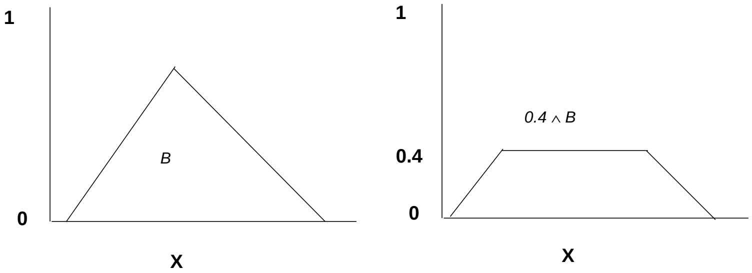


Figure 1: Fuzzy Set B (left) and the fuzzy set $0.4 \wedge B$ (right)

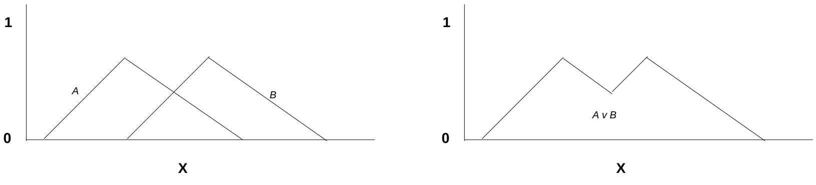
- ***Nilpotent*** *if T is continuous and if each $x \in (0,1)$ is such that $x_T^{(n)} = 0$ for some $n \in \mathbb{N}$.*

Definition 5. *If $B : X \to I$, $a \in I$, and R is any binary operator on I, i.e., $R : I \times I \to I$, then $R(a, B)$ is a fuzzy set on X, i.e., $R(a, B) : X \to I$, defined as $R(a, B)(x) = R(a, B(x)), \forall x \in X$.*

Remark 1. *Thus R can also be seen as $R : I \times \widetilde{F}(X) \to \widetilde{F}(X)$- where $\widetilde{F}(X)$ denotes the set of all fuzzy sets on X. For example if $R(a, b) = \min(a, b)$ then in Figure 1 we have $B \in \widetilde{F}(X)$ and $R(0.4, B) = \min(0.4, B) = 0.4 \wedge B \in \widetilde{F}(X), i.e. R(0.4, B)(x) = \min(0.4, B(x)) = 0.4 \wedge B(x)$, for all $x \in X$.*

Definition 6. *If $A, B : X \to I$, and R is any binary operator on I, i.e., $R : I \times I \to I$, then $R(A, B)$ is a fuzzy set on X, i.e., $R(A, B) : X \to I$, defined as $R(A, B)(x) = R(A(x), B(x)), \forall x \in X$.*

Remark 2. *Thus R can also be seen as $R : \widetilde{F}(X) \times \widetilde{F}(X) \to \widetilde{F}(X)$- where $\widetilde{F}(X)$ denotes the set of all fuzzy sets on X.*



(a) Fuzzy Sets A and B

(b) Fuzzy Set $A \vee B$

Figure 2: Fuzzy Sets A and B

For example if $R(a, b) = \max(a, b)$ then in Figure 2(a) we have $A, B \in \widetilde{F}(X)$ are fuzzy sets on X and Figure 2(b) gives $A^* = R(A, B) = \max(A, B) = A \vee B \in \widetilde{F}(X)$.

Definition 7 ([37] Definition 1.15, Pg 22)**.** *A function $J : I^2 \to I$ is called a fuzzy implication if it has the following properties:*

$$J(p,r) \geq J(q,r)\ if\ q \geq p, \tag{J1}$$

$$J(p,r) \geq J(p,s)\ if\ r \geq s, \tag{J2}$$

$$J(0,r) = 1,\ \forall r \in I, \tag{J3}$$

$$J(p,1) = 1,\ \forall p \in I, \tag{J4}$$

$$J(1,0) = 0. \tag{J5}$$

The following are the two important classes of fuzzy implications well-established in the literature:

Definition 8 ([37] Definition 1.16, Pg 24)**.** *An S-implication $J_{S,N}$ is obtained from a t-conorm S and a strong negation N as follows:*

$$J_{S,N}(a,b) = S(N(a),b), \forall a,b \in I. \tag{1}$$

Definition 9 ([37] Definition 1.16, Pg 24)**.** *An R -implication J_T is obtained from a t-norm T as its residuation as follows:*

$$J_T(a,b) = Sup\{x \in I : T(a,x) \leq b\}, \forall a,b \in I. \tag{2}$$

R- and S-implications satisfy (J1) - (J5). Tables 1 and 2 list few of the well-known S-implications and R-implications, respectively.

Name	$S(a,b)$	$N(a,b)$	$J_{S,N}(a,b)$
Dienes	$max(a,b)$	$1-a$	$max(1-a,b)$
Reichenbach	$a+b-ab$	$1-a$	$1-a+ab$
Lukasiawicz	$min(1,a+b)$	$1-a$	$min(1,1-a+b)$

Table 1: Some of the well known S-implications with their corresponding t-conorms

t-norm	$T(a,b)$	Implication	$J_T(a,b)$
Lukasiawicz	$max(0,a+b-1)$	Lukasiawicz	$min(1,1-a+b)$
Mamdani	$min(a,b)$	Godel	$\begin{cases} 1, & \text{if } a \leq b \\ b, & \text{otherwise} \end{cases}$
Larsen	$min(1,a+b)$	Goguen	$\begin{cases} 1, & \text{if } a \leq b \\ b/a, & \text{otherwise} \end{cases}$

Table 2: Some of the well-known R-implications and their corresponding t-norms

3.2 Fuzzy If-Then Rules

A linguistic statement ”$x\ is\ A$” is interpreted as the variable x taking the linguistic value A. For example, if x denotes ”Temperature” (on a suitable domain), then it can assume the following linguistic values A, viz., high, more or less high, medium, cool, very cold, etc. Each of the linguistic values (say cool) is represented by a fuzzy set on the domain X of the linguistic variable x, i.e., $A : X \to I$. The shape of the graph of the function represents the concept (say high temperature). The concept of high temperature is again context-dependent. For example, high temperature (fever) for a human being is different from the high temperature in a blast furnace, and accordingly the domain of the linguistic variable is selected.

A Fuzzy If-Then rule is of the form

$$\text{If } x \text{ is } A \text{ Then } y \text{ is } B, \tag{3}$$

where x, y are variables and A, B are linguistic expressions / values assumed by the linguistic variables. For example,

”If x (temperature) is A (High)
Then y (Pressure) is B (Low)”

The above is an example of a SISO rule. A Two-Input Single-Output rule is of the form

$$R_1 : \text{If } x \text{ is } A \text{ and } y \text{ is } B \text{ Then } z \text{ is } C,$$

where again A, B, C are linguistic values taken by the linguistic variables x, y, z over their respective domains.

3.3 Different Stages in the inferencing of a Fuzzy System

Let us consider the following system of m fuzzy if-then rules:

$$\begin{aligned} R_1 &: If\ x_1\ is\ A_1^1,\ \ldots, x_n\ is\ A_n^1\ Then\ y\ is\ B_1 \\ &\vdots \\ R_j &: If\ x_1\ is\ A_1^j,\ \ldots, x_n\ is\ A_n^j\ Then\ y\ is\ B_j \\ &\vdots \\ R_m &: If\ x_1\ is\ A_1^m,\ \ldots, x_n\ is\ A_n^m\ Then\ y\ is\ B_m \end{aligned} \tag{4}$$

where $A_i^j \in \widetilde{F}(X_i)$ for $i = 1, 2, \ldots, n$ are the antecedent fuzzy sets over the n non-empty domains $X_1, X_2, \ldots, X_n$. For $j = 1, 2, \ldots, m$, B_j can be a fuzzy set on the non-empty output domain Y, i.e., $B_j \in \widetilde{F}(Y)$, as in the case of a Mamdani Fuzzy System, or $B_j \in Y$ as in the case of a constant-output Takagi-Sugeno-Kang Fuzzy Systems.

In the following we propose a general framework for Inference in Fuzzy Rule Based Systems that captures the working of both the established models of Fuzzy Systems - TSK and Mamdani models of Inference. Towards this end, a Fuzzy System can be seen to consist of the following 5 stages:

3.3.1 Fuzzifier

If the given input is a crisp number $x \in X$, it is fuzzified to get a fuzzy set $\widetilde{X}$ on the corresponding input space X, i.e., $C : X \to \widetilde{F}(X)$, where $C(x) = \widetilde{X}$. Thus given a vector of crisp points $\overline{x} = [x_1, x_2, \ldots, x_n]$, where $x_i \in X_i$, for every input space X_i, we get a vector of Fuzzy sets $\overline{X} = [\widetilde{X_1}, \widetilde{X_2}, \ldots, \widetilde{X_n}]$. The often used [40,79] *Singleton Fuzzifier* of a crisp number x is given as

$$\widetilde{X}(y) = \begin{cases} 1 & \text{if } y = x \\ 0 & \text{otherwise} \end{cases} \qquad (SF^*)$$

Remark 3. *It can be readily seen that the above stage of "Fuzzifier" is the reverse of "Defuzzification" - wherein we obtain a crisp number from a fuzzy set (See 3.3.6 below). Though in many actual implementations of Fuzzy Systems a crisp value is directly given as input, the above stage has been added for generality. Also many times the input given to a fuzzy system is not precise owing to many types of observation errors. For example, a reading from a sensor that becomes an input for the controlling fuzzy system may be inherently imprecise due to instrument errors. In such cases a fuzzy set about the reading may be a more realistic input. In this paper, crisp inputs are identified with their fuzzified version as given by* (SF^*).

3.3.2 Matching

The input fuzzy sets $(\widetilde{X_1}, \widetilde{X_2}, \ldots, \widetilde{X_n})$ are matched against their corresponding if-part fuzzy sets in each of the rule antecedents in the Fuzzy System, i.e.

$$M : \widetilde{F}(X_i) \times \widetilde{F}(X_i) \to I \tag{5}$$

where $M(A_i^j, \widetilde{X_i}) = a_i^j$ for A_i^j and $\widetilde{X_i} \in \widetilde{F}(X_i)$, $j = 1, \ldots, m$.

A few matching functions used in the literature are given later in section 5.4.1.

3.3.3 Combining

In a multi-antecedent fuzzy system, the various matching degrees a_i^j of the n input fuzzy sets to the antecedent of the j^{th} fuzzy if-then rule are combined to give the *"fit values"* μ_j ,

$$\mu : I^n \to I \tag{6}$$

where $\mu(a_1^j, \ldots, a_n^j) = \mu_j$, $j = 1, 2, \ldots, m$. μ can be any t- or t-conorms (see Section 3.1).

3.3.4 Rule Firing

The combined value μ_j fires the rule consequent or the output fuzzy set B_j of the j^{th} rule. This B_j can be a fuzzy set on Y,i.e.,$B_j \in \widetilde{F}(Y)$, or a value in Y, i.e.,$B_j \in Y$. Thus we have

$$f : I \times Z \to Z \tag{7}$$

- When $Z = \widetilde{F}(Y)$ - the set of all fuzzy sets on the output domain Y, i.e.,$B_j \in \widetilde{F}(Y)$, $f(\mu_j, B_j) = f_j \in \widetilde{F}(Y)$ and is defined as $f(\mu_j, B_j(y)) = f_j(y), \forall y \in Y$
- When $Z = Y$, the output domain itself, i.e., $B_j \in Y$, then $f_j \in Y$ and is defined as $f(\mu_j, B_j) = f_j$.

Usually, $f = \pi$, the product, is commonly employed when $B_j \in Y \subseteq \mathbb{R}$, while a t-norm or any Fuzzy Implication Operator (Section 3.1) is the prefered choice if $B_j \in \widetilde{F}(Y)$.

3.3.5 Aggregation of Individual Inferences:

The fired output fuzzy sets (or crisp real numbers) $f_j,\ j = 1, 2, \ldots, m$ are then aggregated to obtain the final inferred fuzzy set (or crisp real number)

$$g : Z^m \to Z \tag{8}$$

where again

- If $Z = \widetilde{F}(Y)$, the infered output set $g(f_1, \ldots, f_m) = B \in \widetilde{F}(Y)$. One can use any of the fuzzy logic operators, t- or t-conorms, to obtain $B \in \widetilde{F}(Y)$.
- If $Z = Y \subseteq \mathbb{R}$ then the Weighted Average or the Weighted Sum are the commonly used aggregation operators for g.

3.3.6 Defuzzification

When $Z = \widetilde{F}(Y)$, $g(f_1, \ldots, f_m) = B \in \widetilde{F}(Y)$ and we need to defuzzify B - a fuzzy set on Y - to a single value $b \in Y$, using an appropriate defuzzification method h as follows:

$$h : \widetilde{F}(Y) \to Y \tag{9}$$

The Centre of Area or the Mean of Maxima methods [42, pp. 336 - 338] are the most widely used Defuzzification methods.

The different stages and the corresponding mappings capturing their actions are given in Table 3.

3.4 Different Models of Fuzzy System in the literature

Following are the two most established models of Fuzzy Systems:

Fuzzifier:	$\widetilde{X} = C(x)$	$C : X \to \widetilde{F}(X)$
Matching:	$a_i^j = M(A_i^j, \widetilde{X}_i)$	$M : \widetilde{F}(X_i) \times \widetilde{F}(X_i) \to I$
Combining:	$\mu_j = \mu(a_1^j, \ldots, a_n^j)$	$\mu : I^n \to I$
Firing:	$f_j = f(\mu_j, B_j)$	$f : I \times Z \to Z, Z = \widetilde{F}(Y)\ or\ Y$
Aggregation:	$B = g(f_1, \ldots, f_m)$	$g : Z^m \to Z$
Defuzzification:	$b = h(B)$	$h : \widetilde{F}(Y) \to Y$

Table 3: Different stages of a Fuzzy System

3.4.1 Mamdani Fuzzy System

E.H. Mamdani and S. Assilian [57] proposed the first type of Fuzzy Rule Based Systems. The rules in a Mamdani Fuzzy System are specified linguistically both for antecedents and consequents. Given a vector of crisp inputs $\overline{x'} = [x'_1, x'_2 \ldots, x'_n]$, where $x'_i \in X_i$, the final output fuzzy set B on Y for the fuzzy rule base in (4) is obtained as follows:

$$B(y) \quad = \bigvee_{j=1}^{m} \{[\bigwedge_{i=1}^{n} a_i^j] \wedge B_j(y)\}, \forall\ y \in Y \tag{10}$$

where $a_i^j = A_i^j(x'_i)$.

Though the Mamdani model is usually used with crisp inputs, it can handle both crisp and fuzzy inputs. In the case of a fuzzy inputs, say $x_1\ is\ A_1,\ \ldots, x_n\ is\ A_n$, where A_i is a fuzzy set on the domain X_i, the final output fuzzy set B on Y for the fuzzy rule base in (4) is given by (10), but with a_i^j given by (11)

$$a_i^j = max_{x \in X_i}\{min(A_i^j(x), A_i(x))\} \tag{11}$$

Also in the case of a crisp input, the crisp input can be singleton fuzzified by *(SF*)* (Section 3.3.1) into a fuzzy set and can be given as an input to the fuzzy system. Thus given a vector of crisp inputs $\overline{x'} = [x'_1, x'_2, \ldots, x'_n]$, where $x'_i \in X_i$, for every input space X_i, we get a vector of fuzzy inputs $\overline{X} = [\widetilde{X_1}, \widetilde{X_2}, \ldots, \widetilde{X_n}]$. It can be easily seen that if instead of the crisp inputs x'_i, if their corresponding singleton fuzzified inputs are given, i.e., $A_i = \widetilde{X}_i$ are inputs, $a_i^j = A_i^j(x'_i) = max_{x \in X_i}\{min(A_i^j(x), A_i(x)\} = max_{x \in X_i}\{min(A_i^j(x), \widetilde{X}_i(x)\}$. Thus we can always consider an input for the Mamdani model of fuzzy system to be fuzzy, with the understanding that any crisp input is singleton fuzzified according to *(SF*)* and (10) can be employed with (11).

Let the Matching Function $max_{x \in X}\{min(A(x), B(x))\}$ of two fuzzy sets $A, B : X \to I$ be denoted by $M_1(A, B)$. Now, comparing the inference in (10) to the different stages in Section 3.3, it can be seen that $M = M_1, \mu = \wedge, f = \wedge, g = \vee$ and $Z = \widetilde{F}(Y)$.

3.4.2 Takagi - Sugeno - Kang Fuzzy System

Instead of working with the linguistic rules of the kind employed in Mamdani Fuzzy Systems, Takagi and Sugeno [71] proposed a new model based on rules whose antecedent is composed of linguistic variables and the consequent is represented by a real function of the input variables. TSK model differs from the Mamdani model both in the form of their rules and the inference operators used. If in the case of Mamdani model of a SISO fuzzy system a fuzzy rule has the form (3)

$$If\ x\, is\, A\ Then\ y\, is\, B$$

where A and B are fuzzy sets on X and Y, respectively, then in the case of the TSK model the rules have the form (12)

$$If\ x\, is\, A\ Then\ y = b(x) \tag{12}$$

and the input is a crisp value for x. Their conclusion contains the real valued function $b(x)$ and not a fuzzy set. This function can be non-linear, although usually linear functions are applied. Then the TSK rules have the form:

$$If\ x\, is\, A\ Then\ y = px + q \tag{13}$$

where the input is a crisp value for x and p, q are constants. In general the rules of a SISO and MISO TSK fuzzy systems are of the form given by (14) and (15), respectively.

$$R_j\ :\quad If\ x\, is\, A_j\ Then\ y\ = b_j(x) \tag{14}$$

$$R_j\ :\quad If\ x_1\ is\ A_1^j,\ \ldots,\ \ x_n\ is\ A_n^j\ Then\ y\ = b_j(\overline{x}) \tag{15}$$

for $j = 1, \ldots, m$ and the input vector $\overline{x'} = [x'_1, x'_2, \ldots, x'_n]$ and each x'_i is a crisp value in X_i for $i = 1, \ldots, n$.

Let us again consider a fuzzy rule base of m rules of the form (15) and a vector of crisp inputs $\overline{x'} = [x'_1, x'_2, \ldots, x'_n]$, where $x'_i \in X_i$, be given. In the TSK model of fuzzy systems, the final crisp output is obtained as the Weighted Sum of *"fit values"* and the rule consequents as given in (16).

$$F(\overline{x'}) = \sum_{j=1}^{m} \mu_j(\overline{x'}) \cdot b_j(\overline{x'}) \tag{16}$$

where $\mu_j(\overline{x'}) = \Pi_{i=1}^{n} a_i^j = \Pi_{i=1}^{n} A_i^j(x'_i) = A_1^j(x'_1) \cdot A_2^j(x'_2) \cdot \ldots \cdot A_n^j(x'_n)$.

As in Section 3.4.1, by taking the singleton fuzzified crisp input vector $\overline{x'}$, as given by *(SF*)*, it can be seen that, if $A_i = \widetilde{X}_i$ are inputs, $a_i^j = A_i^j(x'_i) = max_{x \in X_i}\{min(A_i^j(x), A_i(x)\} = max_{x \in X_i}\{min(A_i^j(x), \widetilde{X}_i(x)\}$. Thus again one can always consider the singleton fuzzified fuzzy set $\widetilde{X}_i$ of a crisp input x'_i as being the input for a TSK model of fuzzy system. Also product is the antecedent combiner, i.e., $\mu = \prod$. Though the product between the "fit value" of the given input to the antecedents of rule j, $\mu_j(\overline{x'})$ and its consequent $b_j(\overline{x'})$ is an effect of the Weighted Sum aggregation employed and is not a rule connective, per se, one can perhaps consider it such so that $f = \pi$ for the TSK model in the above framework, i.e., $f : I \times Z \to Z$ is such that $f(\mu_j(\overline{x'}), b_j(\overline{x'})) = \mu_j(\overline{x'}) \cdot b_j(\overline{x'})$, where $Z = Y \subseteq \Re$, the actual domain of the output fuzzy sets.

Now, comparing the inference in (16) to the different stages in Section 3.3, it can be seen that $M = M_1, \mu = \pi, f = \pi, g = \Sigma$ and $Z = Y \subseteq \Re$.

From the above two sections, it is clear that the different stages in the inference of an output, given an input, in a fuzzy system can be mapped to different functions capturing the actions performed at every stage.

Definition 10. *A model of Inference in a fuzzy system is given by the quintuple $Q = \{M, \mu, f, g, Z\}$ where M, μ, f, g are the corresponding operators of the above framework and Z is the domain of consequents of the rule.*

Thus Mamdani Model of inference in a fuzzy system is defined as the quintuple $Q_M = \{M_1, \wedge, \wedge, \vee, \widetilde{F}(Y)\}$ while the TSK model of inference in a fuzzy system is given by $Q_{TSK} = \{M_1, \prod, \prod, \sum, Y \subseteq \Re\}$. We do not consider the fuzzifier stage since a crisp input to the fuzzy system can be thought of as a singleton fuzzified input fuzzy set using (SF^*). Table 4 summarises the above discussion, where $\prod$ = Product, $\sum$ = Sum, $\vee$ = max, $\wedge$ = min.

Name / Type	M	μ	f	g	Fuzzifier	Z
TSK	M_1	$\prod$	$\prod$	Σ	SF^*	$Y \subseteq \Re$
Mamdani	M_1	$\wedge$	$\wedge$	$\vee$	SF^*	$\widetilde{F}(Y)$

Table 4: M, μ, f, g and Z for the different models of fuzzy systems in Section 3.4

3.5 A Rule Reduction Technique for a Class of Fuzzy Systems

More often than not, the number of fuzzy sets, k, defined on the single output domain Y, is typically much less than the number of rules m, i.e., $k \ll m$. This suggests that the antecedents of more than one rule lead to the same consequent. To eliminate this redundancy, we propose a new type of off-line rule reduction where the rules with the same consequent but different antecedents are merged into a single rule. Then we will have only as many rules as there are output membership functions, in fact only those that are part of the original fuzzy system.

The issue involved here is that despite the merging of the above rules, there should be no loss of inference, i.e., the inference obtained from the original rule base and that obtained from the reduced rule base should be identical. This necessitates the functions M, μ, f and g to possess some properties. These are explored in the next section.

Remark 4. *Also in the rest of the paper, we will only consider fuzzy rules (SISO or MISO, as the case may be) of the type where both the antecedents and consequents* *are fuzzy sets on their respective domains. The inputs may be crisp or fuzzy. In the case of crisp inputs, we will consider their fuzzified version as obtained using* (SF^*) *on the input.*

4 Conditions on the General Framework for Lossless Rule Reduction

The rule reduction procedure we propose is an offline procedure, i.e., from the given original rule base we club the rules with same consequent, but different antecedents, to produce new rules to replace old rules. In this, section we determine the structure of the antecedents of the newly formed rules. The following theorem gives sufficient conditions that the operators of the above proposed framework should satisfy to obtain *lossless* or *inference invariant* rule reduction by combining antecedents of rules that have identical consequents.

4.1 The Restrictions on g, f, μ and M :

Theorem 1. *Let a model of inference in a fuzzy system be defined by* $Q = \{M, \mu, f, g, Z = \widetilde{F}(Y)\}$*, where* Y *is the output domain. The following conditions on the operators* M, μ, f *and* g *are sufficient to ensure inference invariant rule reduction, by combining antecedents of rules that have identical consequents, in any MISO fuzzy system:*

There exist operators g, o_g, o_μ *which are commutative and associative binary operators on* I *and for any* $a, b, a_1, a_2, b_1, b_2 \in I$, A_1, A_2, A *which are fuzzy sets defined on an input domain* X *and* $C \in \widetilde{F}(Y)$,

$$g[f(a,C), f(b,C)] = f(a\ o_g\ b, C) \tag{17}$$

$$\mu(a_1, b_1)\ o_g\ \mu(a_2, b_2) = \mu(a_1\ o_\mu\ a_2, b_1\ o_\mu\ b_2) \tag{18}$$

$$M(A_1, A)\ o_\mu\ M(A_2, A) = M(A_1\ o_\mu\ A_2, A) \tag{19}$$

Remark 5. *In the LHS of (19)* o_μ *is a binary operator on I while in the RHS of (19)* o_μ *is the extension of* o_μ *to fuzzy sets on X (See Definition 6 and Remark 2 in Section 3.1).*

Proof. Without loss of generality, let us take a 2-input 1-output fuzzy system consisting of three rules, where X_1 and X_2 are the input domains and Y the output domain. Consider the fuzzy system given by the following rules, written in a simplified form:

$$\begin{aligned} R_1 &: A_1, B_1 \to C \\ R_2 &: A_2, B_2 \to C \\ R_3 &: A_3, B_3 \to D \end{aligned} \tag{20}$$

where A_1, A_2, A_3 are fuzzy sets on X_1; B_1, B_2, B_3 are fuzzy sets on X_2 and C, D are fuzzy sets on Y.

Let us consider the inference in the above MISO - fuzzy system in the presence of an input, say "x_1 *is* A *and* x_2 *is* B", which is represented as $X = (A, B)$, where $A \in \widetilde{F}(X_1)$ and $B \in \widetilde{F}(X_2)$. The MISO inference from the original rule base (20) is given as:

$$\begin{aligned} g\{ \quad & f[\mu(M(A_1, A), M(B_1, B)), C], \\ & f[\mu(M(A_2, A), M(B_2, B)), C], \\ & f[\mu(M(A_3, A), M(B_3, B)), D]\} \end{aligned} \tag{21}$$

Letting $\mu(M(A_1, A), M(B_1, B))$ as a and $\mu(M(A_2, A), M(B_2, B))$ as b we have from (17), with g being associative,

$$\begin{aligned} (21) \quad = \quad & g\{g\{f[\mu(M(A_1, A), M(B_1, B)), C], \\ & f[\mu(M(A_2, A), M(B_2, B)), C]\}, \\ & f[\mu(M(A_3, A), M(B_3, B)), D]\} \\ = \quad & g\{f[\mu(M(A_1, A), M(B_1, B)) \\ & \; o_g \; \mu(M(A_2, A), M(B_2, B)), C], \\ & f[\mu(M(A_3, A), M(B_3, B)), D]\} \end{aligned} \tag{22}$$

Again letting $M(A_i, A) = a_i \in I, M(B_i, B) = b_i \in I, \; i = 1, 2$, we have using (18) and (19)

$$\begin{aligned} (22) \quad = \quad & g\{f[\mu(M(A_1, A) \; o_\mu \; M(A_2, A), \\ & \qquad M(B_1, B) \; o_\mu \; M(B_2, B)), C], \\ & \; f[\mu(M(A_3, A), M(B_3, B)), D]\} \end{aligned} \tag{23}$$

$$\begin{aligned} = \quad & g\{f[\mu(M(A_1 \; o_\mu \; A_2, A), M(B_1 \; o_\mu \; B_2, B)), C], \\ & f[\mu(M(A_3, A), M(B_3, B)), D]\} \end{aligned} \tag{24}$$

Thus the rule base in (20) can be reduced to the following rule base containing just two rules:

$$R_1^* \quad : A_1 \; o_\mu \; A_2, B_1 \; o_\mu \; B_2 \rightarrow C \tag{25}$$
$$R_3 \quad : A_3, B_3 \rightarrow D$$

It can be easily seen that for a given input $X = (A, B)$, the inference obtained from the reduced rule base (25) under the given model of inference Q is identical to (24). □

The above requirements on the general framework give us a class of Fuzzy Systems that allow lossless rule reduction by combining rules with same consequent.

5 Analysis of the Requirements for Lossless Rule Reduction

In this section, we explore each of the above conditions for Lossless Rule Reduction in the setting of Fuzzy Logic operators.

5.1 Conditions on g, o_g and o_μ

In this study we consider only continuous t-norms and t-conorms for g, o_g and o_μ, which are by definition both commutative and associative. This enables us to extend them to functions from I^n to I in the case of n-input domains.

5.2 On the Equation (17)

Typically in a Fuzzy System f (the rule firing operation) is interpreted either as a t-norm, for example Mamdani's *min*, or a Fuzzy Implication operator. In this work we investigate the solution of (17) both with f as a Fuzzy Implication Operator and as a t-norm. We explore equation (17) in the following way:

- Fix f to be in a specific class of Fuzzy Implications or any t-norm and vary g and hence o_g over conitnuous t-norms and t-conorms.

In this study, we consider the two most established and well studied families of Fuzzy Implication Operators, viz., R- and S-Implications. (See Definitions 8 and 9 in Section 3.1).

When f is fixed in (17), g and o_g are taken to be any S/T-norms, we have the following 4 possibilities:

$$f(T(p,q),r) = S(f(p,r), f(q,r)) \tag{26}$$

$$f(S(p,q),r) = T(f(p,r), f(q,r)) \tag{27}$$

$$f(T_1(p,q),r) = T_2(f(p,r), f(q,r)) \tag{28}$$

$$f(S_1(p,q),r) = S_2(f(p,r), f(q,r)) \tag{29}$$

5.2.1 $f = J$ a Fuzzy Implication

Fixing f to be any Fuzzy Implication J, we get the following four equations from the above:

$$J(T(p,q),r) = S(J(p,r), J(q,r)) \tag{30}$$

$$J(S(p,q),r) = T(J(p,r), J(q,r)) \tag{31}$$

$$J(T_1(p,q),r) = T_2(J(p,r), J(q,r)) \tag{32}$$

$$J(S_1(p,q),r) = S_2(J(p,r), J(q,r)) \tag{33}$$

Recently with $f = J$ interpreted as an R- or an S-implication and $g = S$, an t-conorm and $o_g = T$, a t-norm, Trillas and Alsina [76] have investigated (30) and proven the following:

Theorem 2. *An S- or an R-implication J satisfies (30) iff $S = max$ and $T = min$.*

In [7] the authors have proven the following Theorem 3 concerning equation (31) obtained by letting $f = J$ to be an R- or an S-implication and $o_g = S$, an t-conorm and $g = T$, a t-norm,

Theorem 3. *An S- or an R-implication J satisfies (31) iff $S = max$ and $T = min$.*

Also we have the following result:

Lemma 1. *For no Fuzzy Implication J, t-norm T (t-conorm S, respectively) do equations (32) ((33), respectively) hold.*

Proof. Let $p = 1, q = r = 0$. Then using the property of t-norms, $T(1, 0) = 0$ and (J3), we have that,

$$\begin{aligned} LHS\ of\ (32) &= J(T_1(1, 0), 0) = J(0, 0) = 1 \\ RHS\ of\ (32) &= T_2(J(1, 0), J(0, 0)) = T_2(0, 1) = 0 \end{aligned}$$

LHS = *RHS* implies that $1 = 0$, which is absurd. Similarly, that (33) does not have a solution can be seen by again fixing $p = 1, q = r = 0$. □

5.2.2 $f = T$ a t-norm

In [76] it is also shown that (30) does not hold for the Mamdani's Minimum $f = J = \wedge$ and the Larsen's Product $f = J = \prod$ operators. That (26) and (27) do not hold when f is any t-norm T can be easily seen by taking $p = r = 1$ and $q = 0$. Thus fixing $f = T$ to be a t-norm,we need to consider only the equations (28) and (29) which become:

$$T(T_1(p, q), r) = T_2(T(p, r), T(q, r)) \tag{34}$$

$$T(S_1(p, q), r) = S_2(T(p, r), T(q, r)) \tag{35}$$

We have the following theorems:

Theorem 4. *(34) is valid iff when $T_1 \equiv T_2 = min$.*

Proof. Claim: $T_1 \equiv T_2$ on $I \times I$.

Let $r = 1$. Then $\forall p, q \in I$, we have

LHS of (34) = $T(T_1(p, q), 1) = T_1(p, q)$

RHS of (34) = $T_2(T(p, 1), T(q, 1)) = T_2(p, q)$ = LHS of (34) $\forall p, q \in I$ iff $T_1 \equiv T_2$.

Now, let $p = q = 1, r \in I$. Then

LHS of (34) = $T(T_1(1, 1), r) = T(1, r) = r$.

RHS of (34) = $T_1(T(1, r), T(1, r)) = T_1(r, r) = r$, $\forall r \in I$ iff $T_1 = min$, the only idempotent t-norm. □

Theorem 5. *(35) is valid iff when $S_1 \equiv S_2 = S$ and T distributes over S.*

Proof. *Claim:* $S_1 \equiv S_2$ on $I \times I$.

Let $r = 1$. Then $\forall p, q \in I$, we have
LHS of (35) $= T(S_1(p, q), 1) = S_1(p, q)$
RHS of (35) $= S_2(T(p, 1), T(q, 1)) = S_2(p, q) =$ LHS of (35) $\forall p, q \in I$ iff $S_1 \equiv S_2$.
Thus the equation (35) becomes

$$T(S(p, q), r) = S(T(p, r), T(q, r)) \tag{36}$$

which is true iff T distribuites over S. □

Corollary 1. *(36) is true if $S = max$.*

5.3 On the Equation (18)

Continuing along the same vein, we have investigated the generalised bisymmetry equation (18)

$$\mu(a_1, b_1)\ o_g\ \mu(a_2, b_2) = \mu(a_1\ o_\mu\ a_2, b_1\ o_\mu\ b_2)$$

involving o_g, μ and o_μ, with $a_1, a_2, b_1, b_2 \in I$.

Definition 11. *A function $B : [a, b]^2 \to [a, b]$ is said to be bisymmetric if*

$$B(B(x, y), B(u, v)) = B(B(x, u), B(y, v)),\ \forall x, y, u, v \in [a, b].$$

For a comprehensive coverage on Bisymmetry Equations refer [1 - 4, 74]. Also [38,39] list many results on bisymmetry equations on the unit interval. Allowing o_g, μ and o_μ to be t- and t-conorms, we get the following 8 possible cases in all, which for convenience we have grouped into two sets:

Group 1

$$T_1(T_2(a_1, b_1), T_2(a_2, b_2)) = T_2(T_3(a_1, a_2), T_3(b_1, b_2)) \tag{37}$$
$$S_1(S_2(a_1, b_1), S_2(a_2, b_2)) = S_2(S_3(a_1, a_2), S_3(b_1, b_2)) \tag{38}$$

Group 2

$$T_1(S(a_1, b_1), S(a_2, b_2)) = S(T_3(a_1, a_2), T_3(b_1, b_2)) \tag{39}$$
$$T_1(T_2(a_1, b_1), T_2(a_2, b_2)) = T_2(S(a_1, a_2), S(b_1, b_2)) \tag{40}$$
$$T_1(S_1(a_1, b_1), S_1(a_2, b_2)) = S_1(S_2(a_1, a_2), S_2(b_1, b_2)) \tag{41}$$
$$S_1(T(a_1, b_1), T(a_2, b_2)) = T(S_2(a_1, a_2), S_2(b_1, b_2)) \tag{42}$$
$$S_1(S_2(a_1, b_1), S_2(a_2, b_2)) = S_2(T(a_1, a_2), T(b_1, b_2)) \tag{43}$$
$$S_1(T_1(a_1, b_1), T_1(a_2, b_2)) = T_1(T_2(a_1, a_2), T_2(b_1, b_2)) \tag{44}$$

We show that only 2 of the above 8 equations, the ones belonging to *Group 1* have solutions, while the rest of the equations belonging to *Group 2* do not have solutions as given by the following theorems, the proofs of which can be found in the Appendix.

Theorem 6. *If T_1, T_2 and T_3 are any t-norms then the equation (37) obtained by letting $o_g = T_1, \mu = T_2$ and $o_\mu = T_3$ in (18) is valid iff $T_1 \equiv T_2 \equiv T_3$ on I^2 .*

Theorem 7. *If S_1, S_2 and S_3 are any t-conorms then the equation (38) obtained by letting $o_g = S_1, \mu = S_2$ and $o_\mu = S_3$ in (18) is valid iff $S_1 \equiv S_2 \equiv S_3$ on I^2 .*

Theorem 8. *The equations belonging to* Group 2 *do not have solutions.*

5.4 On the Equation (19)

In this section, we investigate equation (19), namely,

$$M(A_1, A)\ o_\mu\ M(A_2, A) = M(A_1\ o_\mu\ A_2, A)$$

where M is a matching function that compares two fuzzy sets on the same domain, i.e., $M : \widetilde{F}(X) \times \widetilde{F}(X) \to I$, with o_μ a t- or t-conorm, in which case we get the following equations (45) and (46):

$$T[M(A_1, A), M(A_2, A)] = M(T(A_1, A_2), A) \tag{45}$$

$$S[M(A_1, A), M(A_2, A)] = M(S(A_1, A_2), A) \tag{46}$$

5.4.1 A few Matching functions existing in the literature

Below we list a few of the matching functions commonly employed in the literature.

- Zadeh's Sup-min : $M_1(A, A') = max_x\ min(A(x), A'(x))$
- Magrez - Smets' Measure [56]:$M_2(A, A') = max_x\ min(\overline{A(x)}, A'(x))$, where $\overline{A(x)}$ is the negation of $A(x)$.
- Sup-T : $M_3(A, A') = max_x\ T(A(x), A'(x))$, where T is any t-norm.
- Sup-T-N:$M_4(A, A') = max_x\ T(\overline{A(x)}, A'(x))$.
- Inf- max :$M_5(A, A') = min_x\ max(A(x), A'(x))$.
- Inf - max- N: $M_6(A, A') = min_x\ max(\overline{A(x)}, A'(x))$.
- Inf-S : $M_7(A, A') = min_x\ S(A(x), A'(x))$, where S is any t-conorm.
- Inf - S - N: $M_8(A, A') = min_x\ S(\overline{A(x)}, A'(x))$.

Note: M_3 and M_4 (M_7 and M_8) are generalisations of M_1 and M_2 (M_5 and M_6), respectively, while M_5, M_6, M_7 and M_8 are duals of M_1, M_2, M_3 and M_4.

The proofs of the following results can be found in the Appendix.

Theorem 9. *M_1, M_2, M_3 and M_4 satisfy equation (46) iff $S = max$.*

Theorem 10. *M_5, M_6, M_7 and M_8 satisfy equation (45) iff $T = min$.*

Remark 6. *M_1, M_2, M_3 and M_4 (M_5, M_6, M_7 and M_8) do not satisfy (45) (resp. (46)) since there does not exist any t-conorm S (resp. t-norm T) such that $S(min_x a_x, min_x b_x) = min_x S(a_x, b_x)$ (such that $T(max_x a_x, max_x b_x) = max_x T(a_x, b_x)$). We refer the readers to [20, 21] for the corresponding proofs.*

Combining the results of section 5.1 - 5.4, we get the following table - Table 5 - of operators available for equations (17),(18) and (19), where J_S, J_R denote S- and R-implication, while T and S denote a t-norm and t-conorm, respectively.

f	g	o_g	μ	o_μ	Conditions	Examples of M
J_S or J_R	$\vee$	$\wedge$	$\wedge$	$\wedge$	-	M_5, M_6, M_7, M_8
J_S or J_R	$\wedge$	$\vee$	$\vee$	$\vee$	-	M_1, M_2, M_3, M_4
T	$\wedge$	$\wedge$	$\wedge$	$\wedge$	-	M_5, M_6, M_7, M_8
T	S	S	S	S	T dist over S	$M_1, M_2, M_3, M_4; S = \vee$

Table 5: Table of operators for (17),(18) and (19) to be satisfied

6 Examples of a few Fuzzy Systems from the above class

In this section we show how the results from Section 5 can be applied to particular models of inferencing in Fuzzy Systems. For throughout this section, we consider the following SISO fuzzy system with 3 rules as given in (47).

$$\begin{aligned} R_1 &: A_1 \rightarrow B \\ R_2 &: A_2 \rightarrow B \\ R_3 &: A_3 \rightarrow C \end{aligned} \tag{47}$$

where A_1, A_2, A_3 are fuzzy sets on X; B, C are fuzzy sets on Y and $\rightarrow$ is any rule firing operation relating the antecedent to the consequent.

6.1 Mamdani Model of Inference in Fuzzy Systems

Consider the following set of m Single-Input Single-Output (SISO) fuzzy if-then rules of Mamdani type:

$$If\ x\ is\ A_j\ Then\ y\ is\ B_j,\ \ j = 1, 2, \cdots, m$$

where A_i, B_i are fuzzy sets on the input and output domains X, Y, respectively. From Section 3.4.1, we know a Single-Input Single-Output (SISO) Mamdani type Fuzzy System has the final output fuzzy set B given by

$$B(y) \quad = \bigvee_{j=1}^{m}\{[A_j(x) \wedge B_j(y)\}, \forall\, y \in Y \tag{48}$$

which corresponds to $Q_M = \{M_1, na, \wedge, \vee, \widetilde{F}(Y)\}$.

Remark 7. *Since in the case of SISO rule base, the antecedent combiner μ does not play a role we have indicated it as Not Applicable - na - in Q_M.*

6.1.1 Lossless Rule Reduction in Mamdani Model of Inference in SISO Fuzzy Systems

Theorem 11. *Inference Invariant Rule Reduction is possible in Mamdani Model of Inference, in the case of SISO fuzzy rules, by combining the antecedents of rules that have identical consequent.*

Proof. We know that f - the rule firing operator - is the t-norm *min* in (48). In the presence of an input, say *x is A*, denoted as $X \ = A$, we have from (48), the final output fuzzy set B' is given by

$$\begin{aligned} B'(y) \quad = \quad & [M_1(A_1, A) \wedge B] \\ & \vee[M_1(A_2, A) \wedge B] \\ & \vee[M_1(A_3, A) \wedge C] \end{aligned} \tag{49}$$

¿From (49) by the distributivity of ∧ over ∨ we have (50),

$$\begin{aligned} B'(y) \quad = \quad & \{[M_1(A_1, A) \ \vee (M_1(A_2, A)] \wedge B\} \\ & \vee\ [M_1(A_3, A) \wedge C] && (50) \\ = \quad & [M_1(A_1 \vee A_2, A) \wedge B] \\ & \vee\ [(M_1(A_3, A) \wedge C] && (51) \\ = \quad & [M_1(A_1^*, A) \wedge B] \vee [(M_1(A_3, A) \wedge C] && (52) \end{aligned}$$

We know from Theorem 9 that $M_1(A_1, A) \vee M_1(A_2, A) = M_1(A_1 \vee A_2, A)$, using which we obtain (51) from (50). In (52) $A_1^* = A_1 \vee A_2$, which is again a fuzzy set on X, by Definition (6) and Remark 2.

Thus instead of the SISO fuzzy rule base of 3 rules (47), the following reduced rule base with two rules can be used, without any loss of inference for a given input, while employing the Mamdani Model of Inference.

$$\begin{aligned} R_1 \quad &: \quad A_1^* \to B \\ R_3 \quad &: \quad A_3 \to C \end{aligned} \tag{53}$$

□

6.2 General Mamdani Model of Inference in Fuzzy Systems

A slight generalisation of the Mamdani model of inference can be seen as follows: Let $Q_M^T = \{M, na, T^*, S^*, \widetilde{F}(Y)\}$ denote a General Mamdani Model of Inference where T^* is any t-norm that distributes over the t-conorm S^*. Then the following can be easily shown as above:

Theorem 12. *Inference Invariant Rule Reduction is possible in Mamdani Model of Inference, in the case of SISO fuzzy rules, by combining the antecedents of rules that have identical consequent, if the Matching function M is such that*

$$S^*[M(A_1, A), M(A_2, A)] = M(S^*[A_1, A_2], A). \tag{54}$$

In the case $S^* = max$, the matching function M, among others, can be one of M_1, M_2, M_3, M_4.

6.3 Modified Mamdani Model of Inference in Fuzzy Systems

By a Modified Mamdani Model of Inference we refer to the following quintuple $Q_M^J = \{M, na, J, \wedge, \widetilde{F}(Y)\}$, where M is any Matching function and J is either an R- or an S-implication. In this model of inference an Implication Operator J is employed to relate the antecedent and the consequent of the fuzzy rules. The final output fuzzy set B in Q_M^J for a SISO rule base is given by

$$B(y) \quad = \bigwedge\nolimits_{j=1}^{m}\{[A_j(x) \rightarrow B_j(y)\}, \forall\, y \in Y \tag{55}$$

where → is either an R- or an S-implication.

Recently Li et al [54,55] have shown that the above Modified Mamdani Model of Inference in Fuzzy Systems with R- or S-implications for the rule firing operation and with Trapezoidal or Triangular membership functions are Universal Approximators both in the case of SISO and MISO fuzzy systems. (In the case of MISO systems the antecedent combiner $\mu = \wedge$). These considerations make Modified Mamdani Model of Inference very attractive. In this section we show that lossless rule reduction is possible in Modified Mamdani Model of Inference in the SISO case.

6.3.1 Lossless Rule Reduction in Modified Mamdani Model of Inference in SISO Fuzzy Systems

Theorem 13. *Inference Invariant Rule Reduction is possible in Modified Mamdani Model of Inference, in the case of SISO fuzzy rules, by combining the antecedents of rules that have identical consequent, if the Matching function M obeys (56).*

$$M(A_1, A) \vee M(A_2, A) = M(A_1 \vee A_2, A). \tag{56}$$

Proof. Let us now interpret f - the rule firing operator - as an R- or an S-implication in (55). Let us again consider the above SISO fuzzy system of 3 rules as given in

(47), but where the → is any R- or S-implication. Also let the Matching function M obey (56).

In the presence of an input, say $x\ is\ A$, denoted as $X = A$, we have from (55), the final output fuzzy set B' is given by

$$B'(y) = [M(A_1, A) \to B] \wedge [M(A_2, A) \to B] \\ \wedge [M(A_3, A) \to C] \quad (57)$$

¿From (57) by using Theorem 3 we obtain (58),

$$\begin{aligned} B'(y) &= \{[M(A_1, A) \vee (M(A_2, A)] \to B\} \\ &\quad \wedge [M(A_3, A) \to C] && (58) \\ &= [M(A_1 \vee A_2, A) \to B] \\ &\quad \wedge [M(A_3, A) \to C] && (59) \\ &= [M(A_1^*, A) \to B] \wedge [M(A_3, A) \to C] && (60) \end{aligned}$$

We obtain (59) by using the fact that M obeys (56). In (60) $A_1^* = A_1 \vee A_2$, which is again a fuzzy set on X, by Definition (6) and Remark 2. Thus again instead of the SISO fuzzy rule base of 3 rules (47), we have the reduced rule base (53). □

Examples of Matching functions M that satisfy (56) are M_1, M_2, M_3, M_4.

In general, the number of rules can be reduced to k , where k is the number of output fuzzy sets that featured in the original rule base. Most importantly, this type of rule reduction is lossless w.r.to inference and Table 6 summarises the above discussion for the SISO case with the following:

Condition (i) T^* distributes over S^*

Condition (ii) M satisfies (54).

Condition (iii) M satisfies (56).

Name / Type	Q	M	f	g	$o_g = o_\mu$	Conditions
Original Mamdani	Q_M	M_1	$\wedge$	$\vee$	$\vee$	-
General Mamdani	Q_M^T	M	T^*	S^*	S^*	(i) and (ii)
Modified Mamdani	Q_M^J	M	J	$\wedge$	$\vee$	(iii)

Table 6: M, f, g, o_g and o_μ for the different models of inference discussed in Sections 6.1 - 6.3

7 Conclusion

In this work we have proposed a simple rule reduction technique that combines rules with identical consequents, which is lossless with respect to inference. Towards this end, we proposed a general framework for Inference in Fuzzy Systems and imposed certain requirements on the different inference operators employed in a Fuzzy System. Also we have explored these requirements in the setting of Fuzzy Logic Operators. We have also given a few examples of Models of Inference in Fuzzy Systems that have the required properties for inference invariant rule reduction. We note a few observations below:

- Merging of rules, in some cases, may turn out to be computationally intensive, but this is a one time off-line job which even though might reduce interpretability will make computation of inferences much faster. Perhaps the original rule base can still be preserved for interpretability considerations.

- In some instances, the above method may even increase the number and the complexity of the fuzzy sets defined on different input domains.

In this work, we have considered only S- and R-implications for the fuzzy implication J and have shown the important role played by their distributivity over t-norms and t-conorms in the inference scheme in Section 6.3. Recently, there are a few more families that have been proposed, viz., U-implications and the residual implications of uninorms J_{U*} in [29] and the recently proposed families of f-generated implications J_f and g-generated implications J_g by Yager in [85] and h-generated implications J_h in [8], [9]. The distributivity of J_{U*} and U-implications over uninorms - which are generalisations of t-norms and t-conorms (see [84]) - is studied in [27] and [28] while that of J_f over t-norms and t-conorms is done in [9]. Hence these families of fuzzy implications can also be employed for the inference scheme in Section 6.3.

In this work we have considered in detail the proposed rule reduction technique only in the SISO case explicitly. Recently we have done some work on rule reduction in the MISO case also, as has been demonstrated within the scope of Similarity Based Reasoning in [10].

Acknowledgments

The author wishes to express his gratitude to Professor C. Jagan Mohan Rao for his most valuable suggestions and remarks. The author wishes to acknowledge the excellent environment provided by Dept. of Mathematics and Computer Sciences, Sri Sathya Sai Institute of Higher Learning, during the period of this work, for which the author is extremely grateful.

References

[1] J. Aczel, Lectures on Functional Equations and their Applications, Academic Press (1966).

[2] J. Aczel and Gy. Maksa, Solution of the Rectangular m x n Generalized Bisymmetry Equation and of the Problem of Consistent Aggregation , J. Math. Anal. Appl. 203 (1996) 104 - 126.

[3] J. Aczel, Bisymmetry and Consistent Aggregation: Historical Review and Recent Results , in Choice, Decision, and Measurement, Eds. Morley, A. A. J. and Lawrence Erlbaum, Assoc. Publ., Mahwah, NJ, (1997) 225 - 233.

[4] J. Aczel, Gy. Maksa, and M. A. Taylor, Equations of Generalized Bisymmetry and of Consistent Aggregation: Weakly Surjective Solutions Which May Be Discontinuous at Places , J. Math. Anal. Appl. 214 (1997) 22 - 35.

[5] Michal Baczynski, On a class of Distributive Fuzzy Implications , Intl. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9 (2) (2001) 229 - 238.

[6] Michal Baczynski, Contrapositive Symmetry of Distributive Fuzzy Implications , Intl. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (Suppl) (2002) 135 -147.

[7] Balasubramaniam.J., and C.Jagan Mohan Rao, On the Distributivity of Implication Operators over T- and t-conorms , IEEE Transaction on Fuzzy Systems, 12 (2) (2004) 194 - 198.

[8] Balasubramaniam.J., ”Contrapositive symmetrization of fuzzy implications - Revisited”, Fuzzy Sets and Systems 157 (2006) 2291 – 2310.

[9] Balasubramaniam.J., ”Yagers new class of implications J_f and some classical tautologies”', Info. Sci. *In Press*.

[10] Balasubramaniam.J., Rule Reduction for Efficient Inferencing in Similarity Based Reasoning, Int. Jl. Approx. Reas., *Submitted*.

[11] P. Baranyi and Y. Yam, Singular value-based approximation with Takagi-Sugeno type fuzzy rule base, Proc. of the 6th IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'97), volume I, Barcelona, Spain, 1997, pp. 265 - 270.

[12] P. Baranyi and Y. Yam, Singular value-based approximation with non-singleton fuzzy rule base, Proc. of the 7th Int. Fuzzy Systems Association World Congress (IFSA'97), volume II, Prague, Czech Republic, 1997, pp. 127 - 132.

[13] P.Baranyi and Y.Yam, Fuzzy Rule Base Reduction in Fuzzy IF-THEN Rules in Computational Intelligence : Theory and Applications, Eds. D.Ruan and E.E.Kerre, Kluwer, (2000) 135 - 160.

[14] P.Baranyi et al, Trade-off between Approximation Accuracy and Complexity : HOSVD based Complexity Reduction, Periodica Polytechnica Ser. Transp. Eng. 29 (1-2) (2001) 3 - 26.

[15] P.Baranyi, P.Korondi and H.Hashimoto, Trade-off Between Approximation accuracy and complexity for TS fuzzy models , Asian Journal of Control 6 (21-1) (2004) 21 - 33.

[16] P.Baranyi, Y.Yam, D.Tikk and R.J.Patton, Trade-off between approximation accuracy and complexity: TS controller design via HOSVD based complexity minimization, Studies in Fuzziness and Soft Computing, Vol. 128. Interpretability Issues in Fuzzy Modeling, J. Casillas, O. Cordn, F.Herrera, L.Magdalena (Eds.), Springer-Verlag, 2003. pp. 249-277.

[17] P.Baranyi, Y.Yam, A.R.Várkonyi-Kóczy and R.J.Patton, SVD Based Reduction to MISO TS Fuzzy Models , IEEE Transaction on Industrial Electronics, Vol. 50. No. 1. February 2003, pp. 232-242.

[18] P.Baranyi, Y.Yam, A. R.Várkonyi-Kćzy, R.J.Patton, P.Michelberger and M.Sugiyama, SVD Based Reduction to TS Fuzzy Models , IEEE Transaction on Industrial Electronics, 49 (2) (2002) 433-443.

[19] Boverie,N., Narishkin,D., Lequelle,J., Titli,A., Fuzzy Control of Higher Order Systems using a Parallel Structure of second order blocks , in Preprints from IFAC World Congress, (1993) 573 - 576.

[20] Carlo Bertoluzza, On the Distributivity between t-Norms and t-Conorms, FUZZ-IEEE ’93, 140-147.

[21] Carlo Bertoluzza, V. Doldi, ”On the Distributivity between t-norms and t-conorms”, Fuzzy Sets and Systems 142 (1) (2004) 85-104.

[22] C.Chantana, S.Tongsima and E.H.M.Sha, Minimization of Fuzzy Systems based on Fuzzy Inference Graphs , Research Report TR-96-6, University of Notre Dame.

[23] W.E.Combs and J.E.Andrews, Combinatorial rule explosion eliminated by a fuzzy rule configuration , IEEE Trans. Fuzzy Systems 6 (1998) 1-11.

[24] W.E.Combs, Author’s reply , IEEE Trans. Fuzzy Systems 7 (1999) 371.

[25] W.E.Combs, Author’s reply , IEEE Trans. Fuzzy Systems 7 (1999) 478 - 479.

[26] V.Cross, T.Sudkamp, ”Fuzzy Implication and Compatibility Modification”, FUZZ-IEEE ’93, San Francisco, California March 28 - April 1, 1993 pp. 219 - 224.

[27] Daniel Ruiz, Joan Torrens, Distributive Strong Implications from Uninorms. Proc. AGOP-2005 (2005), 103-108.

[28] Daniel Ruiz, Joan Torrens, Distributive Residual Implications from Uninorms. Proc. EUSFLAT-2005 (2005).

[29] De Baets B., Fodor J., Residual operators of uninorms, Soft Computing 3 (1999), 89–100.

[30] S.Dick and A.Kandel, Comments on ”Combinatorial rule explosion eliminated by a fuzzy rule configuration” , IEEE Trans. Fuzzy Systems 7 (1999) 475 - 477.

[31] Dubois.D. and H.Prade, ” Fuzzy sets in approximate reasoningPart I: Inference with possibility distributions”, Fuzzy Sets and Systems 40 (1991) 143202.

[32] Dubois.D. and H.Prade, ”What are fuzzy rules and how to use them”, Fuzzy Sets and Systems 84 (1996) 169185 (special issue in memory of Prof A.Kaufmann).

[33] Dubois D., Prade H. and Ughetto L. (1997). Checking the coherence and the reduncancy of fuzzy knowledge bases, IEEE Trans. on Fuzzy Systems, 5, 398-417.

[34] Erich Peter Klement, Radko Mesiar and E.Endre Pap, ” Triangular norms. Position paper I: Basic Analytical and Algebraic Properties”, Fuzzy Sets and Systems 143 (2004) 5 - 26.

[35] Erich Peter Klement, Radko Mesiar and E.Endre Pap, ”Triangular norms - Position paper II: General constructions and parameterized families”, Fuzzy Sets and Systems 145 (2004) 411 - 438.

[36] Erich Peter Klement, Radko Mesiar and E.Endre Pap, ” Triangular norms - Position paper III: continuous t-norms”, Fuzzy Sets and Systems 145 (2004) 439 - 454.

[37] Fodor.J.C., and M. Roubens (1994), Fuzzy Preference Modelling and Multicriteria Decision Support, Kluwer, Dordrecht.

[38] Janos C Fodor, On Bisymmetry operations on the Unit Interval , Proc. IFSA ’97, 7th Int. Fuzzy Systems Association World Congress (1997) 277 - 280.

[39] J. Fodor and J.-L. Marichal, On nonstrict means, Aequationes Mathematicae 54 (3) (1997) 308-327.

[40] S. Galichet L. Foulloy, Fuzzy Controllers: synthesis and equivalences, IEEE Trans. Fuzzy Systems 3 (1995) 140-148.

[41] Gegov,A. and Frank,P., Reduction of Multidimensional Relations in Fuzzy Control Systems , Systems and Control Letters, 25 307 - 313.

[42] George.J.Klir and Bo Yuan, Fuzzy Sets and Fuzzy Logic: Theory and Applications, Prentice Hall Inc., Englewoods, 1995.

[43] J.S.Glower and J.Munighan, Designing Fuzzy Controllers from a Variable Structures Standpoint , IEEE Trans. Fuzzy Systems 5 (1) (1997) 138 - 144.

[44] Guven.K.Mustafa and Kevin M. Passino, Avoiding Exponential Parameter Growth in Fuzzy Systems , IEEE Trans. Fuzzy Systems 9 (1) (2001) 194 - 199.

[45] C.C.Hung and B.R.Fernandez, Minimizing Rules of Fuzzy Logic System by using a Systematic Approach , FUZZ-IEEE '93, 38 - 48.

[46] H.Ishibuchi, K. Nozaki, N.Yamamoto and H.Tanaka, Selecting fuzzy if-then rules for classification problems using genetic algorithms , IEEE Trans. on Fuzzy Systems 3 (1995) 260 - 270.

[47] Jia,L., and Zhang.X., Identification of Multivariable Fuzzy Systems through Fuzzy Cell Mapping , Preprints from IFAC World Congress (1993) 389 - 393.

[48] C.L.Karr, Applying genetics to fuzzy logic , AI Expert 6 (1991) 38 - 43.

[49] L. T. Koczy, Fuzzy if-then rule models and their transformation into one another , IEEE Trans. on Syst. Man and Cyber. Part A 26 (5) (1996) 621 - 637.

[50] L.T.Koczy and A.Zorat, Optimal Fuzzy Rule Bases - the Cat and Mouse Problem, FUZZ - IEEE '96, 1865 - 1870.

[51] L.T.Koczy and K.Hirota, Size Reduction by Interpolation in Fuzzy Rule Bases, IEEE Trans. on Syst. Man and Cyber. Part B 27 (1) (1997) 14 - 25.

[52] A.Krone, P.Krause and T.Slawinski, A New Rule Reduction Method for finding Interpretable and Small Rule Bases in High Dimensional Search Spaces, FUZZ-IEEE 2000, 694 - 699.

[53] M.A.Lee and H.Takagi, Integrating design stages of fuzzy systems using genetic algorithms, FUZZ-IEEE '93, 612 - 617.

[54] Y.M.Li, Z.K.Shi, Z.H.Li, Approximation Theory of Fuzzy Systems based upon genuine many-valued implication - SISO Case, Fuzzy Sets and Systems, 130 (2) (2002) 147 - 158.

[55] Y.M.Li, Z.K.Shi, Z.H.Li, Approximation Theory of Fuzzy Systems based upon genuine many-valued implication - MIMO Case, Fuzzy Sets and Systems, 130 (2) (2002) 159 - 174.

[56] P.Magrez, Ph.Smets, Fuzzy Modus Ponens: a new model suitable for applications in Knowledge-bases systems, Int. Jl. of Intelligent Systems 4 (1989) 181 - 200.

[57] E.H. Mamdani and S.Assilian, "An experiment in Linguistic Synthesis with a Fuzzy Logic Controller", Int. Jl. of Man - Machine Studies 7 (1975) 1 - 13.

[58] Mamdani. E. H., Application of Fuzzy Logic in approximate Reasoning using Linguistic Systems, IEEE Transactions on Computers 26 (1997) 1182 - 1191.

[59] J.M.Mendel and Q.Liang, Comments on ”Combinatorial rule explosion eliminated by a fuzzy rule configuration”, IEEE Trans. Fuzzy Systems 7 (1999) 369 - 371.

[60] N.N.Morsi, A.A.Fahmy, On generalised modus ponens with multiple rules and a residuated implication , Fuzzy Sets and Systems 129 (2) (2002) 267 - 274.

[61] G.C.Mouzouris and J.M.Mendel, Designing fuzzy logic systems for uncertain environments using a singular-value-QR decomposition method , in FUZZ-IEEE ’96, 295 - 301.

[62] Raju,G. and Zhou,J, Fuzzy Logic processes controller, IEEE Intl. Conf. On Systems Engg., (1990) 145 - 147.

[63] H.Roubos and M.Setnes, Compact and Transparent Fuzzy Models and Classifiers through Iterative Complexity Reduction, IEEE Trans. Fuzzy Systems 9 (4) (2001) 516 - 524.

[64] R.Rovatti, R.Guerrieri and G.Baccarani, An Enhanced Two-Level Boolean Synthesis Methodology for Fuzzy Rules Minimization, IEEE Trans. Fuzzy Systems 3 (1995) 288 - 299.

[65] E.H.Ruspini, A new approach to clustering, Information and Control 15 (1) (1969) 22 - 32.

[66] M.Setnes, A.Koene, R.Babuska and P.Bruijin, Data-driven Initialisation and structure learning in fuzzy neural networks, FUZZ-IEEE ’98, 1147 - 1152.

[67] Magne Setnes, R.Babuska, U.Kaymak and H.R. van Nauta Lemke, Similarity measures in fuzzy rule base simplification, IEEE Trans. on Syst. Man and Cyber. Part B 28 (3) (1998) 376 - 386.

[68] M.Setnes, V.Lacrose and A.Titli, Complexity Reduction Methods for Fuzzy Systems in Fuzzy Algorithms for Control, Eds. H.Verbruggen, H.J.Zimmermann and R.Babuska, (1999) 185 - 218.

[69] Magne Setnes and R.Babuska, Rule Base Reduction: Some Comments on the Use of Orthogonal Transforms, IEEE Trans. on Syst. Man and Cyber. Part C, 31 (2) (2001) 199 - 206.

[70] C.T.Sun, Rule-Base structure identification in an Adaptive-Network-Based Fuzzy Inference System, IEEE Trans. Fuzzy Systems 2 (1) (1994) 64 - 73.

[71] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Systems, Man and Cybernetics 15 (1) (1985) 116 - 132.

[72] K.Tanaka, T.Taniguchi and H.O.Wang, Generalised Takagi-Sugeno Fuzzy Systems : Rule Reduction and Robust Control, FUZZ-IEEE 2000, 688 - 693.

[73] M. A. Taylor, The Generalized Equation of Bisymmetry: Solutions Based on Cancellative Abelian Monoids, Aequationes Math. 57 (1999) 288 - 302.

[74] M. A. Taylor, The Aggregation Equation : Solutions with Non-intersecting Partial Functions in Functional Equations - Results and Advances, Eds. Z. Daroczy and Z. Pales, Kluwer Academic Publishers (2000) 1 - 10.

[75] E.Trillas and L.Valverde, On Implication and Indistinguishability in the setting of Fuzzy Logic, in Management Decision Support Systems Using Fuzzy Sets and Possibility Theory, Verlag TUV-Rhineland, 1985, pp. 198 - 212.

[76] E.Trillas, C.Alsina, On the Law $(p \wedge q) \rightarrow r \equiv (p \rightarrow r) \vee (q \rightarrow r)$ in Fuzzy Logic , IEEE Trans. Fuzzy Systems 10 (1) (2002) 84 - 88.

[77] I.B.Turksen and Z.Zhong, ”An Approximate Reasoning schema based on Similarity measures and Interval valued fuzzy sets”, IEEE Trans. on Systems, Man and Cybernetics, 18 (6) (1988) 1049 - 1056.

[78] Jose Villar, Conditions for Equivalence between the Compositional Rule of Inference and the Compatibility Modification Inference , FUZZ-IEEE ’96 (1996) 444 - 449.

[79] L.X.Wang and J.M.Mendel, Fuzzy basis functions, universal approximation and orthogonal least squares learning , IEEE Trans. on Neural Networks, 3 (1993) 807 - 814.

[80] Yeung Yam, Fuzzy Approximation Via Grid Point Sampling and Singular Value Decomposition , IEEE Trans. on Syst. Man and Cyber. Part B 27 (6) (1997) 933 - 951.

[81] Yeung Yam, Baranyi, P. and Yang, C.T., Reduction of Fuzzy Rule Base via Singular Value Decomposition ,IEEE Transaction on Fuzzy Systems, 7 (2) (1999) 120-132.

[82] John Yen and Liang Wang, Simplifying fuzzy rule-based models using orthogonal transformation methods , IEEE Trans. on Syst. Man and Cyber. Part B 29 (1) (1999) 13 - 24.

[83] R.R.Yager, Alternative Structures for Knowledge Representation in Fuzzy Logic Controllers, in ”Fuzzy Control Systems”, Eds. A.Kandel and G.Langholz, CRC Press (1994) 99 - 137.

[84] Yager R.R., Rybalov A., Uninorm aggregation operators, Fuzzy Sets and Systems, 80 (1996) 111–120.

[85] Yager. R., ”On some new classes of implication operators and their role in Approximate Reasoning”, Info Sci. 167 (2004) 193 – 216.

[86] L.A.Zadeh, Outline of a new approach to the analysis of complex systems and decision processes , IEEE Trans. Systems, Man and Cybernetics 3 (1973) 28 - 44.

Proof. Proof of Theorem 6
Claim $T_1 \equiv T_2$*:* Let $a_1 = b_2 = 1$. Then $\forall a_2, b_1 \in I$, we have

$$\begin{aligned} LHS &= T_1(T_2(1, b_1), T_2(a_2, 1)) = T_1(b_1, a_2) \\ RHS &= T_2(T_3(1, a_2), T_3(b_1, 1)) = T_2(a_2, b_1) = T_2(b_1, a_2) \end{aligned}$$

which implies $T_1(b_1, a_2) = T_2(b_1, a_2)$ $\forall a_2, b_1 \in I$ and thus $T_1 \equiv T_2 = T$. Now (37) becomes

$$T(T(a_1, b_1), T(a_2, b_2)) = T(T_3(a_1, a_2), T_3(b_1, b_2)) \tag{61}$$

Now, since T is a t-norm,

$$T(T(a_1, b_1), T(a_2, b_2)) = T(T(a_1, a_2), T(b_1, b_2)) \tag{62}$$

and we have from (61) and (62) that $T_1 \equiv T_2 \equiv T_3$ on I^2. □

Proof. Proof of Theorem 8
Let us consider (40) from *Group 2*. Let $a_1 = b_2 = 1$ and $a_2, b_1 \in (0, 1)$. Then we have that

$$\begin{aligned} LHS &= T_1(T_2(1, b_1), T_2(a_2, 1)) = T_1(b_1, a_2) \\ RHS &= T_2(S(1, a_2), S(b_1, 1)) = T_2(1, 1) = 1 \end{aligned}$$

which implies that $T_1(b_1, a_2) = 1$, with $a_2, b_1 \in (0, 1)$, which is absurd. Similarly, all the other equations, (39), (41) - (44) in *Group 2* can be shown to have no solutions. □

Proof. Proof of Theorem 10
We give the proof for $M = M_7$. The proofs for $M = M_5, M_6$ and M_8 are similar.

$$\begin{aligned} LHS\ of\ (45) &= T[\inf_x S(A_1(x), A'(x)), \\ &\quad \inf_x S(A_2(x), A'(x))] \qquad (63) \\ &= \inf_x\ T[S(A_1(x), A'(x)), \\ &\quad S(A_2(x), A'(x))] \qquad (64) \\ &= \inf_x\ S(T[A_1(x), A_2(x)], A'(x)) \qquad (65) \\ &= M_7[T(A_1, A_2), A'] \\ &= RHS\ of\ (45) \end{aligned}$$

Since $T(\inf_x\ a_x, \inf_x\ b_x) \equiv \inf_x\ T(a_x, b_x)$ iff $T = \min$, we have that (64) is equivalent to (63) iff $T = \min$. Also since any t-conorm S is distributive over $T = \min$ [20], we obtain (65) from (64). □