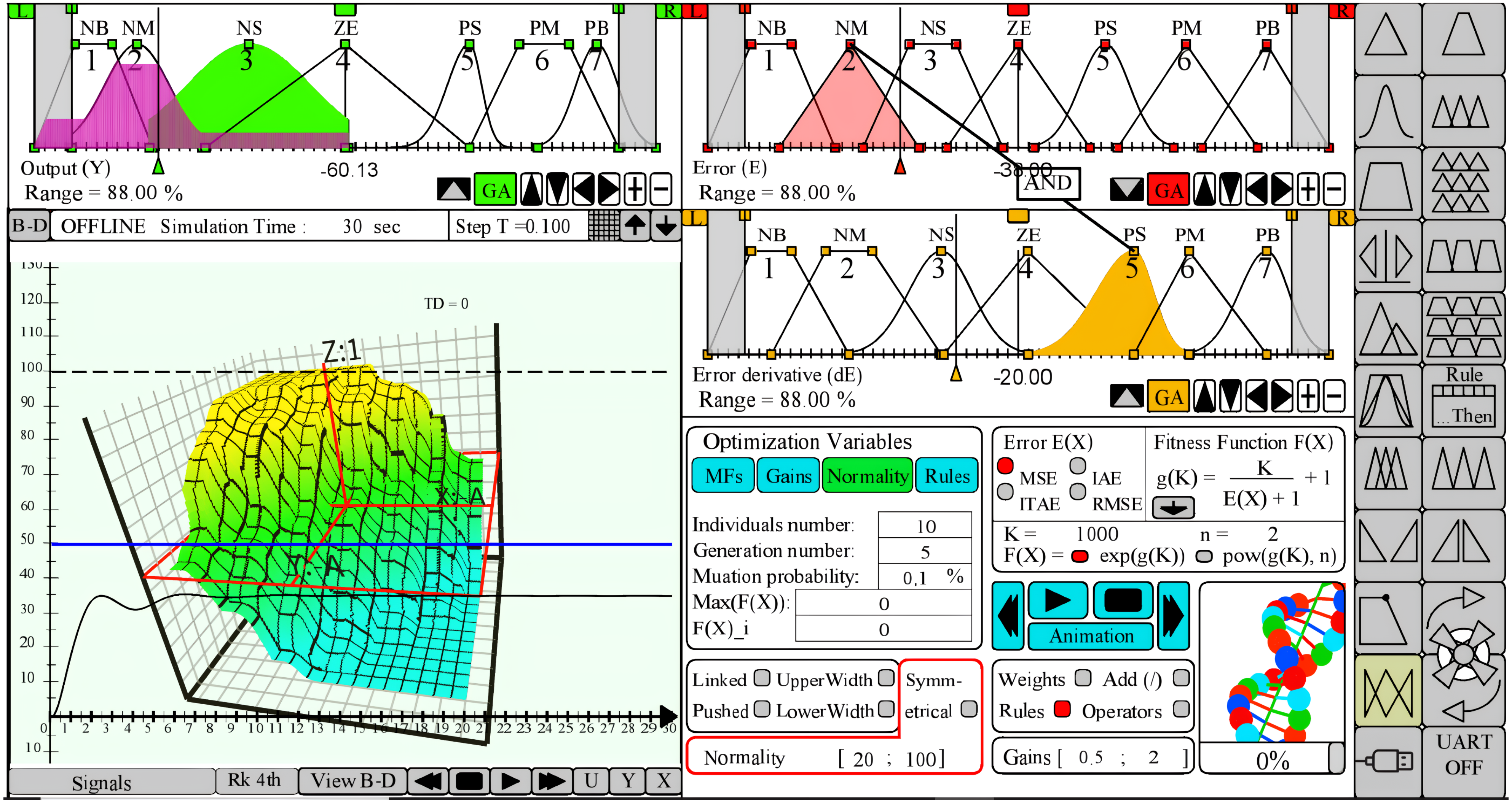
P4FPID: Quick Start Guide

August 27, 2025



NB NM NS ZE PS PM PB
Output (Y)
Range = 88.00 %
-60.13
GA
Error (E)
Range = 88.00 %
-30.00
AND
Error derivative (dE)
Range = 88.00 %
-20.00
B-D
OFFLINE Simulation Time : 30 sec
Step T =0.100
TD = 0
Z:1
X:A
Y:A
Signals
Rk 4th
View B-D
U
Y
X
Optimization Variables
MFs
Gains
Normality
Rules
Individuals number: 10
Generation number: 5
Muation probability: 0,1 %
Max(F(X)): 0
F(X)_i 0
Error E(X)
MSE
IAE
ITAE
RMSE
Fitness Function F(X)
g(K) = K / (E(X) + 1) + 1
K = 1000
n = 2
F(X) = exp(g(K)) pow(g(K), n)
Animation
0%
Linked
UpperWidth
Pushed
LowerWidth
Symm-
etrical
Normality [20 ; 100]
Weights
Add (/)
Rules
Operators
Gains [0.5 ; 2]
Rule
... Then
UART
OFF

Contents

List of Figures

List of Tables

Abbreviations

MF Membership Function

MV Membership Variable (contains a set of MFs)

MVB Membership Variable Box (the GUI box where MV is displayed)

FIS Fuzzy Inference System

B-D Block Diagram

R-K Runge–Kutta

GA Genetic Algorithm

1 Software Overview

P4FPID is software for fuzzy control training and education. The software has been developed using Processing 4.0. P4FPID is a real-time control, simulation, and animation tool dedicated to engineering and educational applications. It focuses on empirical experimentation, where human expertise in control systems is the primary actor.
This document is not peer-reviewed. To refer to the P4FPID software, the corresponding paper can be found in (.... *Paper citation; DOI:XXXX* .....).

2 System Requirements

- Operating System:
  - Windows.
- Hardware:
  - **RAM:** 400 MB free at maximum.
  - **CPU:** Quad-core of 1.6 GHZ base frequency or higher
  - **GPU:** Integrated GPUs are sufficient. A dedicated GPU running as the main renderer is **highly recommended** for higher frame rates.

3 Installation Instructions

P4FPID doesn't need to be installed. It runs on a third-party virtual machine. It requires the installation of the latest Java Development Kit (JDK) for the software to run. You can download it from `https://www.oracle.com/java/technologies/downloads/`. Just copy/paste the executable file onto the computer's hard disk and you are ready to use it.

4 Usage Instructions

Mouse Scrolling Wheel

The usage of the software is simple and direct. In order to better experience the advantages of P4FPID, it is recommended to use a computer mouse with a functioning mouse scroll wheel (not the touchpad in the case of a laptop). In the coming sections, we go step by step, covering all aspects of P4FPID.
Note: There is an inherent problem in Processing when invoking a second window (it shows no stability). Therefore, when importing a P4FPID file from a different location, make sure to check the navigation window on your computer's task view. It doesn't necessarily appear directly, depending on your machine and system.

4.1 Main Interface

Figure 1 shows a sample of a random MF distribution inside an MVB. MF manipulation is subject to a number of constraints set based on the reasonableness of the controller. The following paragraphs summarize the main actions the user can apply to MFs.

Tables 1 and 2 show the functionality of each button of the main interface.

4.1.1 Adding / Substruction

The Plus/Minus buttons in Figure 1 are designed to automatically generate a new MF distribution depending on the desired number between [3–9] (an odd number to maintain symmetry).
Important: The user must select the desired MF type first in order to enable button execution.

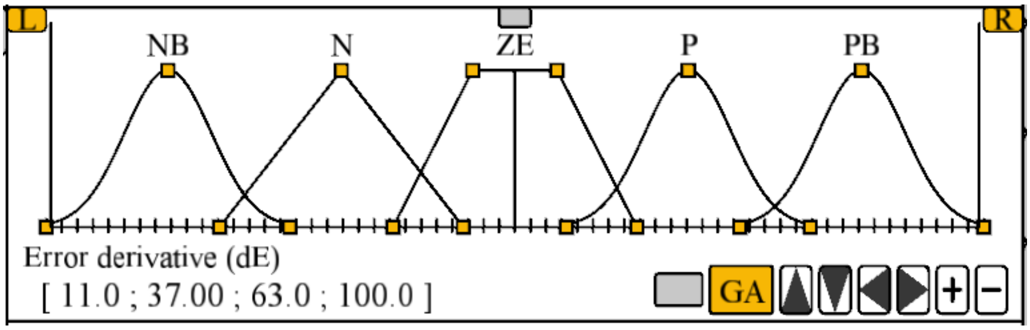


Figure 1: A sample of a Membership Variable Box (MVB)

Table 1: MF manipulation buttons (1)

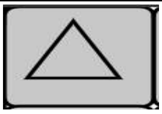
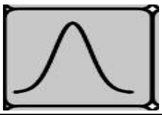
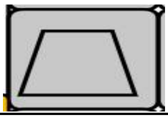
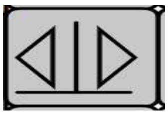
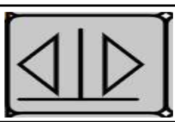
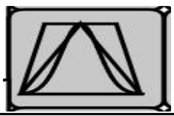
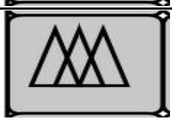
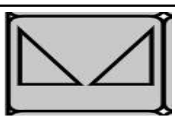
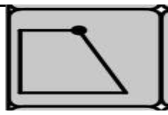
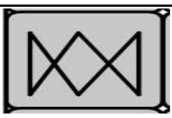
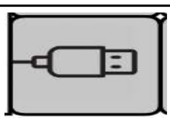
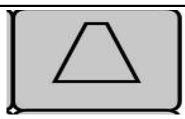
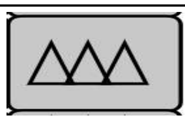
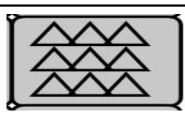
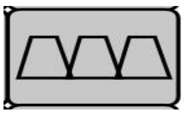
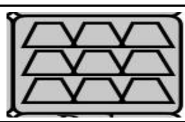
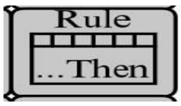
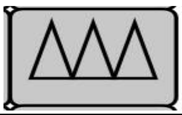
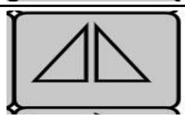
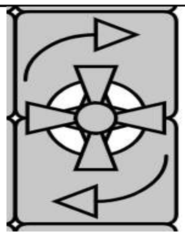
Button	Functionality
	Triangular membership function. **Activation:** direct click or mouse wheel roll inside a MVB. **Disactivation:** direct click or right mouse click anywhere in the interface.
	Gaussian membership function. **Activation:** direct click or mouse wheel roll inside a MVB. **Disactivation:** direct click or right mouse click anywhere in the interface.
	Trapoizadal membership function. **Activation:** direct click or mouse wheel roll inside a MVB. **Disactivation:** direct click or right mouse click anywhere in the interface.
	Symmetry option activation button. **Activation:** direct click. **Disactivation:** direct click.
	Membership function subnormality option. **Activation:** direct click. **Disactivation:** direct click.
	Generating random membership functions. **Activation:** direct click. It depends on the activated MVB to apply the concerned algorithm.
	Link to center side by side membership functions. **Activation:** direct click. *It depends on the activated MVB to apply the concerned algorithm.*
	Push ranges towared edged membership functions. **Activation:** direct click. *It depends on the activated MVB to apply the concerned algorithm.*
	Shrink / Extend all membership functions based on edged trapozidal MFs. **Activation:** direct click. *It must actived each time the user use this option in the concered MVB.*
	Active linkage of membership functions. **Activation:** direct click. **Disactivation:** direct click.
	Initialize Serial Communication. **Activation:** direct click. **Disactivation:** direct click.

Table 2: MF manipulation buttons (2)

Button	Functionality
	Enable upper width adjustment (only for Gaussian and trapozidal MF). **Activation:** direct click or key "w" or "W". **Disactivation:** direct click or right mouse click anywhere in the interface.
	Enable lower width adjustment of the concered MVB / MVB selction for applied action. **Activation:** direct click or key "a" or "A". **Disactivation:** direct click or right mouse click anywhere in the interface.
	Enable lower width adjustment of all MVB / all MVB selction for applied action. **Activation:** direct click or key keyCode "TAB". **Disactivation:** direct click or right mouse click anywhere in the interface.
	Enable upper width adjustment of the concered MVB / MVB selction for applied action. **Activation:** direct click or key "a" or "A" (highest periority). **Disactivation:** direct click or right mouse click anywhere in the interface.
	Enable upperwidth adjustment of all MVB / all MVB selction for applied action. **Activation:** direct click or keyCode "TAB" (highest periority). **Disactivation:** direct click or right mouse click anywhere in the interface.
	Enabling rule assignment when rules table editing is activated. **Activation:** direct click or key "r" or "R". . **Disactivation:** direct click or right mouse click anywhere in the interface.
	Link to edges side by side membership functions. **Activation:** direct click. *It depends on the activated MVB to apply the concerned algorithm.*
	Pushedged membership functions toward ranges. **Activation:** direct click. *It depends on the activated MVB to apply the concerned algorithm.*
	Jump to the next / previous interface. The drawn wheel animates mouse wheel actions which is used for: -changing upper / lower MFs widths. -selection the concerned MVB. -selection / assignment of the fuzzy rules (THEN.. rule). -shrinking, exteding of the X / Y axies, as well as, zoom IN / OUT of the 3D surface.
	Start / stop Serial Communication. **Activation:** direct click. **Disactivation:** direct click.

4.1.2 Extend / Shrink MFs Distribution

The Left/Right buttons in Figure 1 are designed to generate a new MF distribution depending on the degree of entanglement controlled by the left/right buttons. This degree remains fixed each time the user adds/subtracts MFs.
Important: The user must select the desired MF type first in order to enable button execution.

4.1.3 MFs Normality Distribution

The Up/Down buttons are designed to create distribution patterns toward the centered MF or the edge MFs equally.
Important: The user must select a random MF type first in order to enable button execution.

4.1.4 MV's Universe of Discourse Ranges

The button with grey color at the upper center of each MVB is designed to enable changing the range of the universe of discourse by dragging it from the MVB edges using the computer mouse.
Important: The MV range is limited to the first/last edge MF parameter.

4.1.5 Changing MF Shape/Type

The user selects the desired MF shape from the buttons in Figure 1, or by simply rolling the mouse wheel inside an MVB to choose the new MF shape, then directly assigns it to the corresponding MF or MFs (depending on the desired number or MVs).
Important: If the symmetry button is activated, the new shape will be automatically assigned to the symmetrical MF.

4.1.6 Changing MF/MFs Parameters

There are two ways to manipulate MF parameters: the precise way and the intuitive way, noting that both are subject to the previously mentioned constraints. The precise way involves clicking on the box where the parameters of the selected MF are displayed, then writing the new ones. The user must respect the required format to represent them as follows:

Case of Triangular and Gaussian MF:
[1^{st} parameter; 2^{nd} parameter; 3^{rd} parameter; Normality Degree %]

Case of Trapezoidal MF:
[1^{st} parameter; 2^{nd} parameter; 3^{rd} parameter; 4^{th} parameter; Normality Degree %]

1. The first intuitive way involves using the mouse to drag the MF's parameters and normality at the same time if needed. The dragging process is always subject to the mentioned constraints. To drag one parameter, left-click on the concerned one, then drag. To drag all parameters at once, right-click on one of the parameters, then drag. In addition, if the symmetry option is activated, the changes will be applied to the symmetrical MF according to its place among the side MFs with respect to the set constraints.
   Important: For the symmetry option, the symmetrical MF must have the same type in order to receive the intended symmetrical change of parameters.

2. The second intuitive way involves using the mouse wheel to change the MFs' upper/lower widths. The user must select the width type (upper or lower) & the number of MFs to which the change will be applied (see Figure 2), then scroll the mouse wheel.

4.1.7 Harnessing Built-in Algorithms

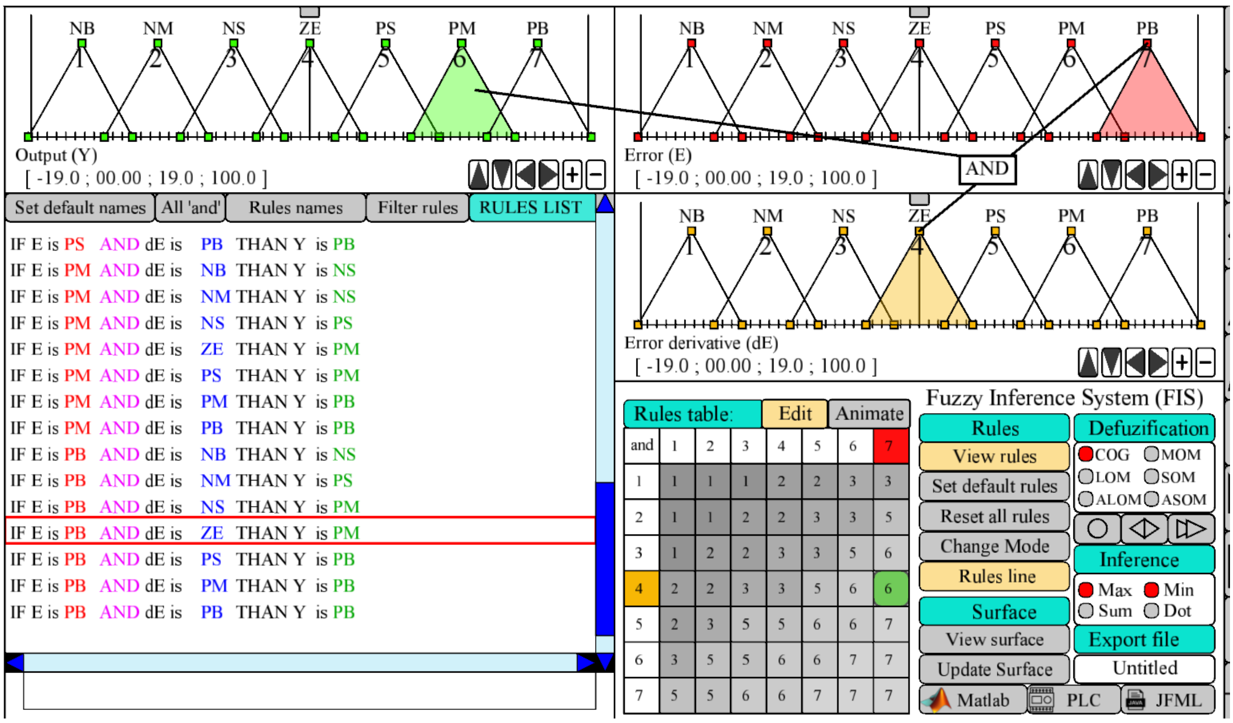
The user can hover the mouse over one of the built-in algorithm buttons in Figures 1 and 2, scroll the mouse to change the concerned MVB (or select all of them by activating buttons B3 or B5), then press the button to apply the desired action.

4.1.8 MF Optimization Using Genetic Algorithm

The buttons with the symbols "R" and "L" are responsible for enabling the optimization of the MFs on the right or left side, respectively. Also, the button with the symbol "GA" is responsible for enabling the optimization of the entire MV. Lastly, the button to the left of the "GA" button is for setting the normality distribution pattern: whether fully random (empty box), toward the centered MF (triangular box), or toward the edge MFs (two opposite triangles). Just click the button to change the mode each time.

4.2 FIS configuration

This section explains how the user configures the FIS parameters and fills in the rules table, as shown in Figure 2.



and	1	2	3	4	5	6	7
1	1	1	1	2	2	3	3
2	1	1	2	2	3	3	5
3	1	2	2	3	3	5	6
4	2	2	3	3	5	6	6
5	2	3	5	5	6	6	7
6	3	5	5	6	6	7	7
7	5	5	6	6	7	7	7

Figure 2: FIS parameters

4.2.1 Interface 1 Buttons and Their Descriptions

Table 3 explains the functionality of each button as follows:

4.2.2 Assigning / Changing Fuzzy Rules, Rules' Weights, and Rules' Connectors

Targeting a specific rule can be done by clicking directly on the concerned rule (selecting both relevant MFs of the rule), either in the rules table or in the rules list if it is shown. Alternatively, the mouse wheel can be used when the pointer is inside the rules list or the rules table.

To assign or change fuzzy rules, there are three available methods:

1. **The first method is using mouse clicks**. Click on the relevant rule by selecting the corresponding *Error* and *Error Derivative*, then choose the intended rule *system output*.

Table 3: FIS GUI Button & Functionality

Button	Functionality Description
`Animate`	Animates inputs/outputs of the fuzzy controllers.
`Edit`	Enables editing the FIS parameters — **must be enabled to continue with the rest**.
`View Rules`	Shows the rules list as pictured in Figure 2.
`Set default rules`	Sets the state-of-the-art default rules.
`Reset all rules`	Resets all rules to "Do Nothing", weights to 1, connectors to "and".
`Change mode`	Changes the table mode to Rules, Rule Weights, or Rule Connectors.
`Rules line`	Shows the rules line for better visualization.
`Set default names`	Sets default (conventional) membership function names.
`All and / All or`	Switches all connectors to "and" or "or".
`MFs names/numbers`	Displays membership functions by name or number.
`Filter Rules`	Displays only activated rules.

2. **The second method is using the mouse wheel**. After navigating to the intended rule using the mouse wheel, the user must activate the "IF Then" button in Figure 2, then scroll the mouse wheel to assign the desired fuzzy rule.
3. **The third method is using keyboard keys**. After selecting the target rule, press the rule number (as shown in Figure 2) to assign it automatically.

Changing rule weights or rule connectors is also done using the second method mentioned above (Method 2). By scrolling the mouse wheel, the rule weights will increase or decrease accordingly, or the rule connectors will be switched.

4.2.3 Defuzzification Methods, Inference / Aggregation Laws

P4FPID offers six defuzzification methods: COG (Center of Gravity), MoM (Mean of Maximums), LOM (Largest of Maximums), SOM (Smallest of Maximums), and two additional methods developed to handle symmetry in the universe of discourse (negative and positive values): ALOM (Absolute Largest of Maximums) and ASOM (Absolute Smallest of Maximums). Additionally, four inference methods are supported: max/min, max/prod, sum/min, and sum/prod. These can be activated with simple mouse clicks.

4.2.4 Rules and Rules Weights Flow Controllers

P4FPID offers shortcut buttons to control the flow of fuzzy rules listed in the tables below (Tables 4 and 5):

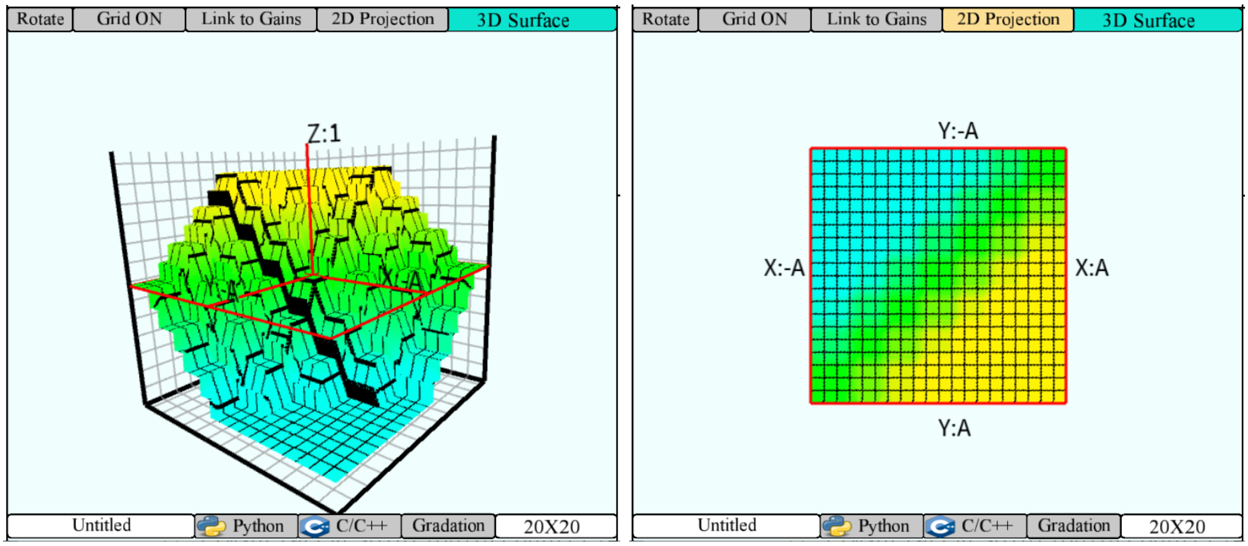
Table 4: Rules Buttons Action

Button	Rules Action
Left Button (with circle inside)	Gradually generalizes the "Zero" rule
Middle Button (with opposite triangles)	Gradually generalizes the rules between the zero and the edge rules
Right Button (with aligned triangles)	Gradually generalizes the edge rules

4.3 3D Surface

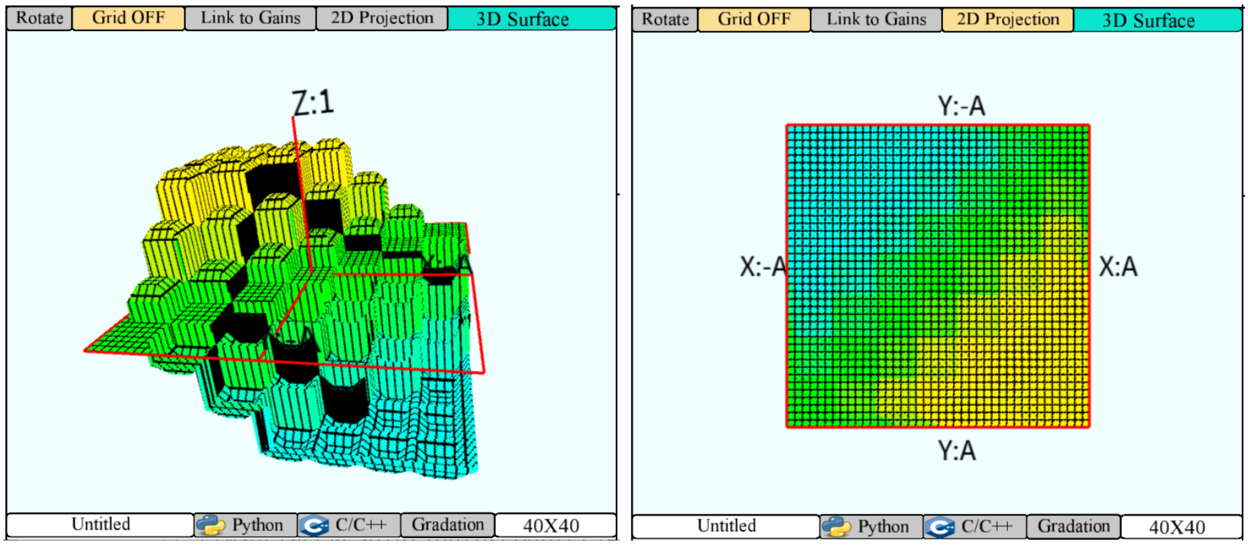
P4FPID offers a three-dimensional visualization and animation of the resulting output of the fuzzy controller. Figures 3 and 4 show two examples of the surfaces and their corresponding 2D projections, respectively, using two gradation levels for light hardware implementation.

Table 6 explains the functionality of each table in the 3D surface zone.



(a) (b)

Figure 3: 3D surface and 2D projection with a 20 × 20 gradation level



(a) (b)

Figure 4: 3D surface and 2D projection with a 40 × 40 gradation level

Table 5: Rules' Weights Buttons Action

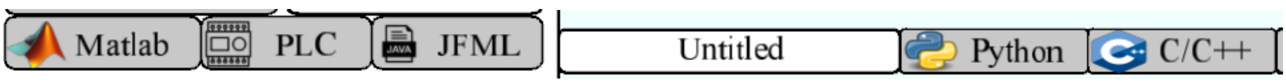
Button	**Rules' Weights Action**
Left Button (with circle inside)	Gradually reduces the other rules' weights, except for the Zero rule's weight.
Middle Button (with opposite triangles)	Gradually reduces the other rules' weights, except for the center rules' weights.
Right Button (with aligned triangles)	Gradually reduces the other rules' weights, except for the edge rules' weights.

Table 6: 3D Surface Buttons

Button	**Functionality**
Rotate	Rotates the 3D surface for better visualization.
Grid	Activates/deactivates the grid around the 3D surface.
Link to Gains	Highlights the corresponding parts of the 3D surface when using control gain factors.
2D Projection	Projects the surface onto a 2D frame.

4.4 Files Export

P4FPID generates three types of file formats for broader interoperability: '.fis' (MATLAB), IEEE-6013 (PLC), and JFML (XML) for the corresponding FIS parameters. In addition, it saves the 3D surface output for different programming languages (C/C++ and Python) in different resolutions (gradation levels) independently. See the concerned buttons in Figure 5.



(a) (b)

Figure 5: File formats storage buttons

Important
The user must first set the name of the file, then click on the desired format to be generated in independent folders inside the execution file directory.

It also saves both the FIS and the simulation B-D parameters in '.csv' table format, which can be invoked at any time. The user has two options. The first one is to save the intended simulation file as the **Default file**, which will be automatically invoked when the software is launched (the default file contains all details of the P4FPID file). The second one is saving the obtained file in the directory where the P4FPID is located. A time-print will be assigned to the file name to not confuse with files with the same name (See the concerned buttons in Figure 6).
Reversibly, the user can follow the data structure logic to make precise modifications to the FIS parameters.

When using real-time control, the software can save the recorded experimental data (by activating the "Record" button) in a '.csv' file, which can then be used by the data record visualizer or any other software.

Important
When activating the "Record" button, the user must first set the file name (using the same text box as for the FIS and simulation B-D parameters), then press the "Save Record" button for the data to be saved under the specified name. If this step is skipped, the data will be saved automatically with a timestamp to avoid overwriting previously recorded data. Therefore, the user can click the "Save Record" button multiple times without data confusion. The data will be saved under the specified folder name, as well as inside the directory of the data record visualizer, allowing it to be automatically invoked if needed.

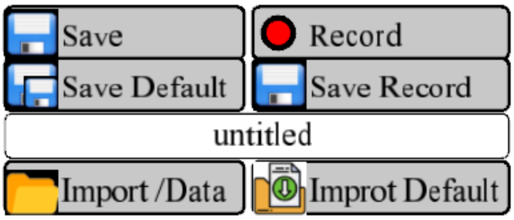


Figure 6: Data storage buttons

4.5 Graph Plotter

Figure 7 shows a screenshot of the graph plotter in P4FPID. Table 7 lists the functionalities of all buttons as follows:

Table 7: Graph Plotter Buttons

Button	**Functionality**
Grid	Activates/deactivates the grid.
Arrow Up / Down	Shifts the graph vertically.
B-D Button	Switches between control block diagrams.
Signals	Drop-down list to display the desired control signals.
RK	Drop-down list to choose the plotting algorithm (Runge-Kutta order).
Animate	Slows down / stops /starts / speeds up the simulation (animate the system output and the control signals).
View B-D	Displays the block diagram simulation.
X	Enables extending/shrinking the time axis using the mouse wheel (hold the "x" key).
X + (key 'x' or 'X')	Auto-adjustment of the time Axis (mouse click).
Y	Enables extending/shrinking the output axis using the mouse wheel (hold the "y" key).
Y + (key 'y' or 'Y')	Auto-adjustment of the Y Axis (mouse click).
U	Enables manipulation of the setpoint signals.
Plus	Adds a setpoint input.
Minus	Deletes a setpoint input.
Push	Pushes all setpoint edges to the end of the simulation time.
Reset	Unifies all setpoint signals into one setpoint.

4.6 Simulation Block

Figure 8 shows a screenshot of the simulation GUI alongside the simulation block diagram. The usage is clear, simple, and straightforward. Everything is separated into independent boxes: the box for entering the control gain factors, the transfer function box (direct typing or by .txt file import button), and the maximum saturation boxes. The user can either type the parameter values directly or set the step size and use the add/subtract buttons to dynamically change the simulation parameters.

4.7 GA Optimization Block

Figure 10 shows a screenshot of the optimization GUI. The user can generate more than 20 optimization scenarios, which are listed as follows:
-About Membership function variables.

1. **MVB:** Choose the MFV in which the user intends to optimize. There are 3 options: the error, the error derivative, and the output. All 3 options can be combined or just one or two of them.

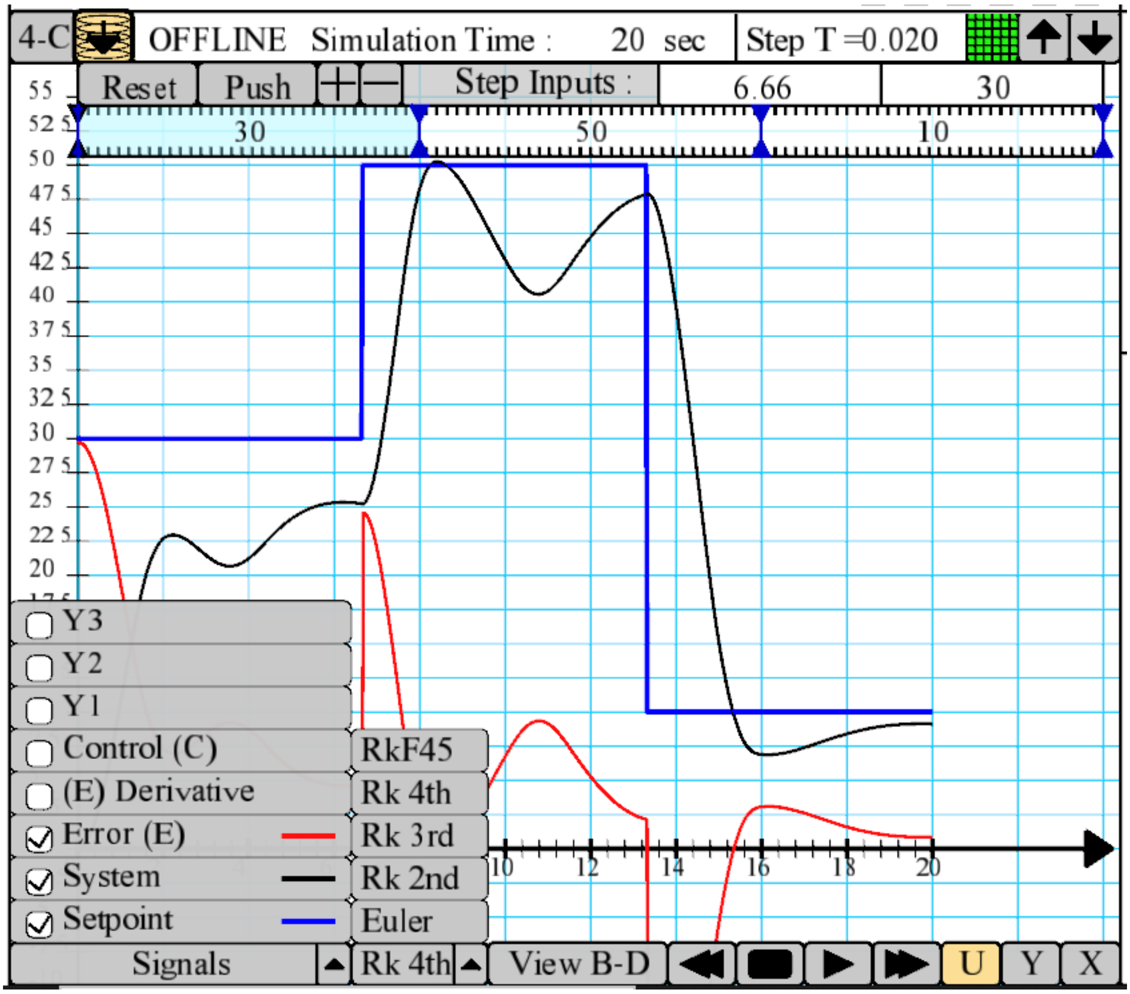


Figure 7: Graph Plotter

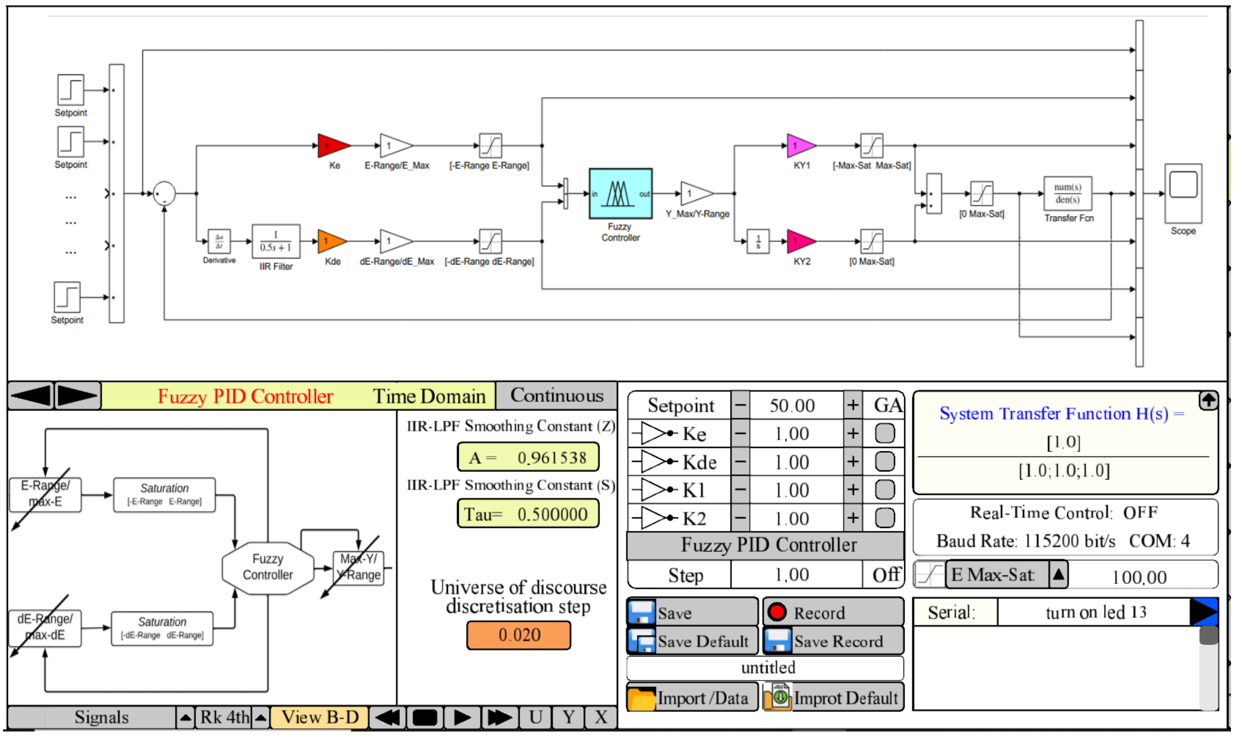


Figure 8: Simulation Interface

2. **MVB Side:** P4FPID offers the user to choose the side of the optimization to be the right or the left side. This option gives more control in case of a linearized system to be optimized according to the nonlinear slope if one side has been mastered.

-About Membership functions partition.

1. **Random MFs Partition** – This is the standard option. It randomly distributes the MFs across the universe of discourse.

2. **Linked MFs Partition** – This option links the MFs together to maintain a uniform, side-by-side pattern.

3. **Edged MFs Partition** – This option pushes the edge MFs toward the center of the defined MV range.

4. **MFs' Lower Width** – This option keeps the core of the MFs fixed while optimizing the lower parameters.

5. **MFs' Upper Width** – This option optimizes the core parameters of the MFs (only for Gaussian and trapezoidal MFs), while keeping the lower parameters fixed.

6. **Symmetrical/Asymmetrical MFs** – P4FPID provides the option to set symmetrical or asymmetrical MFs partition. This setting can be applied in conjunction with the previously mentioned options.

7. **Unified/Different MF Shapes** – All the above optimization options can be performed using either unified or different MF shapes. However, when the symmetry option is selected, all symmetrical MFs within the same MVB must share the same shape.

-About control gain factors. It offers the option to select the concerned control gain to optimize (4 options).

-About MFs' normality.

1. **Toward Center NW Distribution** – This option pushes the MFs' normality weights toward the centered MFs by gradually reducing the edge MFs' normality weights and amplifying the centered ones.

2. **Toward Edges NW Distribution** – This option pushes the MFs' normality weights toward the edge MFs by gradually reducing the centered MFs' normality weights and amplifying the near-edge and edge ones.

3. **Symmetrical/Asymmetrical Normality Distribution** – P4FPID provides the option to set symmetrical or asymmetrical MFs' normality weights. This setting can be applied in conjunction with the previously mentioned options.

-About Rues Table.

1. **Toward the Center distribution** – This option pushes the MFs' normality weights toward the centered MFs by gradually reducing the edge MFs' normality weights and amplifying the centered ones.

2. **Toward the Edges Distribution** – This option pushes the MFs' normality weights toward the edge MFs by gradually reducing the centered MFs' normality weights and amplifying the near-edge and edge ones.

3. **Symmetrical/Asymmetrical Normality Distribution** – P4FPID provides the option to set a symmetrical or asymmetrical distribution for rules, rule weights, and connectors.

Important

For the MF optimization variable, the symmetry option can only be enabled if all MFs have corresponding symmetrical MFs of the same type (shape). The usage sequence is as follows:

1. Choose the optimization variable (MFs, control gain factors, MF normality, or fuzzy rules).
2. Configure the optimization parameters:
   - Number of generations
   - Number of individuals (maximum 200)
   - Mutation probability (%)
   - Error type
   - Fitness function amplification method and gains
   - Crossover style (fixed or random), and inclusion of the Elitism option by clicking on the green button beside the progress percentage, as shown in Figure 9.

Figure 9: GA Optimization Parameters Configuration

3. Choose the sub-parameters/configuration of the selected variable.
4. Choose the optimization mode (background or animation) by activating/deactivating the animation button.
5. Press the Start/Stop button to begin or stop the optimization.
6. Increase or decrease the optimization speed when the animation mode is activated. The animated DNA reflects the current optimization speed.
7. Observe the convergence or divergence of the optimization process after each jump or drop in the fitness function.
8. Observe the optimization animation process for each variable in its corresponding GUI.

4.8 Real-time Control

P4FPID can be used for real-time control in a straightforward manner. The implemented communication protocol is UART (Universal Asynchronous Receiver Transmitter). The user must activate this option before launching the software by modifying the "Com.csv" file—switch the COM state from 0 to 1 and set the common Baud Rate (which must match that of the other communicating device in order to function properly).

The user can fully perform all adjustments to the FIS, Block Diagram, and Graph Plotter parameters in real time. The following code is an example of how to link P4FPID with an Arduino board and conduct a real-time experiment.

```
// Variables to store received data
String receivedPacket = "";
String receivedCommand = "";
bool sendData = false;
bool receivingPacket = false;
float controlSignal;
```

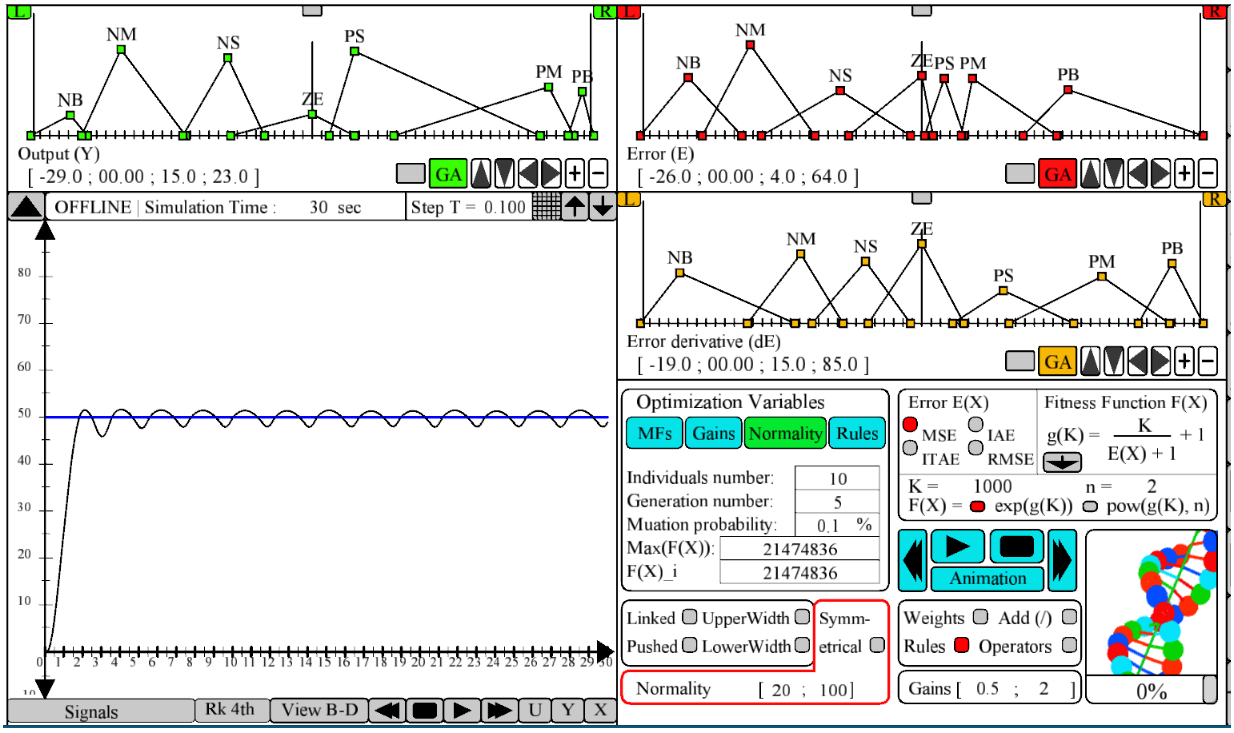


Figure 10: GA Optimization GUI

```

void setup() {
  Serial.begin(115200);  // Initialize serial communication at 9600 bps
  pinMode(13, 1);
  digitalWrite(13, 0);
}

void loop() {
  // Only send data if sendData is true
  if (sendData) {
    float SensorInput = map(analogRead(0), 0, 1023, 0, 100);
    String sentPacket = "(" + String(SensorInput) + ")";
    int length = sentPacket.length();
    char charArray[length + 1];
    sentPacket.toCharArray(charArray, length + 1);
    Serial.write(charArray);
    delay(100);  // Adjust delay as needed
  }
}

// This function is called automatically when serial data is available
void serialEvent() {
  while (Serial.available()) {
    char incomingChar = Serial.read();

    // Check for the start of a packet
    if (incomingChar == '(') {
      receivingPacket = true;  // Start accumulating data
      receivedPacket = "";     // Reset packet
    }

    // Accumulate characters if within a packet
    if (receivingPacket) {
      receivedPacket += incomingChar;

      // Stop accumulating and process the packet when ')' is detected
      if (incomingChar == ')') {
        receivingPacket = false;        // End packet accumulation
```

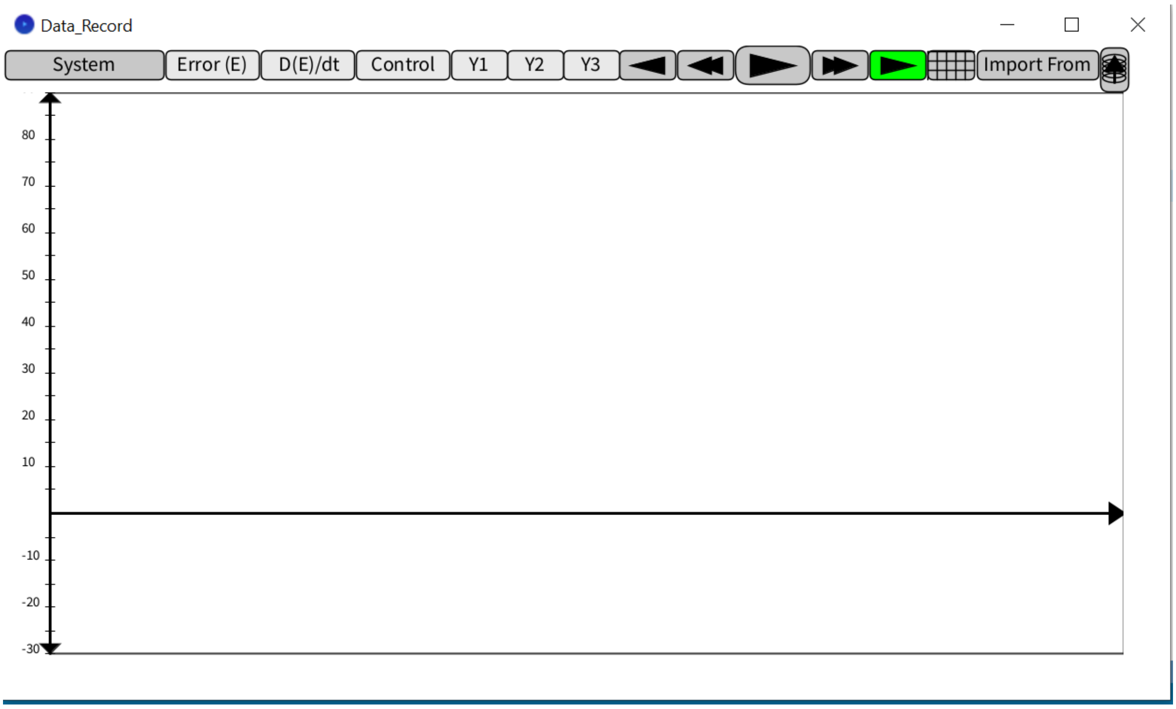


Figure 11: Recorded Data Visualizer

```
        processPacket(receivedPacket);  // Process the complete packet
        receivedPacket = "";            // Reset for the next packet
      }
    }
  }
  if (receivedCommand == "turn on led 13")
    digitalWrite(13, 1);
  else if (receivedCommand == "turn off led 13")
    digitalWrite(13, 0);
}

// Function to process received packet
void processPacket(String packet) {
  // Ensure the packet format starts with '(' and ends with ')'
  if (packet.startsWith("(start") && packet.endsWith(")")) {
    sendData = true;  // Start sending data
  } else if (packet.startsWith("(stop") && packet.endsWith(")")) {
    sendData = false;  // Stop sending data
  } else if (packet.startsWith("(") && packet.endsWith(")")) {
    String input = packet.substring(1, packet.length() - 1);
    if (input.startsWith("<") && input.endsWith(">"))
      receivedCommand = input.substring(1, input.length() - 1);
    else
      controlSignal = input.toFloat();
  }
  // controlSignal
}
```

The user can record the experiment and review it using the data visualizer toolbox, as shown in Figure 11 below.

Important

The user must not change the directory structure of the software files in order for the data visualizer to run successfully.

5 License and Terms of Use

This toolbox is provided under the GNU General Public License v3.0.

6 Contact and Support

P4FPID is an opened project for collaboration. For bug reports or collaboration requests, please contact us:

- Email: ....

- Website: ....

A dedicated YouTube Channel of usage tutorial and Practical work examples is found in this link (..........). Paper reference (............ Doi: ....).