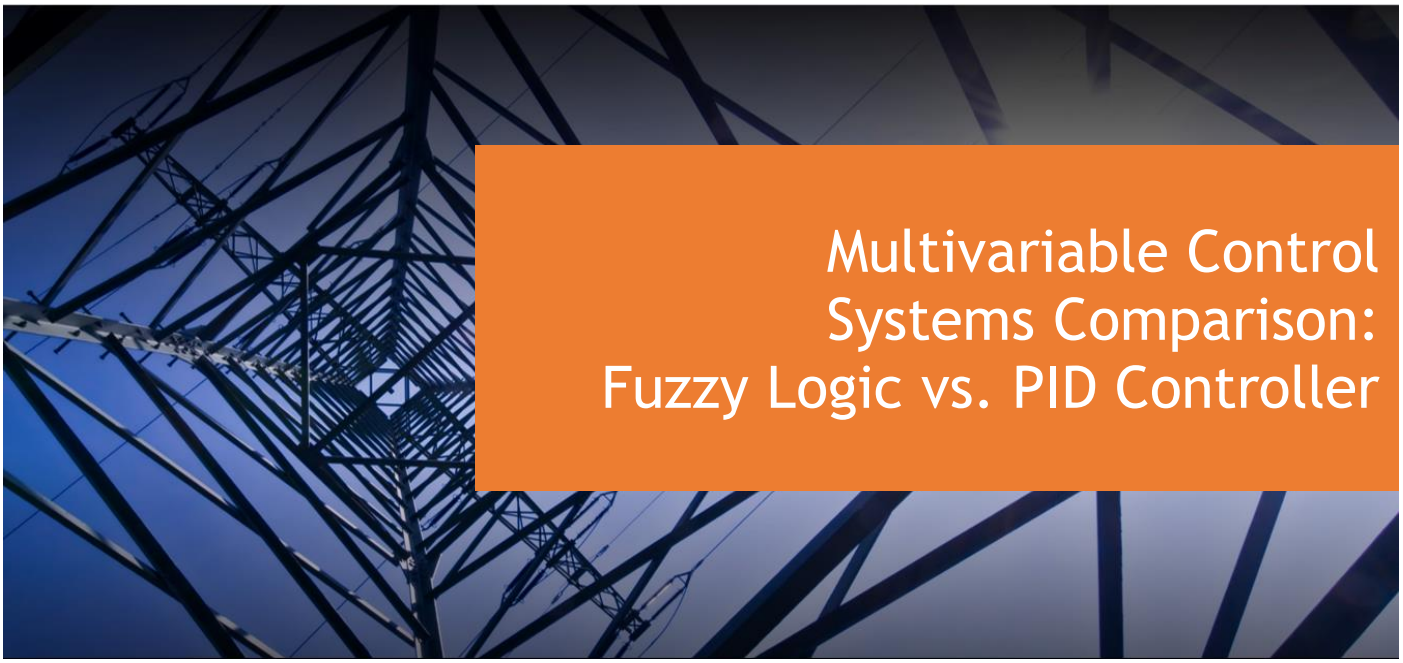


Multivariable Control Systems Comparison: Fuzzy Logic vs. PID Controller

RESEARCH TEAM: Rutul Bhavsar
Patrick McInnis
Laith Al-Musawi
Mariano Arriaga

mohawk IDEAWORKS
ENERGY & POWER INNOVATION CENTRE

FUNDING AGENCY: Ontario Centres of Excellence

REPORT #: EPIC-EP_20-01/2

PREPARED FOR: Xiera Technologies Inc.

Xiera Intelligence in Decision Making

PROJECT PARTNERS / INDUSTRY SUPPORT:

Xiera Technologies Inc.:

Talib Janabi
Alain Marsman

CENNA Engineering:

Mehak Ghawar

Abstract

The objective of this report is to setup a Multi-variable Control System test to compare the capabilities of a conventional PID controller vs. a novel fuzzy logic-based controller (FLC) developed by Xiera Technologies Inc. The selected test cases consist of 2x2 and 3x3 input/output control systems representing industrial processes. The controller is setup and analyzed testing three scenarios. First, a baseline scenario; followed by a disturbance signal scenario, where the feedback signal for individual controllers is altered to measure the closed loop response. Finally, a noise scenario that measures the impact of a Gaussian distribution signal in the feedback loop is presented.

For both systems studied, the FLC system presented a significant improvement in response (settling) time when compared to the PID controller system. Additionally, depending on the system under study, the FLC system also showed improvement by reducing the effect of disturbances and interaction in the system (3x3 case), and reducing the noise signal impact in the controller output (2x2 case).

Table of Contents

1 Motivation

For decades, the conventional Proportional-Integral-Derivative (PID) controller has been used to control industrial processes mostly due to its relatively simple implementation and reliability. However, real industrial processes are frequently complex, have a non-linear behaviour, and/or require multiple control inputs. From a control perspective, these characteristics can represent implementation challenges in the mathematical model representation and in the PID tuning process [1].

On the other hand, intelligent control strategies, such as Fuzzy Logic Control (FLC), can deal with the complexity and non-linear nature of advanced industrial processes. Hence, it can potentially increase process efficiency and economic savings. This advantage is based on the core FLC idea of control action expressed in terms of human operator experience achieving smooth interconnection between distinct control outputs [2]. Yet, FLC parameters (knowledge-base) consist of a number of rules, Membership Functions (MFs), and gains that considerably increase with the level of complexity of the problem and the number of variables to control. As a result, one of the main challenges of FLC implementation is the complexity and time-consuming process of generating and tuning FLC rules, MFs and gains, which are generally tuned manually.

In this work Mohawk College, in collaboration with Xiera, has created a validation platform aimed at investigating the capability of Xiera's FLC auto-tuner, and to compare its performance to a conventional PID controller. Here, both controllers were used to control industrial processes considering Multi-variable Control Systems (MVCS). The FLC parameters were auto-tuned using Xiera's software platform and compared to a PID controller auto-tuned using the MATLAB PID tuner software.

2 Background

2.1 Multivariable Control Systems

Multi-input multi-output (MIMO) control systems are complex to design, as they integrate multiple sensor data aimed to coordinate multiple actuators. In particular, the level of complexity escalates as the crosstalk (interaction) between sensors and actuators increases. In addition, it is worth pointing out that time delays and nonlinear behaviour further complicates the practical implementation of MVCS. For such reasons in most cases, MIMO controllers are custom designed for their application.

From a control perspective, a multivariable system aims to obtain desirable behaviour of several output variables by simultaneously manipulating several input channels. In general, the approach to design a MIMO controller has several requirements:

- That although in most cases, the interaction among different control loops cannot be eliminated; however, control designers aim to minimize this interaction.
- Minimize any nonlinear behaviour in the system.
- Improve the control system by modifying the control variable responses.

One technique used to design MIMO systems is based on integrating two (or more) Single-Input/Single-Output (SISO) controllers together by including transfer functions that model the interaction between the different variables. Figure 1 presents a common approach for a MIMO system implementation, in this case a 2x2 input/output system, which they seek to regulate while also accounting for their interaction. Sometimes the control structure uses a set of 2x2 set of decoupling gains placed after the controllers whose function is to minimize the interaction between the loops.

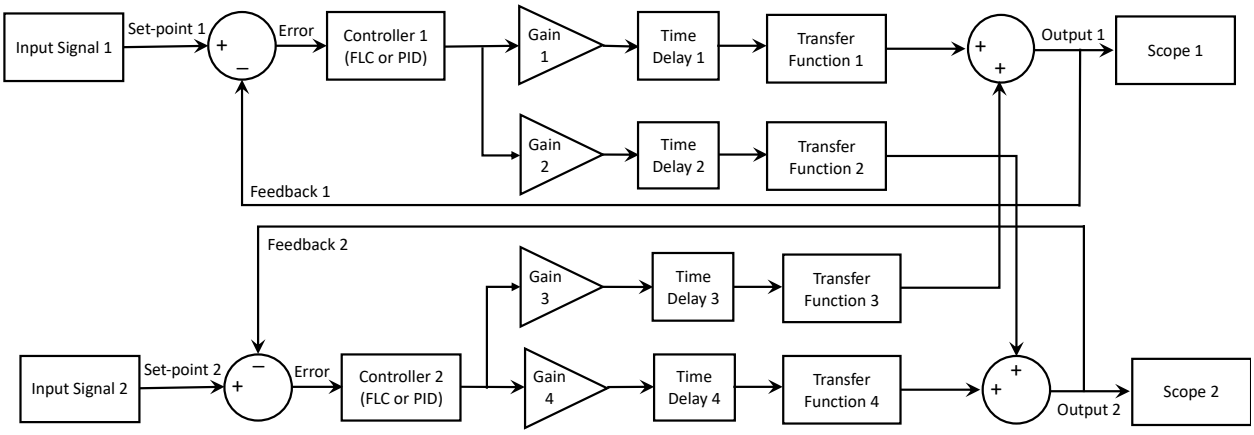


Figure 1: 2X2 MIMO model block diagram

2.2 Proportional-Integral-Derivative Control

PID controllers have been widely used in industry for almost a century [3]; it is estimated that 95% of the industrial controllers in the market are PID-type because of their simplicity, clear functionality, applicability and ease of use [4]. Their use extends from simple single variable to multi-variable control systems; including motor drives, automotive & flight control, as well as industrial process control.

One of the main advantages of PID controller is its low number of tuning parameters (proportional, integral, and derivative), which for simple SISO systems, can be relatively easy to tune manually and/or automatically. Auto-tuning processes are common on control systems software; in the case of this study, MATLAB/Simulink PID tuner was used. The PID tuner, computes a linear plant model from the Simulink model and designs an initial controller aimed to increase the stability of the system [5]. Furthermore, the PID parameters can be manually modified to improve the response time and transient behaviour of the system.

The tuning process for MIMO systems is in most cases complex regardless of the tuning platform being used. For example and relevant to this project, the multivariable tuning process can be performed in MATLAB; however the process is sensitive to nonlinearities that when considering an interdependent process (Figure 1), the control parameters are very difficult to auto-tune. MATLAB does provide a tool to minimize the nonlinearity issue by creating a linear plant model using its Control System Toolbox©. This system can be used together with a Simulink model to tune the compensator parameters using an interactive technique [6]. Nevertheless, the auto-tuning process requires further analysis and competing objectives (e.g., reference tracking, disturbance rejection, and stability margins) that will affect the overall system performance.

2.3 Fuzzy Logic Control

Fuzzy logic, a branch of artificial intelligence, was proposed by Lotfi A. Zadeh of the University of California at Berkeley in 1965. The initial FLC applications date from 1974 and focused on controlling a steam engine model industrial plant [7]. By early nineties, Japan started to commercially use FLC technology for home appliance control [2]. In recent years, FLC has permeated different commercial products ranging from washing machines, video camera, air conditioning, and automobile anti-lock brake systems [8].

Despite system complexity and vagueness of situations, human operators can make decisions by employing heuristics in the form of linguistic control rules. The concept of fuzzy reasoning expresses such linguistic rules in a rigorous mathematical framework. Fuzzy logic is developed to handle the fuzziness found in human concepts such as those embedded in the knowledge base of an intelligent system, as well as being able to build a framework that can handle linguistic quantifiers such as most, very, somewhat etc. In addition, fuzzy logic can address

uncertainties associated with data inaccuracy and process complexity which are problems encountered in industrial applications [9].

Fuzzy logic has yet to be embraced by industry as part of a multivariable system approach. To the best of the authors' knowledge, there are very few reported cases of an industrial multivariable system application using fuzzy logic. This is likely due to the historical complexity of implementing an FLC from an industrial perspective. One documented case was implemented decades ago in a rotary cement kiln in Denmark, achieving better performance when compared to conventional PID control, particularly when tested under noise conditions [10].

2.3.1 Fuzzy Controller Structure

An FLC has three basic blocks (Figure 2):

- Fuzzifier block that maps the measurement signals into fuzzy terms (e.g., high, low etc.),
- Inference engine, or approximate reasoning mechanism, which deduces the control actions in the form of fuzzy terms, and
- Defuzzifier block that translates the fuzzy control actions into crisp output control signals.

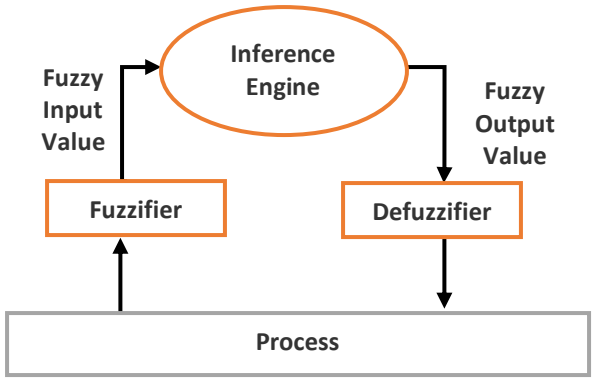


Figure 2: Fuzzy logic controller cycle

2.3.2 The Tuning Problem

There are two problems when fuzzy logic is applied to a real system:

- The definition of the set of rules, part of the inference engine, is a complicated process that depends on expert knowledge and there is no formal procedure to determine the parameters of the fuzzy system [11].

- The inherent complexity of tuning of the parameters associated to the MFs, FLC rules and gains.

When the rules are set properly and parameters tuned correctly, an FLC system can yield a high-performance controller even for complex systems [12]. The problem is that tuning fuzzy controllers is a non-trivial task. At present, it is performed manually by fuzzy logic experts and, even for a simple application, can take a significant amount of time due to the large number of parameters of the knowledge base and the related fuzzy rules, MFs and gains. Due to this tuning complexity, applications of fuzzy logic have been limited to relatively simple systems, such as home appliance products, and very limited application in large and complex industrial systems, to date.

2.4 Xiera's edeX Platform and Auto-Tuner System

2.4.1 edeX Platform

In recent years, Xiera Technologies Inc. (Xiera) has developed an intelligent auto-tuner addressing the time consuming and complexity issue of FLC knowledge-base parameter tuning. The auto-tuner is embedded as part of a hardware and software FLC platform, edeX (Figure 3); targeted to simplify and expand the implementation of FLCs in complex control systems; while addressing the time consuming and complexity issue of FLC knowledge-base tuning.

Xiera's edeX platform is an interactive fuzzy logic design and development environment intended to offer flexibility and ease-of-use for FLC design and tuning. The platform enables users to model and simulate the target process; as well as designing and testing the FLC system with the aid of an auto-tuner application. In addition, the edeX platform supports the FLC validation process by providing data acquisition capabilities when implementing the controller on a real system. edeX is also designed with capabilities to link to third party hardware systems, such as custom-design boards and off the shelf single board controllers (SBCs).
Based on Xiera's experience, edeX can:

- Handle MIMO applications,
- Combine fuzzy and conventional controllers in one control system,
- Design and configure fuzzy and/or conventional controllers,
- Run unlimited number of separate fuzzy/conventional controllers on one microcontroller chip, only limited by the chip memory, and
- Handle 32 configurable I/O's.

The hardware component of the edeX platform consists of:

- The fuzzy controller board (Xiera Controller Module or XCM), which carries the microcontroller programmed with the fuzzy algorithms (Figure 4 [a]).
- Up to eight (8) connectable Evaluation Boards (EVBs), each with modular analog Input/output (IO) capability (Figure 4 [b]).
- The XCM features a USB interface to the edeX software, for:
    - Data-acquisition,
    - Downloading controller parameters to the XCM's microcontroller,
    - Validating the control loop during development.
- A second serial interface to the XCM used to communicate with EVB or third-party host.

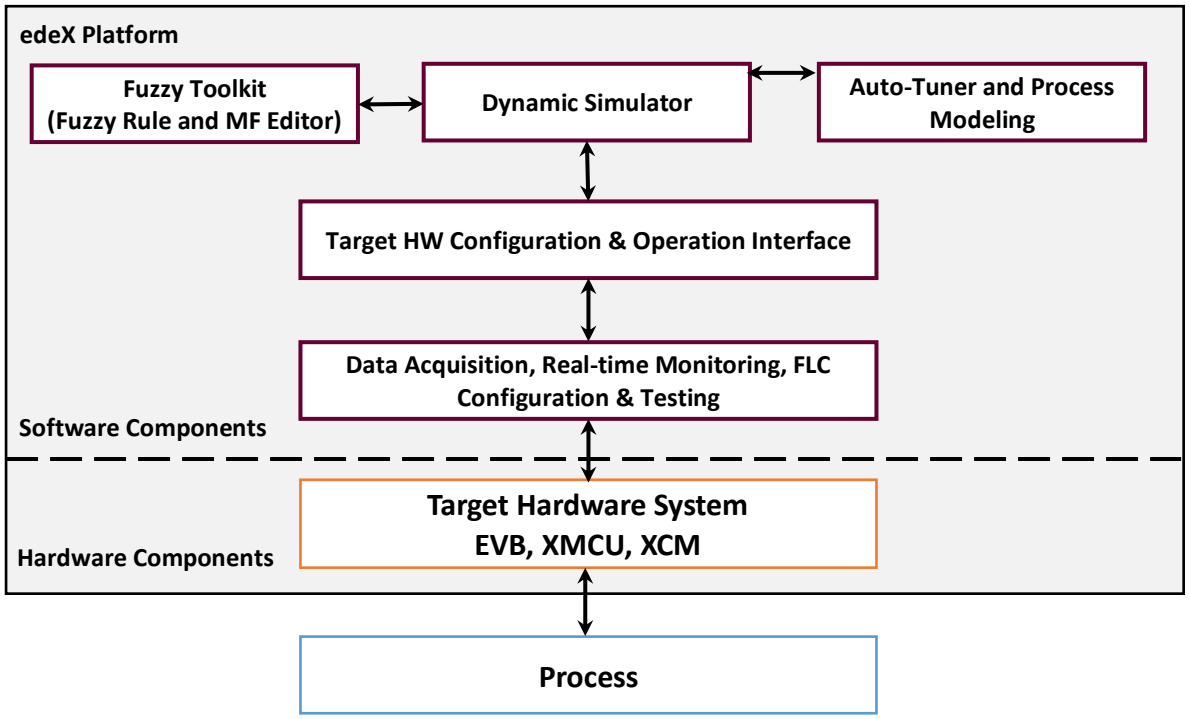
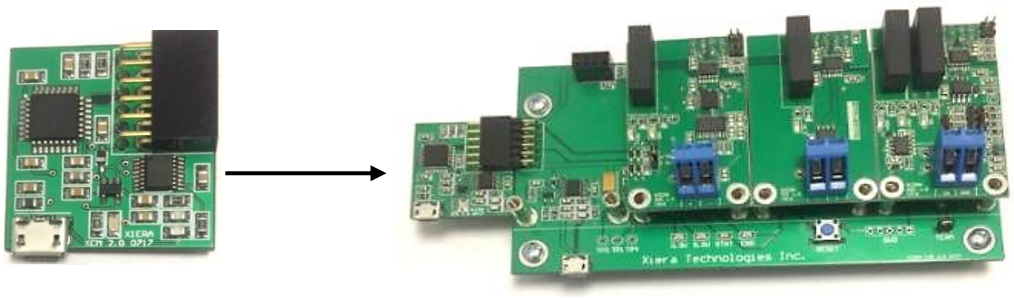


Figure 3 edeX software and hardware components and process interface



[a] [b]

Figure 4: edeX hardware [a] XCM board, [b] XCM board with Evaluation Board and I/O boards

2.4.2 Auto-Tuner System

After the control process model has been properly built on the edeX software platform, the configuration for the edeX auto-tuning parameters can be started. The edeX auto-tuning process can be initiated using default parameters to achieve a stable control process. However, to obtain enhanced control results edeX has the flexibility to modify:

- **Simulation time:** Adjust the sampling frequency of the FLC controller, which can be adjusted depending on the physical characteristics of the control system.
- **Performance evaluation:** Modify the thresholds and constraints involving the auto-tuning (optimization) process including rise time, overshoot, settling time, steady state error, and oscillations.
- **Design variables:** Select the number of parameters/blocks to include as part of the auto-tuning (optimization) process. For example, the user can define to just tune the FLC process gains, or also include MFs and fuzzy rules-related parameters in the auto-tuning process.
- **General Settings:** Specify the number of auto-tuning iterations and the number of best solutions, based on performance evaluation criteria, to keep at the end of the process.

3 MIMO Models Setup, Cases and Performance Criteria

3.1 Test Setup

The MIMO process models and the PID controller were emulated on an OPAL-RT real-time simulator. The high-level concept is to emulate real-time the physical industrial process, interacting with the PID and FLC systems. This project successfully compared two MV system setups: (1) a 2x2 I/O and (2) a 3x3 I/O system configuration. Yet, the implementation methodology and scenarios are similar for both cases.

The industrial process model was developed using MATLAB Simulink and then programmed onto the OPAL-RT simulator. The process input/output signals are then directed to the Hardware-In-the-Loop (HIL) Controller and Data Acquisition Interface (Figure 5 – middle right) which then are connected to the controller[1]. This platform and connection setup is used for all the tests in this report; which also allowed for simple data acquisition and analysis.

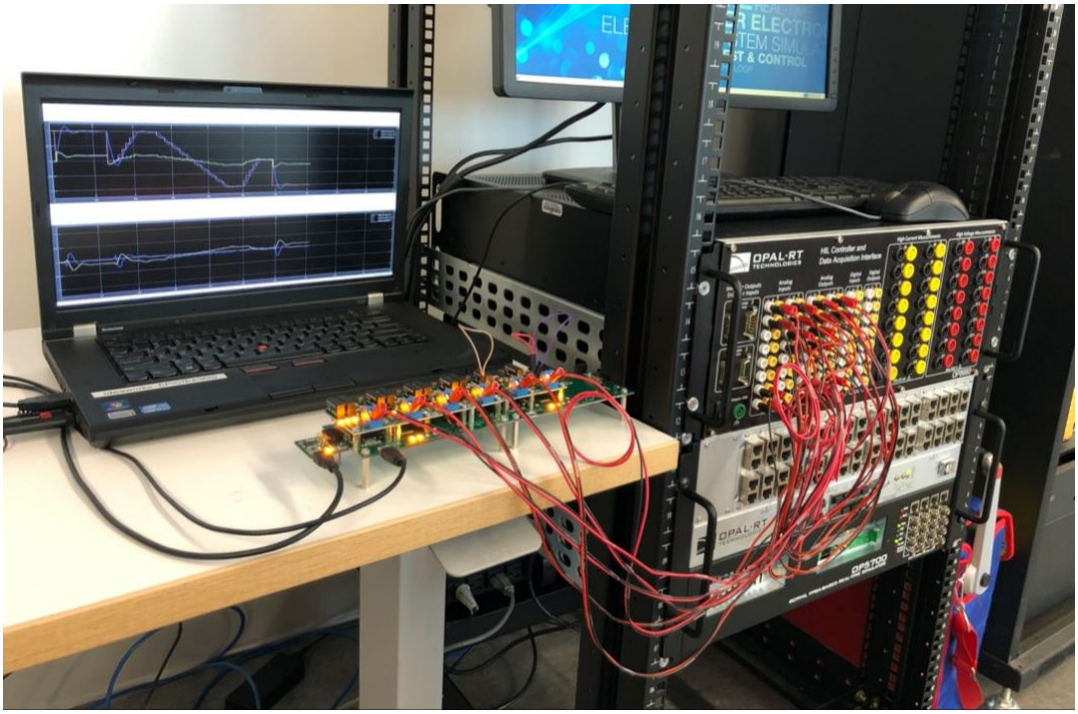


Figure 5: Physical FLC controller connection with OPAL-RT test setup

[1] The PID controller was programmed on a different core of the OPAL-RT processor, thus the I/O signals were outputted and feedback to and from the OPAL-RT module, as shown in Figure 5.

3.2 Selected Cases and Performance Criteria

For the two case studies presented in the next sections, there are three cases analyzed when comparing the results for the PID and FLC test systems:

1) **Base Case:**
    a. Step function applied as a set-point for each loop, to capture results of a virtually perfect system.
2) **Disturbance Case:**
    a. Once the feedback signals for all loop controllers settle to the their steady-state values, a disturbance of 20% was introduced for 5 second on each feedback signal.
    b. The disturbance was introduced on one loop at a time to see its effect on the other loop's feedback signal.
    c. This case represents sudden changes in the system affecting the feedback signal of a controller.
3) **Noise:**
    a. In this case, a noise signal with 10% variance was applied to the feedback signals to replicate noise in any practical environment.

The tests included the calculation of the following control systems performance characteristics using the default calculation used in MATLAB/Simulink [13].

- **Rise time:** The time taken for the output to go from 0% to 90% of the set-point input. In this case a step signal from 0 to 1.
- **Steady-State Error:** The difference between the set-point and the feedback steady state signal values.
- **Settling Time:** The time it takes for the error between the setpoint and the response value to reach a steady-state response; with the difference being within 2% of setpoint.
- **Overshoot:** The percentage difference of the maximum peak value of the signal to the final value (steady state).

4 2x2 MIMO System: Distillation Column Control

The objective of this project is to compare the performance of the FLC against a PID controller using a MIMO model where PID is typically used. Thus, a 2x2 distillation column model was selected to test and compare the performance of the FLC and PID controllers.

4.1 System Definition

The 2x2 MIMO model implemented here has previously been reported in the literature to control a distillation column process [14]. The analyzed distillation process is used in the petroleum and chemical industries to purify their final products. The distillation column is based on an R–S (Reflux –Steam) structure or the energy balance method. The steam flow rate, S, and the reflux flow rate, R, are the control inputs; the objective is to maintain the products concentration (controlled variables) as the feed flow and feed concentration (input parameters) change over time [15].

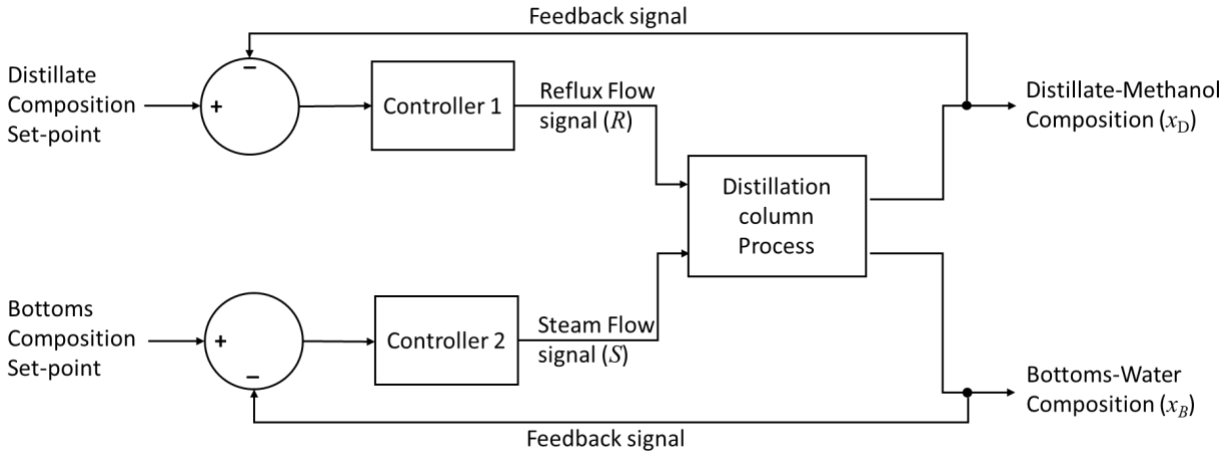


Figure 6: Process control of the analyzed distillation column

The 2x2 transfer function model comes from the Wood-Berry model of the distillation column processes used to separate methanol and water. The system outputs are the distillate and bottoms compositions x_D and x_B respectively, which are controlled by the reflux, R, and steam, S, flow rates. The following is the transfer function model for the process obtained from [14]:

$$G(s) = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix}$$

Combining the transfer function and the block diagram in Figure 7, the following diagram is obtained:

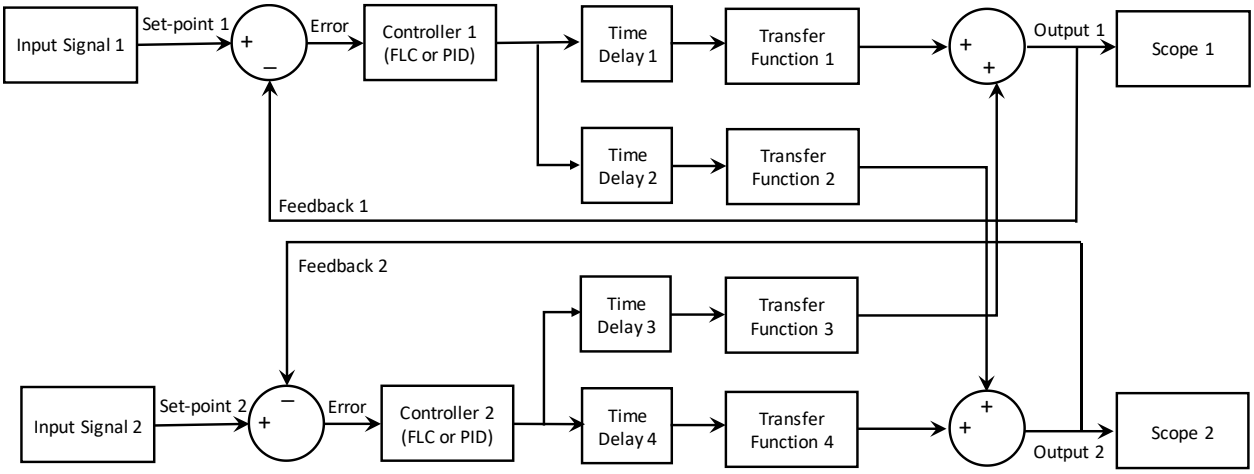


Figure 7: PID & FLC controller's test setup using a Hardware-In-the-Loop setup using OPAL-RT

For both the PID and FLC control, a multi-loop design was used, with separate controllers that process only the feedback signal from each loop independently. The following Table 1 shows the gains, delays, and associated transfer functions for this case study:

Table 1: Values of model's gains, time delays, and transfer function

Block No.	Time Delay (sec)	Transfer Function
1	1	$12.8/(16.7s+1)$
2	7	$6.6/(10.9s+1)$
3	3	$-18.9/(21s+1)$
4	3	$-19.4/(14.4s+1)$

4.2 PID Controller Test Setup and Tuning Process

The HIL module's analog I/O were divided into two systems: (1) Emulating the PID controller and (2) Emulating the distillation column process, as previously shown in *Figure* 7. The PID setup process is as follows:

- The process input signals were provided from the real-time simulator using a step block function.
- The PID controller was emulated using the real-time simulator and its I/O signals were routed via the HIL Controller and Data Acquisition Interface (Figure 5) to emulate the separate hardware.

- The sampling time for the PID controller and process was set to 100ms which is sufficient for capturing any variation of the distillation process.

4.2.1 PID Controller Tuning Procedure

Once the Simulink model has been implemented based on the previous process, the following steps show the parameter tuning process for the PID controller:

- In Simulink, under the Analysis menu, select Control Design → Control System Tuner
- In the Control System Tuner select the blocks that need to be tuned by clicking the selected Blocks. For this model, both PID controllers need to be selected.
- Click on New Goal and select reference tracking. In the Step Tracking Goal dialog, specify the composition signals (x_D and x_B) set-point for tracking.
- Under Specify step-response inputs, click Add signal to list. Then click Select signal from model; select all the step inputs for step-response inputs and all the feedback signals for the step-response outputs.
- Once the tuned blocks and tuning goals are set, click Tune. The tuner will automatically tune the controller.
- The model has now been linearized and an initial auto-tuned solution has been found. For a MIMO system, it is likely that the user would require to perform an iterative process to fine-tune the response signals to obtain the desired outputs. In MATLAB Simulink, this process is done graphically to adjust the controller robustness and speed, which MATLAB then adjust the PID parameters accordingly.

The resulting parameter values for both PID controllers are as follows:

Table 2: Resulting PID controller parameters

PID Parameters	PID Controller 1	PID Controller 2
Proportional (k_P)	0.2822	0.6915
Integral (k_I)	0.0295	0.0583
Derivative (k_D)	-0.2485	-3.5126
Filter Coefficient (N)	0.1193	0.1440

4.3 Fuzzy Logic Controller Test Setup and Tuning Procedure

The FLC setup follows the same block diagram as shown in Figure 7. The main difference is that in this case the FLC controller was standalone physical device; thus, the HIL Controller and Data Acquisition Interface was used to connect the real-time simulator to Xiera's controller (Figure 5). The following are the general settings for the setup:

- The I/O blocks were connected directly to the Xiera's Evaluation Board coming from the real-time simulator.
- The inputs (step signals) are fed from the real-time simulator to the FLC system.
- To simplify the results comparison, the data acquisition process was performed using the real-time simulator. However, the Xiera system is also capable of recording data.
- The simulation time was kept at 100 ms as it was the case for the PID system, and to match the sampling rate of Xiera's edeX platform.

4.3.1 FLC Tuning Process

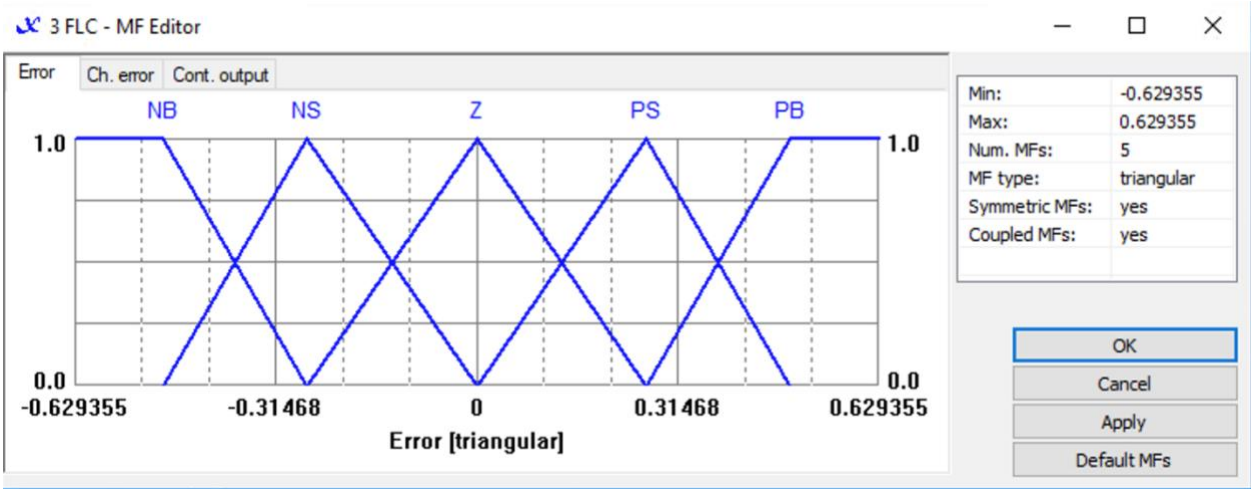
The tuning process for the FLC system was performed using the edeX design software. As an initial step, the performance evaluation constraints were modified (default are zero) to avoid unnecessary longer auto-tuning time. The values for the evaluation constraints were set as follows:

Table 3: FLC Performance Evaluation Constraints

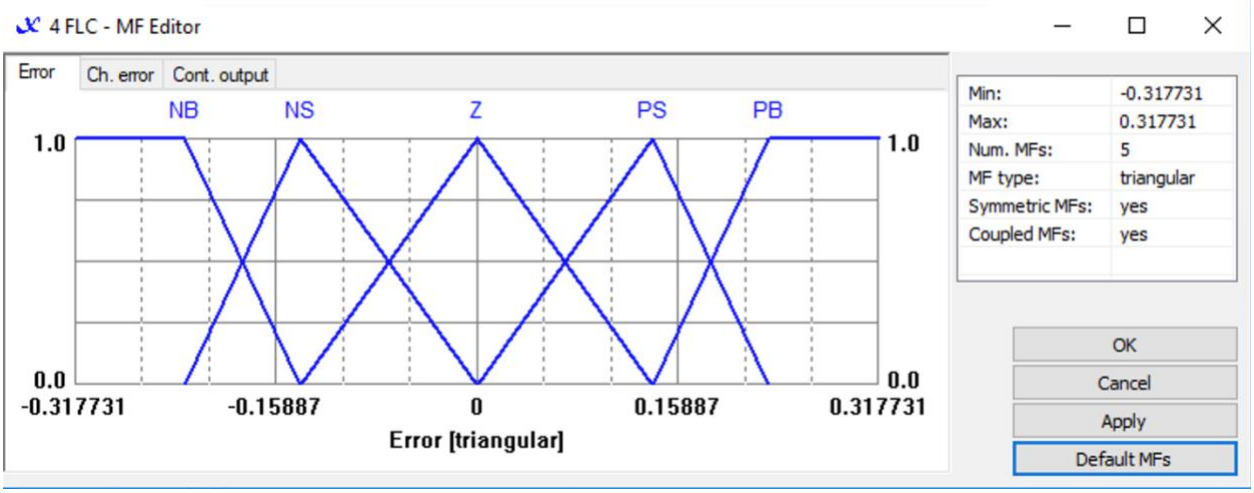
Constraint Type	Constraint Value (Loops 1 & 2)
Overshoot (%)	0.01
Rise-time (s)	30.0
Settling time (s)	35.0
Steady-state error (%)	0.01
Oscillations (%)	0.01
Asymptotic slope (%)	0.01

As a second step, the FLC rule actions, gains and integral parameters were selected to be tuned as a part of the optimization process. Finally based on Xiera's tuning experience, the number of MFs for both controller's Error & Cont. output were changed from 3 to 5; The type of MFs was left unchanged.

The resulting knowledge-based rules, error MFs, change-of-error MFs, and FLC output MFs are as follows:

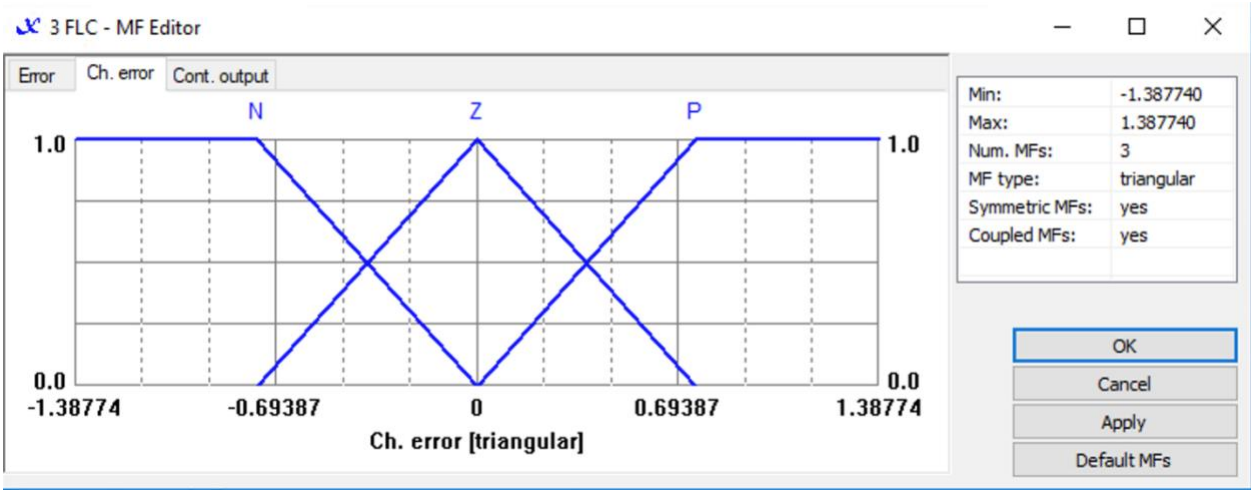


(a)

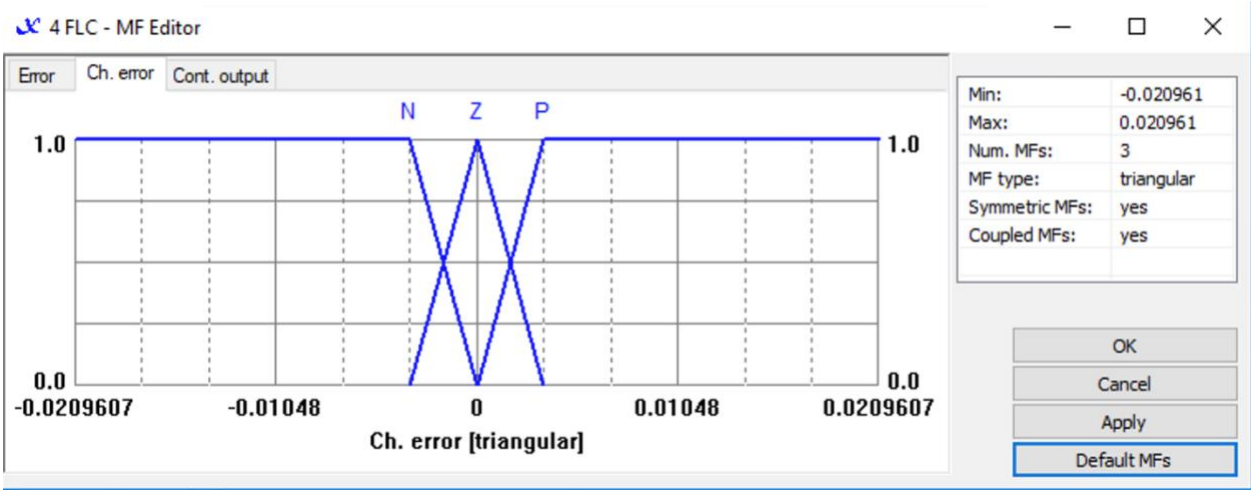


(b)

Figure 8: FLC error membership functions (a) FLC 1 and (b) FLC 2

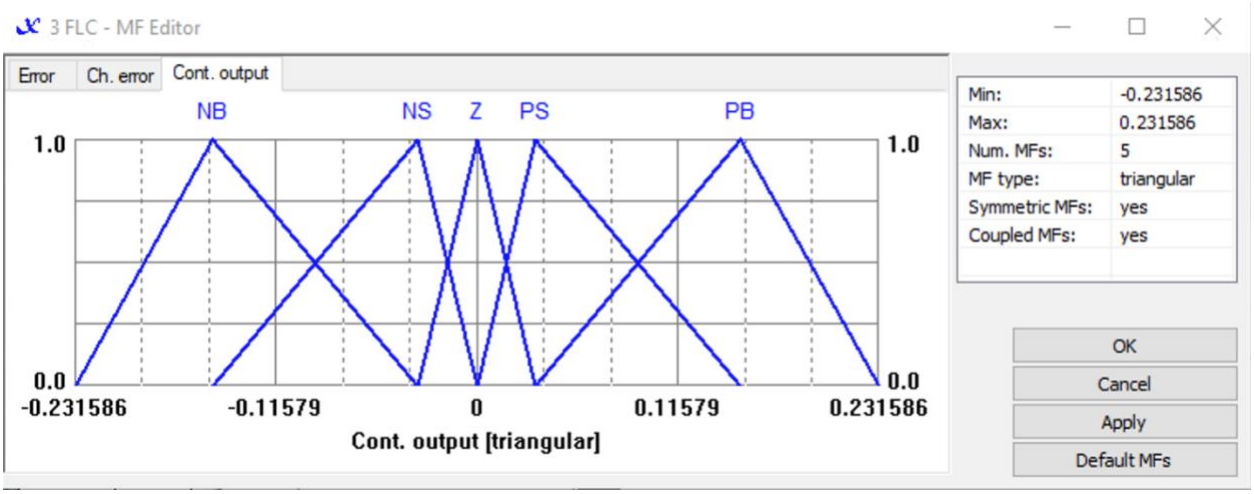


(a)

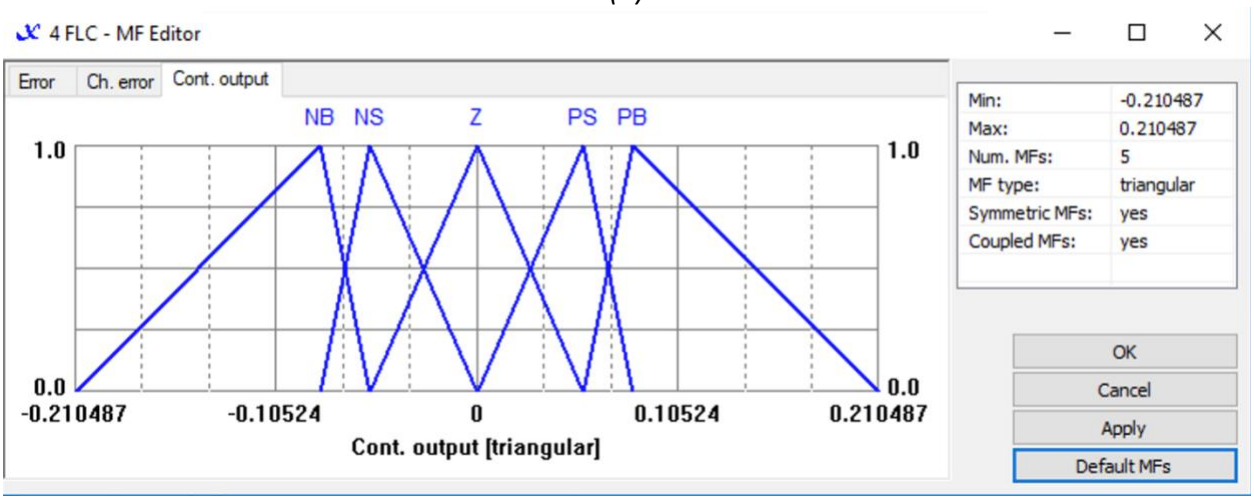


(b)

Figure 9: FLC change of error membership functions (a) FLC 1 and (b) FLC 2

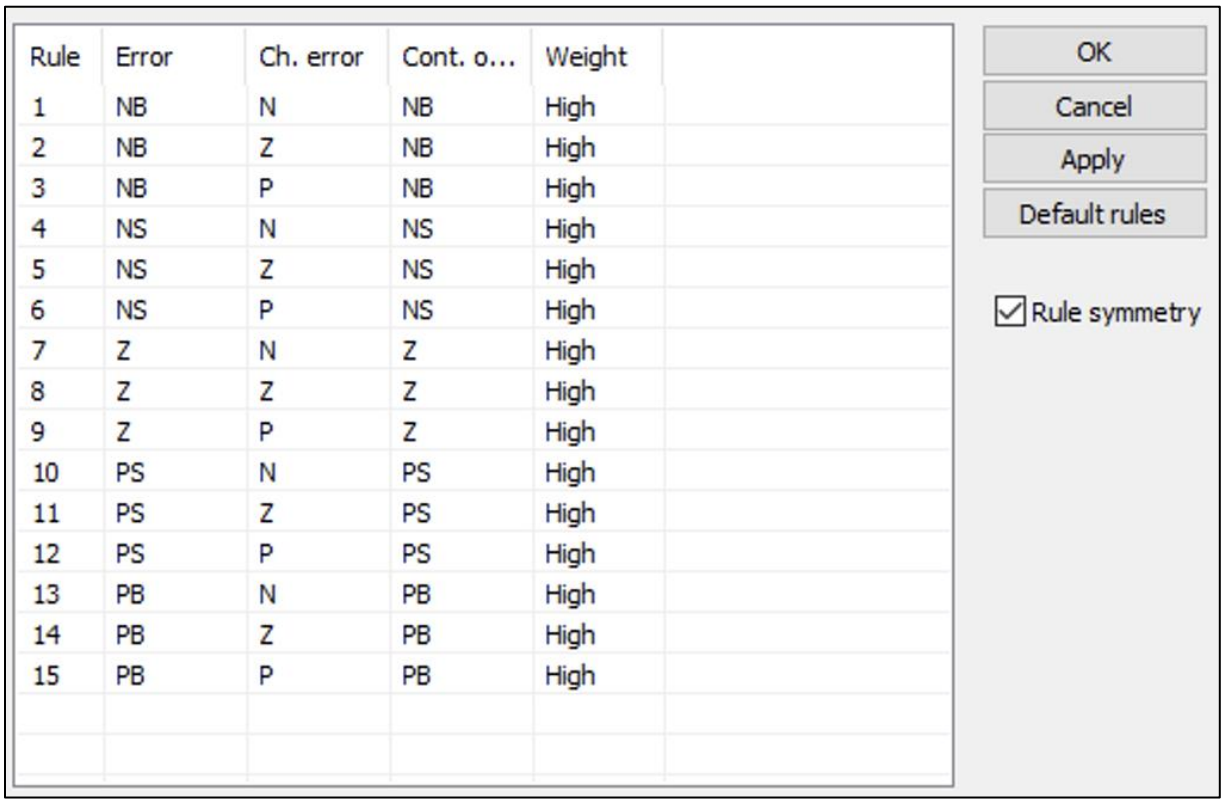


(a)



(b)

Figure 10: FLC output membership functions (a) FLC 1 and (b) FLC 2



Rule	Error	Ch. error	Cont. o...	Weight
1	NB	N	NB	High
2	NB	Z	NB	High
3	NB	P	NB	High
4	NS	N	NS	High
5	NS	Z	NS	High
6	NS	P	NS	High
7	Z	N	Z	High
8	Z	Z	Z	High
9	Z	P	Z	High
10	PS	N	PS	High
11	PS	Z	PS	High
12	PS	P	PS	High
13	PB	N	PB	High
14	PB	Z	PB	High
15	PB	P	PB	High

Figure 11: FLC 1 and FLC 2 rules

4.4 PID and Fuzzy Logic Controller Test Results

This section presents the results for the PID and FLC test system with the objective of comparing their performance. As mentioned in Section 3.2, there are three cases analyzed (base, disturbance, and noise case) and for each of them the comparison was made considering the characteristics of the feedback response signal (rise time, steady-state error, settling time and overshoot).

4.4.1 Base Case Results

The PID controllers were tuned using the procedure described in Section 4.2.1 and the results are shown in Figure 12. The three signals shown are the set-point (black), the controller's output signal (red), and the feedback (process output) signal (blue). The units in the y-axis are in volts representing a 'per unit' base for the results to simplify the comparison among the three signals. The result of the PID controllers shows that the rise time for loop-1 feedback signal is ≈13.1 sec and loop-2 feedback signal is ≈7.2 sec with an overshoot value of 32.2% and 42.9%, respectively. In addition, the signals settling time for loop-1 is ≈108 sec and loop-2 is ≈100 sec after going through an oscillation phase.

Figure 13 shows the response of the auto tuned FLC for the same case. The results show that the rise time for the loop-1 and loop-2 feedback signals is ≈30.7 sec and ≈34 sec, respectively. The systems settling time of loop-1 and 2 are ≈60 sec and ≈65 sec, respectively, with a 0% overshoot and steady-state error.

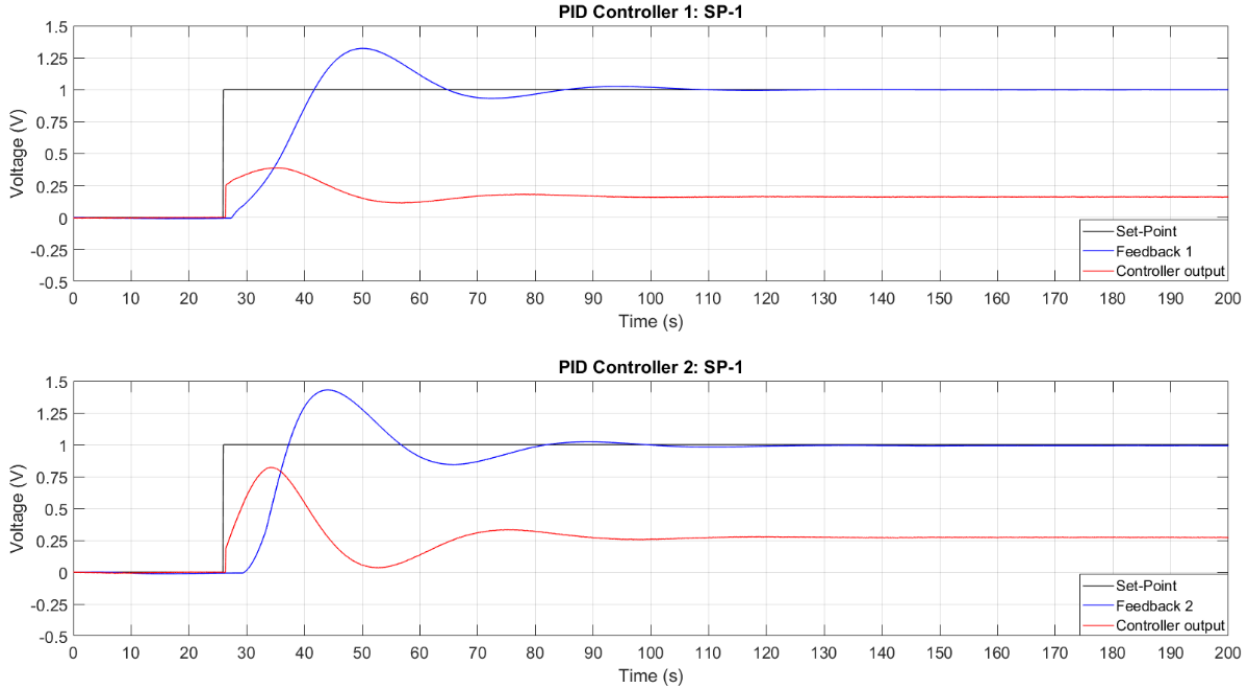


Figure 12: PID controller step response: Base case

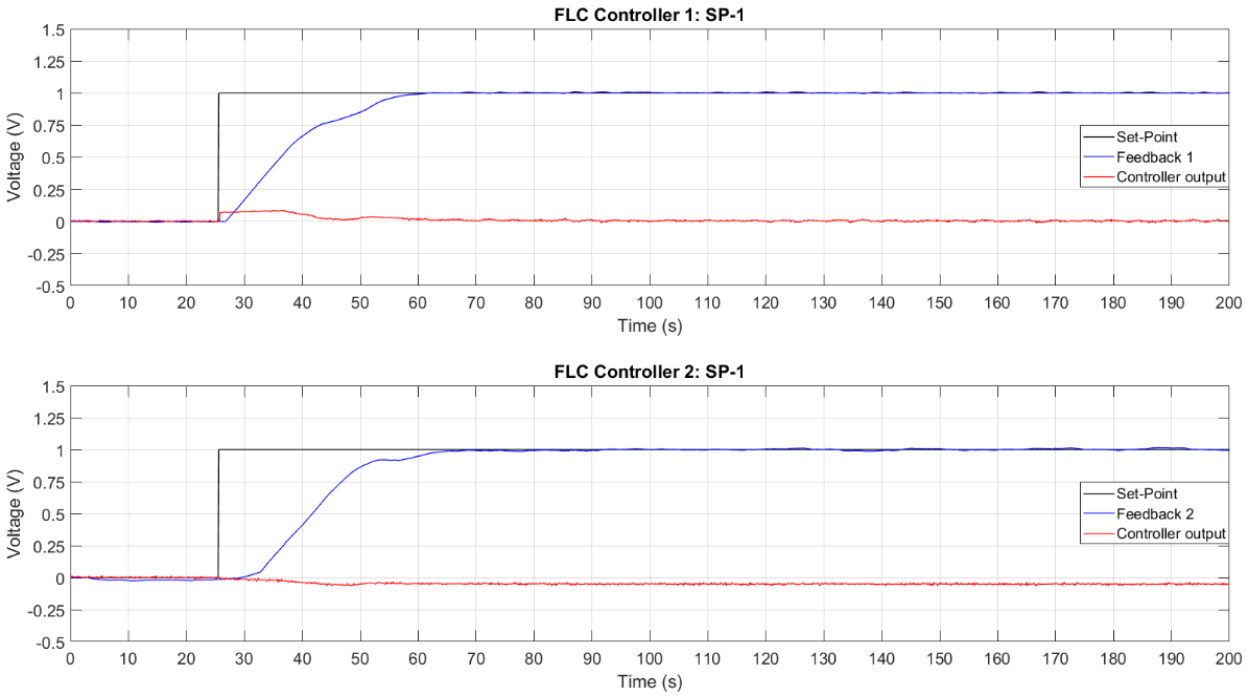


Figure 13: FLC step response: Base case

Given, the performance criteria; the rise time of PID controller-1 and controller-2 is 57.2% and 78.8% faster than the FLC system, respectively. However, the FLCs outperformed the PID controllers from an overshoot perspective since there was zero overshoot; and from a settling time, perspective where the FLC-1 and 2 were 44.4% & 35% faster than the PID controllers.

Table 4: Result summary: Base case

Loop-1	Rise time (s)	Settling time (s)	Overshoot (%)
PID	13.1	108	32.2
FLC	30.7	60	0
Loop-2	**Rise time (s)**	**Settling time (s)**	**Overshoot (%)**
PID	7.2	100	42.9
FLC	34	65	0

4.4.2 Disturbance Case Results

The Disturbance Case analyzed the performance of PID and FLC systems when a 20% disturbance (relative to the set-point reference) is applied for 5 seconds to the feedback signal. The disturbance was applied independently to each loop in turn, making sure that the system reaches steady state before applying the disturbance signal to the feedback signal of the other loop. The main objective was to identify the effect of a disturbance in both the target loop as well as the second loop, due to their interaction.

Figure 14 presents the PID controller response for the disturbance case. The disturbance was first applied to the feedback of loop-2 and the effect on loop-1 can be observed at t ≈132.5 sec. The disturbance caused a slight overshoot of 5.2% on the feedback signal of loop-1 and the signal settled after ≈ 28 sec. Also, the removal of disturbance from loop-2 caused its feedback signal to show an undershoot of 13.24% and took ≈20.6 secs before it settled.

When the disturbance was applied to loop-1, there was a resulting 7.5% undershoot in the feedback signal of loop-2 and the signal settled after ≈36 sec. Also, the removal of disturbance from loop-1 caused the feedback signal to show a 9.95% undershoot which took ≈23.3 secs before settling.

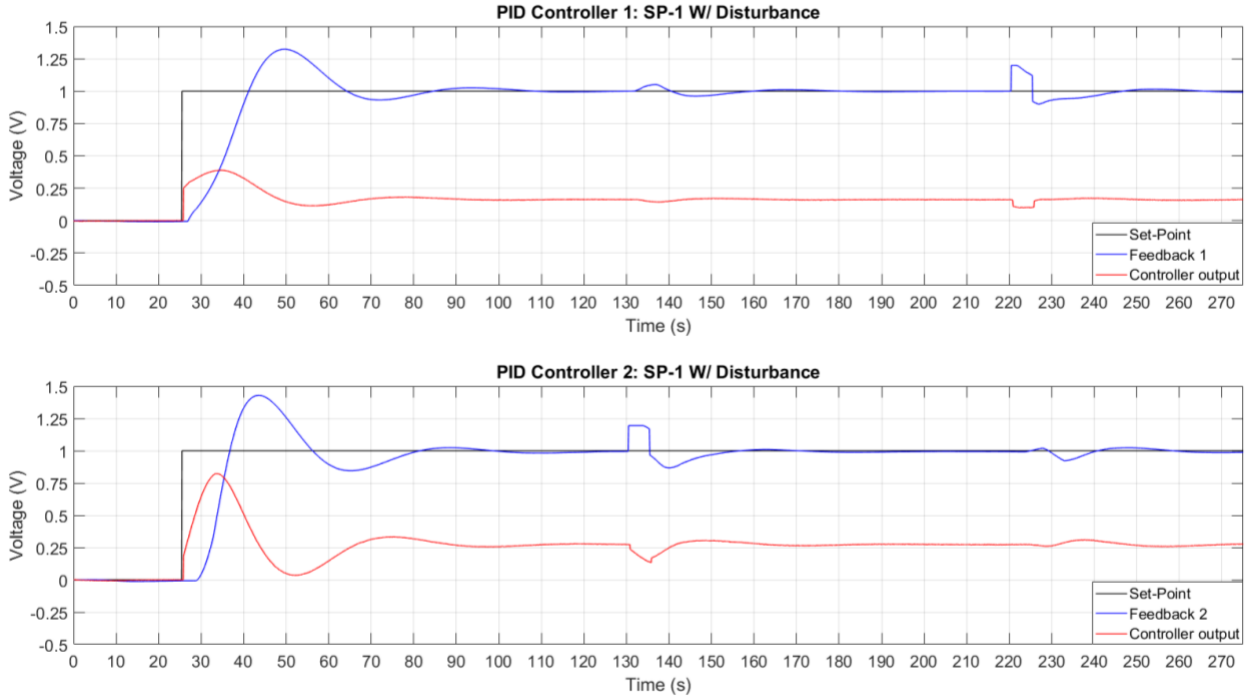


Figure 14: PID controller response: Disturbance case

Figure 15 presents the FLC controller response for the disturbance case. Like the PID controller, the disturbance was first applied to loop-2 and an effect on loop-1 can be observed at t ≈132.5 sec. The disturbance caused a slight overshoot of 4.8% on the feedback signal of loop-1 and the signal settled after ≈11 sec, after the disturbance is removed. Also, the removal of disturbance from loop-2 caused its feedback signal to show an undershoot of 12.19% and took ≈37 secs before it settled.

When the disturbance was applied to loop-1, there was a resulting undershoot of 10.6% on feedback signal of loop-2 and the signal settled after ≈25.6 sec. Also, the removal of the disturbance from controller-1 caused the feedback signal to show an undershoot of 11.65% and took ≈10.4 secs before it settled.

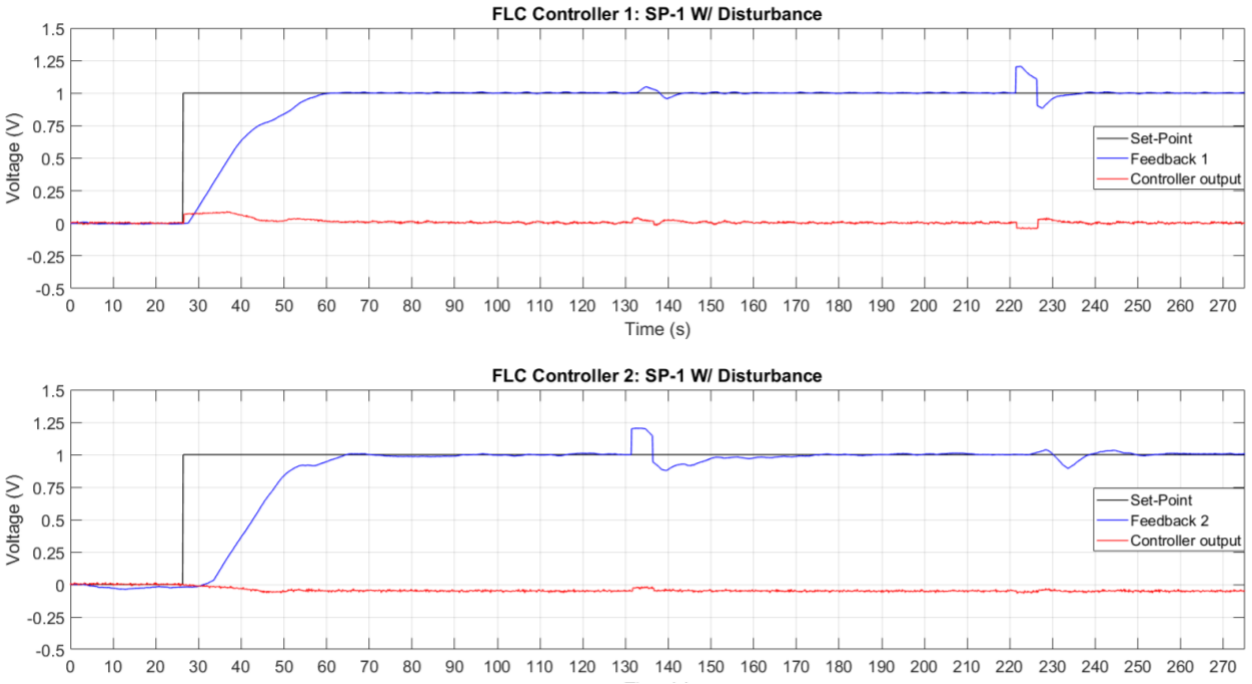


Figure 15: FLC response for disturbance case

Table 5 and *Table 6* summarize the results obtained for the disturbance case.

Table 5: Result summary: Loop-1 disturbance case

Disturbance applied to Loop-1		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	23.30	-
FLC	10.40	-
Resulting effect on Loop-2		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	36	-
FLC	25.60	-

Table 6: Result summary: Loop-2 disturbance case

Disturbance applied to Loop-2		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	20.60	-
FLC	37	-
Resulting effect on Loop-1		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	28	5.20
FLC	11	4.80

4.4.3 Noise Case Results

The Noise Case shows the effect of adding a 0.1 standard deviation noise signal to the feedback signal for all controllers. The results for the PID and FLC runs are show in Figure 16 and Figure 17. The response with noise is the same as without noise for each of the PID or FLC cases individually. Thus, with the noise level selected, there is no apparent difference between using a PID or FLC in the general response. However, when comparing the controller output of the PID and FLC controller, a significant attenuation difference is observed.

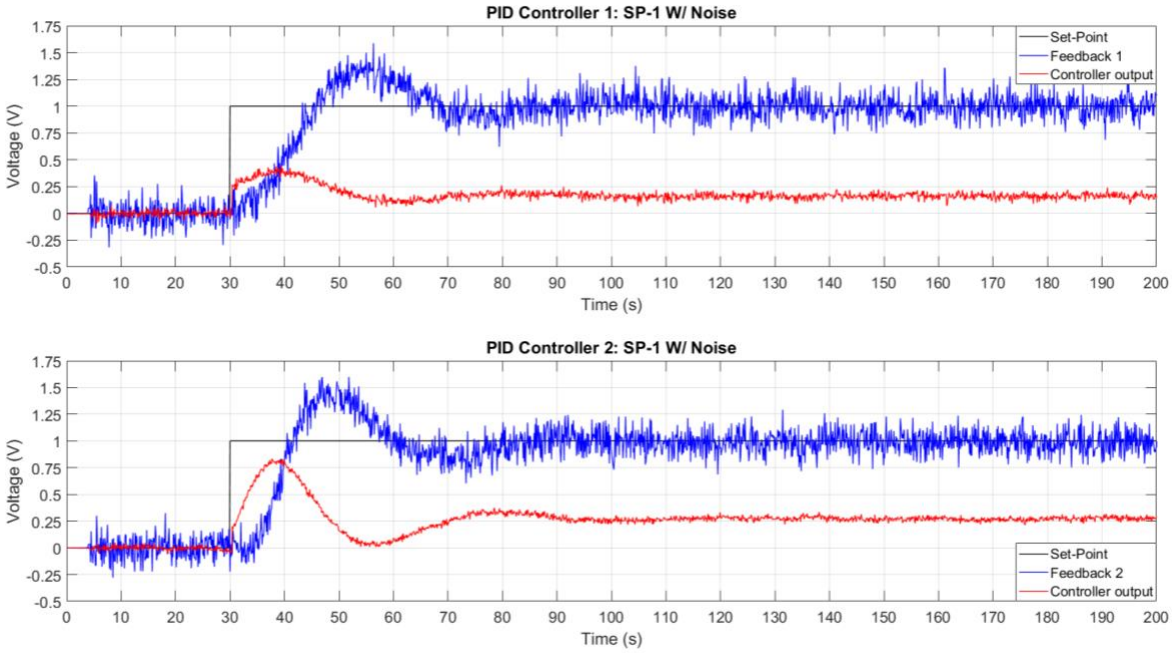


Figure 16: PID response: Noise case

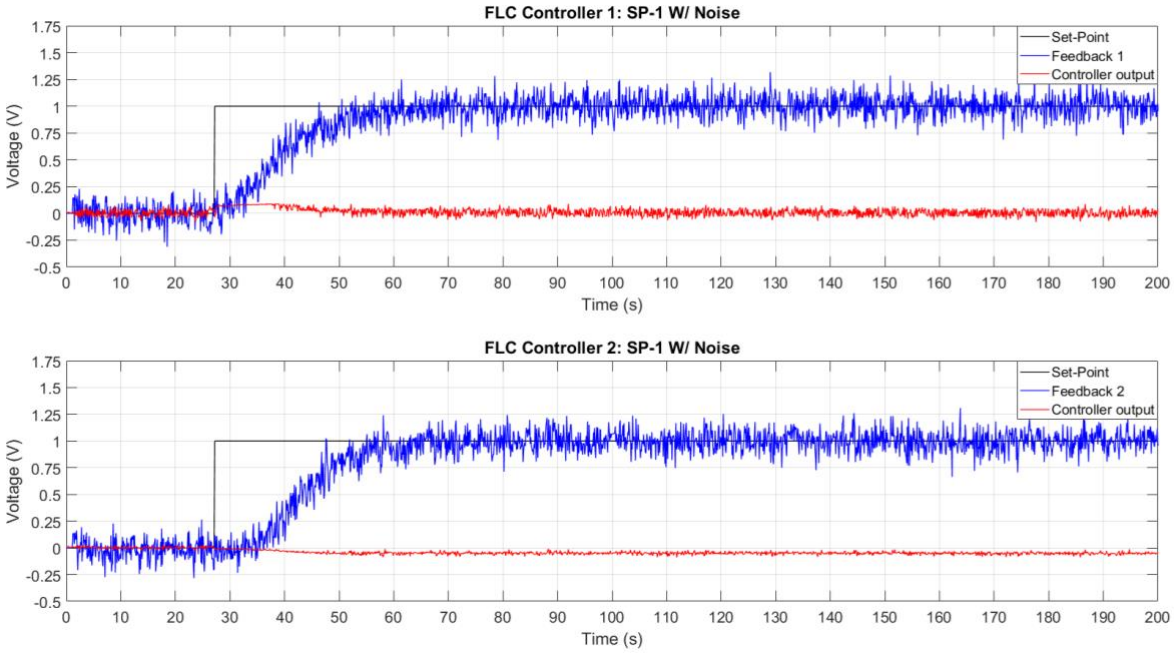


Figure 17: FLC response: Noise case

For comparison purposes, this report uses the Root Mean Square Error (RMSE) to compare the controller output signals between the baseline and noise case scenarios. The objective was to obtain a numeric value to easily quantify the impact of the noise on the controller. Table 7 shows the results of calculating the RMSE for both cases. The results show that there is a higher noise attenuation effect using the FLC; thus for Controller 1 and 2, the effect of the noise on the control output of the FLC is 60% and 92% less than the PID controller, respectively.

Table 7: Result summary: Noise case – Root Mean Square Error

Controller-1	
PID (RMSE)	0.0821
FLC (RMSE)	0.0327
$(RMSE_{FLC} - RMSE_{PID})/RMSE_{PID}*100$ [%]	-60%
Controller-2	
PID (RMSE)	0.1748
FLC (RMSE)	0.0127
$(RMSE_{FLC} - RMSE_{PID})/RMSE_{PID}*100$ [%]	-92%

5 3x3 Multi-variable Control System

The objective of this project is to compare the performance of the FLC against a PID Controller using a more complex 3x3 MIMO process model. This system represents a simulation of a process with high interaction and time delays.

5.1 System Definitions

The model is selected to test and compare the performance of PID vs. fuzzy logic control for a complex process with 3 inputs and 3 outputs, having high interactions between the loops and time delays in all the branches of the systems. The time delays add to the nonlinear behaviour normally found in industrial processes, such as thermal and fossil plants, boilers, distillation columns and chemical plants. Figure 18 and Figure 19 show the control model using PID and FLC controllers, respectively.

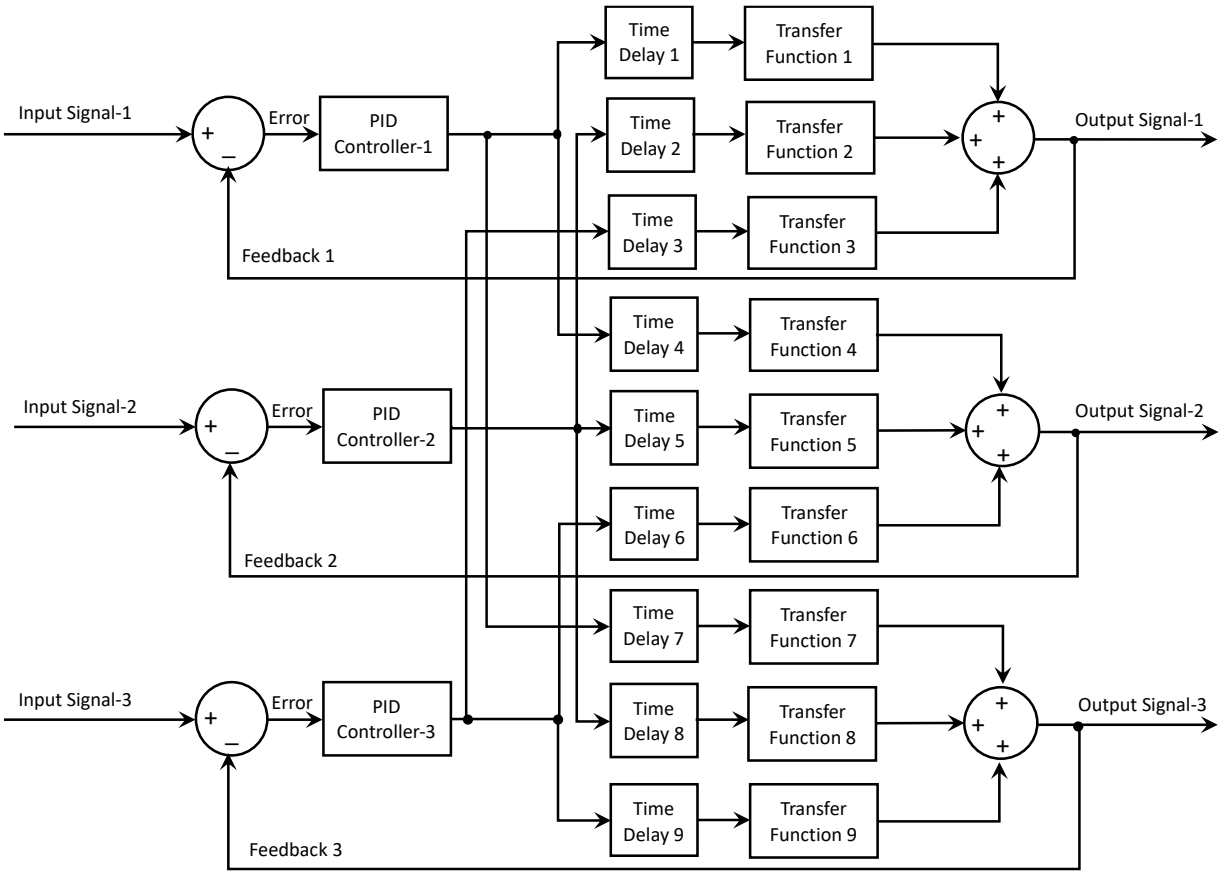


Figure 18: 3X3 System model Setup with PID controller

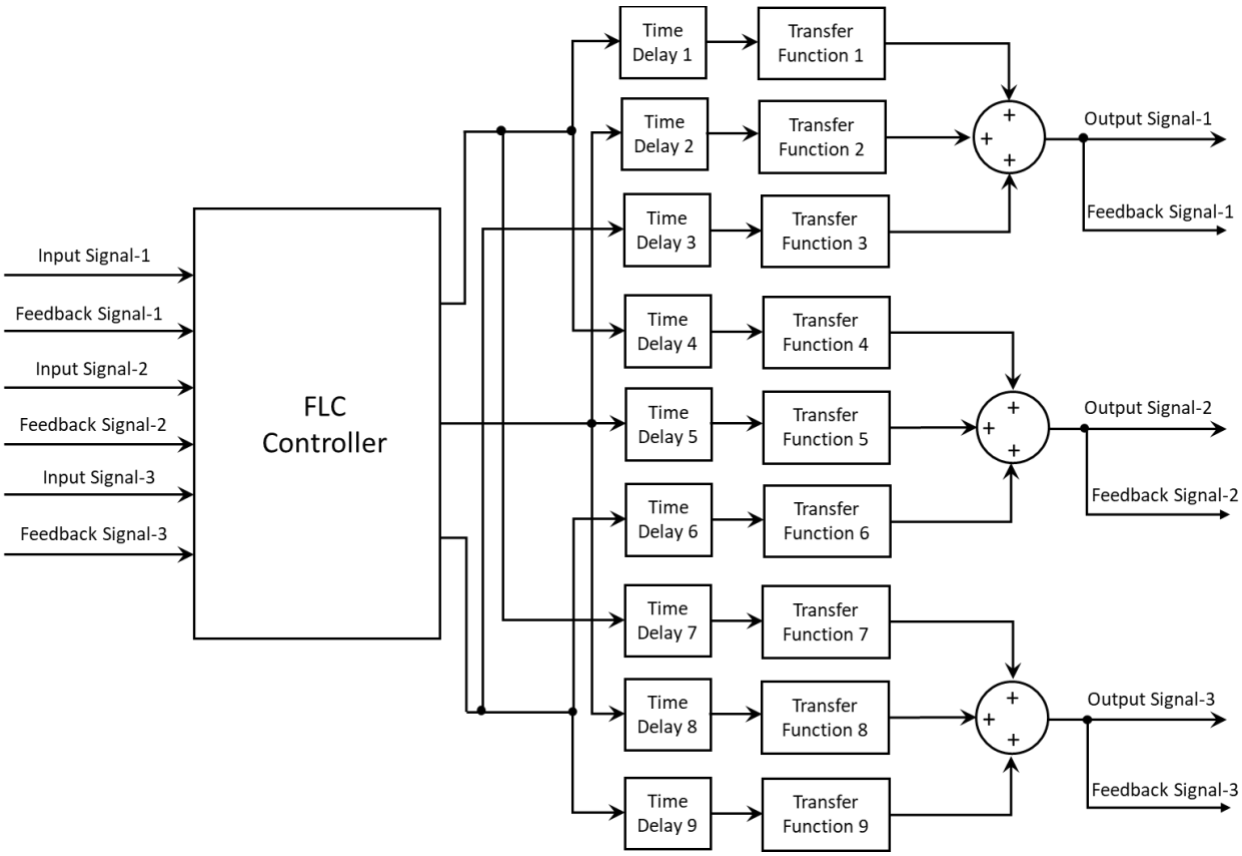


Figure 19: 3X3 System model Setup with FLC controller

Table 8 shows the values of the time delays and transfer functions used in the process model.

Table 8: List of model's time delay and transfer function values

Block No.	Time Delay (sec)	Transfer Function
1	5	$(-1.15)/(s^2 + 1.5s + 0.45)$
2	1	$0.95/(s + 1.15)$
3	1.5	$1.15/(s + 0.523)$
4	2	$1.12/(s + 0.501)$
5	1.25	$1.151/(s^2 + 0.52s + 0.351)$
6	2	$1.103/(s + 0.989)$
7	1	$1.019/(s + 1.116)$
8	1	$1.1/(s + 0.975)$
9	10	$(-0.915)/(s^2 + 1.12s + 0.51)$

5.2 Controllers Test Setup & Tuning Parameters

The 3x3 test model and performance criteria for both the PID controller and FLC follow the same procedure as in the 2x2 MIMO tuning process; however, there is a fundamental difference in the controller design between the two approaches. While the PID control is implemented as multiple loops (where each PID is only able to process the feedback from a single loop), the fuzzy logic controller uses a true multivariable approach. The FLC block processes all three-loop feedback signals as a unit, and its rule-based nature allows it to generate each loop's control signal while taking into account the behaviour of the other two interacting loops. The simulation time of the model was set to 250ms, which is sufficient for capturing any variation of the analyzed process.

For detailed information about the model setup, cases and performance criteria can be found under section 3 and for tuning process of the individual controllers, refer to sections 4.2.1 and 4.3.1.

The resulting controller parameters for PID controller and FLC are as shown in the following tables:

Table 9: Resulting PID controller parameters

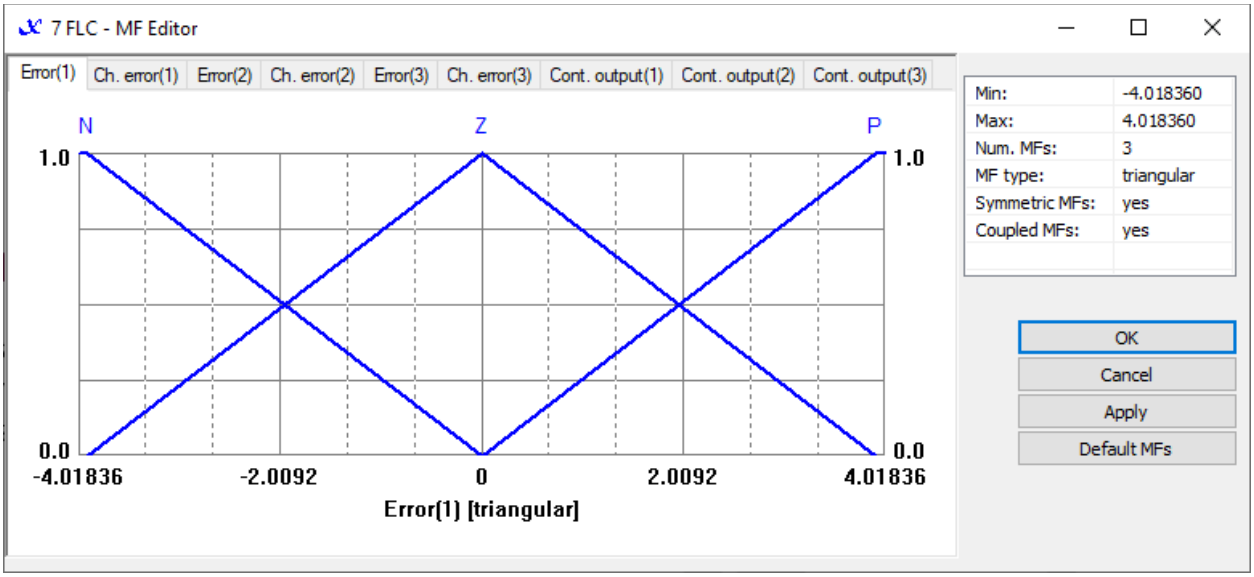
PID Parameters	PID Controller 1	PID Controller 2	PID Controller 3
Proportional (k_P)	-0.17507	0.14350	-0.02307
Integral (k_I)	-0.00597	0.00322	-0.01527
Derivative (k_D)	7.02435	-3.25949	0
Filter Coefficient (N)	0.02409	0.04402	100

Table 10: FLC Performance Evaluation Constraints

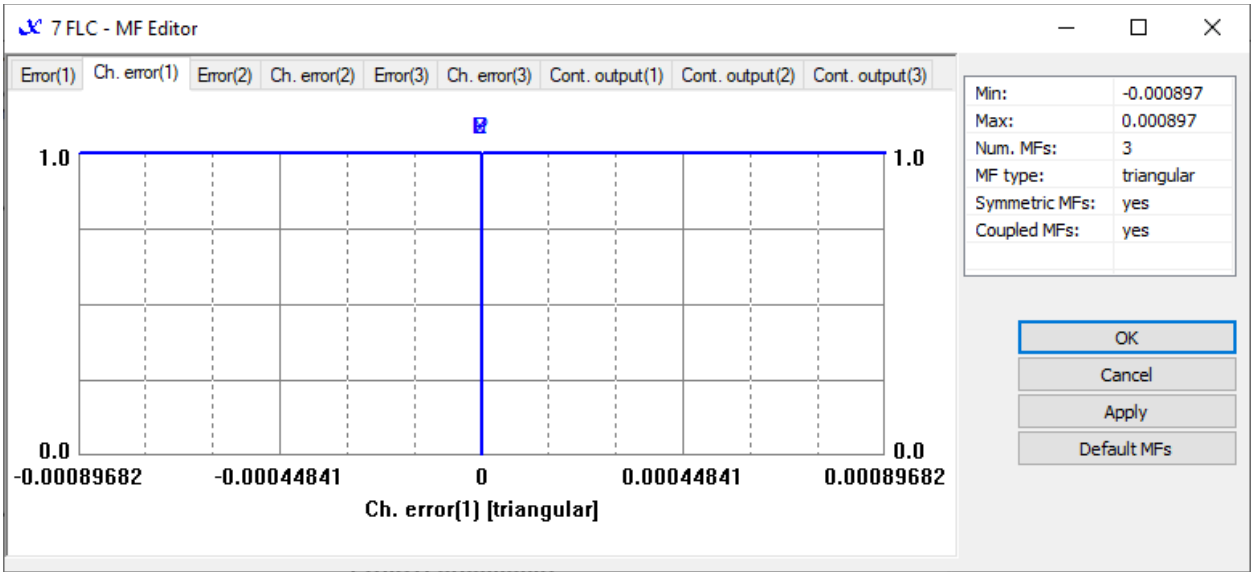
Constraints Type	Constraints Value (Loops 1, 2 & 3)
Overshoot (%)	0.01
Rise-time (s)	40.0
Settling time (s)	80.0
Steady-state error (%)	0.01
Oscillations (%)	0.01
Asymptotic slope (%)	0.01

As with the 2x2 case; the FLC rules, membership functions, gains and integral parameters were selected to be tuned as a part of the optimization process.

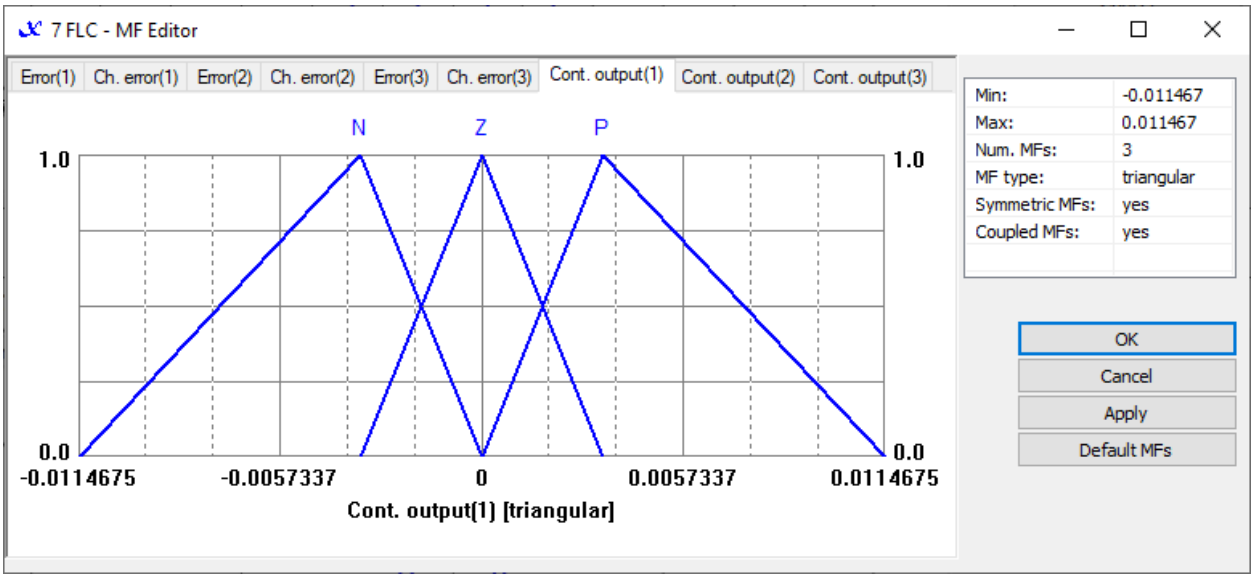
As show in Figure 19, the 3x3 model requires two inputs per signal (set-point and feedback) which are internally processed by the FLC to calculate the error and change-of-error for each loop. These two latter parameters are the six input variables used for the fuzzification process and resulting in three fuzzy outputs which are the control signals for each loop (Figure 20 to Figure 22). The number of membership functions were kept to default values (three per fuzzy variable), which results in 3^6=729 rules in the knowledge-base. Note that only half of the rules are independent since symmetrical MFs was selected as part of the optimization.



(a)

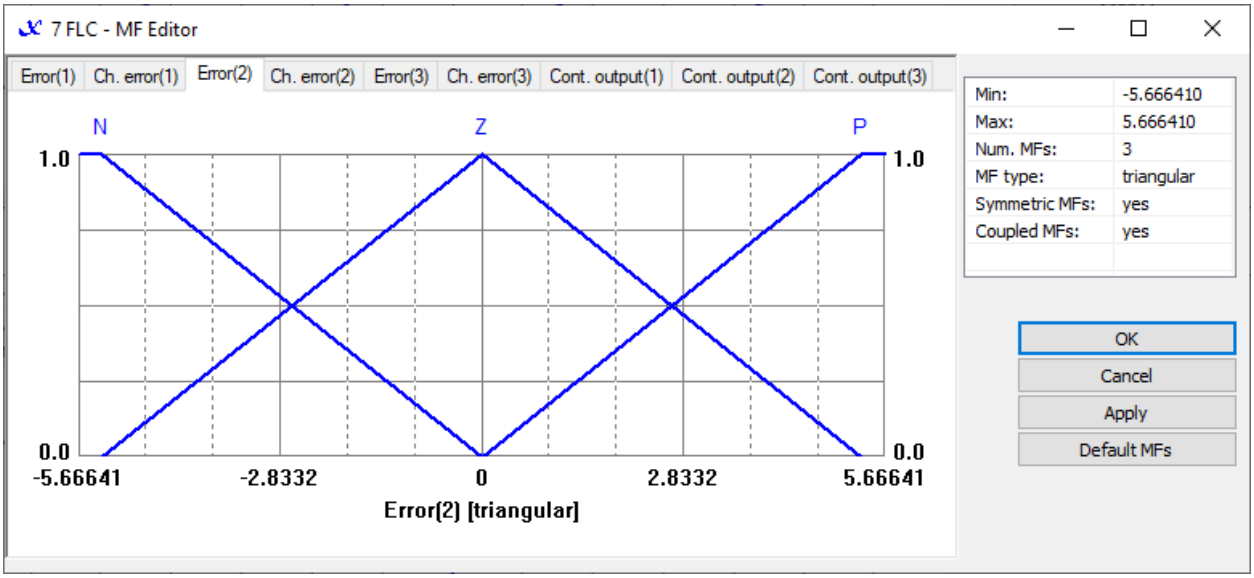


(b)

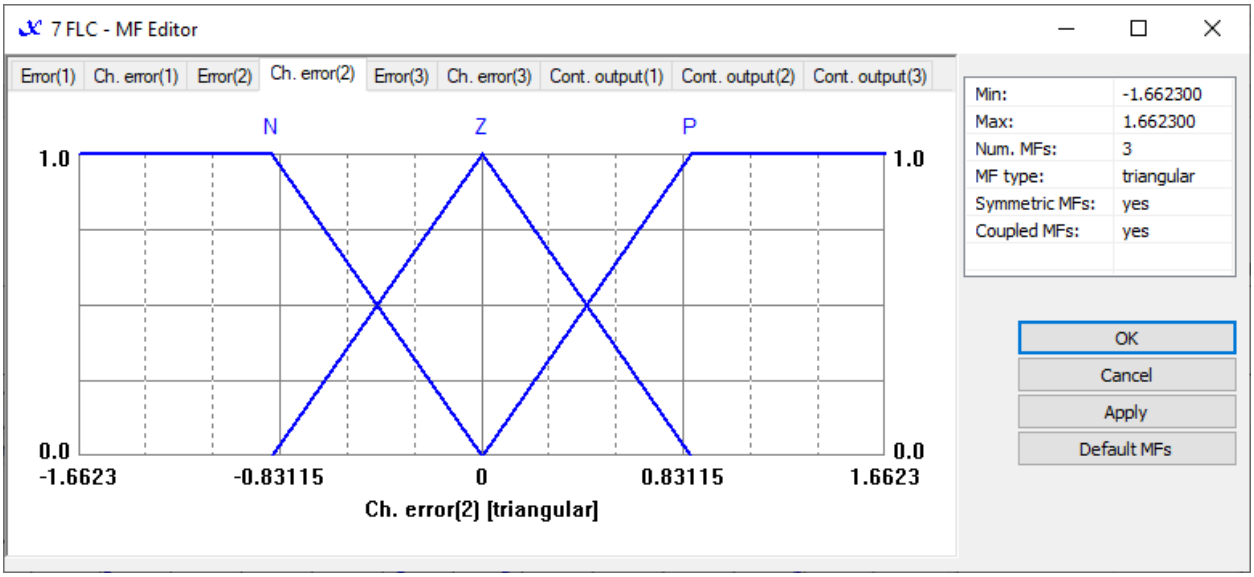


(c)

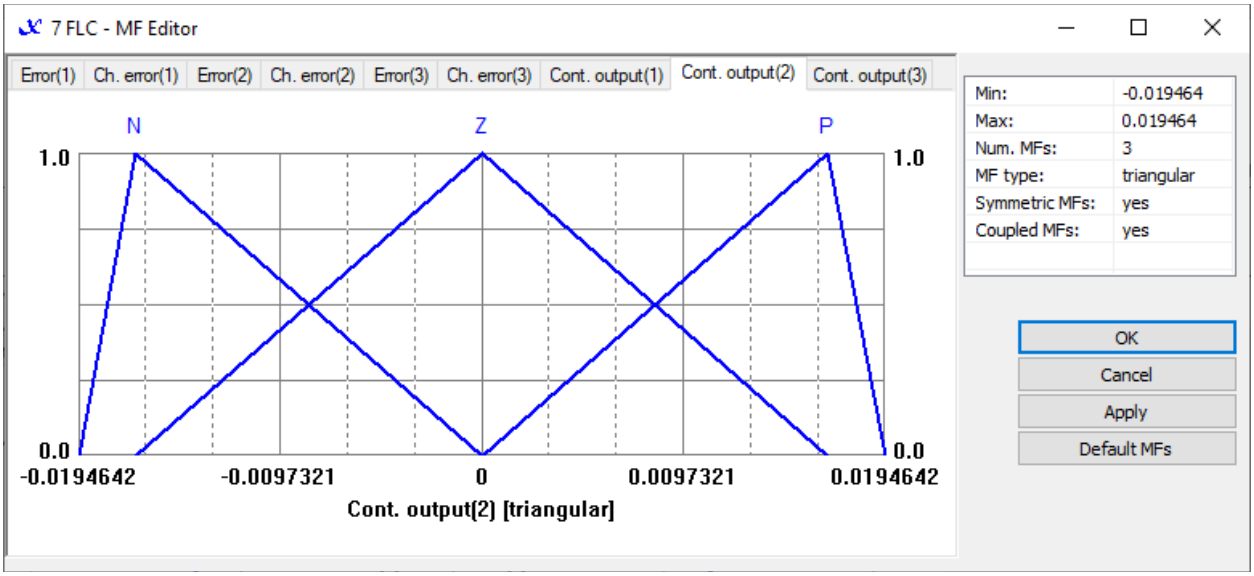
Figure 20: FLC membership functions (a) error (b) change of error and (c) controller output for loop 1


7 FLC - MF Editor
Error(1)
Ch. error(1)
Error(2)
Ch. error(2)
Error(3)
Ch. error(3)
Cont. output(1)
Cont. output(2)
Cont. output(3)
N
Z
P
1.0
0.0
-5.66641
-2.8332
0
2.8332
5.66641
Error(2) [triangular]
Min: -5.666410
Max: 5.666410
Num. MFs: 3
MF type: triangular
Symmetric MFs: yes
Coupled MFs: yes
OK
Cancel
Apply
Default MFs

(a)

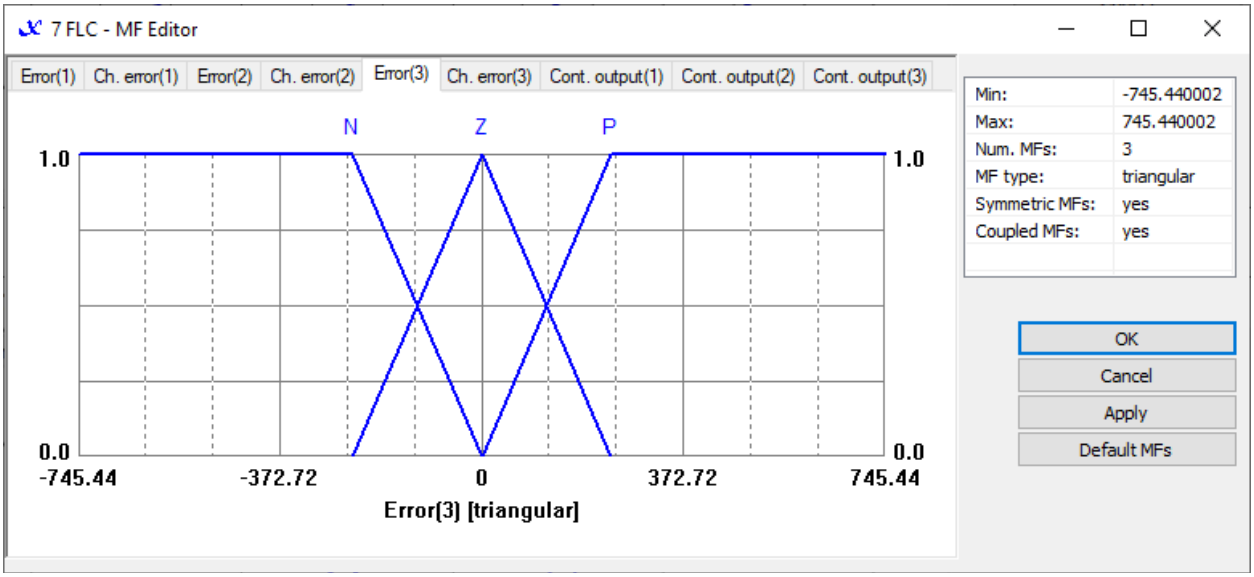

7 FLC - MF Editor
Error(1)
Ch. error(1)
Error(2)
Ch. error(2)
Error(3)
Ch. error(3)
Cont. output(1)
Cont. output(2)
Cont. output(3)
N
Z
P
1.0
0.0
-1.6623
-0.83115
0
0.83115
1.6623
Ch. error[2] [triangular]
Min: -1.662300
Max: 1.662300
Num. MFs: 3
MF type: triangular
Symmetric MFs: yes
Coupled MFs: yes
OK
Cancel
Apply
Default MFs

(b)

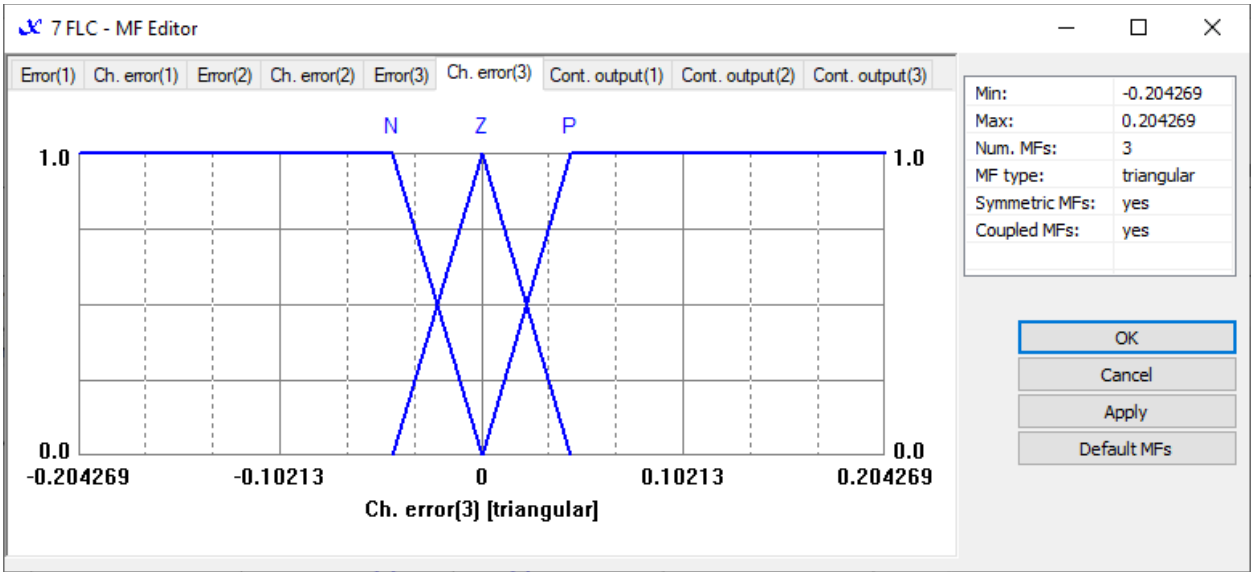


(c)

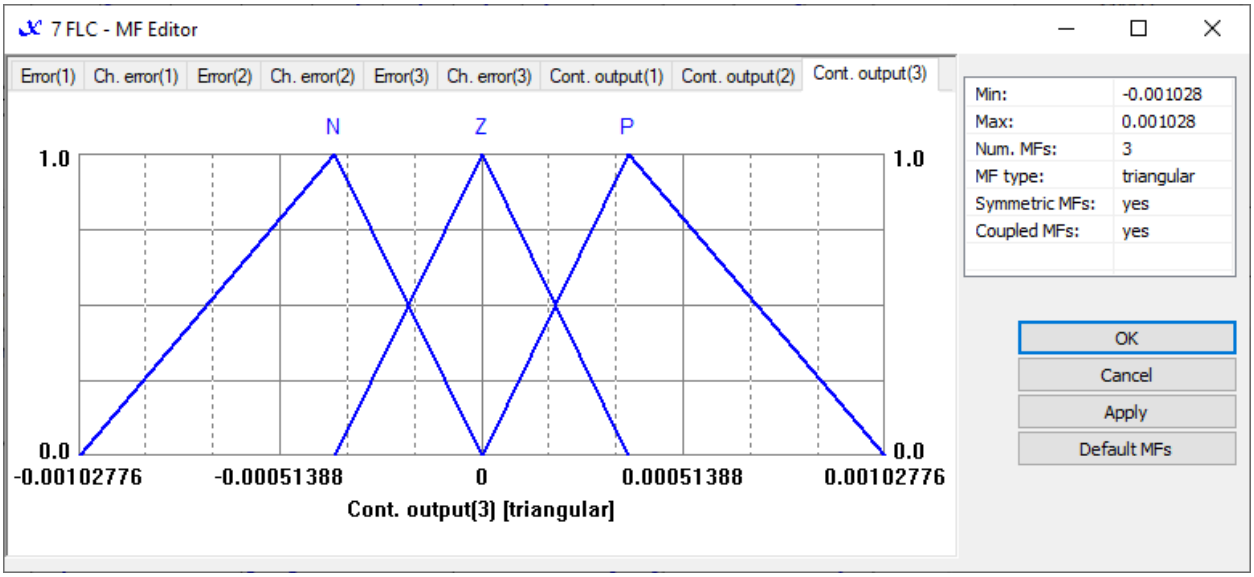
Figure 21: FLC membership functions (a) error (b) change of error and (c) controller output for loop 2



(a)



(b)



(c)

Figure 22: FLC membership functions (a) error (b) change of error and (c) controller output for loop 3

5.3 PID and Fuzzy Logic Controller Test Results

This section presents the results for the PID and FLC test system with the objective of comparing their performance. As mentioned in Section 3.2, there are three cases analyzed (base, disturbance, and noise case) and for each of them the comparison was made considering the characteristics of the feedback response signal (rise time, steady-state error, settling time and overshoot).

5.3.1 Base Case Results

The PID controllers were tuned using the procedure described in Section 4.2.1 and the results are shown in Figure 23 to Figure 25. The three signals shown are the set-point (black), the controller's output signal (red), and the feedback (process output) signal (blue). The units in the y-axis are in volts representing a 'per unit' base for the results to simplify the comparison among the three signals. The result of the PID controllers shows that the rise time for loop-1 feedback signal is ≈153.6 sec, loop-2 feedback signal is ≈373.9 sec and loop-3 is ≈71.3 Sec with an overshoot value of 5.2%, 0%, and 6.9% respectively. In addition, the signals settling time for loop-1 is ≈500 sec, loop-2 is ≈484.3 sec and loop-3 is 306 sec.

The response of the FLC for the same case show that the rise time for the loop-1, loop-2, and loop-3 feedback signals is ≈39.1 sec, ≈46.6 sec, and ≈97.3 sec., respectively. The systems settling time for all three loops are same as the rise time of the system with a 0% overshoot.

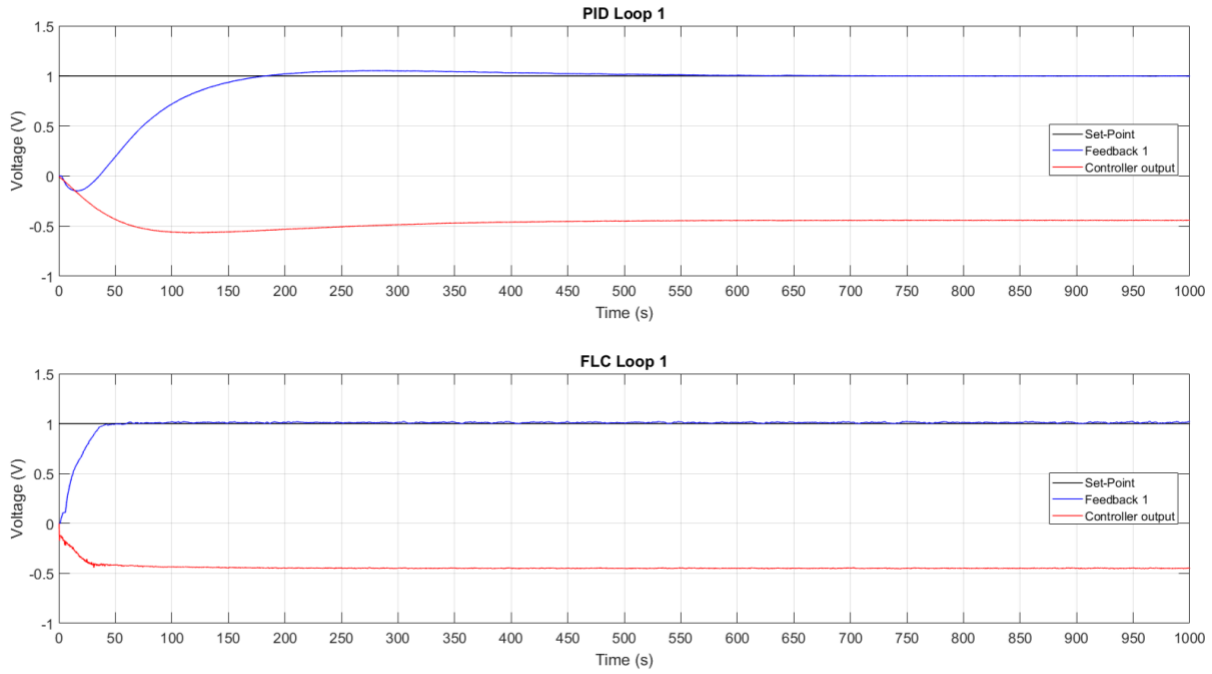


Figure 23: Response comparison for loop 1

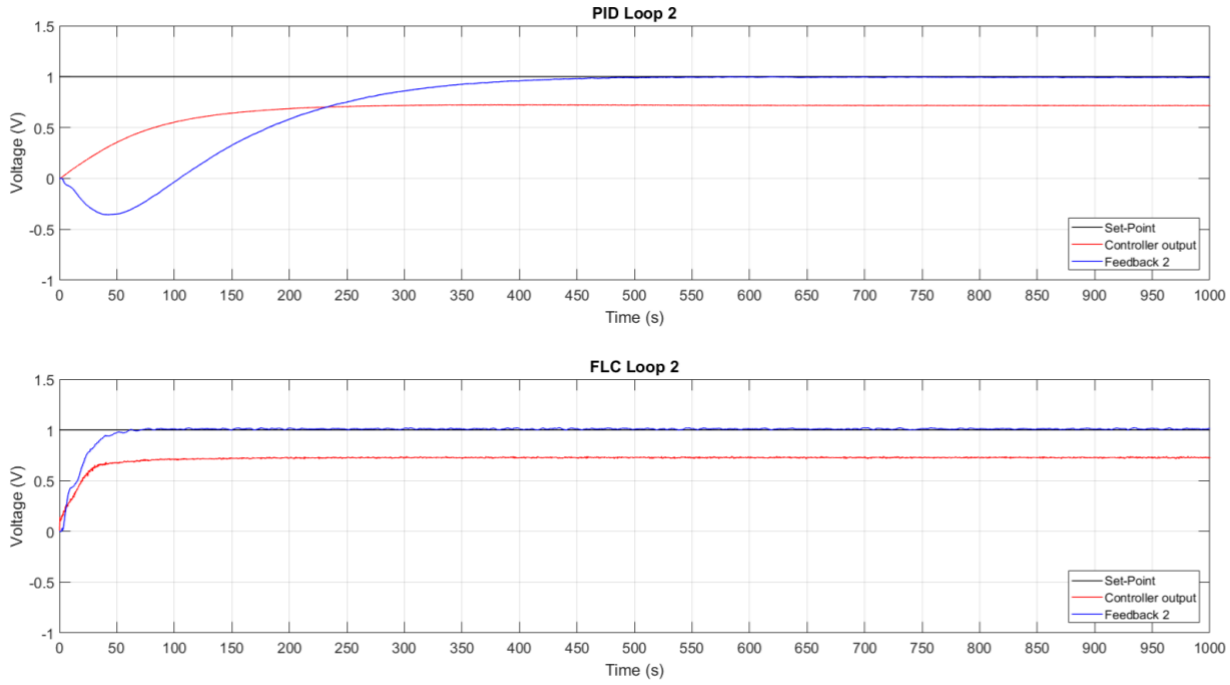


Figure 24: Response comparison for loop 2

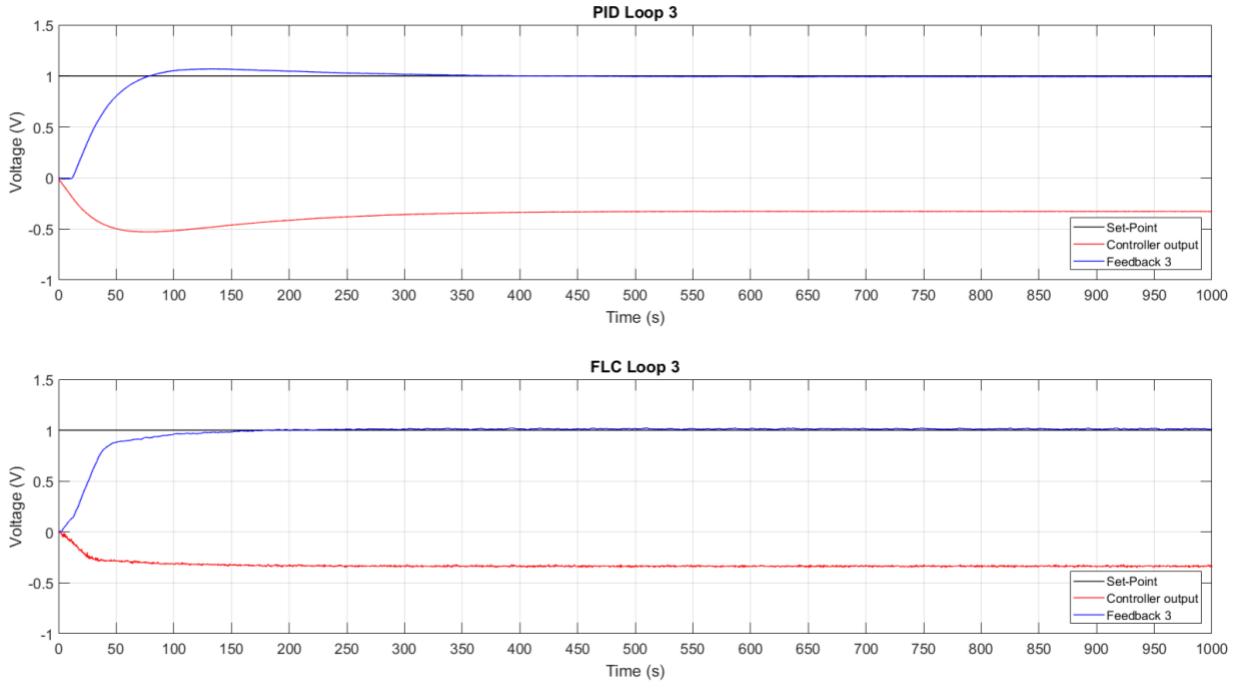


Figure 25: Response comparison for loop 3

Based on the selected performance criteria, the rise time of FLC loop-1 and loop-2 is 76.23% and 87.40% faster than the PID system, respectively and for loop-3 the rise-time of the PID system is 50.09% faster than FLC. However, the FLCs outperformed the PID controllers from an overshoot perspective since there was zero overshoot; and from a settling time, perspective where the FLC loops-1, 2 & 3 were 91.30%, 87.40% and 48.20% faster than the PID loops.

Table 11: Result summary: Base case

Loop-1	**Rise time (s)**	**Settling time (s)**	**Overshoot (%)**
PID	153.6	500	5.2
FLC	39.1	43.5	-
Loop-2	**Rise time (s)**	**Settling time (s)**	**Overshoot (%)**
PID	373.9	484.3	-
FLC	46.6	61	-
Loop-3	**Rise time (s)**	**Settling time (s)**	**Overshoot (%)**
PID	71.3	306	6.9
FLC	97.3	158.5	-

It is evident that from the rise and settling time results, the FLC system presents significant improvements when compared to the traditional PID controller. The FLC system is able to handle effectively the nonlinear, multivariable nature of the system and compensate for the

complexity of the interactions between the control loops. Notice that there are two main characteristics of the selected case study that significantly increase the complexity of the system (1) Two of the transfer functions in the direct branches have gains of opposite sign to the transfer functions representing the interactions, and (2) There is a the wide range of time delays, between 1 and 10 seconds, in the various system pathways (Table 8).

The independently controlled PID loops display non-minimum phase responses in loops 1 and 2 which combined with the linear algorithm of the PID, leads to longer rise and settling times, as well as the visible overshoots. In contrast, the FLC's nonlinear operation is able to generate the necessary transient control signals faster and reach the steady-state level earlier. The calculated control signal in loop 3 has caused it to plateau too early, giving a longer rise time; this characteristic may be improved with further tuning.

5.3.2 Disturbance Case Results

The Disturbance Case analyzed the performance of PID and FLC systems when a 20% disturbance (relative to the set-point reference) was applied for 25 seconds to the feedback signal. The disturbance was applied independently to each loop in turn, making sure that the system reaches steady state before applying the disturbance signal to the feedback signal of the other loop. The main objective was to identify the effect of a disturbance in both the target loop as well as the second loop, due to their interaction.

Figure 26 presents the PID controller response for the disturbance case. The disturbance was first applied to the feedback signal of loop-3 and an effect on loop-1 & 2 can be observed at t ≈750 sec. The disturbance caused an overshoot of 10.4% & 6.2% on the feedback signal of loop-1 and loop-2 respectively. The signal settled after ≈ 45.2 sec & 53.7 sec for loop-1 & 2 respectively. Also, after removal of disturbance of the feedback signal of loop-3 took 61.5 sec to recover.

The disturbance was then applied to the feedback signal of loop-2 and an effect on loop-1 & 3 can be observed at t ≈900 sec. However, there was no noticeable effect on the feedback signal of loop-1 & 3 and the removal of disturbance took ≈81.7 sec to recover. Similarly, the application of disturbance on the feedback signal of loop-1 at t ≈1050 sec, shows slight overshoot of 5.0% on loop-2 feedback signal with no significant effect on loop-3 feedback signal. The removal of disturbance signal took ≈ 60 Sec for the feedback signal of loop-1 to recover.

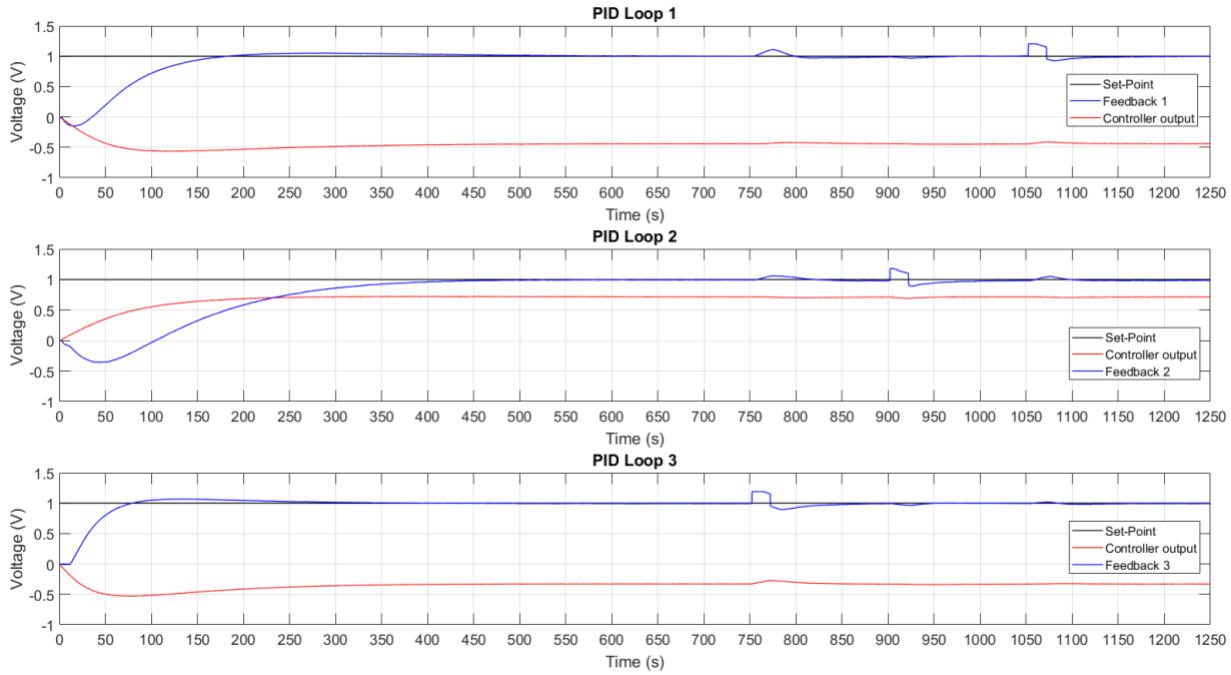


Figure 26: PID controller response: Disturbance case

Figure 27 presents the disturbance case response of the FLC. The disturbance was first applied to loop-3, but there was no effect of disturbance on the feedback signals of loop 1 & 2. It took ≈ 23.5 sec to recover after removal of the disturbance from loop 3.

When the disturbance was applied to loop-2 there was no effect on loops 1 & 3, after removal of disturbance signal from loop-2, it took ≈ 20.8 sec to recover after removal of disturbance signal. Similarly, when the disturbance was applied to loop-1 there was no effect on loops 2 & 3, after removal of disturbance signal from loop-1, it took ≈ 18 sec to recover after removal of the disturbance signal.

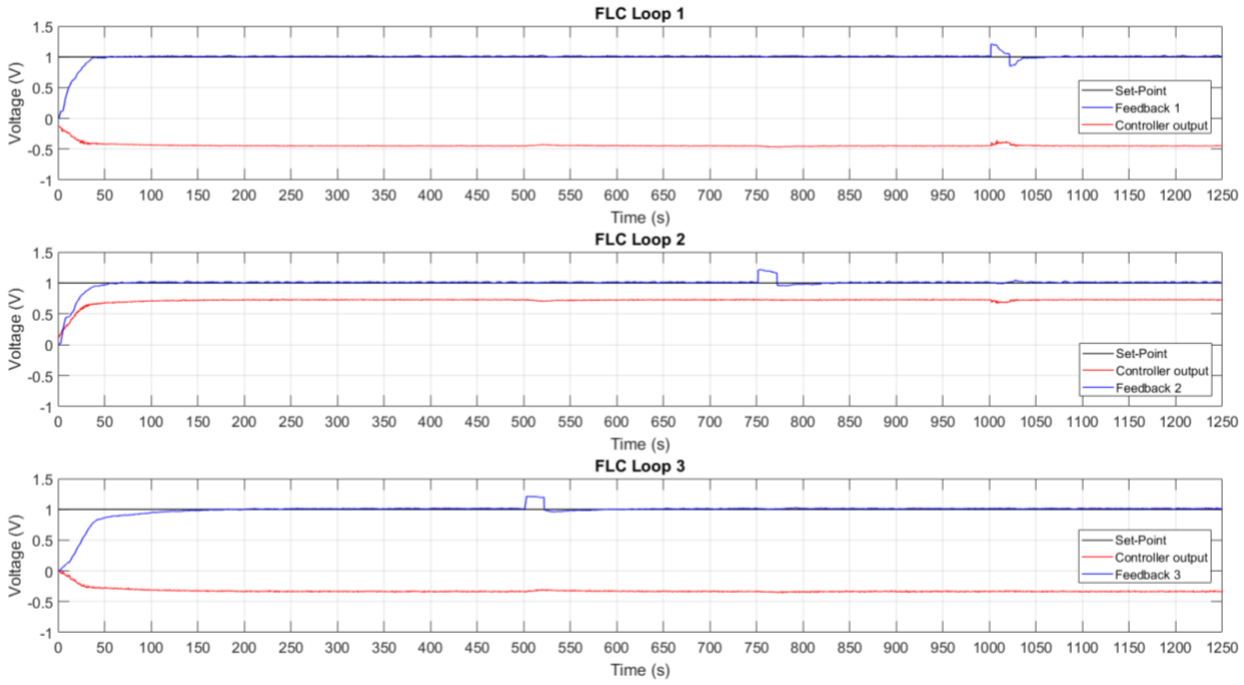


Figure 27: FLC response for disturbance case

Table 12, Table 13 and Table 14 summarize the results obtained for the disturbance case.

Table 12: Result summary: Loop-1 disturbance case

Disturbance applied to Loop-1		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	60	-
FLC	18	-
Resulting effect on Loop-2		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	35	5.0
FLC	-	-
Resulting effect on Loop-3		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	-	-
FLC	-	-

Table 13: Result summary: Loop-2 disturbance case

Disturbance applied to Loop-2		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	81.7	-
FLC	20.8	-
Resulting effect on Loop-1		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	-	-
FLC	-	-
Resulting effect on Loop-3		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	-	-
FLC	-	-

Table 14: Result summary: Loop-3 disturbance case

Disturbance applied to Loop-3		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	61.5	-
FLC	23.5	-
Resulting effect on Loop-2		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	53.7	6.2
FLC	-	-
Resulting effect on Loop-1		
Controller	**Recovery time (s)**	**Overshoot (%)**
PID	45.2	10.4
FLC	-	-

5.3.3 Noise Case Results

The Noise Case shows the effect of adding a 0.1 standard deviation noise signal to the feedback signal for all controllers. The results for the PID and FLC runs are show in Figure 28 and Figure 29. The results for both cases are very similar to the base case response but with the additional noise. Thus, with the noise level selected, there is no apparent difference between using a PID or FLC in the general response. However, when comparing the controller output of the PID and FLC controller, a significant attenuation difference is observed.

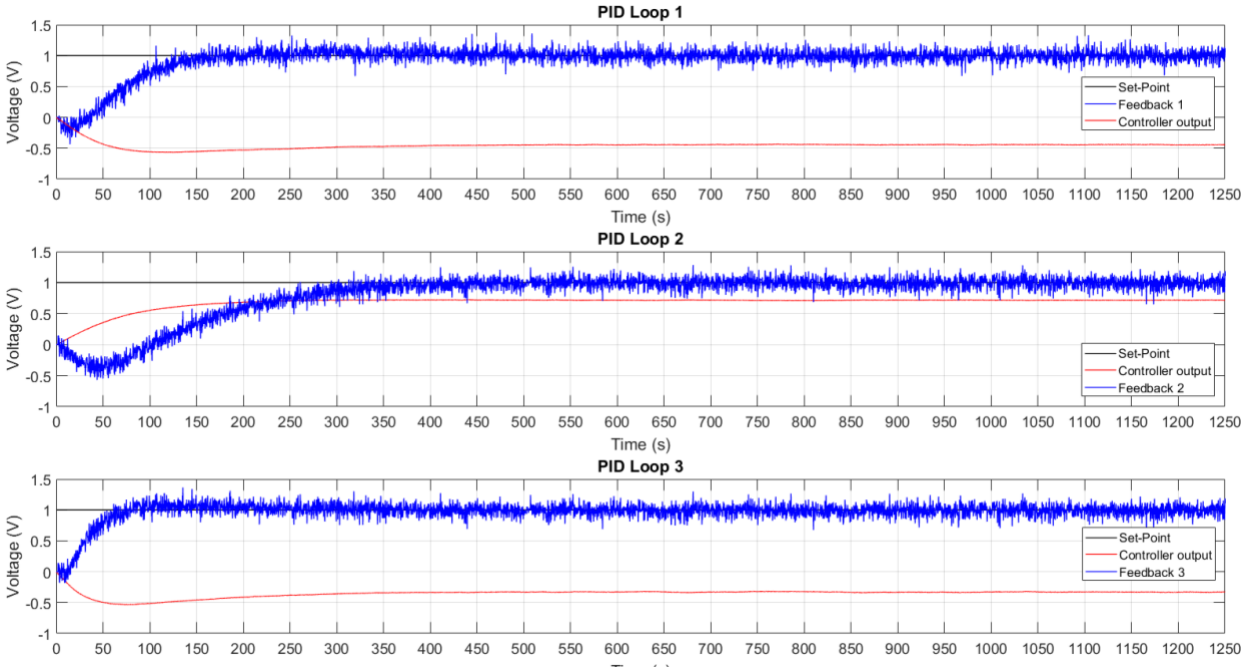


Figure 28: PID response: Noise case

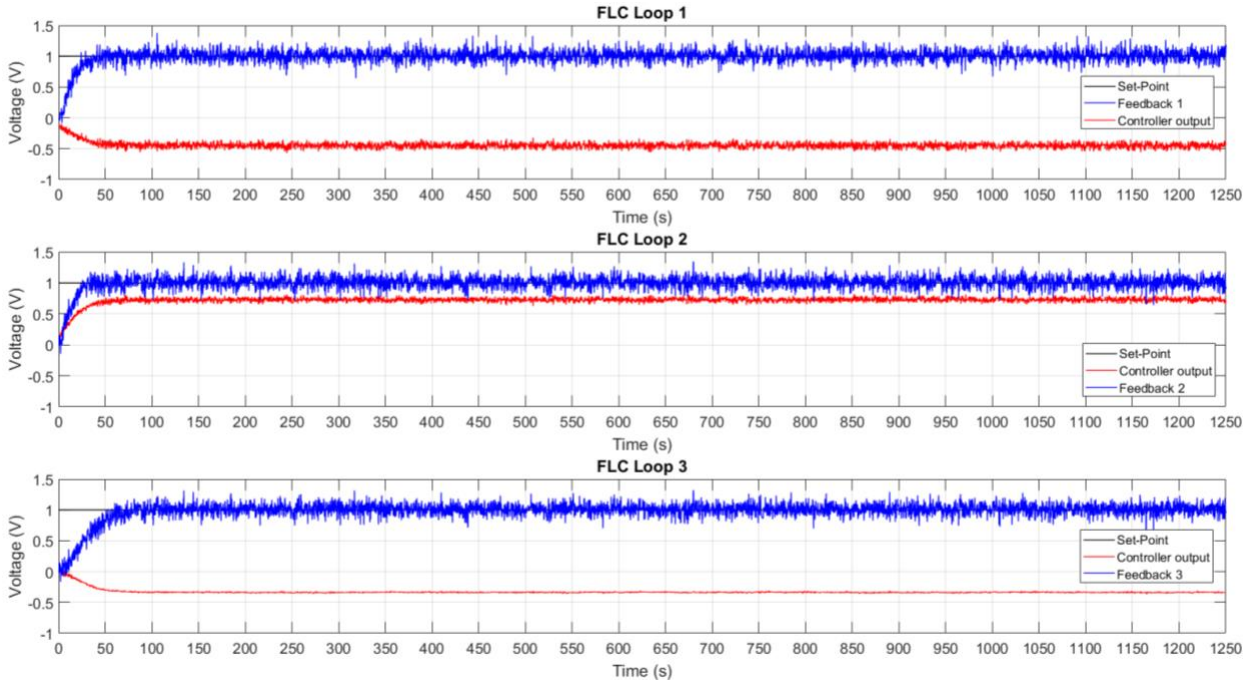


Figure 29: FLC response: Noise case

Table 15 shows the results of calculating the RMSE between the Baseline and the Noise Case scenario for both controller types. The results show that the PID controller performs better for loop 1 & loop 2, while the higher noise attenuation effect is seen in loop 3 of the FLC.

Table 15: Result summary: Noise case – Root Mean Square Error

Loop-1	
PID (RMSE)	0.0217
FLC (RMSE)	0.0388
$(RMSE_{FLC} - RMSE_{PID})/RMSE_{PID}*100$ [%]	78%
Loop-2	
PID (RMSE)	0.0172
FLC (RMSE)	0.0295
$(RMSE_{FLC} - RMSE_{PID})/RMSE_{PID}*100$ [%]	71%
Loop-3	
PID (RMSE)	0.0327
FLC (RMSE)	0.0143
$(RMSE_{FLC} - RMSE_{PID})/RMSE_{PID}*100$ [%]	-56%

6 Conclusion

This report presented the methodology and results for testing a Multi-Variable Control System for two case studies including a 2x2 and a 3x3 control system aimed to replicate two industrial control systems. The performed tests included a base case considering a step signal applied to all inputs, a feedback disturbance test, and a noise case. The main objective of these tests was to compare the control performance between the PID and FLC systems, which is summarized as follows:

- For the 2x2 system when comparing the PID and FLC system, the FLC system presents a significant improvement in settling time and noise attenuation. The settling time for the FLC is 35% to 44% faster and the noise impact in the controller output signal is 60% to 90% smaller. When comparing the impact of disturbance in the signals, the controller response is very similar for both, PID and FLC systems.

- For the 3x3 system, the settling time and response to disturbance from the FLC system is significantly better than the PID system. The settling time for the controllers is 48% to 90% faster than the PID system and the FLC controllers are able to handle disturbance issues without affecting the process. Finally, the noise impact on FLC and PID controllers is comparable for both cases.

References

[1] R. Babuska and E. Mamdani, "Fuzzy Control," 21 October 2011. [Online]. Available: http://www.scholarpedia.org/article/Fuzzy_control. [Accessed 10 December 2019].

[2] K. E. Arzen and M. Johansson, "Fuzzy Control: From Heuristic PID to Optimization-Based Nonlinear Control," in *Fuzzy Logic Control Advances in Application*, World Scientific Publishing Co. Pte. Ltd., 2001, p. 331.

[3] A. K. Ho, "PDHonline Course E331 (3 PDH)," 2014. [Online]. Available: https://pdhonline.com/courses/e331/e331content.pdf. [Accessed 11 December 2019].

[4] J. Zhang, N. Wang and S. Wang, "A developed method of tuning PID controllers with fuzzy rules for integrating process," in *American Control Conference*, Boston, 2004.

[5] The MathWorks, Inc., "PID Controller Tuning in Simulink," 2019. [Online]. Available: https://www.mathworks.com/help/slcontrol/gs/automated-tuning-of-simulink-pid-controller-block.html.

[6] The MathWorks, Inc., "Control System Toolbox," March 2020. [Online]. Available: https://www.mathworks.com/products/control.html.

[7] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Made Studies,* vol. 7, no. 1, January 1975.

[8] H. Singh, M. M. Gupta, T. Meitzler, Z.-G. Hou, K. K. Garg, A. M. G. Solo and L. A. Zadeh, "Real-Life Applications of Fuzzy Logic," *Hindwani,* 2013.

[9] O. M. Okwu and A. N. Nwachukwu, "A review of fuzzy logic applications in petroleum exploration, production and distribution operations," *Journal of Petroleum Exploration and Production Technology,* p. 14, 2018.

[10] P. M. Larsen, "Industrial applications of fuzzy logic controller," *International Journal of Man-Machine Studies,* pp. 3-10, 1980.

[11] C. Kavka, P. Roggero and J. Apolloni, "An architecture for fuzzy logic controller evolution and learning in microcontroller based environment," 2003.

[12] A. T. Azar and S. Vaidyanathan, Advances in Chaos Theory and Intelligent Control, Switzerland: Springer International Publishing , 2016.

[13] The MathWorks, Inc., "Stepinfo," 2019. [Online]. Available: https://www.mathworks.com/help/ident/ref/stepinfo.html.

[14] R. K. Wood and M. W. Berry, "Terminal composition control of a binary distillation column," *Chemical Engineering Science,* vol. 28, pp. 1707-1717, 02 July 1973.

[15] S. Raghupathy and M. K. Sanu, "Design and development of model predictive controller for binary distillation column," *International Journal of Science and Research,* pp. 445 - 451, 2013.

Appendix I: Real-Time Simulator Characteristics and Specifications

Matlab/Simullink software version details:

```
>> ver
-----------------------------------------------------------------------------------------------------
MATLAB Version: 9.2.0.556344 (R2017a)
MATLAB License Number: 40755337
Operating System: Microsoft Windows 10 Pro Version 10.0 (Build 17134)
Java Version: Java 1.7.0_60-b19 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode
-----------------------------------------------------------------------------------------------------
MATLAB                                                Version 9.2          (R2017a)
Simulink                                              Version 8.9          (R2017a)
ARTEMIS Blockset                                      Version 7.3.2.1271     (R2017a)
Control System Toolbox                                Version 10.2         (R2017a)
Data Acquisition Toolbox                              Version 3.11         (R2017a)
Database Toolbox                                      Version 7.1          (R2017a)
Fuzzy Logic Toolbox                                   Version 2.2.25       (R2017a)
Instrument Control Toolbox                            Version 3.11         (R2017a)
MATLAB Coder                                          Version 3.3          (R2017a)
OPC Toolbox                                           Version 4.0.3        (R2017a)
Optimization Toolbox                                  Version 7.6          (R2017a)
RT-EVENTS Blockset                                    Version 4.3.2.749     (R2017A.x)
RT-LAB                                                Version v11.3.3.62     (R2017a.x)
Signal Processing Toolbox                             Version 7.4          (R2017a)
Simscape                                              Version 4.2          (R2017a)
Simscape Power Systems                                Version 6.7          (R2017a)
Simulink 3D Animation                                 Version 7.7          (R2017a)
Simulink Coder                                        Version 8.12         (R2017a)
Simulink Control Design                               Version 4.5          (R2017a)
Simulink Design Optimization                          Version 3.2          (R2017a)
Simulink Desktop Real-Time                            Version 5.4          (R2017a)
Spreadsheet Link                                      Version 3.3.1        (R2017a)
Statistics and Machine Learning Toolbox               Version 11.1         (R2017a)
Symbolic Math Toolbox                                 Version 7.2          (R2017a)
System Identification Toolbox                         Version 9.6          (R2017a)
eFPGAsim                                              Version v1.5.4.41-4     (R2017a)
```

Figure 30: MATLAB add-ons tools and version

The real-time simulator form OPAL-RT consists of two pieces of hardware; OP5700 RCP/HIL FPGA-Based Real-Time Simulator and OP8660 HIL Controller and Data Acquisition Interface. The OP5700 simulator contains a target computer in the lower section, which uses OPAL-RT's RT-LAB tools to run simulations and the upper section contains the FPGA and the I/O conditioning modules (Figure 31).
The target computer in lower section can be connected to a network of simulator or can be used as a standalone system and some of its features are:

- ATX motherboard
- Linux-based real-time operating system
- Intel® Xeon® E5 CPU with 4, 8, 16 and 32 processor cores, up to 3.2GHz
- 10MB Cache Memory per 4 cores
- up to 32GB of DRAM,

- 512GB SSD disk,
- 6 PCIe slots1, used to connect the internal FPGA board and PCIe or PCI third party I/O and communication cards.

Figure 31: Opal-RT real time simulator

Features of the FPGA and the I/O conditioning modules include:

- Xilinx® Virtex®7 FPGA programmable from the target computer via PCIe. The FPGA is used to
- execute models designed with the OPAL-RT's RT-XSG tool, manage the I/O lines and execute
- embedded FPGA-based simulations. It exchanges data with the real-time simulations running on
- the target computer CPUs via the PCIe link.
- Flat carrier board capable of connecting any combination of up to 8 digital and analog conditioning modules.
- Each module controls 16 or 32 lines for a total of up to 256 I/0 lines.
- 16 SFP ports for high speed communication with other FPGA-based systems or with external devices.

In general, some of the main features of the simulator are as shown in the table below:

OP5700 SPECIFICATIONS

Product name	OP5700
CPU	Intel® Xeon® E5, 4 core / 3GHz to 32 cores / 2.3GHz
FPGA	Xilinx® Virtex®7 FPGA on VC707 board, 485T, 485 760 Logic cells, 2 800 DSP slices
I/O lines	256 lines, routed to 8 analog or digital, 16 or 32 channel, conditioning modules
High speed communication ports	16 SFP sockets, up to 5GBps
I/O connectors	4 panels of 4 DB37F connectors
Monitoring connectors	4 panels of RJ45 connectors
PC interfaces	Standard PC connectors (monitor, keyboard, mouse and network)
PCIe slots	6 on-board PCIe slots. 4 or 5 slots are available for third party cards or riser boards depending on CPU configuration.
Hard disk	512 GB SSD
Power supply	Universal input and active power factor correction 650W continuous power Input : 100-240VAC, 50-60Hz, 10A/5A Power: 600W
Dimensions	22.27 x 47.7 x 49.3cm (8.75" x 18.8" x 19.4") HxWxD
Weight	17kg (37.5 lbs)
Operating temperature	10 to 35 °C (50 to 95°F)
Storage temperature	-55 to 85°C (-67 to 185°F)
Maximum rated ambient temperature	40C° (104°F)
Relative humidity	10 to 90% non-condensing
Maximum altitude	2000 m (6562 ft.)

Figure 32: OP5700 Specifications

The **OP8660 HIL Controller and Data Acquisition Interface** is designed to be used with a real-time simulator (such as OP5700) to provide supplementary signal conditioning.
The rear of the chassis provides DB37 connectors to connect the OP8660 to the real-time simulator, while the front provides connectors (banana jack or DB9) for connecting devices such inverters, encoders, monitoring and measuring devices for monitoring or testing. Which in our case is the FLC controller.

The unit includes four high current and high voltage input conditioning modules, which convert high current and high voltage signals coming from the external device to ±10V voltage signals compatible with the real-time simulator's inputs. The HIL Controller is useful link between the unit under test (ECU, motor controller, etc.) and the simulator, you can insert a fault at any point in the test to assess how the unit reacts to the fault. Some of the main features and specifications of this unit are as below:

- DB9 inverter and encoder connectors.
- Banana jack high current and high voltage measurement connectors.
- Banana jack analog input (+/-16V) monitoring connectors
- Banana jack analog output (+/-16V) interface connectors
- Banana jack digital input (0-30V) monitoring connectors
- Banana jack digital output (0-5V) interface connectors

- DB37 connectors for quick connections to the real-time simulator (all DB37 use common pin assignments).

General Specifications:

Product name	OP8660 HIL Controller
Part number	310-0055
Form factor	4 U
Dimensions	13.33 x 48.26 x 30.8cm HxWxD (5.25” x 19” x 12.125”)
I/O connectors	DB37F, DB9, banana jacks
Operating temperature	10 to 40 ºC (50 to 104ºF)
Storage temperature	-55 to 85ºC (-67 to 185ºF)
Relative humidity	10 to 90%, non-condensing
Maximum altitude	2,000 m (6562 ft.)

Sensor Specifications:
This specification applies to the High Current and High Voltage Measurement connectors in the front of the OP8660.

	Current Sensors Specification	Voltage Sensors Specification
Input range:	15 A	Up to 600 volts
Signal output range:	± 10 Volts	± 10 Volts
Isolation:	Galvanic, 2.5 Kv	Greater than 200 volts after the resistive divider
Bandwidth:	DC to 100 kHz	DC to 100 kHz
Linearity:	< 0.2%	< 0.2 %
Rise time:	< 2 Microseconds	< 2 microseconds
Power supplies:	±15 Volts	±15 Volts