

APPLICATION OF FUZZY LOGIC TOOLBOX FOR MODELLING FUZZY LOGIC CONTROLLERS

Krzysztof Olesiak

Czestochowa University of Technology, Faculty of Electrical Engineering Poland

Abstract. *Computer technology, which has been developing very fast in the recent years, can be also fruitfully applied in teaching. For example, the software package Matlab is highly useful in teaching students at Bachelor Programs of Electrical Engineering and Automatics and Robotics. Fuzzy Logic Toolbox of the Matlab package can be used for designing and modelling controllers. Thanks to a large number of pre-defined elements available in the libraries, it is possible to create even highly complicated models of systems without much effort. Fuzzy Logic Toolbox is especially useful for exploring the basic rules of designing fuzzy logic controllers. The rules involve selecting input and output membership functions, determining their location with respect to one another and defining their ranges. When the membership functions are introduced, a rule base is defined and a defuzzification method is selected. For any defuzzification method, a control surface is obtained, which can be modified by changing the rule base and/or the input and output parameters of the membership function.*

Keywords: *defuzzification method, fuzzy logic controller, fuzzy set, linguistic variable, membership function, rule base.*

Introduction

Computer technology is nowadays widely applied in teaching students of BSc and MSc engineering programs. For instance, the program Matlab/Simulink has been applied for solving problems of modelling drive systems (Olesiak, 2014; Jakubiec, 2015), DasyLab offers computerised methods of carrying out measurements of selected physical quantities (Ptak, 2015), and interactive computer simulations have been applied in many ways in higher education (Praužner, 2016; Praužner, 2015).

From an analytic point of view, designing and tuning fuzzy logic controllers is a time-consuming and complex problem. This is caused by the specific properties of linguistic rules defining the controller algorithm, by the number of inputs and outputs, and by the various possibilities of selecting a membership function for the variables of the control process (Carvajal et al., 2000).

Fuzzy Logic Toolbox of the Matlab package can be used for designing simple fuzzy logic controllers of the SISO type (single-input single-output), as well as more complex controllers of the MISO type (multiple-input single-

output). It is possible to choose between the Mamdani and the Sugeno structure for a controller. In the former structure, the antecedents and consequents of rules take a fuzzy form, whereas in the case of the latter structure, only antecedents are fuzzy, with consequents being expressed as classic functions.

Input and output linguistic variables

The process of designing a fuzzy logic controller starts with selecting a structure and the number of inputs and outputs of linguistic variables. Subsequently, methods are selected for carrying out the product and sum operations on the fuzzy sets. This is done on the basis of the following dependences (Xu et al., 2000):

$$\forall x \in X: \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (1)$$

$$\forall x \in X: \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (2)$$

where: where: x – element, X – universe of discourse, μ_A , μ_B – membership functions of fuzzy sets A and B , $\mu_A(x)$, $\mu_B(x)$ – degree of membership for an element in fuzzy sets.

The next step is to select algorithms of implication and aggregation and to carry out a preliminary selection of a defuzzification method. As a result, a fuzzy controller of the MISO type can be obtained, as presented in Fig.1.

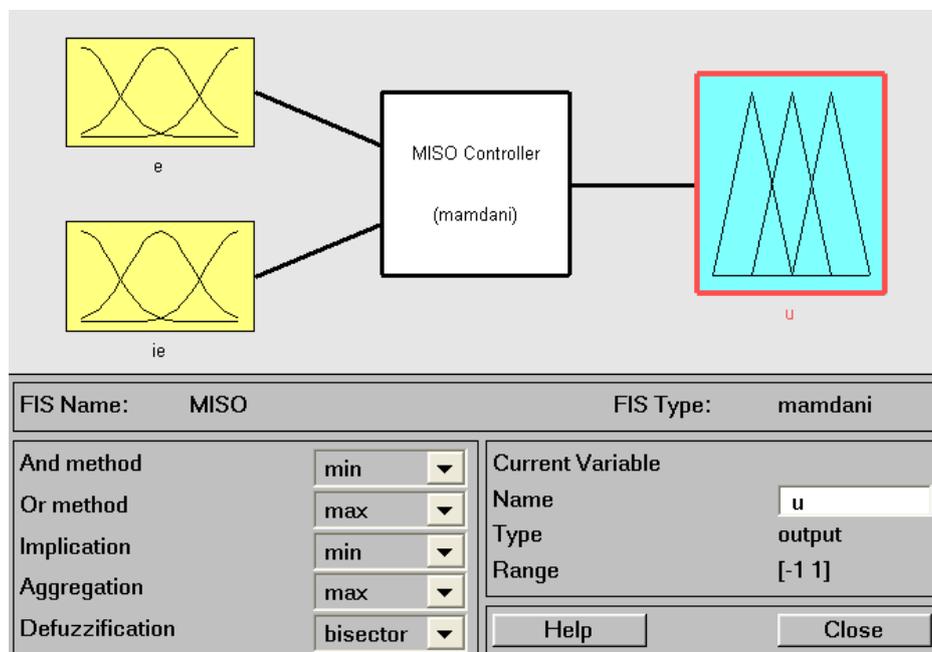


Figure 1 Fuzzy logic controller of the MISO type with linguistic variables modules

The controller has two modules of input linguistic variables e and ie , and one module of output linguistic variable u . For each of these modules, it is necessary to select a type of membership functions, their number and location with respect to one another, determined by the height of the intersection point of neighbouring functions. In the case of input linguistic variables, triangular functions were applied, as presented in Figs 2 and 3.

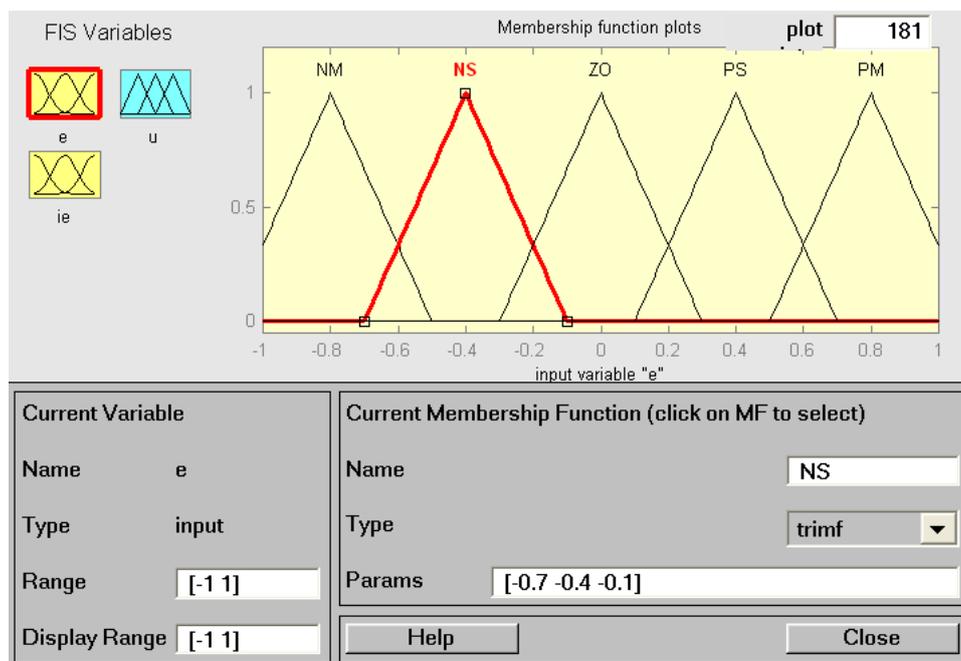


Figure 2 Input membership functions of the linguistic variable e

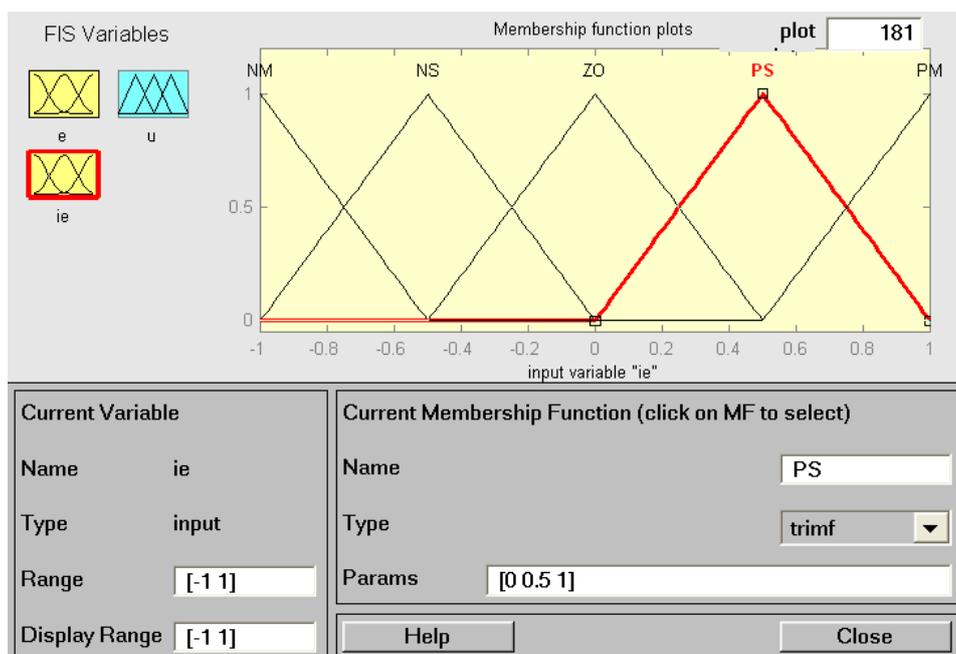


Figure 3 Input membership functions of the linguistic variable ie

Input linguistic variables presented in Figures 2 and 3 include 5 triangular membership functions defining the error e and the error integral ie . The notation used for the functions is as follows: NM – negative mean, NS – negative small, ZO - zero, PS – positive small, PM – positive mean.

For the output linguistic variable u , Gaussian membership functions were applied. The functions are distributed evenly in a standardised interval $[-1, 1]$. Seven functions were used altogether, with the following two symbols added to the previously enumerated ones: NB – negative big and PB – positive big. The output linguistic variable is presented graphically in Figure 4.

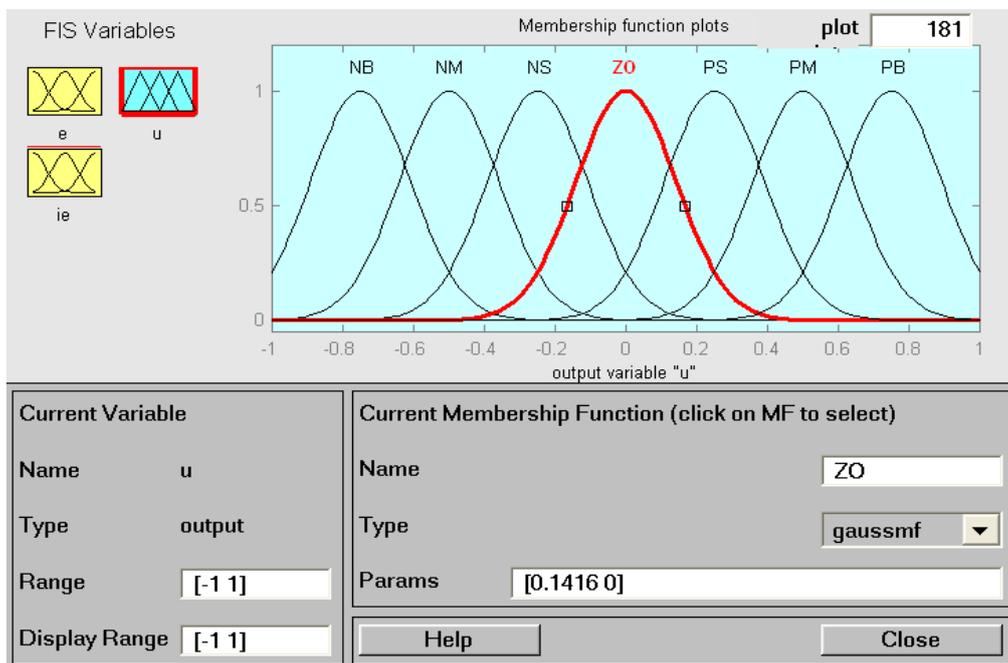


Figure 4 Output membership functions of the linguistic variable u

Once the input and output membership functions have been introduced, it is necessary to define the rule base of the fuzzy logic controller. To this end, a standard Mac Vicar-Whelan table was used after being modified taking into account the number of input and output membership functions defining linguistic variables. The rule base is presented in Table 1.

Table 1 Rule base of the MISO type fuzzy logic controller

Error e	Error integral ie				
	<i>NM</i>	<i>NM</i>	<i>ZO</i>	<i>PS</i>	<i>PM</i>
<i>NM</i>	NB	NB	NM	NS	ZO
<i>NS</i>	NB	NM	NS	ZO	PS
<i>ZO</i>	NM	NS	ZO	PS	PM

Error <i>e</i>	Error integral <i>ie</i>				
	<i>NM</i>	<i>NM</i>	<i>ZO</i>	<i>PS</i>	<i>PM</i>
<i>PS</i>	NS	ZO	PS	PM	PB
<i>PM</i>	ZO	PS	PM	PB	PB

A rule base defining the mutual dependences between input linguistic variables and the output linguistic variable has to be derived with the use of a rule editor suitable for the designed model of a fuzzy logic controller. With respect to all the rules, a weight coefficient equal to one was used, as well as a method of combining rule antecedents by means of the product formula. Even though the sequence of introducing the rules is insignificant from the viewpoint of control characteristics, they are typically introduced according to the row or column sequence. Ordering the rules when they are being introduced facilitates checking their correctness at a later stage. An illustration of the edited rule base is presented in Figure 5.

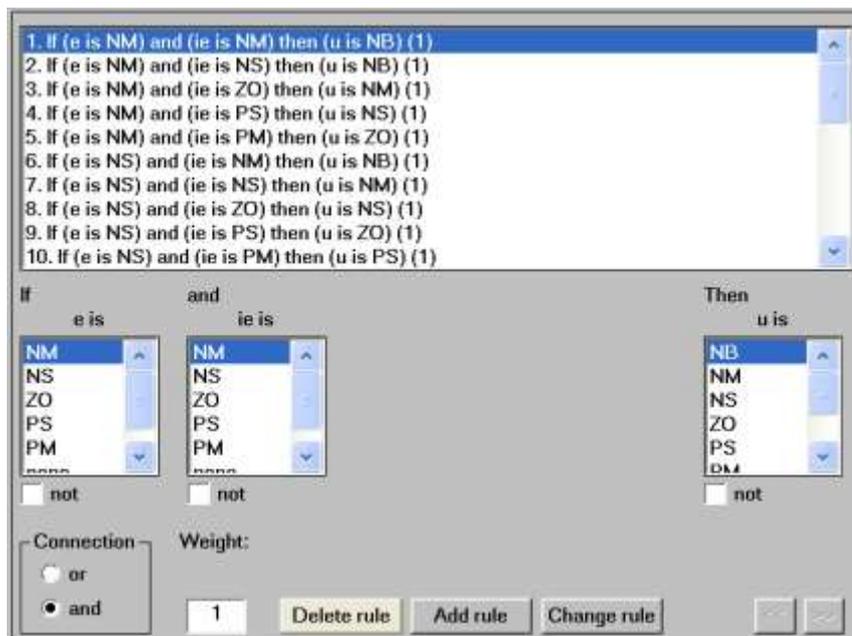


Figure 5 Rule base editor of the fuzzy logic controller

Defuzzification methods

The next step in designing a fuzzy logic controller is the choice of a defuzzification method for transforming a fuzzy set defined by a number of membership functions into a scalar value. Fuzzy Logic Toolbox offers five defuzzification methods, some of which are continuous and some discontinuous. With respect to defuzzification, continuity of a method can be defined by means of the following general dependence (Sivanandam et al., 2006):

$$\begin{aligned} &\forall \varepsilon > 0 \exists (\delta > 0 \wedge d > 0): \\ &|x_1(d) - x_1(d-1)| < \delta \wedge \dots \wedge |x_n(d) - x_n(d-1)| < \delta \\ &\Rightarrow |u(d) - u(d-1)| < \varepsilon \end{aligned} \quad (3)$$

where: δ – change of the input signals, ε – change of the output signal, d – selected discretization step, x_1, \dots, x_n – input linguistic variables, u – output linguistic variable.

Fuzzy controller surface is obtained by means of the Center of Sums method. This method is similar to the Center of Area (CoA) method, but much faster. It also provides continuity and participation in the inference process all active rules. Common areas of the membership functions are included in the defuzzification process more than once using the Center of Sums method.

For a discrete case, the following dependence is used (Sivanandam et al., 2006):

$$u = \frac{\sum_{i=1}^m u_i \cdot \sum_{k=1}^n \mu_{CLU(k)}(u_i)}{\sum_{i=1}^m \sum_{k=1}^n \mu_{CLU(k)}(u_i)} \quad (4)$$

where: u – output linguistic variable, $\mu_{CLU(k)}$ – compressed membership function for each k -th rule, u_i - functions of output fuzzy sets.

The application of the Centre of Sums method is illustrated graphically in Figure 6.

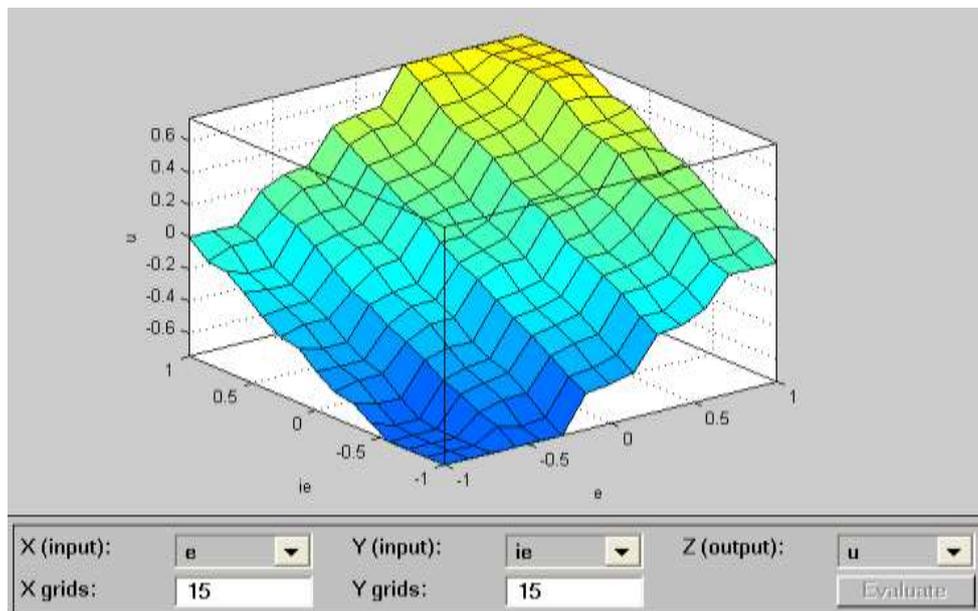


Figure 6 Dependence of the output signal on input signals in a fuzzy controller for the defuzzification method Center of Sums

In the fuzzy controller under scrutiny, the scalar value was also obtained by means of the Middle of Maximum method, represented by the following expression (The Math Works, 2012):

$$u = \frac{\inf \{ \mu_U(u) = hgt(U) \} + \sup \{ \mu_U(u) = hgt(U) \}}{2} \quad (5)$$

where: $\mu_U(u)$ – membership function of output signal, $hgt(U)$ – height of fuzzy set U .

The application of the Middle of Maximum method for obtaining the fuzzy controller surface is shown in Figure 7.

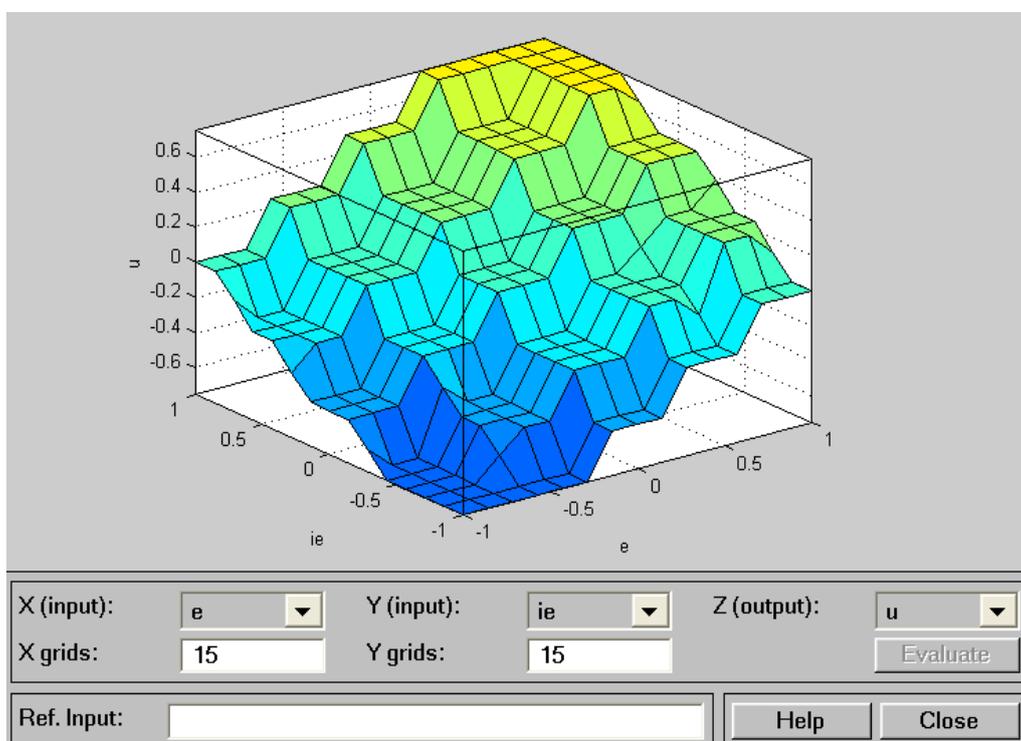


Figure 7 Dependence of the output signal on input signals in a fuzzy controller for the defuzzification method Middle of Maximum

Conclusions

The Fuzzy Logic Toolbox software offers a possibility of modifying a number of parameters, such as type of membership function, function location within a selected interval, height of the intersection point of neighbouring functions, the number of rules and the defuzzification method applied. By modifying these parameters, it is possible to observe their impact on the shape of the control surface of the fuzzy logic controller.

Due to the fact that there exists a wide spectrum of possibilities in designing and analysing the operation of fuzzy logic controllers, Fuzzy Logic Toolbox is a highly effective and helpful tool for performing various didactic and educational tasks. This explains why it is widely applied by instructors of various subjects taught at technical universities.

References

- Carvajal, J., Chen, G., & Ogmen, H. (2000). Fuzzy PID controller: Design, performance evaluation, and stability analysis. *Information Sciences*, No. 123, pp. 249-270.
- Jakubiec, B. (2015). Multiphase permanent magnet synchronous motor drive for electric vehicle. *Przegląd Elektrotechniczny (Electrical Review)*, 91, No. 12, pp. 125-128.
- Olesiak, K. (2014). Selected Problems of the Asynchronous Drive Control with the Three-phase Soft-start System. Proceedings of the Conference: Symposium on Mechatronics Systems, Mechanics and Materials, Jastrzebia Gora, Poland, October 09-10 2013, In: *Mechatronic Systems, Mechanics And Materials II, Book Series: Solid State Phenomena, Vol. 210*, pp 245-251.
- Prauzner, T. (2015). Analysis of the results of the pedagogical research and EEG in the aspect of effective modern teaching aids in the technical education. In: *Society. Integration. Education, Proceedings of the International Scientific Conference. Volume IV, Latvia, Rezekne, May 22th-23th 2015*, pp. 480-489.
- Prauzner, T. (2016). Interactive computer simulation as a response to contemporary problems of technical education. In: *Society. Integration. Education, Proceedings of the International Scientific Conference. Volume II, May 27th – 28th 2016, Rēzekne, Latvia*. pp. 579-588.
- Ptak, P. (2015). Application of DasyLab in teaching electrical engineering. In: *Society, Integration, Education, Proceedings of the International Scientific Conference. Volume IV, Latvia, Rezekne, May 22th-23th 2015*, pp. 490-499.
- Sivanandam, S. N., Sumathi, S., & Deepa, S. N. (2006). *Introduction to Fuzzy Logic using MATLAB*. Springer-Verlag, Berlin.
- The Math Works (2012). *Fuzzy Logic Toolbox for use with Matlab - User's Guide*. 3 Apple Hill Drive Natick, Massachusetts, USA.
- Xu, J. X., Hang, C. C., & Liu, C. (2000). Parallel structure and tuning of a fuzzy PID controller. *Automatica*, Vol. 36 (2000), pp. 673-684.