Configurable Hardware Accelerator Architecture for a Takagi-Sugeno Fuzzy Controller

Oana Boncalo
University Politehnica Timisoara
Timisoara, Romania
oana.boncalo@cs.upt.ro

Alexandru Amaricai
University Politehnica Timisoara
Timisoara, Romania
alexandru.amaricai@cs.upt.ro

Zsofia Lendek
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
zsofia.lendek@aut.utcluj.ro

***Abstract*—In this paper, we present a parametric hardware accelerator for Takagi-Sugeno fuzzy controllers. The architecture consists of an application specific weighting function computation block, generic control output computation unit, and a programmable register file based interface. The proposed hardware design methodology is applied to a two degree of freedom robot arm controller. FPGA implementation results indicate that the hardware TS fuzzy controller supports throughputs up to 1.5 Msamples/sec, with maximum working frequencies of around 150 MHz.**

***Index Terms*—Fuzzy control, robot arm control, FPGA acceleration**

I. Introduction

Takagi-Sugeno (TS) fuzzy models can be used to tackle non-linear systems, such as a wide range of mechanical systems – robot arm, drones, etc – by employing a convex combination of local linear models [1], [2]. Thus, model-based controllers that can guarantee stability and performances such as robustness to model uncertainty and disturbance attenuation can be developed. The stability and performance requirements of TS fuzzy controllers can be included in the controller design process. This is an important advantage with respect to other non-linear model based controllers, such as fuzzy PID or gain scheduling approaches. However, the resulting controllers may suffer from high complexity.

One way to address this complexity is to employ hardware acceleration, either using ASICs or FPGAs. The main gains using dedicated circuitry are represented by increased throughputs – or sampling rates –, as well as improved energy/performance trade-offs. In this paper, we propose a configurable TS fuzzy controller architecture that can be customized with minimum redesign effort for: (i) different applications that require non-linear modelling by re-designing the membership function generation block, (ii) different values for physical parameters of the modelled non-linear system, by employing a programmable modified register file for the storage of different application parameters, and (iii) different cost-throughput constraints by means of architecture parameters. We provide numerical results of the proposed architecture for a two degree-of-freedom (2-DOF) robot arm controller; for this type of controller, we provide the implementation results on a Virtex-7 FPGA device for two cost-throughput configurations.

This paper is organized as follows: Section II presents the theoretical aspects associated to the TS fuzzy controllers, as well as the controller used for 2-DOF robot arm; Section III is dedicated to the proposed configurable hardware architecture; related work is presented in Section IV; concluding remarks are given in Section V.

II. Takagi-Sugeno Fuzzy Controllers

A. Takagi-Sugeno Fuzzy Model

A domain in which fuzzy systems have been extensively utilized and brought significant improvements is nonlinear control [2]. In this case, the usual "fuzzy", linguistic interpretation is rarely employed, but the name fuzzy is used to denote the combination of functions. Although the underlying rules still exist, the classical fuzzification/ inference/ defuzzification trio is generally described directly by the corresponding mathematical formulas. Next to model-free [3], data-driven [4], or expert-defined (neuro-)fuzzy controllers, model-based control design methods that ensure stability and performance measures have also been developed.

In the research described hereafter, we use dynamic Takagi-Sugeno (TS) fuzzy models [1] for the controller design of discrete-time nonlinear systems of the form:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= g(x(k), u(k)) \end{aligned} \tag{1}$$

where f denotes the state transition function, describing the evolution of the states over time, g is the measurement function, relating the measurements to the states, x is the vector of the state variables, u is the vector of the input or control variables, and y denotes the measurement vector. System (1) may also be affected by disturbances.

To design a controller in the TS framework, we first represent system (1) by a TS fuzzy model of the form:

$$\begin{aligned} x(k+1) &= \sum_{i=1}^{r} w_i(z(k))(A_i x(k) + B_i u(k)) \\ y(k) &= \sum_{i=1}^{r} w_i(z(k)) C_i x(k) \end{aligned} \tag{2}$$

where r is the number of local models, A_i, B_i, and C_i are the matrices of the ith local model, z is the vector of the scheduling variables, which may depend on the states, inputs, measurements, or other exogenous variables, and w_i are normalized membership functions, i.e., $w_i(z) \geq 0$ and $\sum_{i=1}^{r} w_i(z) = 1$, $\forall z$.

The TS model is a universal approximator [5], and many nonlinear systems can be exactly represented in a compact set of state variables as TS systems [6]. Such a model presents several advantages. System (2) is the convex combination of local linear models, which facilitates stability analysis and controller and observer design. In addition, many stability and design conditions for TS systems can be formulated as linear matrix inequalities (LMIs) [7], [8], for which efficient algorithms exist.

TS fuzzy models may also appear in descriptor form [9], in particular when considering mechanical systems. In the descriptor form, the system also contains non-linear terms on the left-hand side and the corresponding TS model is of the form:

$$\begin{aligned} \sum_{i=1}^{r_e} w_i^e(z(k)) E_i x(k) &= \sum_{i=1}^{r} w_i(z(k))(A_i x(k) + B_i u(k)) \\ y(k) &= \sum_{i=1}^{r} w_i(z(k)) C_i x(k) \end{aligned} \tag{3}$$

i.e., the left-hand side will also be a convex combination of local models, possibly depending on membership functions different from those on the right-hand side. This separation of the nonlinearities leads to a smaller number of local models [10] and a reduced number of LMI constraints [11], [12], i.e., reduced complexity compared to a classic TS model.

Thus, a descriptor model is characterized by the right-hand side non-linearities $\text{nl}_j(\cdot) \in [\underline{\text{nl}}_j, \overline{\text{nl}}_j]$, and the left-hand side non-linearities $\text{nl}_j^e(\cdot) \in [\underline{\text{nl}}_j^e, \overline{\text{nl}}_j^e]$. An exact descriptor TS representation is constructed by computing first the weighting functions:

$$\begin{aligned} h_j^0(\cdot) &= \frac{\overline{\text{nl}}_j - \text{nl}_j(\cdot)}{\overline{\text{nl}}_j - \underline{\text{nl}}_j} \quad h_j^1(\cdot) = 1 - h_j^0(\cdot) \\ v_j^0(\cdot) &= \frac{\overline{\text{nl}}_j^e - \text{nl}_j^e(\cdot)}{\overline{\text{nl}}_j^e - \underline{\text{nl}}_j^e} \quad v_j^1(\cdot) = 1 - v_j^0(\cdot) \end{aligned} \tag{4}$$

Based on these weighting functions, the membership functions w_i and w_i^e, are computed as follows:

$$w_i(z) = \prod_{j=1}^{p} h_j^{i_j}(z_j) \qquad w_i^e(z) = \prod_{j=1}^{p_e} v_j^{i_j}(z_j) \tag{5}$$

The computation of the weighting functions $h_j^0(\cdot)$ and $v_j^0(\cdot)$ is dependent on the physical properties of the modelled process. Furthermore, their value changes each sample, being dependent on the current measured state of the process.

B. Controller Design

Next, we consider controller design for system (3). In order to ensure that the controller stabilizes the closed-loop system, a classic controller structure is the Parallel Distributed Compensator (PDC), which consists of linear state feedbacks gains blended together using the the nonlinear membership functions $w_i(z), w_i^e(z)$, and has the form:

$$u(k) = -\sum_{i=1}^{r} w_i(z(k)) \sum_{i=1}^{r_e} w_i^e(z(k)) K_{ij} x(k) \tag{6}$$

The controller gains K_{ij} are computed based on Lyapunov synthesis and by solving a set of LMIs [7]. These gains are constant and they are usually computed offline. In what follows, we denote:

$$\begin{aligned} u(k) &= -K_{hv} x(k) \\ K_{hv} &= \sum_{i=1}^{r} w_i(z(k)) \sum_{i=1}^{r_e} w_i^e(z(k)) K_{ij} \end{aligned} \tag{7}$$

Therefore, computing the actual control input $u(k)$ to be applied to the system requires the following steps:

- Compute the weighting functions $h_j^0(\cdot)$ and $v_j^0(\cdot)$, corresponding to the modelled process
- Compute the membership functions w_i and w_i^e, as products between weighting functions $h_j^0(\cdot), h_j^1(\cdot)$ / $v_j^0(\cdot), v_j^0(\cdot)$, as described in (5)
- Compute the final gain matrix K_{hv}, and the control output $u(k)$, using (7)

Note that the first step is application dependent, while the computation of membership functions, final gain matrix and the control output is the same for all types of TS fuzzy controllers.

C. Numeric Example - 2-DOF Robot Arm Control

In this subsection, we exemplify how a TS representation is derived and, based on this representation, a TS controller is computed for a 2 degree-of-freedom (2-DOF) robot arm controller. The state vector of the robot arm is given by angles and the angular velocities of the two joints – $x = (q, \dot{q})$. Of these, the two angles are measured. The control inputs are the motor torques of the two joints - $u = \tau$. The reference to be followed is given by a pair of desired angles for the two joints. The dynamic model associated to the 2-DOF robot arm is:

$$M(q)\ddot{q} = -D(q, \dot{q})\dot{q} + I\tau \tag{8}$$

where M represents the mass matrix and D contains the Coriolis, centrifugal and friction forces. The physical parameters are given in Table I. This model can also be written as

$$E(x)\dot{x} = A(x)x + Bu \tag{9}$$

where

$$E(x) = \begin{bmatrix} I & 0 \\ 0 & M(x) \end{bmatrix} A(x) = \begin{bmatrix} 0 & I \\ 0 & -D(x) \end{bmatrix} B = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{10}$$

The mass matrix and the Coriolis, centrifugal and friction matrix are expressed as:

$$
\begin{aligned}
M(x) &= \begin{bmatrix} m_1(x) & 0 \\ 0 & \frac{M_2 L_2^2}{4} + I_{2y} \end{bmatrix} \\
D(x) &= \begin{bmatrix} d_1(x) & 0 \\ d_2(x) & b_2 \end{bmatrix}
\end{aligned} \tag{11}
$$

with

$$
\begin{aligned}
m_1(x) &= I_{1x} + I_{2z} + \cos(x_2)^2 (I_{2x} - I_{2z}) + \\
&\quad + M_2(L_1) + \frac{L_2 \cos(x_2)}{2})^2 \\
d_1(x) &= -x_4(\sin(2x_2)(\frac{M_2 L_2^2}{4} + I_{2x} - I_{2z}) + \\
&\quad + L_1 L_2 M_2 \sin(x_2)) + b_1 \\
d_2(x) &= x_3(\frac{\sin(2x_2)}{2}(I_{2x} - I_{2z} + \frac{L_2}{4}) + \\
&\quad + \frac{L_2 M_2 L_1 \sin(x_2)}{2})
\end{aligned} \tag{12}
$$

The mass matrix presents one non-linearity, while the $D(x)$ matrix presents two non-linearities. Therefore, the TS fuzzy model associated to this process has three weighting functions $v^0(x)$, $h_1^0(x)$ and $h_2^0(x)$, with the following expressions:

$$
\begin{aligned}
h_1 &= -x_4(\sin(2x_2)(M_2 \frac{L_2^4}{4} + I_{2x} - I_{2z}) + L_1 L_2 M_2 \sin(x_2)) \\
h_2 &= x_3 \frac{\sin(2x_2)}{2}(I_{2x} - I_{2z} + L_2^2 \frac{M_2}{4}) + L_2 M_2 L_1 \frac{\sin(x_2)}{2} \\
v &= I_{1x} + I_{2z} + M_1 \frac{L_1^2}{4} + \cos^2(x_2)(I_{2x} - I_{2z}) + \\
&\quad + M_2(L_1 + L_2 \frac{\cos(x_2)}{2})^2
\end{aligned} \tag{13}
$$

A set-point tracking controller that is able to follow some desired reference angles has been designed using the approach described in [13].

This controller has a form similar to (7):

$$
u = -K_{hv}[x(k)^T, \, (x^I(k))^T]^T \tag{14}
$$

where x denotes the states of the physical system (angles and angular velocities), $x^I(k)$ is a vector of auxiliary variables ("controller states") used in order to ensure that the tracking is realized without steady-state errors and $K_{hv} = \sum_{i=1}^{r} w_i(z(k)) \sum_{i=1}^{r_e} w_i^e(z(k)) K_{ij}$ is a convex combination of the controller gain matrices. The controller states are updated each sample k as:

$$
x^I(k+1) = x^I(k) + y_{ref} - Cx \tag{15}
$$

where y_{ref} denotes the reference to be followed and the matrix C is the output (measurement) matrix, i.e., $C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$.

The membership functions w and w^e are computed based on the weighting functions h_1, h_2, and v as follows: $w_1^e = v$, $w_2^e = 1 - v$, $w_1 = h_1 h_2$, $w_2 = h_1(1 - h_2)$, $w_3 = (1 - h_1)h_2$, $w_4 = (1 - h_1)(1 - h_2)$. For an easier implementation, the weighting functions are rewritten as:

$$
\begin{aligned}
h_1 &= c_1 + x_4(k)\sin(2x_2(k))c_2 + x_4(k)\sin(x_2(k))c_3 + \\
&\quad + c_4 \cos(x_2(k) + c_5 \cos(x_2(k)))^2 \\
h_2 &= c_6 + c_7 x_3^r(k)\sin(2x_2(k)) + c_8 x_3(k)\sin(x_2^r(k)) \\
v &= c_9 + c_{10}\cos(x_2(k)) + c_{11}\cos(x_2(k))^2
\end{aligned} \tag{16}
$$

The parameters c_i are constant for a specific set of physical parameters, and can be computed offline, see Table II.

TABLE I
2-DOF ROBOT ARM PHYSICAL PARAMETERS

$L_1[m]$	length first-second joint
$L_2[m]$	length second joint end-effector
$M_1[kg]$	mass first joint
$M_2[kg]$	mass second joint
$g[m/s^2]$	gravitational acceleration
$I_{1x}[kgm^2]$	moment of inertia
$I_{1y}[kgm^2]$	moment of inertia
$I_{1z}[kgm^2]$	moment of inertia
$I_{2x}[kgm^2]$	moment of inertia
$I_{2y}[kgm^2]$	moment of inertia
$I_{2z}[kgm^2]$	moment of inertia
$b_1[-]$	friction coefficient, first joint
$b_2[-]$	friction coefficient, second joint

TABLE II
WEIGHTING FUNCTION PARAMETERS

Notation	Formula
c_1	$\frac{\max(h_1) - (I_{1x} + I_{2z} + M_1 \frac{L_1^2}{4} - M_2 L_1^2)}{\max(h_1) - \min(h_1)}$
c_2	$\frac{T_s(M_2 \frac{L_2^2}{4} + I_{2x} - I2z)}{\max(h_1) - \min(h_1)}$
c_3	$\frac{T_s L_1 L_2 M_2}{\max(h_1) - \min(h_1)}$
c_4	$\frac{-L_1 L_2 M_2}{\max(h_1) - \min(h_1)}$
c_5	$\frac{-M_2 \frac{L_2^2}{4} + I_{2x} - I_{2z}}{\max(h_1) - \min(h_1)}$
c_6	$\frac{\max(w_{2,1})}{\max(h_2) - \min(h_2)}$
c_7	$\frac{-T_s(I_{2x} - I_{2z} + L_2^2 \frac{M_2}{4}}{2(\max(h_2) - \min(h_2))}$
c_8	$\frac{-T_s L_1 L_2 M_2}{2(\max(h_2) - \min(h_2))}$
c_9	$\frac{(\max(v_1) - (I_{1x} + I_{2z} + L_1^2 \frac{M_1}{4} + L_1^2 M_2))}{\max(v_1) - \min(v_1)}$
c_{10}	$\frac{-L_1 L_2 M_2}{\max(v_1) - \min(v_1)}$
c_{11}	$\frac{-(I_{2x} - I_{2z} + M_2 \frac{L_2^2}{4})}{\max(v_1) - \min(v_1)}$

In order to obtain increased throughput, as well as efficient FPGA resource consumption, a fixed point implementation for the controller has been considered. An analysis of the required precision, based on Matlab simulations, suggests that a 24-bit precision implementation, with 16 bits corresponding to the fractional part, yields negligible controller performance loss with respect to the floating point version. For computing the trigonometric functions 2^{nd} and 3^{rd} order Taylor series approximations proved sufficient.

III. Configurable Hardware Takagi-Sugeno Fuzzy Controller

A. Generic Architecture

Our main goal is to develop a generic and parametrizable architecture for the TS fuzzy controllers that can be tuned and programmed for a specific control application with minimum redesign effort. The proposed hardware controller consists of the following modules - Fig. 1 :

1) **Application specific fuzzy scalar computation block** - this module computes the weighting functions h_i and v_i. It is application dependent, and has to be redesigned for each control application.
2) **Generic parameterizable control block** - this module can be customized based on the following parameters: (i) number of control outputs n_u - size of the u vector, (ii) number of non-linearities in the controlled process n_{nl}, and (iii) parallelism degree in the scalar-matrix and vector-matrix multiplications p_m. Modifying the control application, as well as the throughput/cost trade-offs requires the regeneration of this module with the corresponding set of parameters. It consists of:
   - **Fuzzy scalar multiplication module** - computes the multiplications between the weighting functions required in the computation of the final gain matrix K_{hv}; the inputs of this block are the fuzzy scalars h_i and v_i, while the outputs are the products $\prod_{i \neq j; k \neq l} h_i(1-h_j)v_k(1-v_l)$; the $2^{n_{nl}}$ products are stored in a register file; the $2^{n_{nl}}$ multiplications are performed in a serial manner;
   - **Gain matrix computation block** - computes the final gain matrix K_{hv} based on (7) using the 2^{n_l} fuzzy scalar products and the $2^{n_{nl}}$ K_{ij} matrices; p_m multiply-add fused units are employed, computing in parallel p_m elements of K_{hv};
   - **Output computation block** - performs the multiplication between the K_{hv} matrix and the x, x^I vector, according to (14); it outputs the control vector u; this block is made of p_m multiply-add fused units
   - **Difference computation block** - computes the next internal state of the controller, see (15), based on the process state x, current controller state x^I and the input reference y_{ref};
3) **Programmable parameter modified register file** - this module consists of register files used to store the parameters c_i, the gain matrices K_{ij} and the matrix C. These register files can be accessed via a slave bus interface (*e.g.* AMBA AXI interface), and therefore can be software programmed. It has two access ports: (i) a 1-word port used to access the modified register file via the slave interface, and (ii) a multiple-word port used to access the modified register file from the processing units of hardware controller (Fig. 2); the number of words corresponding to the processing access port is parametrizable;

The parameters of the hardware controller are:

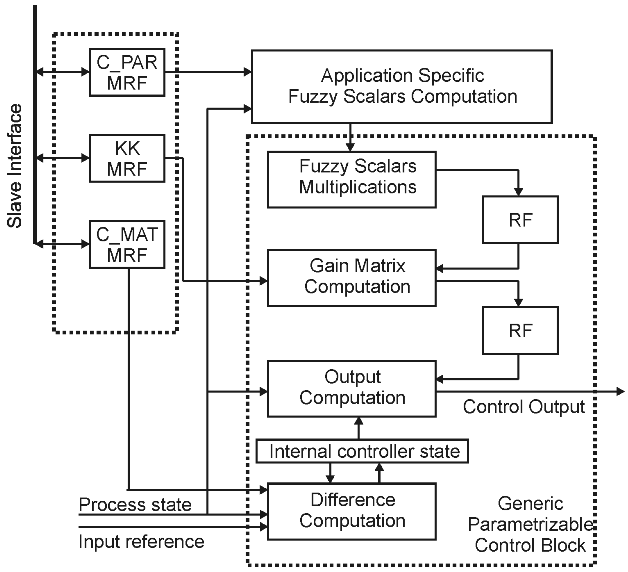

Fig. 1. Generic TS fuzzy controller architecture

1) the data quantisation, and the size of the fractional part expressed as the number of bits;
2) the number of non-linearities in the model of the control process n_{nl}; this number is equal to the number of fuzzy scalars h_i and v_i; the number of products $\prod_{i \neq j; k \neq l} h_i(1-h_j)v_k(1-v_l)$, as well as the number of gain matrices K_{ij} is equal to $2^{n_{nl}}$;
3) the number of control outputs n_u;
4) the parallelism degree for scalar-matrix and vector-matrix multiplication p_m - the number of elements in the result matrix/vector that are computed simultaneously; this parameter is limited by the number of control outputs n_u (the number of elements in the final output vector u);

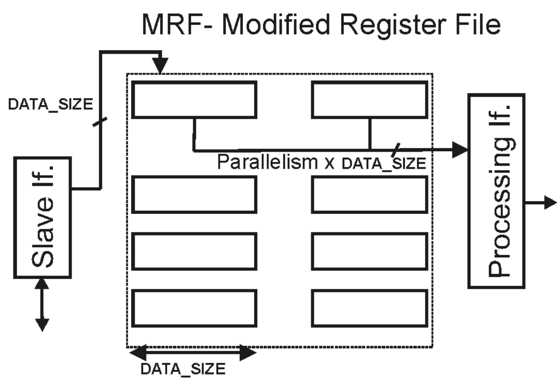

Fig. 2. Modified register file

The proposed architecture provides the following levels of flexibility:

1) For a specific application, the modification of different physical parameters within the controlled process (*e.g.* for the 2-DOF robot arm, the variation of masses, lengths, friction coefficients of joints in the robot arm

or moments of inertia) leads to the modification of controller parameters c_i and controller gain matrices K_{ij}; these changes will require the re-programming via the slave bus interface of the modified register files used to store these values;

2) In order to change the control application, the following steps have to be carried out:
   - Redesign of the application specific fuzzy scalar computation block that computes the fuzzy scalars h_i and v_i, according to the control application;
   - Re-generation of the gain matrix computation block, with the modification of the following two parameters: (i) the number of control outputs, and (ii) the number of non-linearities in the controlled process;
3) In order to modify the throughput characteristics of the controller architecture, the re-generation of the gain matrix computation block and the modified register file, by appropriately changing the p_m parameter used in scalar-matrix and vector-matrix multiplications is needed; an increase in the parallelism degree in the scalar-matrix and vector-matrix multiplications leads to increased in performance and larger implementation cost;

B. Two degree of freedom robot arm controller case study

The 2-DOF robot arm controller accelerator presents the generic architecture described in the previous subsection, and depicted in Fig. 1, with the following parameters for the generic parametrizable control block: number of non-linearities $n_{nl} = 3$, and the number of control outputs $n_u = 2$. Since $n_u = 2$, the values for the parallelism degree of the scalar-matrix and the vector-matrix p_m parameter are either 1 - fully serial -, or 2.

The fuzzy scalar computation block computes the weighting functions h_1, h_2, and v, according to (16). This module comprises of two parts: (i) the computation of the trigonometric functions $\cos(x)$, $\sin(x)$, and $\sin(2x)$, and (ii) the computation of the fuzzy scalars using a multiply-add based pipeline. The three trigonometric function are computed in parallel using the Taylor series based approximations.The three fuzzy scalars are computed in a serial manner, while the multiply-add operations corresponding to each scalar are performed in parallel (Fig. 3). Therefore, a maximum of 5 c_i parameters are required to be read simultaneously from the modified register file. Thus, the modified register file storing them has a 5-word port associated to the processing unit.

C. Implementation Results

We have implemented the 2-DOF controller design for Xilinx Virtex-7 VX485T-2 FPGA device, using the Xilinx Vivado 2017.1 tool. The data quantisation used is 24 bit length, with 16 bits for the fractional part. The design is pipelined, such that, the maximum number of multiplier or multiply-add units per pipeline stage is one. Results are depicted in Table III for $p_m = 1$ and $p_m = 2$. Sampling rate is estimated as the number of samples that can be processed by the controller per second. It is dependent on the maximum frequency of the controller, and the number of clock cycles required to process a set of input samples.

TABLE III
IMPLEMENTATION RESULTS FOR 2-DOF ROBOT ARM CONTROLLER

Matrix Parallelism	Slices	DSP Blocks	Frequency	Sampling rate [Ksamples/sec]
1	1948	34	142 MHz	807
2	1948	38	150 MHz	1515

Table III indicates that the main difference in terms of cost between the two configurations is represented in the number of DSP blocks, while the usage of logic and register based memory resources is almost the same. The difference in cost between the two configuration is of only 4 DSP blocks due to the high cost in terms of DSPs of the fuzzy scalar computation block - 26 blocks. The increase in parallelism for matrix computations - a cost increase of 11% in the amount of DSP blocks - leads to almost doubling in the controller sampling rate.

IV. RELATED WORK

Hardware acceleration for fuzzy controllers has been designed for a wide range of application that require nonlinear modelling. These include the type-1 fuzzy controllers in [14], [15], as well as the type-2 fuzzy and neuro-fuzzy controllers in [4], [16], [17] that can be used when a model is not available. A comparison between the proposed fuzzy controller and these approaches is difficult to perform due to:

- The different type of implemented controllers - although the proposed implementation can be categorized as a type-1 fuzzy controller, it does not use linguistic fuzzy rules. The latter relies on comparators, such as the min-max modules in [14]–[17]; in the proposed controller, the fuzzy rules are embedded in the computation of the weighting functions h and v; furthermore, the application field for the controllers presented in [14]–[17] is different;
- Different FPGA devices used for the evaluation - the design in [14] is implemented on a Xilinx Spartan-3 device, the one in [15] is implemented on a Xilinx Virtex-5, the one in [16] is implemented on a Xilinx Virtex II devices, the controller proposed in [3], [4] are implemented for ASIC, while the proposed approach is implemented on a Xilinx Virtex-7;

A review of the existing FPGA and hardware implementations for fuzzy controllers indicates that the proposed approach represents the first accelerator design of a TS fuzzy controller without an inference memory storing the linguistic rules.

V. CONCLUSIONS

In this paper, we presented a configurable hardware architecture for a specific class of TS fuzzy controllers, having the fuzzy rules embedded in the multiply-add based computation of specific weighting functions. To the best of our knowledge this is the first attempt to implement in hardware such a TS fuzzy controller with guaranteed stability and performance measures.

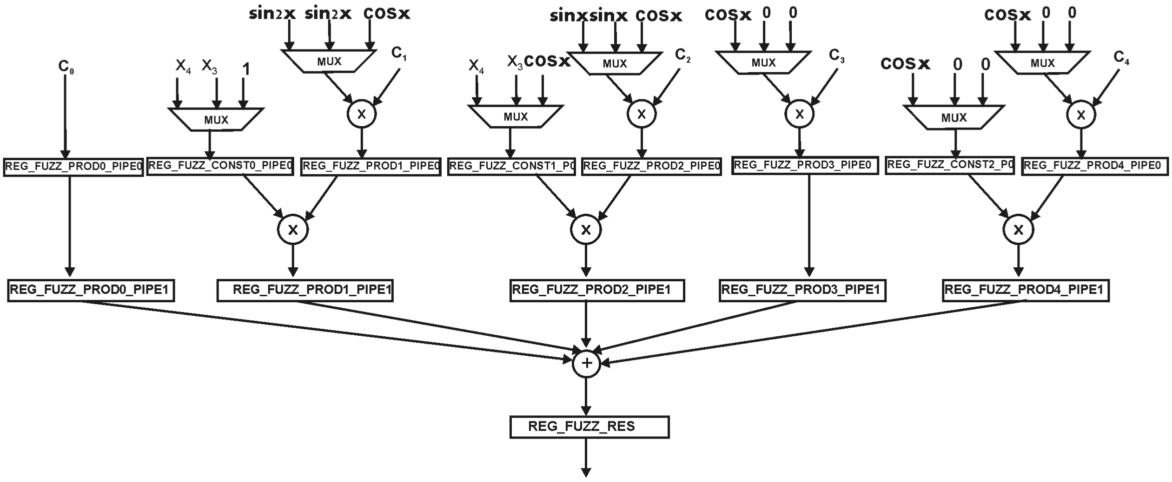

Fig. 3. Computation of weighting parameters h_1, h_2, v - the parameters c_i are read from the C_PAR MRF as follows: $\{c_1, c_2, c_3, c_4, c_5\}, \{c_6, c_7, c_8, 0, 0\}, \{c9, c_{10}, c_{11}, 0, 0\}$

The main goal is to obtain a high degree of flexibility in the controller architecture that allows its tuning with minimum redesign effort for different control applications, different physical parameters specific to a controlled process, as well as different cost-throughput trade-offs. The proposed accelerator uses a single module, mainly the computation block of the application dependent weighting functions h_i and v_i, that needs to be redesigned for different control applications.

We use the proposed TS fuzzy controller architecture design methodology for the implementation of a 2-DOF robot arm controller. For this use-case, we have provided two configurations with different parallelism degree in the scalar-matrix and vector-matrix multiplications. FPGA implementation results indicate that by increasing this degree we have roughly doubled the controller throughput, at the cost of around 11% increase in the amount of DSP blocks used.

Acknowledgment

This work has been supported by the European Space Agency Innovation Triangle Initiative project REDOUBT.

References

[1] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

[2] K. Tanaka and H. O. Wang, *Fuzzy Control System Design and Analysis: A Linear Matrix Inequality Approach*. New York, NY, USA: John Wiley & Sons, 2001.

[3] C.-F. Juang and J.-S. Chen, "Water bath temperature control by a recurrent fuzzy controller and its FPGA implementation," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 3, 2006.

[4] C. Juang and K. Juang, "Circuit implementation of data-driven tsk-type interval type-2 neural fuzzy system with online parameter tuning ability," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4266–4275, May 2017.

[5] C. Fantuzzi and R. Rovatti, "On the approximation capabilities of the homogeneous Takagi-Sugeno model," in *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, New Orleans, LA, USA, September 1996, pp. 1067–1072.

[6] H. Ohtake, K. Tanaka, and H. Wang, "Fuzzy modeling via sector nonlinearity concept," in *Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, vol. 1, Vancouver, Canada, July 2001, pp. 127–132.

[7] S. Boyd, L. El Ghaoui, E. Féron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, ser. Studies in Applied Mathematics. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1994.

[8] K. Tanaka, T. Ikeda, and H. Wang, "Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based designs," *IEEE Transactions on Fuzzy Systems*, vol. 6, no. 2, pp. 250–265, 1998.

[9] T. Taniguchi, K. Tanaka, K. Yamafuji, and H. Wang, "Nonlinear model following control via Takagi-Sugeno fuzzy model," in *Proceedings of the 1999 American Control Conference*, vol. 3, San Diego, CA, USA, June 1999, pp. 1837–1841.

[10] V. Estrada-Manzo, Zs. Lendek, T. M. Guerra, and P. Pudlo, "Controller design for discrete-time descriptor models: a systematic LMI approach," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 5, pp. 1608–1621, October 2015.

[11] T. M. Guerra, M. Bernal, A. Kruszewski, and M. Afroun, "A way to improve results for the stabilization of continuous-time fuzzy descriptor models," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 5960–5964.

[12] T. Taniguchi, K. Tanaka, and H. O. Wang, "Fuzzy descriptor systems and nonlinear model following control," *IEEE Transactions of Fuzzy Systems*, vol. 8, no. 4, pp. 442–452, 2000.

[13] Zs. Lendek, E. Páll, Z. Nagy, and J. Lauber, "Local stabilization of discrete-time TS descriptor systems," *Engineering Applications of Artificial Intelligence*, vol. 67, pp. 409–418, Jan. 2018.

[14] S. Sanchez-Solano, A. J. Cabrera, I. Baturone, F. J. Moreno-Velo, and M. Brox, "Fpga implementation of embedded fuzzy controllers for robotic applications," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 4, pp. 1937–1945, Aug 2007.

[15] S. Gdaim, A. Mtibaa, and M. F. Mimouni, "Design and experimental implementation of dtc of an induction machine based on fuzzy logic control on fpga," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 3, pp. 644–655, June 2015.

[16] R. Seplveda, O. Montiel, O. Castillo, and P. Melin, "Embedding a high speed interval type-2 fuzzy controller for a real plant into an fpga," *Applied Soft Computing*, vol. 12, no. 3, pp. 988 – 998, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S156849461100473X

[17] M. D. Schrieber and M. Biglarbegian, "Hardware implementation and performance comparison of interval type-2 fuzzy logic controllers for real-time applications," *Applied Soft Computing*, vol. 32, pp. 175 – 188, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494615001787