Webology (ISSN: 1735-188X)
Volume 19, Number 4, 2022

Software Platform For The Design Of Diffuse Pid Controllers On Arduino Cards

Johan Camilo Santiago Rodriguez*, MSc. Oscar Manuel Duque Suarez, Msc. July Andrea Gomez Camperos*****

*Universidad de Pamplona, Facultad de Ingenierías y Arquitectura, Programa de Ingeniería mecatrónica. Km 1 Vía Bucaramanga Ciudad Universitaria, Pamplona, Norte de Santander, Colombia.

** SENA, SENNOVA, Tecnoacademia Cúcuta, Cúcuta, Regional Norte de Santander,

*** Universidad Francisco de Paula Santander Seccional Ocaña, Grupo de Investigación en Nuevas Tecnologías, Sostenibilidad e Innovación (GINSTI), Colombia

---

ABSTRACT:
When controlling systems that present chaotic operating zones that produce undesired behavior and the plant is required to present a smooth change when entering this zone, fuzzy controllers have proven to be effective in these types of situations where the only requirement to the design of this is the knowledge of the system and the experience that it has in the control. This project will allow students to consolidate their knowledge in the design of fuzzy controllers and to carry out research with this type of controllers, through a platform made in open software where a license is not required for its use. This platform can design fuzzy controllers and making the fuzzy controller operate autonomously without relying on a computer equipment to perform the tests and commissioning of said plant on a low-cost hardware open board from the Arduino series.

Keywords: Fuzzy controller, platform, chaotic, ground.

1. INTRODUCTION

fuzzy controllers usually require a computer team to perform the mathematical processes of fuzzy logic and then send them to the embedded card in charge of receiving and sending information to the plant to control. The students at the university of Pamplona at the time of carrying out these practices are supported so that those students of scarce resources do not limit themselves to only observe their colleagues work because the minimum requirements for the team to execute the program They are properly superior to your computer equipment.

The software platform developed in this project allows the student not to require a computer equipment to operate the fuzzy controller and the team requirements for the design are more accessible to the student, no license is required for non-commercial use. On the DICODI software platform you can design fuzzy controllers, evaluate the controller according to the assigned rules, export a text file containing a code C of Arduino with the knowledge base of the controller and instructions to operate autonomously without requiring a computer computer (output universes have a range of 0 to 255).

2. FUZZY LOGIC

Fuzzy logic, belonging to the area of Artificial Intelligence, was initially investigated by Lofti Zadeh in 1965 in his article "Fuzzy Sets" (Zadeh, 1965). Initially conceptualized as a "fuzzy whole", this mathematical article presents its definition as "a class with varying degrees of belonging". This mathematical framework allows us to make a systematic treatment to the imprecision and uncertainty produced, mainly, by variables with a high rate of ambiguity. According to classical ensembles, the weather is either cold or hot. However, with diffuse logic the climate can be hot and cold at the same time to a certain degree of belonging in both sets (Manuel and Freddy, 2021).

In addition, Zadeh mentions that "this framework provides a natural means to deal with problems in which the source of imprecision is the absence of clearly defined criteria of class membership rather than the presence of random variables" (Zadeh, 1965). Some uses given to diffuse logic are the optimization of instantaneous changes in the production and reinjection of wells under the dynamics of geothermal deposits (Füsun S.,2020), to control the position of a solar panel as time advances to obtain the most solar energy (Lu Li and Yinshi Li, 2020), for the stationary flight of a grape robot (Calderón and Alejandra, 2015).

2.1. DIFFUSE CONTROLLER

A fuzzy controller is composed of four main parts: fuzzification interface, knowledge base, decision logic and defuzzification interface. The fuzzification interface measures the values of the input variables to perform a scale mapping that transfers the range of values of the variables to a fuzzy discourse universe. Fuzzification converts input data into linguistic values that are the labels of membership functions or fuzzy sets. as shown in figure 1

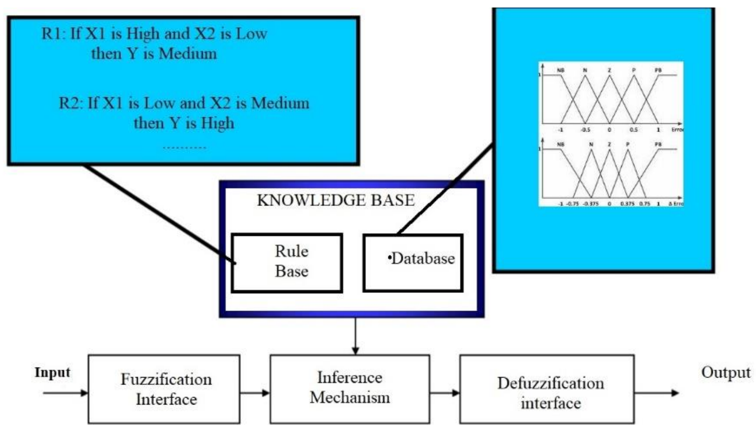

Figure 1. Structure of a System Based on Fuzzy Rules- Source: Samuel Sanahuja

The knowledge base contains all the information of the application to be monitored, as well as the controller's goals. It consists of a database and a base of linguistic rules to control the variable. The database provides definitions for rule-setting and manipulation of fuzzy data. The rules base characterizes the control goals and policy experts use to carry out control, using propositions. A fuzzy control algorithm must be able to infer a corresponding control action for each state of the process to be controlled, a property called a unit. The database strategy comprises the support for the definition of fuzzy sets.

The inference mechanism for making decisions within a fuzzy controller is the core of it. It simulates the logic used by people to make decisions, based on diffuse concepts and the inference of control actions, using implications and rules established according to the knowledge base. The defuzzification interface handles scale mapping that converts the range of values of the output variables to their corresponding discourse universes. Defuzzification is the tool to obtain the clear control action from a diffuse control action (Pedro Ponce, 2010). The methods regularly used for defuzzification are the area centre, maximum centre, maximum left, and maximum right.

2.1.1. Elements of a Fuzzy Controller

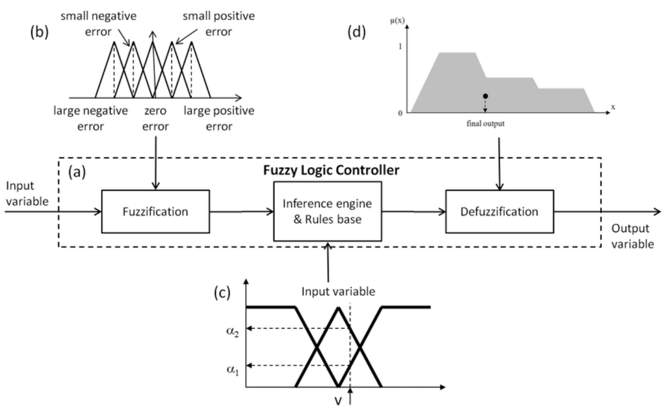

Figure 2. Representation of the elements of a diffuse controller

Linguistic variable: A linguistic variable is an expression or word of natural or artificial language that serves to represent any element that is too complex, or of which there is no specific definition.
Universe of speech: Range of information necessary for the system to present a correct functioning. Diffuse Set: It is a collection of elements of a universe X in which there is a gradual progression defined by a membership or membership function (i Casals, 1997). Membership or membership function: A membership function is one that indicates the degree of membership of an ensemble in the universe of speech. this can be seen in figure 2

3. METHODOLOGY FOR THE CREATION OF THE PLATFORM

The methodology used for the creation of the DICODI platform was the RUP methodology with follow-up to the ISA standard S101 (ANSI/ISA-101.01-2015) and was used for the development of the DICODI platform:
• Start Phase
• Stage of development
• Build Phase

• Transition Phase (not executed)

3.1. Start Phase

In this phase, the elements necessary for the user to design the fuzzy controller were defined:

• Library Opensource to facilitate the programming of user interaction and display of elements.
• Parameters of the set.
• Range of the speech universe.
• Add and remove speech universes.
• Add and remove membership functions.
• Display of the current speech universe and indicator of this.
• Evaluation of rules entered by the user
• Add and delete rules.

3.2. Elaboration phase

In the elaboration phase, the basic commands of the processing language began to be known, through a series of examples of the software page these instructions were quickly assimilated. In order to be able to develop the objects of the platform several libraries made by programmers for the creation of HMI or guides were analyzed, Of those observed which presented a learning curve more suitable for the time of realization of the project was the library G4P of Peter Lager.

G4P is a complex library, which provides an extensive collection of GUI 2D controls for the sketch, in figure 3 we have an interface to create the platform visually, This library comes with many examples to show the use of these controls and on the creator page we can find information about commands by G4P.

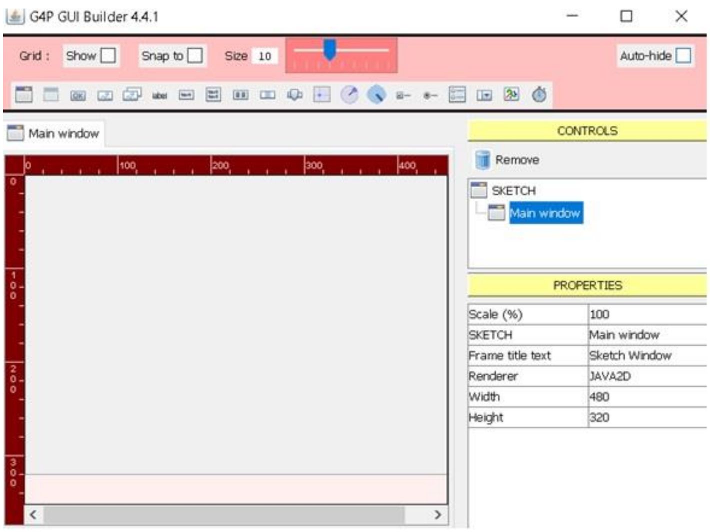

Figure 3. G4P interface for platform development

3.3. Construction Phase

In the construction phase of the platform was started by a first prototype where the elements were placed in a pleasant way, the purple color as a base and the yellow color was established when the action of clicking on the element is performed.

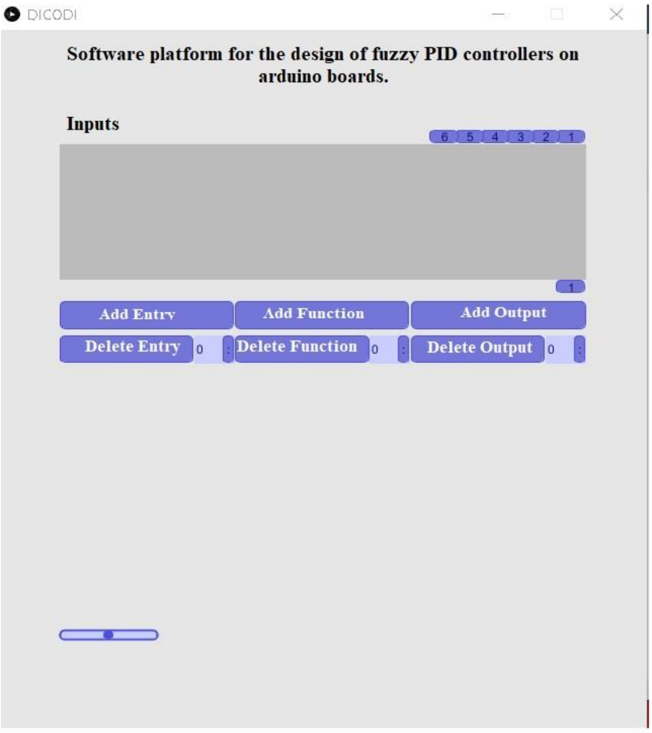

Figure 4. First prototype of the DICODI platform

Figure 4 visualizes the first prototype where it was learning and adapting to this new programming knowledge where when programming came certain ideas on some aspects of the platform such as:

- Inputs and outputs will not be named only number to facilitate programming.
- To make configurations in a membership function must be located, in the universe of speech to be modified and with an indicator the user will understand the location if it is an input or output with the number of this.
- The display box of the membership functions is not interactive like those of MATLAB, to make the modification you will have to select in its respective box the function or the type of change that will be made.
- To distinguish between an output or input a colour shall be used for each which is very different from each other.
- Deleting the input or output in which they are located will update the information displayed in the guide to one that is in stock
- The minimum number of inputs or outputs or functions of belongings is 1, this ensures that there is information to show in the box.
- The user will be able to change the names of the sets, parameters of the membership functions, the type of membership function and the range of the speech universe.
- Give the user the ability to choose the type of membership functions to add and change this in existing functions.

In this second prototype many visual improvements are presented compared to the previous one where only some elements were had, in this the functions of each of the buttons were built, labels, droplist, text boxes and were initialized in addition to including the ideas obtained when making the first prototype.

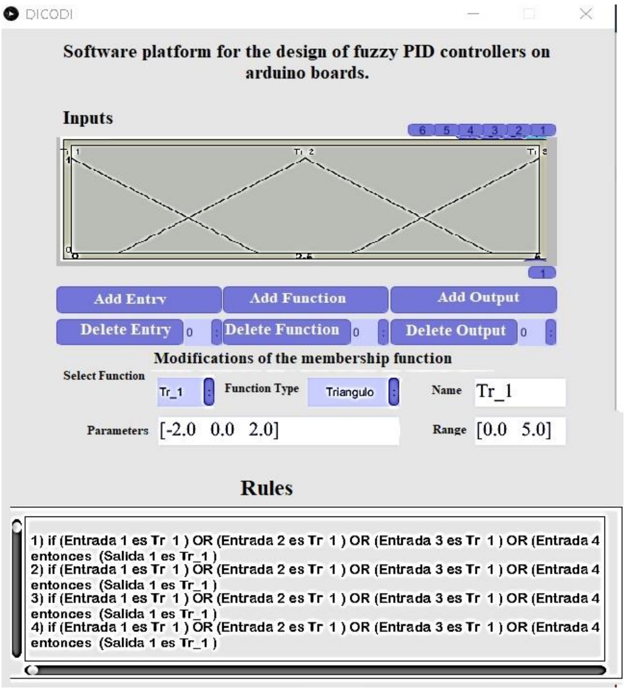

Figure 5. Second prototype of DICODI platform

Figure 5 is the second sketch in which small details were perfected that were found by testing the built functions and more details were found to perfect and updates were made the following:

- If the center of the transfer function is exceeded from the lower and upper limits of the universe, the limit that is exceeded shall be taken as the input value for fuzzily and at the maximum output to operate.
- Incorporation of interaction in the section of modification of membership functions and adjustment of data in the text box of parameters and range.
- Errors were identified when the text box was oversized.
- Two sliders were implemented so that the user can visualize the rules created by him and can identify the rule to be deleted, etc.
- The slider has motion effect on the rules and in the drop-down lists of the output and input.
- Added an option for the rule operation type ie OR or AND operation type.
- A calculate button to fuzzify and defuzzify a value according to established rules.
- At the bottom of the platform is an option to export the controller.
- In some computers, the platform could not be properly perceived, meaning that the viewing area was cut and could not be operated or used.

Figure 6 is the final version corrected the errors existing in the previous ones and added options so that the user knows what values the function that is modifying has.

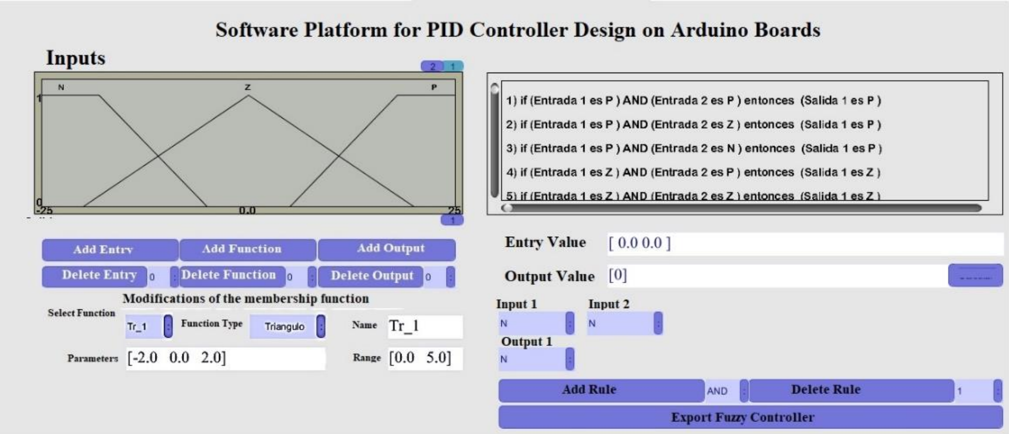

Figure 6. Final version of the DICODI platform

Other modifications to prototype number 2 were as follows:

- Expanded the boxes where parameters must be entered so that the user can enter slightly larger values and more input values when evaluating the fuzzy at a given value.
- It scrolled the mapping part, evaluating rules and exporting the code to the right side so that the user in case the screen resolution is lower than 1920x1080.
- The name box can exceed the limits since it has no verification of the data entered which the text boxes of the parameters, range and input value(s) have.
- The output value can be modified, although this does not have any action just show the result which is updated when you press the calculate button.
- At the time of entering the parameters where the limits are exceeded means that if the figure to be modified is a triangle and I enter a fourth data it will disappear when the text of the box is not edited. This same action is present for the range and value of inputs.
- When changing the limits of the range box when you stop editing this goes through a check that makes the comparison if the lower limit is lower than the upper one, if it is met it modifies the data of that entry otherwise restores the previous value it had before editing.

4. CODE DESIGN TO EXPORT TO ARDUINO

For the code design, figure 1 was taken as a reference to design each part of the fuzzy controller. The flow diagram in Figure 7 explains the process of fuzzizing inputs:

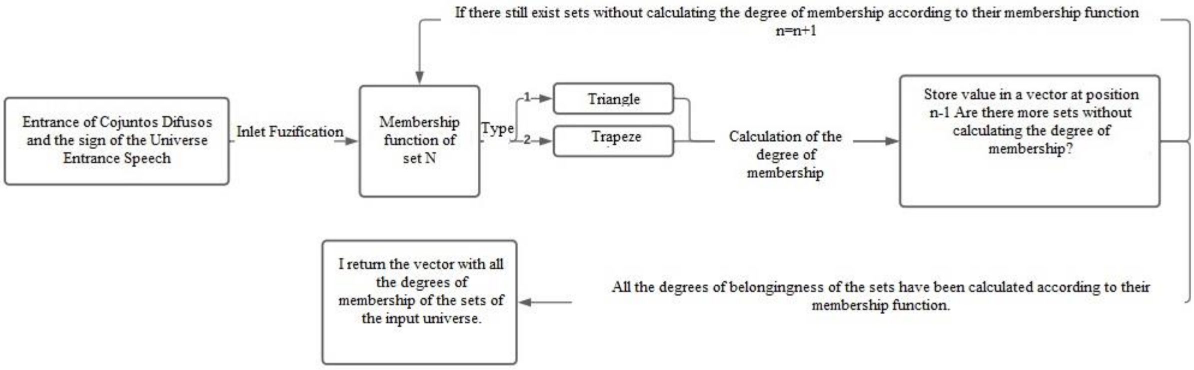

Figure 7. Input fuzzification flowchart

The controller knowledge base is declared by starting the program as global variables so that they can be used without any problems, in figure 8 the code can be seen

```
float E[][6]={
{1,2,-43.75,-27.07,-18.0,-5.0},
{1,1,-20.0,0.0,20.0,-25.0},
{1,2,5.0,18.0,27.08,43.78},
{2,2,-43.76,-27.08,-18.0,-5.0},
{2,1,-20.0,0.0,20.0,-25.0},
{2,2,5.0,18.0,27.0,43.69}};
float S[][6]={
{1,1,40.0,45.0,50.0,0.0},
{1,1,53.5,56.5,59.5,0.0},
{1,1,65.0,75.0,85.0,0.0}};
byte R[][100] ={
{1,3,3,3},
{1,3,2,3},
{1,3,1,3},
{1,2,3,2},
{1,2,2,2},
{1,2,1,2},
{1,1,3,1},
{1,1,2,1},
{1,1,1,1}};
byte CE[]={3,3},CS[]={3},Tam[]={2,1,9};
float LimE[2][2]={{-25.0,25.0},{-25.0,25.0}};
float x[]={Ingrese Entrada 1,Ingrese Entrada 2},OUTF[1];
float S_Temp[100][5],U_Temp[50];
```

Figure 8. Knowledge base exported from the DICODI platform

The flow diagram in Figure 9 shows the process used to evaluate rules and assign the degree of belonging to the outputs:

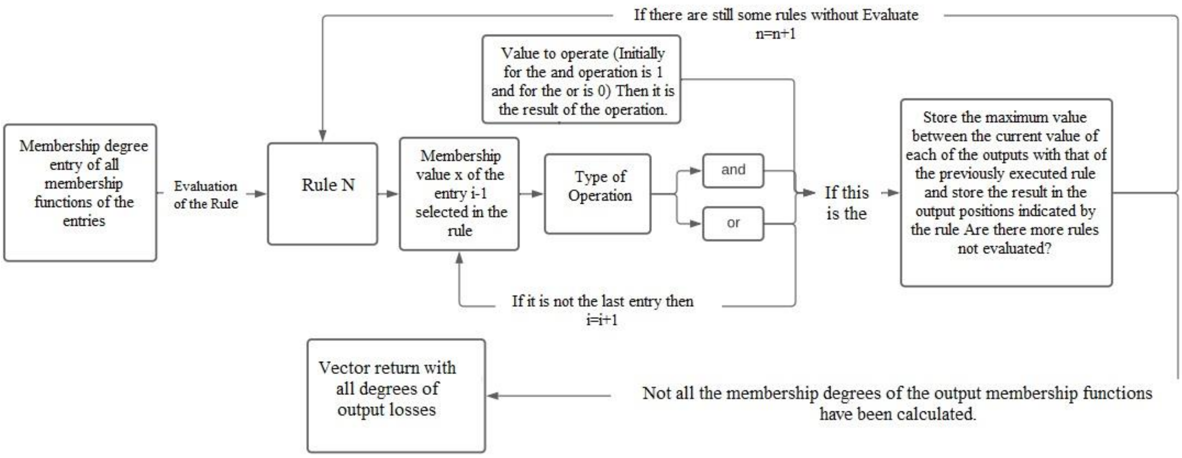

Figure 9. Flow diagram of the evaluation of rules and assignment of degree of belonging to the outputs

the flow diagram in Figure 10 explains the process that is carried out for the defuzzification of the outputs:

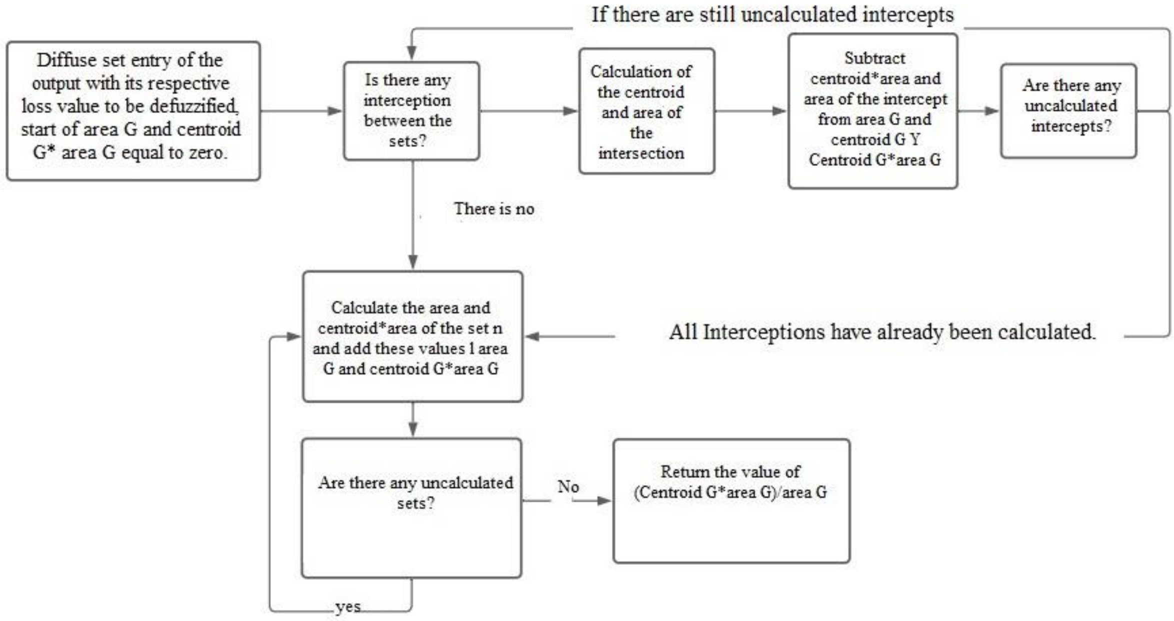

Figure 10. Output defuzzification to obtain a quantifiable value

In order to carry out a correct operation of the fuzzy controller all the different intercepts between the membership functions programmed in the.ino file were analyzed, where 6 types of intersections were found.

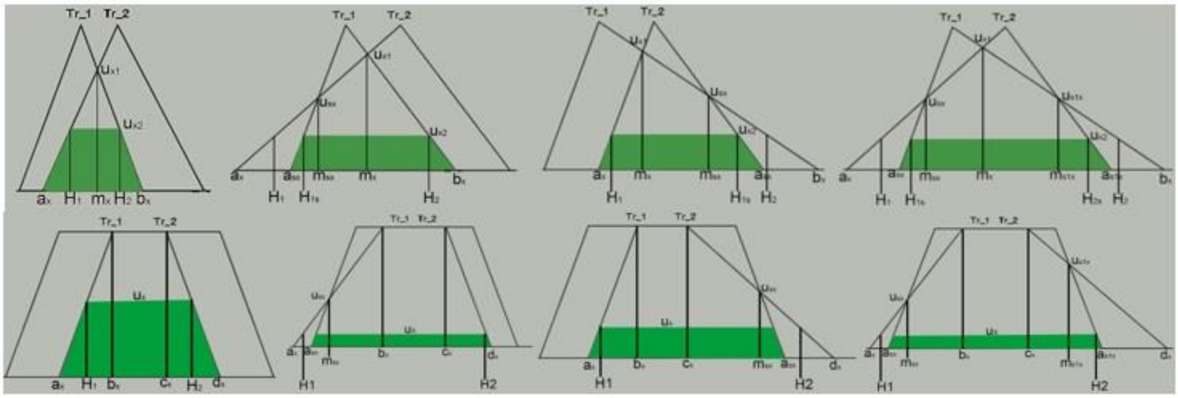

Figure 11. Cases according to the interception

The calculations for the defuzzification of the exits were made by formulas sectioning the figures formed by the types of intersections in Figure 11, that is without using a single method for calculating areas that have to store a certain number of points of the universe of speech with their respective degree of membership of the shaded figure.

The diffuse controller exported by the platform contains:

- Knowledge base (Exported by the platform)

- Mamdani mechanism (Exported by the platform)
- Input fuzzification (Functions contained in the Functions_Fuzzy.ino file) Output defuzzification (Functions contained in the Functions_Fuzzy.ino file)

4. RESULTS

To verify the correct execution of the algorithms designed, two tests were carried out, the first one took the Matlab program as a reference and the second one implemented the code exported by the platform in a generator engine plant.

4.1. Validation of the DICODI platform compared to the Matlab fuzzy logic toolbox tool

The Fuzzy Logic Toolbox provides MATLAB functions, apps, and a Simulink block for analyzing, designing, and simulating fuzzy logic-based systems. This toolbox allows you to model complex system behaviors using simple logical rules and then implement these rules into a fuzzy inference system.

Figure 12 shows the output universe and the rules applied to the controller to be compared with the platform (Figure 13).

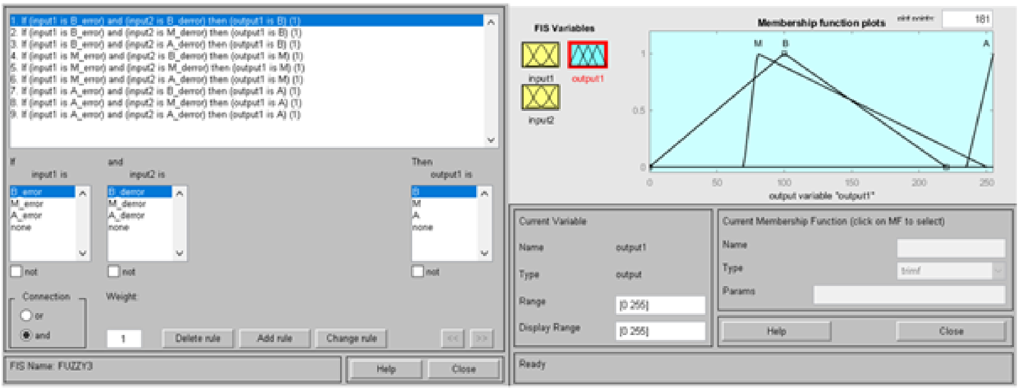

Figure 12. Rules and Fuzzy Controller Output

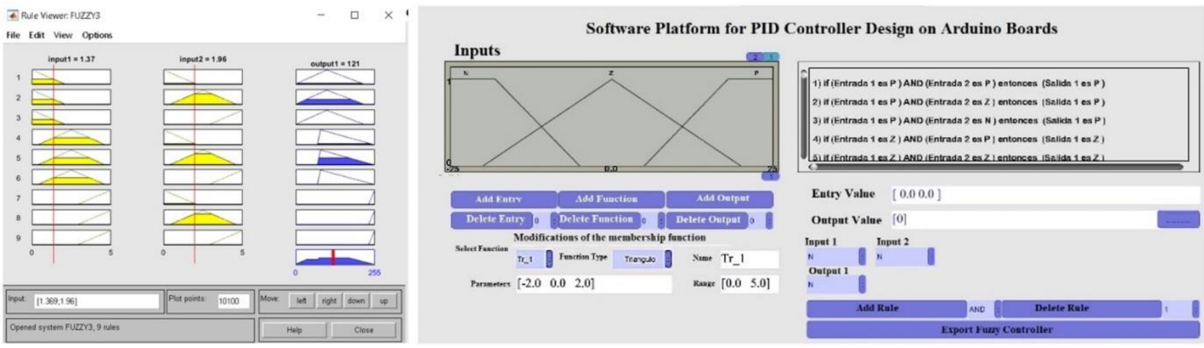

Figure 13. Comparing the defuzzified value

The error rate present between the platform and the reference tool is less than 0.5% so it is considered that the platform fulfills its main function.

4.2. Validation of the exported controller with the generator engine plant

For the design of the diffuse controller a classic controller was used where the ranges of the speech universes would be known and the sets, rules, membership functions, etc.

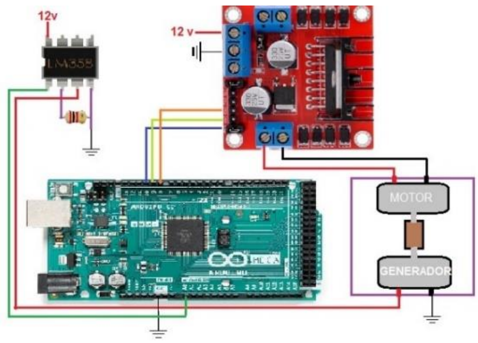

Figure 14. Schematic of the generator engine plant

From the design of the controller in the DICODI platform for the plant Figure 14, the basic knowledge of the controller and the instructions for its implementation were obtained in a text file as can be seen in the figure 15.

```
float E[][6]={
{1,2,-43.75,-27.07,-18.0,-5.0},
{1,1,-20.0,0.0,20.0,-25.0},
{1,2,5.0,18.0,27.08,43.78},
{2,2,-43.76,-27.08,-18.0,-5.0},
{2,1,-20.0,0.0,20.0,-25.0},
{2,2,5.0,18.0,27.0,43.69}};
float S[][6]={
{1,1,40.0,45.0,50.0,0.0},
{1,1,53.5,56.5,59.5,0.0},
{1,1,65.0,75.0,85.0,0.0}};
byte R[][100] ={
{1,3,3,3},
{1,3,2,3},
{1,3,1,3},
{1,2,3,2},
{1,2,2,2},
{1,2,1,2},
{1,1,3,1},
{1,1,2,1},
{1,1,1,1}};
byte CE[]={3,3},CS[]={3},Tam[]={2,1,9};
float LimE[2][2]={{-25.0,25.0},{-25.0,25.0}};
float x[]={Ingrese Entrada 1,Ingrese Entrada 2},OUTF[1];
float S_Temp[100][5],U_Temp[50];
void setup(){
float Uin[2][50],Uout[1][50];
Entrada(CE,x,E,LimE,Uin,Tam[0]);
Salida_Reglas(R,Uin,CE,CS,Uout,Tam);
byte con=0;
for(byte i=0;i<Tam[1];i++){
TemporalMatriz(S_Temp,S,con,con+CS[i]);
TemporalVector(U_Temp,Uout,i,CS[i]);
```

Figure 15. Knowledge base and instructions for implementation of the fuzzy controller

When implementing the fuzzy controller in the generator power plant the response was obtained from figure 16, where the controller managed to stabilize the setpoint without presenting on impulses.

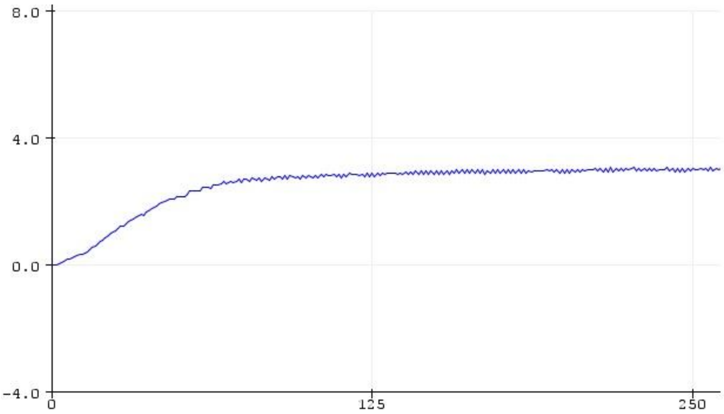

Figure 16. Response from the generator engine plant

The Arduino code used in the generator engine plant has a weight of approximately 60 KB which was reduced by optimizing the Arduino code (Electrodaddy), where the only Arduino capable of supporting the controller would be the Arduino Mega 2560. When adapting the code following the recommendations of the user manual it should be able to be implemented in an Arduino mega with flash memory less than 16 KB.

5. CONCLUSIONS

- The fuzzy controller code structure is very optimal, meaning that the algorithm does not waste time on unnecessary actions and maintains a very low error rate compared to the MATLAB logic toolbox.
- The DICODI platform is not the best platform for designing fuzzy controllers, its programming was not very efficient but this fulfils its purpose of serving to create fuzzy controllers.
- To create and test a platform takes a lot of time so some features that can commonly be seen in a platform were not possible to implement as is the saving of design information, but this is compensated with the quality of the exported fuzzy driver code.

REFERENCES

Manuel Delgado y Freddy Paz, “Sistemas de Lógica Difusa para la Evaluación de Usabilidad de Sitios Web de Gobierno Electrónico: Una Revisión Sistemática”, Revista Ibérica de Sistemas e Tecnologias de Informação, E41 (2021), <https://search.proquest.com/openview/8c5bbc92b4525a0fa4d531557d57ef95/1?pq-origsite=gscholar&cbl=1006393>

Pedro Ponce Cruz, Inteligencia Artificial con aplicaciones a la ingeniería. Primera Edición. México: Alfaomega Grupo Editor, 2010. 376

Samuel Sanahuja, “Sistemas de Control con Lógica Difusa: Métodos de Mamdani y de Takagi-Sugeno Kang (TSK)”, Repositorio de la universitat JAUME I, (2016), ¿<http://repositori.uji.es/xmlui/bitstream/handle/10234/173788/TFG_2017_Diciembre Sanahuja_Sa muel.pdf?sequence=1&isAllowed=y>

Arquímedes Ferrera, “Uso de un Sistema Difuso (FL) para el manejo de fallas basado en Mantenimiento Centrado en Confiabilidad (RCM)”, <https://predictiva21.com/sistema-difuso-fallas-mantenimiento-confiabilidad>

i Casals, Orestes Mas. Sistemas difusos dinámicos para el tratamiento de información temporal imprecisa. Barcelona, 1997, 242p. Trabajo de investigación (Dr. Ingeniero de Telecomunicación). Universitat Politècnica de Catalunya. Departament de Teoria del Senyal i Comunicacions

Füsun S. Tut Haklıdır “The importance of long-term well management in geothermal power systems using fuzzy control: A Western Anatolia (Turkey) case study”, Energy, 213 (2020), 118817 <https://doi.org/10.1016/j.energy.2020.118817>

Lu Li y Yinshi Li * “A feedforward-feedback hybrid control strategy towards ordered utilization of concentrating solar energy”, Renewable Energy, 154 (2020), 305-315 <https://doi.org/10.1016/j.renene.2020.03.004>

Sánchez Calderón y Yill Alejandra “Diseño e implementación de un controlador PID difuso para un robot UAV tipo Quadrotor”, Repositorio Universidad Piloto de Colombia Colección de Ingeniería Mecatrónica, (2015), <http://repository.unipiloto.edu.co/handle/20.500.12277/1059>

L.A.Zadeh “Fuzzy sets”, Information and Control, 8 (1965), 338-353< https://doi.org/10.1016/S0019-9958(65)90241-X>

WEBSITES

1) Matlab.Herramienta fuzzy logic toolbox https://es.mathworks.com/products/fuzzy-logic.html
2) Electrodaddy. “Optimización del código Arduino”, https://www.electrodaddy.com/optimizacion-del-codigo-arduino/
3) Processing, “Reference”, < https://processing.org/reference/>
4) G4P, “G4P (GUI for Processing)”, http://www.lagers.org.uk/g4p/ref/annotated.html