

Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing. Fuzzy provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions.

Unlike classical logic, which requires a deep understanding of a system, exact equations, and precise numeric values, Fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from our knowledge and experience. Fuzzy Logic allows expressing this knowledge with subjective concepts such as very hot, bright red, and a long time, which are mapped into exact numeric ranges.

In the field of engineering, control applications such as temperature control, traffic control, or process control are the most prevalent of current fuzzy logic applications. Industrial control systems are commonly grouped into combinational systems and systems operating in a sequential fashion. In a combinational control system, the output signals depend entirely on the state of its input signals, while in a sequential system, some or all of its outputs depend on previous machine states as well as on current inputs.

Fuzzy Logic has been found to be very suitable for embedded control applications. Several manufacturers in the automotive industry are using fuzzy technology to improve quality and reduce development time. In aerospace, fuzzy enables very complex real time problems to be tackled using a simple approach. In consumer electronics, fuzzy improves time to market and helps reduce costs. In manufacturing, fuzzy is proven to be invaluable in increasing equipment efficiency and diagnosing malfunctions.

Fuzzy Automaton also called as the Finite-state machine or Sequential machine plays a significant role in Complex sequential circuits. The concept of uncertainty of being in a particular state when combined with the event-driven behavior of dynamic systems can create an attractive area for new fuzzy logic applications.

In this project a fuzzy automaton model has been developed using the Graphical User Interface facilities available in java and to connect it as an agent in the Complex Distributed System. Here the complex system used is the Stressometer developed by the ABB Automation products used for flatness control in a Steel manufacturing Industry.

This whole project was divided into four parts which includes Developing the program, Dividing the program into Data Entry Part and Computation part, Connecting to the complex Distribution system as an agent, Developing the graphical portion to provide the inputs as well as to Display the results in a graphical way, and to test the program for its proper operation so that it can be used in further developments.

Hybrid Fuzzy Boolean Finite State Machine

The model of HFB FSM that is implemented by a Boolean automata using two-valued logic is given by a set of formulas as given below.

$$\begin{aligned}Z_F &= X_F \circ R^* \\R^* &= G(R_S) \\Z_C &= DF(Z_F) \\U_B &= f_u(y_B) \\X_B &= B(X_F) \\Z_B &= B(Z_F) \\Y_B &= f_y(X_B, W_B, Z_B, y_B)\end{aligned}$$

Where X_F and Z_F stand for a finite set of fuzzy inputs and outputs, respectively, W_B and U_B stand for a finite set of two-valued logic inputs and outputs, respectively. Defuzzified outputs are denoted by Z_C , R^* is a composite linguistic model, and O is the operator of composition. Each crisp state of the HFB FSM is characterized by an overall linguistic model R_S , or a set of linguistic sub-models in the case of multiple-input-single-output (MISO), and multiple-input-multiple-output (MIMO) systems.

A fuzzy state is defined by a crisp (two-valued) state and a state membership function as shown below.

$$S_{F_k} : S_k, g_{S_k}$$

Where S_{F_k} stands for fuzzy state k , S_k represents crisp state k , and g_{S_k} is the state membership function associated with S_k . G stands for the matrix of state membership functions, X_B , Z_B , Y_B , and y_B are two-valued Boolean input, output and state variables, respectively. B stands for a Fuzzy-to-Boolean transformation algorithm to map a change in the status of a fuzzy variable into state changes of a finite set of corresponding Boolean variables. The Z_C crisp values of the fuzzy outputs are obtained by evaluating a defuzzification strategy, DF .

On the basis of the concept of a fuzzy state, the HFB FSM stays in a number of crisp states. Simultaneously, to a certain degree in each. One of these states is referred to as a dominant state for which the state membership function is a 1 (full membership). For each fuzzy state of the HFB FSM model, a R_i^* composite linguistic model is created from a finite set of R_S overall linguistic models ($i=1, \dots, p$). Let the HFB FSM be in fuzzy state S_{F_k}

$$R_k^* = \max [\min (\beta_1^k, R_{S_1}), \min (\beta_2^k, R_{S_2}), \dots, \dots, \min (\beta_k^k, R_{S_k}), \dots, \min (\beta_p^k, R_{S_p})]$$

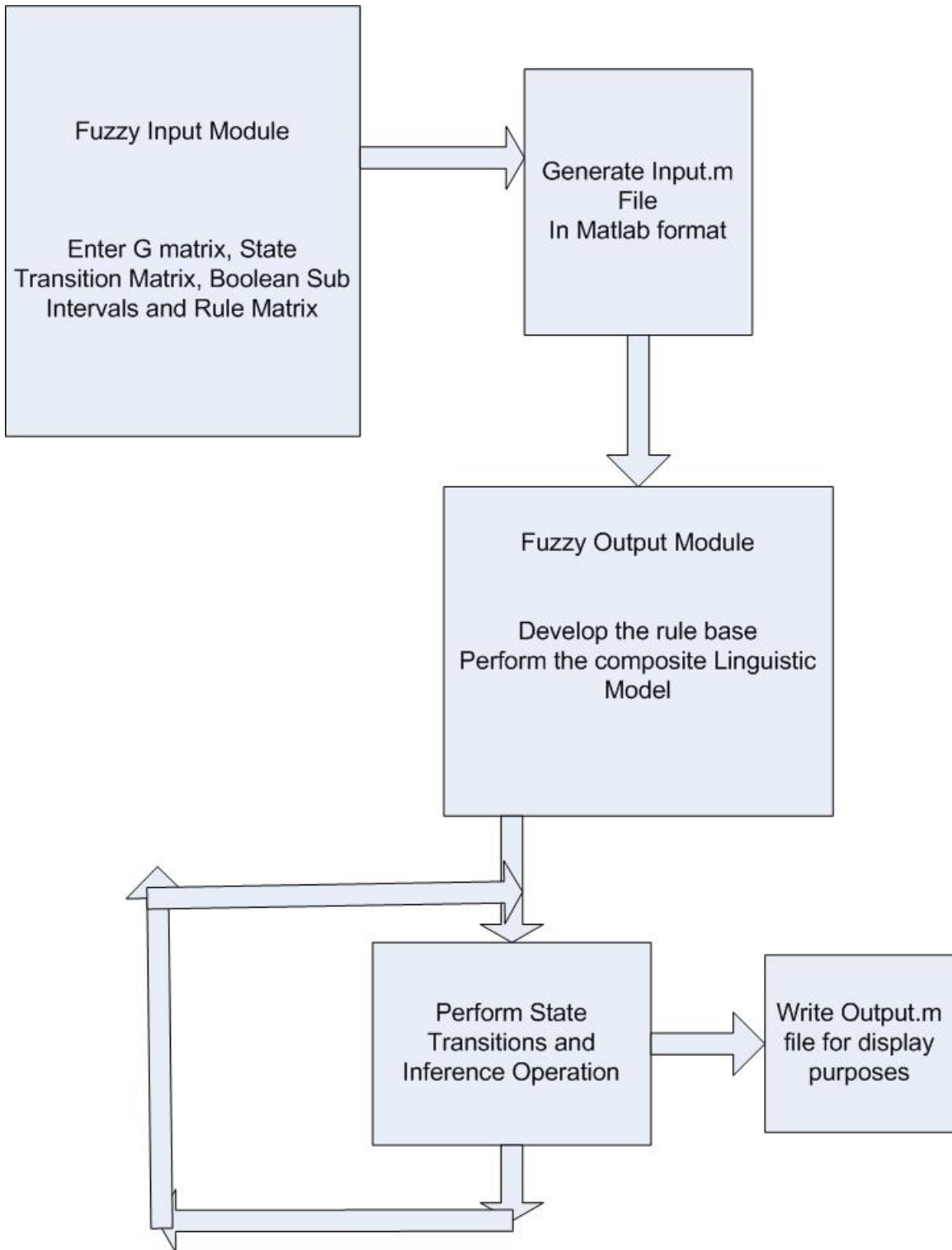
where $\beta_1^k, \beta_2^k, \dots, \beta_p^k$ stand for the degrees of state membership function g_{S_k} and $R_{S_1}, R_{S_2}, \dots, R_{S_p}$ are the overall rules in crisp states S_1, S_2, \dots, S_p , respectively.

The state transients of the HFB FSM are specified by means of a sequence of changes in the states of the fuzzy inputs and outputs, as well as of the two-valued inputs. The changes in the states of the fuzzy inputs and outputs are mapped into the corresponding sequence of changes of Boolean input and output variable sets, respectively. That is, the change in the state of fuzzy input X_F is mapped into state changes of a set of two-valued Boolean variables. The emphasis here is on detecting a change in state, not on quantifying the value of the fuzzy input in two-valued terms.

Let z_c be crisp value of fuzzy output Z_F , let λ_z denote the height of the maxima and α_i denote the threshold value assigned to Boolean sub-interval i . The discrete valued domain set of α is the same as that of the membership functions of inputs and outputs. Let n be the number of disjoint Boolean sub-intervals, and let Z_{B_i} denote the two-valued Boolean output variable which is assigned to disjoint sub-interval i .

$$Z_{B_i} = 1, \text{ iff the } Z_C \text{ position falls into Boolean sub-interval } i \text{ (} i=1, \dots, n \text{) and } \lambda_z \geq \alpha_i, \text{ otherwise } Z_{B_i} = 0 \text{ } Z_{B_j} = 0 \text{ for all } j \neq i \text{ (} j=1, \dots, n \text{).}$$

That is, the change in the state of a fuzzy output variable is mapped into state changes of a finite set of Boolean output variables.



Block Diagram of the GEFASA module implemented