High Speed Implementation of Piecewise-Quadratic Takagi-Sugeno Systems with Memory Saving

RICCARDO ROVATTI
DEIS
University of Bologna
Viale Risorgimento 2, 40136 Bologna
ITALY
rrovatti@deis.unibo.it http://www.chaos.cc

Abstract: – A time-efficient algorithm for the computation of Takagi-Sugeno inferences is presented which is suitable for implementation on advanced DSPs. With such a procedure, an inference depending on 6 inputs can be computed in less than 1.5μ. This speed is counterbalanced by possible huge storage requirements and the problem of reducing the number of data to be stored is also briefly addressed.

Key-Words: – Digital Signal Processing, Fuzzy Systems, Takagi-Sugeno inference, Piecewise-quadratic interpolation

1 Introduction

Recent developments in the field of fuzzy systems implementation (e.g. [1][2][3]) deal with the mixing of classical hardware support for conventional linear processing with high-performance solution for inference computation.

This hybrid approach aims at providing a unique hardware-software module which is able to execute classical linear-processing primitives as well as specialized routines performing fuzzy inference to fit the need of a an embedded controller/classifier/processor attending many tasks with different characteristics in different time slices.

To at address the need for fast processing depending on an large number of variables in the framework of Takagi-Sugeno inference mechanisms (whose implementation peculiarities seem to be vastly neglected by fuzzy hardware literature) we here extend the concept of piecewise-linear inference [4] to piecewise-quadratic inference.

With this, we obtain a regular inference scheme which can be given an extremely efficient implementation. Such an implementation needs a computational effort which does not increase exponentially with the number of inputs but requires large amounts of memory.

To partially cope with storage requirements we also devise a mechanism to trade memory for speed. To do so we exploit the relationship between samples of the same fuzzy systems at neighboring points. Such a relationship allows to compute the samples that we may have chosen not to store, slowling the inference procedure.

2 Piecewise-Quadratic Inference

The fuzzy system under consideration maps the real vector $\underline{x} = (x^1, \ldots, x^n) \in [0,1]^n$ into a single real variable. For each x^i, define m_i fuzzy sets $A^i_1, \ldots, A^i_{m_i}$ depending on an increasing sequence of points $0 = a^i_1 \leq a^i_2 \leq \cdots \leq a^i_{m_i} = 1$ so that A^i_j has a triangular shape centered in a^i_j and vanishing before a^i_{j-1} and after a^i_{j+1}.

With these classical triangular membership functions we may construct the preconditions of the rules defining the systems. They are all the possible n-conjunctions $x^1 \in A^1_{j_1} \wedge \cdots \wedge x^n \in A^n_{j_n}$ modeled by the piecewise-linear operator Θ proposed and discussed in [4]

$$\Theta[t^1, \ldots, t^n] = \frac{1}{2^n} \sum_{I_1, I_2} \max\{\min_{i \in I_1}\{t^i\} + \min_{i \in I_2}\{t^i\} - 1, 0\} \tag{1}$$

where the sum is taken over all the possible partitions of the set $\{1, \ldots, n\}$ in two subsets I_1 and I_2.

We here concentrate on Takagi-Sugeno inference in which consequences are expressed as first-order polynomials of the inputs $C_{j_1,\ldots,j_n}(\underline{x})$ to obtain

$$f(\underline{x}) = \frac{\sum_{j_1,\ldots,j_n} C_{j_1,\ldots,j_n}(\underline{x}) \Theta[A^1_{j_1}(x^1), \ldots, A^n_{j_n}(x^n)]}{\sum_{j_1,\ldots,j_n} \Theta[A^1_{j_1}(x^1), \ldots, A^n_{j_n}(x^n)]} \tag{2}$$

The inference procedure develops in strict analogy with what can be done in the piecewise-affine

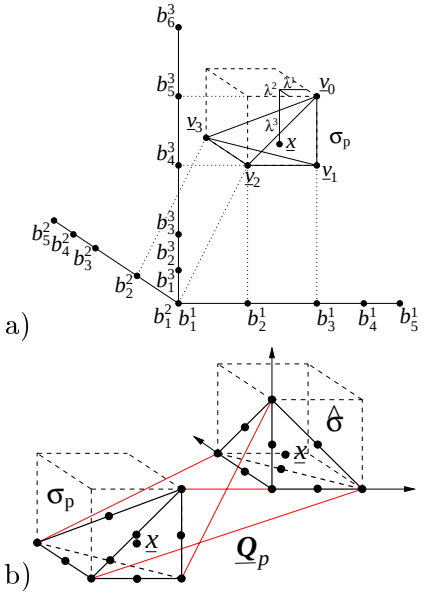

Figure 1: a) The Takagi-Sugeno fuzzy system is quadratic in the simplex σ_p. b) Simplex normalization easing quadratic interpolation

case [5]. First a finer sequences of points b^i_j is constructed which is made of the a^i_j along with the middle points of the intervals $[a^i_j, a^i_{j+1}]$, i.e. $b^i_j = a^i_{(j+1)/2}$ when j is odd and $b^i_j = \frac{1}{2}(a^i_{j/2} + a^i_{j/2+1})$ when j is even ($j = 1, \ldots, 2m_i - 1$).

Then we assume that $\bar{\jmath}_i$ is such that $b^i_{\bar{\jmath}_i} \leq x^i < b^i_{\bar{\jmath}_i+1}$ and set

$$\lambda^i = \begin{cases} 2A^i_{(\bar{\jmath}_i+3)/2}(x^i) & \text{if } \bar{\jmath}_i \text{ is odd} \\ 2A^i_{\bar{\jmath}_i/2}(x^i) & \text{if } \bar{\jmath}_i \text{ is even} \end{cases} \tag{3}$$

Then, let $p : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$ be a permutation such that $\lambda^{p(1)} \geq \lambda^{p(2)} \geq \cdots \geq \lambda^{p(n)}$. The $n+1$ points $\underline{v}_0, \ldots, \underline{v}_n$ are such that $v^i_0 = b^i_{\bar{\jmath}_i}$ if $\bar{\jmath}_i$ is odd while $v^i_0 = b^i_{\bar{\jmath}_i+1}$ if it is even, and n more points $\underline{v}_1, \ldots, \underline{v}_n$ are obtained as

$$v^i_k = \begin{cases} b^i_{\bar{\jmath}_i} + b^i_{\bar{\jmath}_i+1} - v^i_{k-1} & \text{if } i = p(k) \\ v^i_{k-1} & \text{otherwise} \end{cases} \tag{4}$$

It can be proved [4] that $\underline{x}$ belongs to the simplex σ_p which is the convex hull of $\underline{v}_0, \ldots, \underline{v}_n$, and that the λ^i are the normalized coordinates of the input point in σ_p. From the same analysis we also get that if $C_{j_1,\ldots,j_n}(\underline{x})$ are first-order polynomials, $f(\underline{x})$ is quadratic in each σ_p. Figure 1-a) shows where the simplex σ_p is located when $n = 3$ and the input $\underline{x}$ is such that $\bar{\jmath}_1 = 2$, $\bar{\jmath}_2 = 1$ and $\bar{\jmath}_3 = 4$.

Note now that any quadratic form is the linear combination of the n square terms $(x^i)^2$, of the $n(n-1)/2$ mixed terms $x^{i'}x^{i''}$, of n further linear terms x^i and of a final constant for a total of $N = n(n+3)/2 + 1$ coefficients. Hence, in each σ_p, the system is the second-order interpolation of its own samples in N points $\underline{u}_i$. Let now $\underline{w}$ be a vector function such that $\underline{w}(x^1, \ldots, x^n) = [1, x^1, \ldots, x^n, (x^1)^2, \ldots, (x^n)^2, x^1x^2, \ldots, x^{n-1}x^n]$, define the matrix $\underline{\mathbf{W}}_{\bar{\jmath}_1,\ldots,\bar{\jmath}_n,p} = [\underline{w}(\underline{u}_1)^t, \ldots, \underline{w}(\underline{u}_N)^t]^t$ and the vector $\underline{f}_{\bar{\jmath}_1,\ldots,\bar{\jmath}_n,p} = [f(\underline{u}_1), \ldots, f(\underline{u}_N)]^t$. With this one has

$$f(\underline{x}) = \underline{w}(\underline{x})^t \underline{\mathbf{W}}^{-1}_{\bar{\jmath}_1,\ldots,\bar{\jmath}_n,p} \underline{f}_{\bar{\jmath}_1,\ldots,\bar{\jmath}_n,p} \tag{5}$$

where the dependence of the matrix $\underline{\mathbf{W}}$ and the vector $\underline{f}$ on the cell indexed by $\bar{\jmath}_1, \ldots, \bar{\jmath}_n$ and on the σ_p within that cell is highlighted.

A convenient choice for the sampling set is the one including the vertexes of σ_p and the midpoints of its sides. A first property of this choice is that $\underline{\mathbf{W}}$ is surely non-singular. Actually, from the positions of the possible σ_p it can be easily seen that sampling f at the simplex vertexes and at the side midpoints means sampling it on a hyper-rectangular grids which is the Cartesian product of the sequences made of b^i_j and the midpoints between b^i_j and b^i_{j+1} for a total of $\prod_{i=1}^n (4m_i - 3)$ samples.

So far, (5) entails the construction and inversion of a matrix which strongly depends on the simplex containing the input point. This procedure can be reformulated in few simpler steps.

To do so assume first that the cell containing $\underline{x}$ is shifted to the origin and normalized to the unit cube by proper scaling so to change $\underline{\mathbf{W}}_{\bar{\jmath}_1,\ldots,\bar{\jmath}_n,p}$ into $\underline{\mathbf{W}}_p$. Then let $\underline{\mathbf{Q}}_p$ be the linear transformation that maps $\underline{v}_0$ in the origin and the straight lines $\underline{v}_k - \underline{v}_0$ for $k = 1, \ldots, n$, to the Cartesian axes so that σ_p is mapped to a *standard* simplex $\hat{\sigma}$ with n orthogonal unit sides. Figure 1-b) shows how this transformation looks like for the case in Figure 1-a).

Exploiting $\underline{\mathbf{Q}}_p$ one may define $\underline{\hat{\mathbf{W}}} = [\underline{w}(\underline{\mathbf{Q}}_p\underline{u}_1)^t, \ldots, \underline{w}(\underline{\mathbf{Q}}_p\underline{u}_N)^t]^t$ to obtain

$$f(\underline{x}) = \underline{w}(\underline{\mathbf{Q}}_p\underline{x})^t \underline{\hat{\mathbf{W}}}^{-1} \underline{f}_{\bar{\jmath}_1,\ldots,\bar{\jmath}_n,p} \tag{6}$$

This new formulation of the interpolation procedure is favorable as the term $\underline{f}_{\bar{\jmath}_1,\ldots,\bar{\jmath}_n,p}$ which accounts for the needed samples is left unchanged while $\underline{\hat{\mathbf{W}}}^{-1}$ is now fixed and can be pre-computed.

Moreover, $\underline{\mathbf{Q}}_p$ turns out to be a trivial transformation. To see why note that the coordinate vector $[\lambda^{p(1)}, \lambda^{p(2)}, \ldots, \lambda^{p(n)}]$ is a linear transformation of $\underline{x}$ (3) and identifies any point $\underline{x} \in \sigma_p$ [4] (see Figure 1-a)). Moreover, in this coordinate system $\underline{v}_0$ corresponds to $[0, 0, \ldots, 0]$, $\underline{v}_1$ corresponds to $[1, 0, \ldots, 0]$, $\underline{v}_2$ corresponds to $[1, 1, \ldots, 0]$ up to

$\underline{v}_n$ which corresponds to $[1, 1, \ldots, 1]$. From the definition of σ_p we also get that $\underline{\mathbf{Q}}_p\underline{v}_0 = [0, 0, \ldots, 0]$, $\underline{\mathbf{Q}}_p\underline{v}_1 = [1, 0, \ldots, 0]$, $\underline{\mathbf{Q}}_p\underline{v}_2 = [0, 1, \ldots, 0]$ up to $\underline{\mathbf{Q}}_p\underline{v}_n = [0, 0, \ldots, 1]$. Hence $\underline{\mathbf{Q}}_p$ is uniquely identified in terms of the $\lambda^{p(i)}$ coordinates vector. In fact, in that coordinate system we have that the i-th component of $\underline{\mathbf{Q}}_p\underline{x}$ is $\lambda^{p(i)} - \lambda^{p(i+1)}$ for $i < n$ and $\lambda^{p(n)}$ for $i = n$. This structure of $\underline{\mathbf{Q}}_p$ is extremely favorable as the quantities λ^i have already to be computed and sorted to find σ_p and thus $\underline{f}_{\bar{\jmath}_1,\ldots,\bar{\jmath}_n,p}$ and are already available to be fed into the $\mathcal{O}(n)$ procedure giving $\underline{\mathbf{Q}}_p\underline{x}$.

As a last step, the structure of the matrix $\underline{\hat{\mathbf{W}}}^{-1}$ must be investigated. Such a structure is better understood if the the points $\underline{\hat{u}}_i = \underline{\mathbf{Q}}_p\underline{u}_i$ are sorted so that $\underline{\hat{u}}_0$ is the origin, $\underline{\hat{u}}_1, \ldots, \underline{\hat{u}}_n$ are the middle points of the orthogonal sides, $\underline{\hat{u}}_{n+1}, \ldots, \underline{\hat{u}}_{2n}$ are the other vertexes of those sides and $\underline{\hat{u}}_i$ for $i > 2n$ are all the other sampling points.

With this, trivial computations reveal that $\underline{\mathbf{W}}^{-1}$ can be written as

$$\left[\begin{array}{c|c|c|c} 1 & 0 & 0 & 0 \\ \hline -3 & & & \\ \vdots & 4\underline{\mathbf{I}}_n & -4\underline{\mathbf{I}}_n & 0 \\ -3 & & & \\ \hline -2 & & & \\ \vdots & -4\underline{\mathbf{I}}_n & 2\underline{\mathbf{I}}_n & 0 \\ -2 & & & \\ \hline 4 & & & \\ \vdots & -4\underline{\mathbf{C}}_n & 0 & 4\underline{\mathbf{I}}_{n(n-1)/2} \\ 4 & & & \end{array}\right]$$

where $\underline{\mathbf{I}}_n$ is the n-dimensional identity matrix while $\underline{\mathbf{C}}_n$ is a $n(n-1)/2 \times n$ matrix whose rows feature only two non vanishing entries which are equal to 1 and appear in all the possible positions. Obviously, the large number of null entries in $\underline{\mathbf{W}}^{-1}$ and its intrinsic regularity benefit the actual computation whose quadratic complexity depends only on $\underline{\mathbf{C}}_n$ and $\underline{\mathbf{I}}_{n(n-1)/2}$.

3 Memory Saving

The non-exponential time-complexity of the proposed inference depends on the possibility of retrieving all the $f(\underline{u}_i)$ for $i = 0, 1, \ldots, N-1$ in constant time, i.e. on the availability of a possibly huge memory support.

Yet, for every $\underline{x} \in [a^1_{j_1}, a^1_{j_1+1}] \times \cdots \times [a^n_{j_n}, a^n_{j_n+1}]$, the value $f(\underline{x})$ depends only on the coefficients of the 2^n first-order polynomials $C_{j_1,\ldots,j_n}, C_{j_1+1,\ldots,j_n}, \ldots, C_{j_1+1,\ldots,j_n+1}$ so that the 5^n

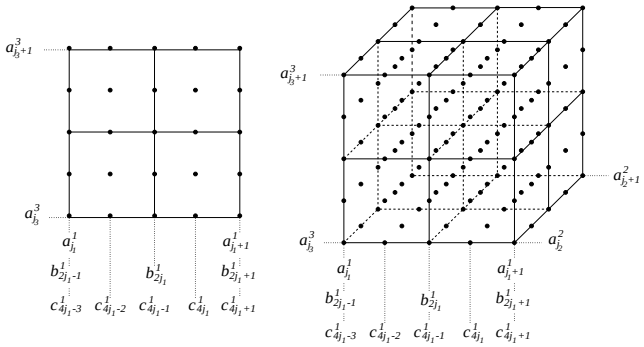

Figure 2: Though quadratic in each of the $2^3 3!$ in which this cell can be partitioned, f is quadratic also along each of the segments drawn in the figure

samples needed by the proposed procedure depend on only $(n+1)2^n$ independent parameters.

A very useful consequence of this dependence is that some samples can be surely computed from the knowledge of other samples. To formalize these useful relationships let us define the sample grid as the cartesian product of the sequences c^i_j defined so that $c^i_j = b^i_{(j+1)/2}$ when j is odd and $c^i_j = \frac{1}{2}(b^i_{j/2} + b^i_{j/2+1})$ when j is even $(j = 1, \ldots, 4m_i - 3)$.

Then, let us concentrate on the hyperrectangular cell $\underline{x} \in [c^1_{4j_1-3}, c^1_{4j_1+1}] \times \cdots \times [c^n_{4j_n-3}, c^n_{4j_n+1}]$, partitioned into $2^n n!$ simplexes in which f is quadratic. It can be proved that f is quadratic also on larger subsets, namely on all the sides of the coarser grid which is the cartesian product of $\{b_{2j_1-1}, b_{2j_1}, b_{2j_1+1}\} \times \cdots \times \{b_{2j_n-1}, b_{2j_n}, b_{2j_n+1}\}$. Hence, once that 3 samples out of the 5 laying on any of the previous segments, are known, also the other 2 are known.

Figure 2 shows these segments for $n = 2$ and $n = 3$ highlighting that some samples in the cell do not lay on any of them. In any case, a simple computation indicates that, for a n-dimensional cell, $3^n + 2n3^{n-1}$ sampling points belong to at least one of these segments.

We may now assume that the samples corresponding to the coarser grid are stored (for a total of $\prod_{i=1}^n (2m_i - 1)$ samples). It can be easily seen that $n+1$ out of the $n(n+3)/2+1$ samples in each simplex are located at points of this grid and that n further samples can be interpolated in constant time knowing other n values on the same grid. The remaining $n(n-1)/2$ samples are then retrieved from memory to allow final intrpolation.

With this scheme, the number of samples to be store is $\prod_{i=1}(4m_i - 3) - \sum_{j=1}^n (2m_j - 2) \prod_{i \neq j}(2m_i - 1)$ while the time needed for an inference is increased by the need of n further fetches from memory and subsequent quadratic interpolation. To as-

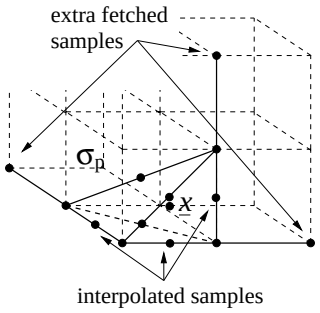

Figure 3: Segments used for inteprolation of missing samples starting from smaples on the coarse grid in a case with $n = 3$

certain what the impact of this interpolation is on the time complexity of the inference procedure we may concentrate on any of the segments in Figure 2 and identify the points in which the samples are known with their distance from the end of the segment which belongs to the simplex under consideration, i.e. 0, 1/2 and 1. The point at which we need the sample value correspond to a coordinate value of 1/4. Figure 3 shows the usual $n = 3$ case and the segments entailed in the inference depending on a point $\underline{x}$ belonging to a particular simples. The additional samples on the coarse grid to be fetched are highlighted along with the segments to which they and the missing samples belong.

We may now restrict f to one of these segments and note that, as f is quadratic then

$$\det \begin{bmatrix} 0 & 0 & 1 & f(0) \\ 1/4 & 1/2 & 1 & f(1/2) \\ 1 & 1 & 1 & f(1) \\ 1/16 & 1/4 & 1 & f(1/4) \end{bmatrix} = 0$$

from which we easily obtain $f(1/4) = (1/8)\,[3f(0) + 6f(1/2) - f(1)]$. As such an expression contains only multiplications and a division by 8 is can be straigthforwardly implemented in most common DSPs with an overhead of 2 multiplications, 2 sums and 1 shift.

4 Simulation Results

The proposed procedure has been manually coded into the native assembly language of the TMS320C6201 exploiting only one of the two available data paths so to leave the other for applications running in parallel with fuzzy inference. Loops have been unrolled by a preprocessing stage which overlaps segments of code designed to schedule the different tasks on the 4 available units and optimizing their exploitation.

n	cycles (full storage)	cycles (reduced storage)	saved storage
1	61	70	8
2	83	101	112
3	122	149	1176
4	156	192	10976
5	207	252	96040
6	253	307	806736

Table 1: Simulated performance of the proposed inference

Table 1 reports the number of clock cycles needed to perform a single inference with n from 1 to 6 with and without the addition of the sample interpolation procedure. Both results are obtained simulating the running code on the trial version of the development tools for the the TMS320C6201. The number of samples whose memorization is unnecessary is also reported assuming $m_1 = m_2 = m_n = 5$.

Note how on a 200MHz device an inference depending on 6 inputs can be computed in about 1.5μs. As a comparison, note that for any input point there are 64 firing rules whose first-order polynomial consequence entails at least 6 multiplications and additions. Thus a straightforward computation would entail not less than 768 cycles, i.e more than 3.8μs even with this gross underestimation.

References

[1] H. Watanabe, "Some Considerations of Design of Fuzzy Information Processors from a Computer Architectural Point of View," in *Fuzzy Engineering towards Human Friendly Systems*, ed. T. Terano, M. Sugeno, M. Mukaidono K. Shigemasu, IOS Press, Tokyo, 1992

[2] A. Costa, A.D. Gloria, P. Faraboschi, A. Pagni, G. Rizzotto, "Hardware Solutions for Fuzzy Control," Proc. of the IEEE, vol. 83, pp. 422-434, 1995

[3] R. Rovatti, M. Vittuari, "Linear and Fuzzy Piecewise-Linear Signal Processing with an Extended DSP Architecture", FUZZ-IEEE'98, Anchorage, Alaska, May 1998

[4] R. Rovatti, "Piecewise Multilinear and Piecewise Linear Fuzzy Systems as Universal Approximators in Sobolev Norms," IEEE TRANSACTIONS ON FUZZY SYSTEMS, vol. 6, pp. 235-249, 1998

[5] R. Rovatti, A. Ferrari, M. Borgatti, "Automatic Implementation of Piecewise-Linear Fuzzy Systems Addressing Memory-Performance Trade-Off," in *Fuzzy Hardware: Architectures and Applications*, ed. A. Kandel and G. Langholz, Kluwer Academic Publishers, 1998