HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

OFFICE FOR INTERNATIONAL STUDY PROGRAMS

PROJECT:

DESIGN A DIRECT ADAPTIVE FUZZY CONTROLLER FOR A DC MOTOR

Student's name: Nguyễn Văn Phong

Student ID: 185660

Class: CC18OTO

Instructor: Ph.D. Trần Đăng Long

Ho Chi Minh City, 2022

VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY
HO CHI MINH UNIVERSITY OF TECHNOLOGY
Faculty of Transportation Engineering
Department of Automotive Engineering

SOCIALIST REPUBLIC OF VIETNAM
Independence – Freedom – Happiness

PROJECT MISSION

1. **Student's name:** Nguyễn Văn Phong **- Student ID:** 1852660
2. **Major:** Automotive Engineering **- Class:** CC18OTO
3. **Thesis title:** Design a direct adaptive fuzzy controller for a DC motor
4. **Content:**
   - ❖ Introduction of the project
   - ❖ Scope of implementation
   - ❖ Working conditions and technical requirements
   - ❖ Theoretical basis
     - Direct adaptive fuzzy controller for a DC motor
   - ❖ General layout design
     - General layout diagram
   - ❖ Result:
     - Simulate the working process of the system using Matlab/Simulink software.
     - Compare Direct adaptive fuzzy controller with PID controller.
5. **Product:**

✓ Presentation report.	✓ Poster.
✓ App designer production.	✓ Simulink simulation.

6. **Assigned day:** April, 2022.
7. **Finished day:** June, 2022.

The content and requirements of the thesis is already approved by the Head of Department of Automotive Engineering.

HCMC, day..... month…… year 2021 *HCMC, day… . month…… year 2021*

Head of Department **Instructor**

Contents

I. INTRODUCTION:

1.1. Objective:

Despite the rapid development of inverter-induction motor drive systems in the field of electric drives, DC motor drives continue to play an important role in many industrial systems such as robotics, machine tools, and especially in low power electrical drive systems. In fact, the DC motor not only has excellent control characteristics, but also a wide range of supply voltage and power for a variety of applications.

Traditional controller design often encounters a deadlock when encountering problems with high system complexity, frequent change of state and object structure, and high accuracy requirements. high and one method for overcoming this challenge is direct adaptive fuzzy control.

As a result, the goal of this project is to design a direct adaptive fuzzy controller for a DC motor.

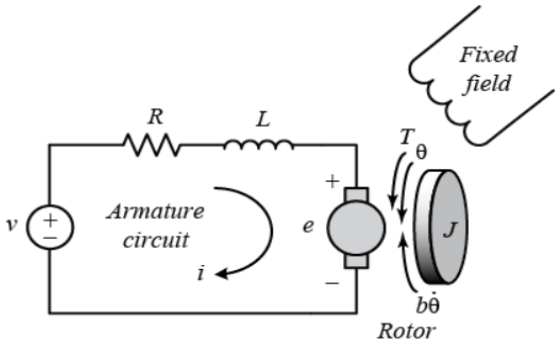


Figure 1. The electric equivalent circuit and the free-body diagram of the rotor of DC motor

1.2. Scope of implementation:

This project aim is to simulate in order to fully understanding working principle, operating of a direct adaptive fuzzy controller for a DC motor in reality. Therefore, we decide to use Matlab/Simulink to simulate this system and use App designer in Matlab to create user interface to export data to Simulink simulation.

1.3. Working conditions:

- Affected by noise generated during simulation.
- DC motor working in no load condition.

1.4. Technical requirements:

- Working normally in above conditions.
- Control the speed of DC motor follows the desired speed in different operation.
- Measuring speed range: 0 – 1000 RPM with the error of the controller within the allowable range.

II. THEORETICAL BASIS:

2.1. Adaptive controller:

2.1.1. Controller adaptability

Conventional controllers, after being designed, guarantee good control quality only when the object has properties similar to those of the model used at the time of design. This model is the model that estimates or approximates the object, which is determined according to certain

conditions (working conditions, working mode of the object). In fact, the conditions and working modes of the subject are always changing, just as the subject itself is always changing with time, aging, and the effects of noise. As a result, the object properties are also constantly changing and may no longer be reasonably described by the approximation model used in the design. In addition, in the process of working, the whole system is affected by noise that distorts the measured values as well as the control values. Then the controller will not be able to maintain the desired control quality. Therefore, a high-quality controller needs to be able to adjust its own control parameters, to ensure the desired control quality against these uncertain changes. Such controllers are called adaptive controllers.

An adaptive control system is a special feedback nonlinear control system with internal states divided into two groups with different rates of change. In which, the parameters of the controller in the state group have a slow rate of change compared to the state group of the object. Therefore, inside an adaptive controller there are two processes working at different speeds: the feedback process working at high speed and the parameter setting process of the controller working at lower speed.

The adaptive system can use only classical controllers, the adaptation methods are clearly described by precise mathematical equations, or it can use fuzzy systems to make the system operation more flexible operation due to the integration of expert experience into the system. With or without the participation of fuzzy systems, adaptive control system is a highly developed control system with special potential, but associated with these advantages is the volume of design computation very large.

2.1.2. Compare adaptive controller with PID controller on the theoretical basis:

PID (Proportional Integral Derivative) controller is a combination of three controllers: proportional, integral and differential, capable of adjusting to the lowest possible error, increasing response speed, reducing overshoot, and limiting oscillation. The PID controller is a process control technique that engages in “proportional, integral, and differential” processing actions. That is, the resulting error signals will be minimized by the effect of the proportional effect, the effect of the integral effect and clarified by a rate obtained with the fractional effect before.

The PID can be understood in a simple way as follows:

- P: It is a proportional adjustment method, which helps to generate an adjustment signal proportional to the input error according to the sampling time.
- I: Is the integral of the error over the sampling time. Integral control is a tuning method to generate tuning signals so that the error is reduced to 0. This tells us the total instantaneous error over time or the accumulated error in the past. The smaller the time, the stronger the integral adjustment effect, corresponding to the smaller deviation.
- D: Is the differential of the error. The differential control generates an adjustment signal that is proportional to the rate of change of the input bias. The larger the time, the stronger the differential tuning range, which corresponds to the faster the regulator responds to input changes.

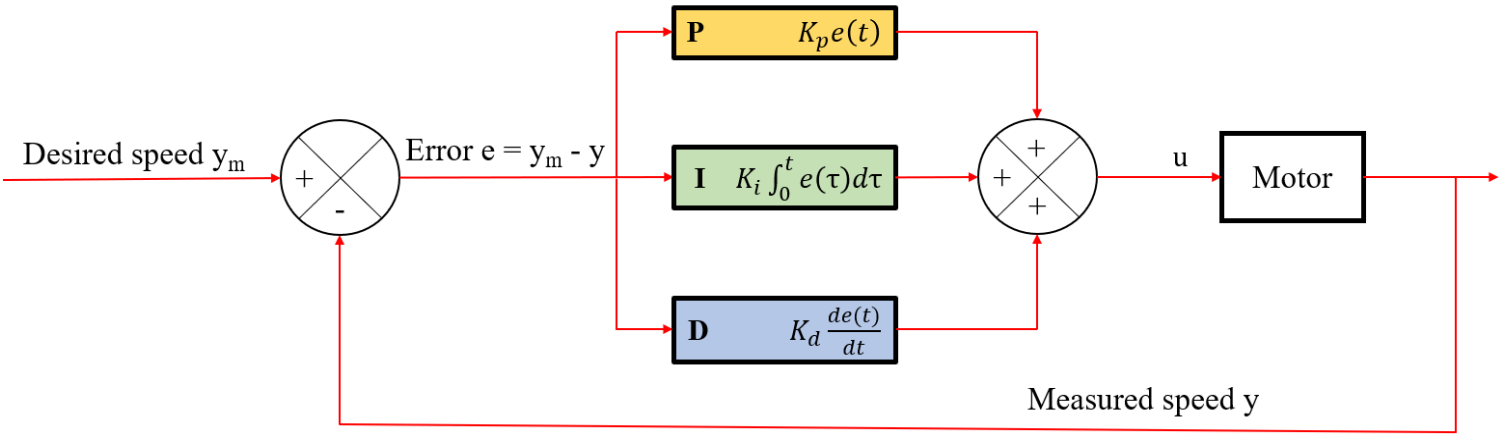


Figure 2. Structure of PID controller

Because there is no adaptability, the disadvantages of the PID controller is shown when the object's parameters are changed or when there is an impact of noise on the system. While the adaptive controller can respond to the output signal y, it always follows the desired signal y_m with any change in the parameters of the object (DC motor) or is less affected by noise, but with the PID control is not like that, it is very affected by noise and for each change of object parameters, a appropriate set of P, I, D parameters must be found. That is the biggest advantage of the adaptive controller over the PID controller that we will demonstrate through simulation in Simulink.

2.2. Fuzzy controller:

Fuzzy control is a special form of nonlinear control that can be applied to any object. In principle, the fuzzy control system is no different from other conventional automatic control systems. The difference here is that the fuzzy controller works like a "brain" in the form of artificial intelligence, works depending on experience and methods of drawing conclusions according to human thinking, then installed into the computer on the basis of fuzzy logic. Fuzzy control system is a control system designed without knowing the object model in advance.

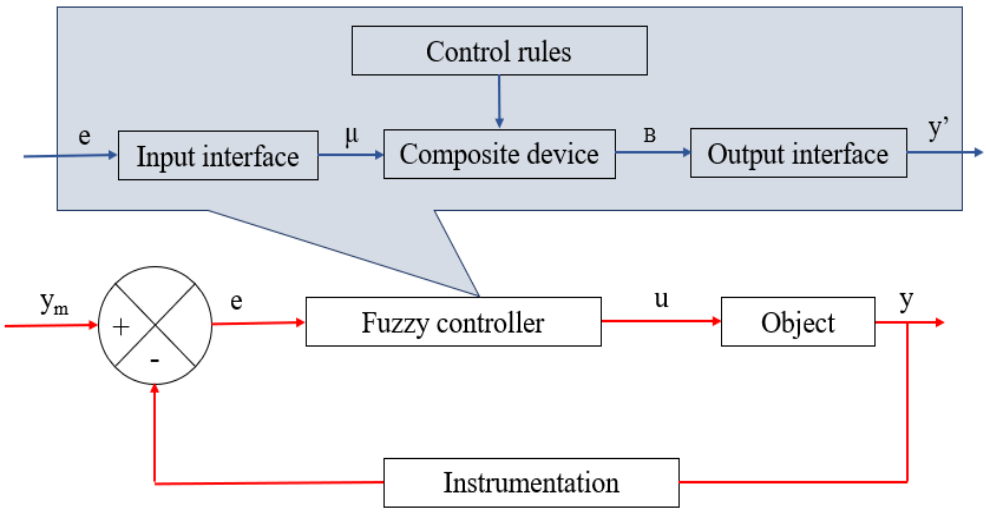
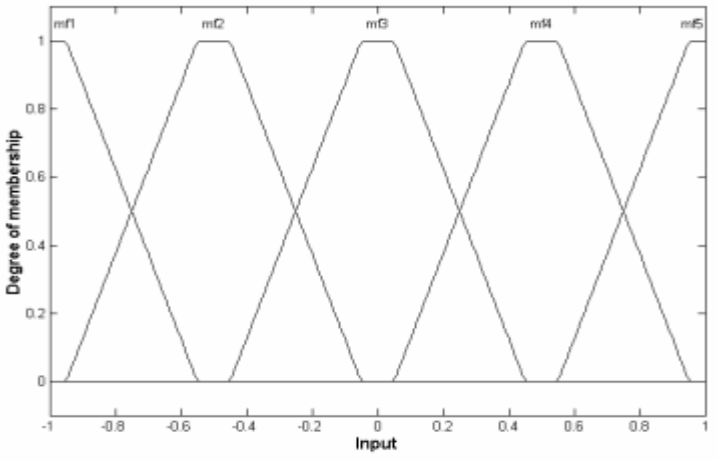


Figure 3. Structure of a fuzzy controller

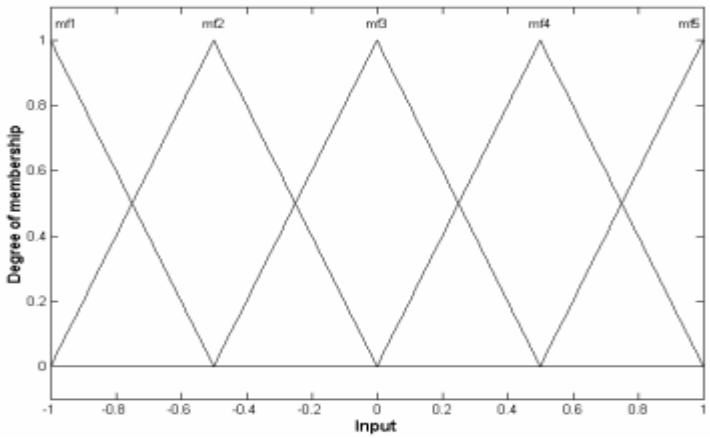
The fuzzy control system is designed including:

- The input interface includes fuzzify and additional auxiliary stages to perform dynamic problems such as differential, integral, ...
- Composite device whose nature the composition rule implementation is built on the basis of control rules.
- Output interface includes defuzzification and direct interface with the object.

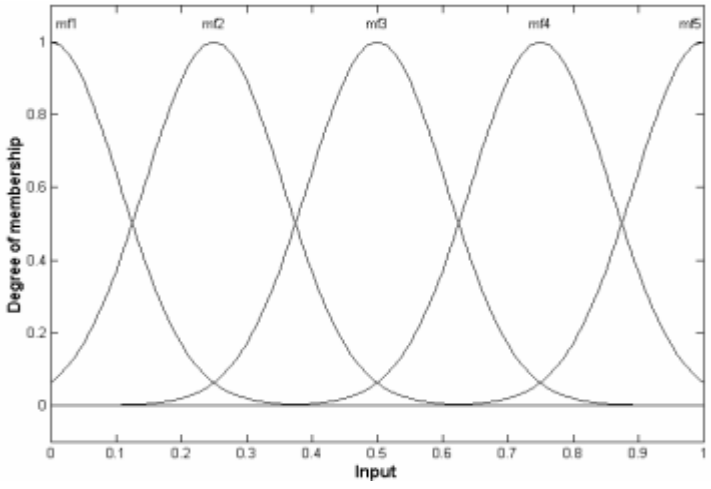
The operation of a fuzzy system can be briefly described as follows: Fuzzy system uses fuzzy sets to fuzzify input explicit values by evaluating the degree (degree of membership) of these explicit values. Based on the evaluation results of input values and established logical relationship between input and output (fuzzy rules), the output value of fuzzy system is determined.



(a) Trapezoid



(b) Triangle



(c) Gaussian

Figure 4. Fuzzy sets

The principle of synthesizing a fuzzy controller is completely based on mathematical methods based on the definition of input/output linguistic variables and the choice of control rules. In addition, the structure and way of working of the fuzzy system is based on human knowledge and experience about the object, or the object's controller, making the design process simpler,

independent of much into the process of identifying or solving complex control problems. In addition, the fuzzify of signals or states in the system makes the system "less sensitive" to changes in objects, interfering with the system or objects. These are the great advantages of fuzzy logic control systems.

2.3. Direct adaptive fuzzy controller:

Fuzzy controllers are designed with the aim of being able to work well in conditions affected by noise or unknown changes in the object or system. Meanwhile, the goal of adaptive control systems is to ensure control quality under such operating conditions. Therefore, a high-quality fuzzy controller needs to be adaptable.

The adaptive fuzzy controller has a special nonlinear structure, so it is suitable for any object, while the normal adaptive controller changes depending on the object. In addition, human knowledge and experience about the working characteristics and control principles of the object are incorporated into the control law, which is not possible with conventional adaptive controllers. These two differences, especially the second one, are the advantages of adaptive fuzzy controller over conventional adaptive controllers.

Depending on the information and understanding of people about the working characteristics of the object or the principle of controlling the object (the controller), adaptive fuzzy sets are divided into three types: Indirect adaptive fuzzy controller, Direct adaptive fuzzy controller and Combined indirect/direct adaptive fuzzy controller.

This project will focus on direct adaptive fuzzy controller using fuzzy system with fuzzy rules describing the working principle of the controller, directly giving the desired control law. This controller includes a fuzzy system with the task of approximating the desired control law, a mechanism (adaption law) to adjust the parameters of the fuzzy system built on the Lyapunov stability criterion and the object control experience incorporated into the fuzzy rules.

2.3.1. Fuzzy System:

In order to approximate the unknown control law, to ensure the control quality when the object parameters change or have the impact of noise during the working process, at the same time, in order to take advantage of the knowledge and experience of object control, a fuzzy system $u_D(X, \theta)$ with configurable θ parameters is used as the controller.

The fuzzy system $u_D(X, \theta)$ has the following characteristics:

• n inputs: each state x_i is an input; each i-th input has m_i fuzzy set $A_i^{l_i}$, index $i = 1..n$ indicates the order of the input, index $l_i = 1 \dots m_i$ indicates the order of the fuzzy set in m_i fuzzy set of i-th input.

• One output is a control signal u. The output fuzzy sets are singleton lines with the corresponding value $\theta^{l_1 l_2 \dots l_n} \in R$ with $l_i = 1 \dots m_i$ and $i = 1..n$. These $\theta^{l_1 l_2 \dots l_n}$ values are modifiable parameters of the fuzzy system and are included in the parameter vector $\theta \in R^{(\Pi_{i=1}^{n} m_i) \times 1}$.

• Collection of fuzzy IF-THEN rule is described as follows:

$$\text{IF } x_1 \in A_1^{l_1} \text{ and } \dots \text{ and } x_n \in A_n^{l_n} \text{ THEN } u = \theta^{l_1 \dots l_n}$$

Where: $l_i = 1 \dots m_i$ and $i = 1..n$. The object control experience is incorporated into the fuzzy

system through the process of selecting the fuzzy set of the inputs and the initial value of the parameter vector θ. There are a total of $\Pi_{i=1}^{n} m_i$ fuzzy IF-THEN rules (equal to the number of parameters $\theta^{l_1 l_2 \dots l_n}$).

• Using the Max–Product operation and defuzzification by the elevation method, the apparent value of the output is determined as follows:

$$u = u_D(X, \theta) = \frac{\sum_{l_1=1}^{m_1} \dots \sum_{l_n=1}^{m_n} \theta^{l_1 \dots l_n} . [\Pi_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)]}{\sum_{l_1=1}^{m_1} \dots \sum_{l_n=1}^{m_n} [\Pi_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)]})$$

Set $\xi_{l_1 \dots l_n}(X) = \frac{\Pi_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)}{\sum_{l_1=1}^{m_1} \dots \sum_{l_n=1}^{m_n} [\Pi_{i=1}^{n} \mu_{A_i^{l_i}}(x_i)]}$ and put in vector $\xi(X) \in R^{(\Pi_{i=1}^{n} m_i) \times 1}$. The apparent output value is rewritten as a vector:

$$u = u_D(X, \theta) = \theta^T . \xi(X)$$

Thus, the fuzzy system $u_D(X, \theta)$ will use all the states of the object as input, the output fuzzy sets are the parameters to be adjusted, and the fuzzy rule set includes all the composition rules. The use of singleton output fuzzy sets, the Max-product operation and the elevation method for defuzzification make the system describe in vector form and the computation is done conveniently and easily.

The initial values of some parameters $\theta^{l_1 \dots l_n}$ are selected based on experience, knowledge of controlling the object, the remaining $\theta^{l_1 \dots l_n}$ parameters are chosen at random (or in some way). Therefore, the experience and knowledge of object control are combined into the fuzzy controller through the selection of initial values for the parameters $\theta^{l_1 \dots l_n}$.

2.3.2. Adaption law:

The fuzzy system $u_D(X, \theta)$ is built with the initial values in the parameter vector θ mostly chosen at random, meanwhile, the task of this fuzzy system must ensure approximation of the unknown desired control law to the y output of the object will follow the desired output y_m. In addition, during the working process, the parameters of the object may change due to the control working, the noise affecting the object changing the rule, or the desired output signal changing the characteristics (variable faster or slower, cyclical or non-cyclical). Therefore, the parameter θ vector needs to be adjusted continuously to always converge to the ideal parameter vector θ^*, ensuring that the output error of the object always converges to zero and the system is always stable. To fulfill this requirement, it is necessary to have an adaptation rule or parameter tuning law to adjust the parameter vector θ to converge to the ideal parameter vector θ^*.

With a direct adaptive fuzzy controller, the parameter tuning law so that the parameter vector θ converges to the ideal parameter vector θ^*, is determined according to the Lyapunov stability criterion.

The parameter adjustment rule is built on the Lyapunov stability criterion with a positive definite Lyapunov function V(E) chosen as follows:

$$V(E) = \frac{1}{2} . E^T . P . E + \frac{b}{2.\gamma} . (\theta^* - \theta)^T . (\theta^* - \theta)$$

Where: $\gamma > 0$ is the update coefficient or the convergence coefficient, E is error vector, $P \in R^{n \times n}$ is a real, positive, symmetric matrix that satisfies the equation $\Lambda^T . P + P . \Lambda = -Q$, with $Q \in$

$R^{n\times n}$ is real, positive, symmetric matrix, $\Lambda = \begin{bmatrix} 0 & 1 \\ -k_2 & -k_1 \end{bmatrix}$. In order for $\dot{V}(E)$ to determine the negative, the parameter adjustment rule is defined as follows:

$$\dot{\theta} = \gamma . E^T . p_n . \xi(X)$$

With $P_n \in R^{n\times 1}$ is the n-th column of the matrix P. Then, E will go to zero, and the output y of the object will stick to the desired output y_m. The parameter adjustment rule is built on Lyapunov standard to ensure the system to achieve a stable state during working process.

The adaption law determined according to the Lyapunov stability criterion ensures the stability of the system, does not require parameter estimators, and the computational volume is small.

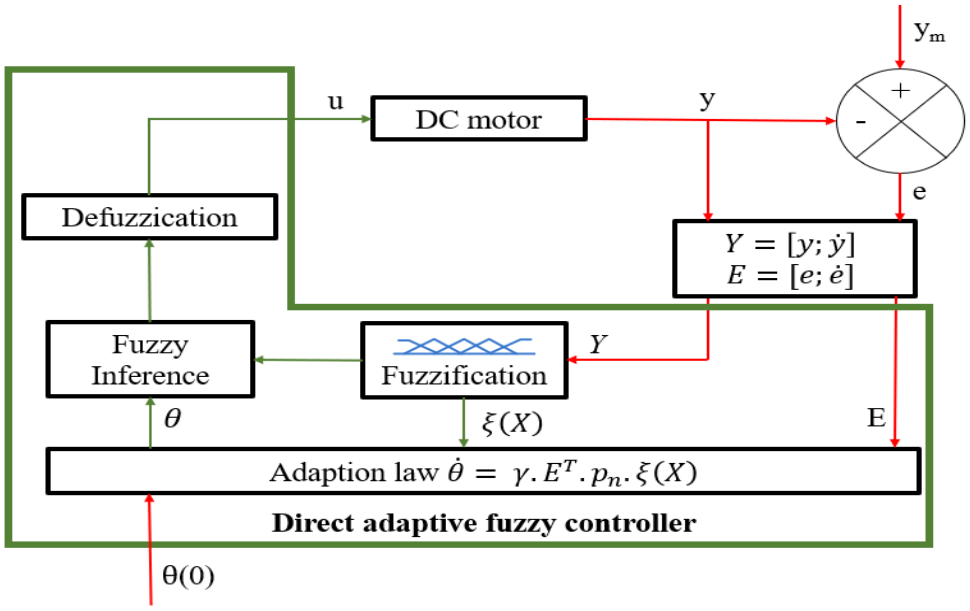


Figure 5. Control structure of direct adaptive fuzzy controller

III. GENERAL LAYOUT DESIGN:

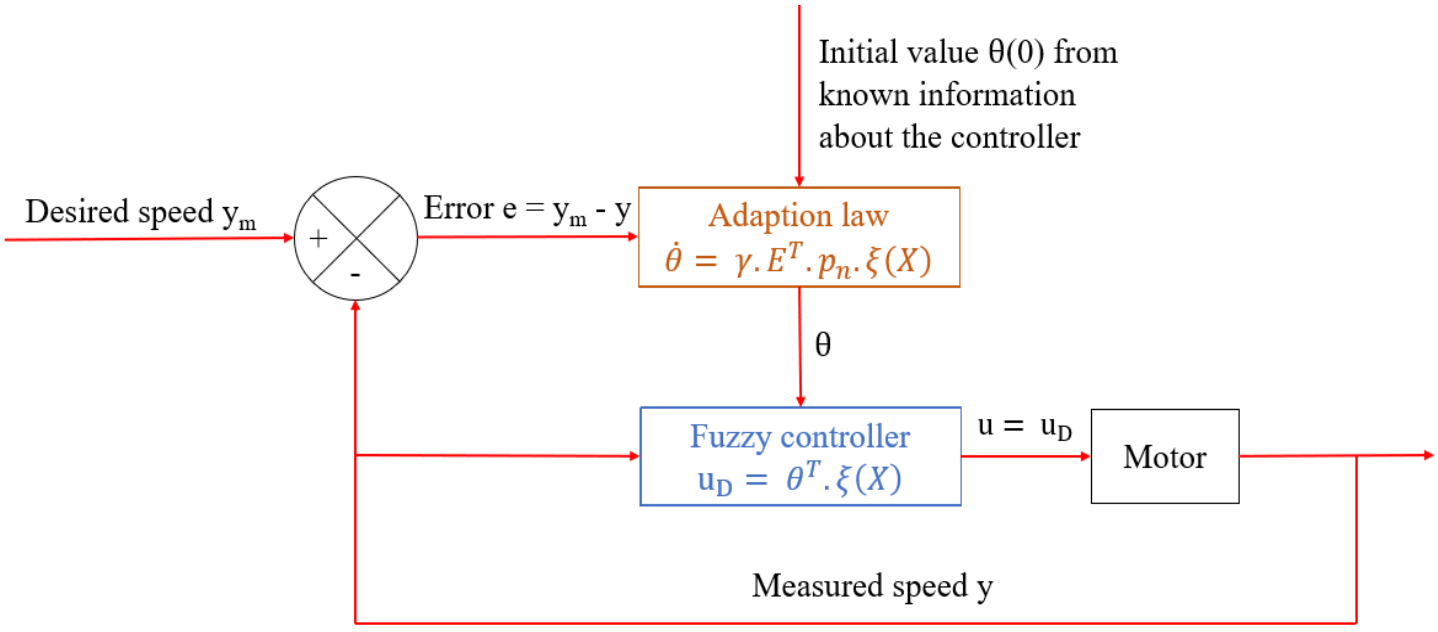


Figure 6. General layout diagram of direct adaptive fuzzy controller for a DC motor

This system includes 2 main parts:

- Adaption law:
  - Continuously adjust the parameter θ to always converge to the ideal parameter, ensure that the error of the output of the object (motor) always converges to 0 and the system is always stable.
- Fuzzy controller:
  - Approximating unknown control law, ensuring control quality when object parameters change or have the impact of noise during working process, and at the same time, in order to take advantage of knowledge and experience of control object.
  - Calculating the control signal u.

IV. MATLAB/SIMULINK SIMULATION:

4.1. Simulink block diagram:

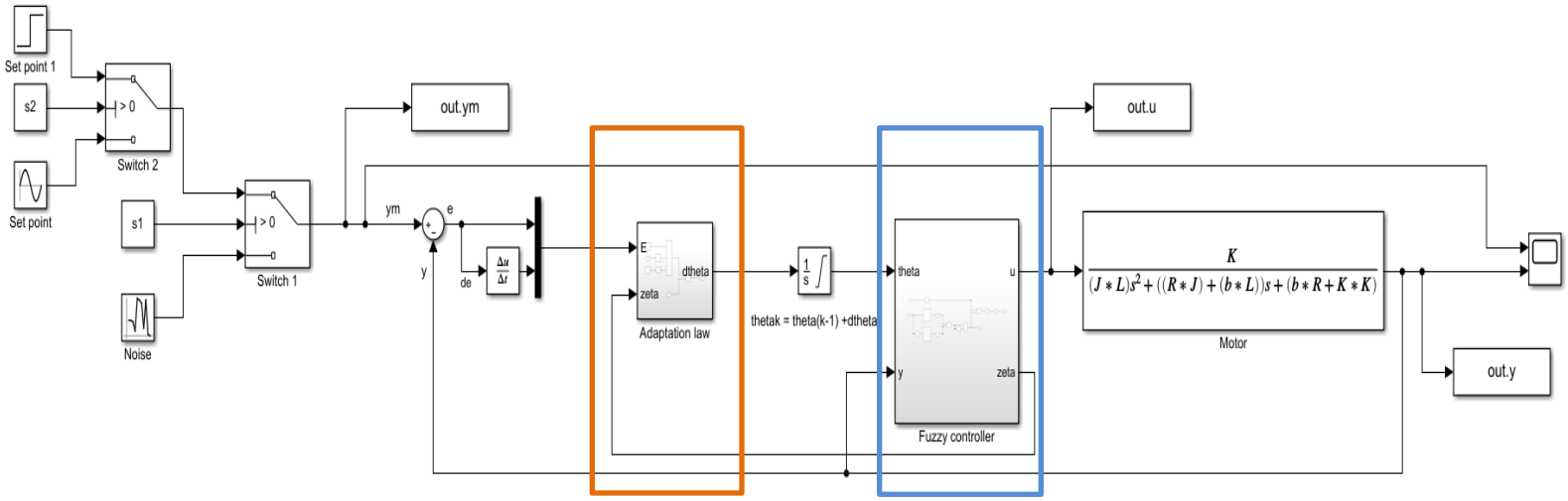


Figure 7. Simulink block diagram in Matlab

4.2. Diagram function:

- Set point:

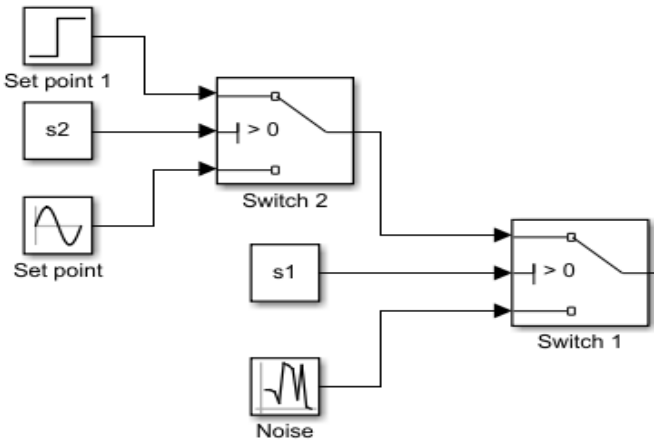


Figure 8. Set point in simulink

- Function: Provide the desired speed (y_m) which varies according to the sine function, step function or random number (noise).

- Adaption law:

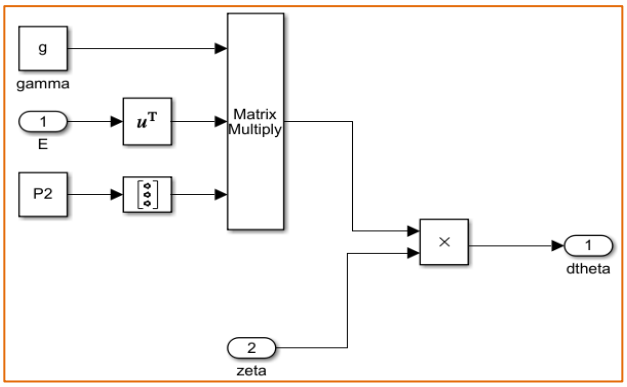


Figure 9. Adaption law in simulink

- Function: Calculation to adjust the parameter vector θ.

  Where: γ (gamma) = 20, P_2 = [0.25;0.5025], E is error vector, and zeta $\xi(X)$ is calculated from fuzzy controller.

- Fuzzy controller:

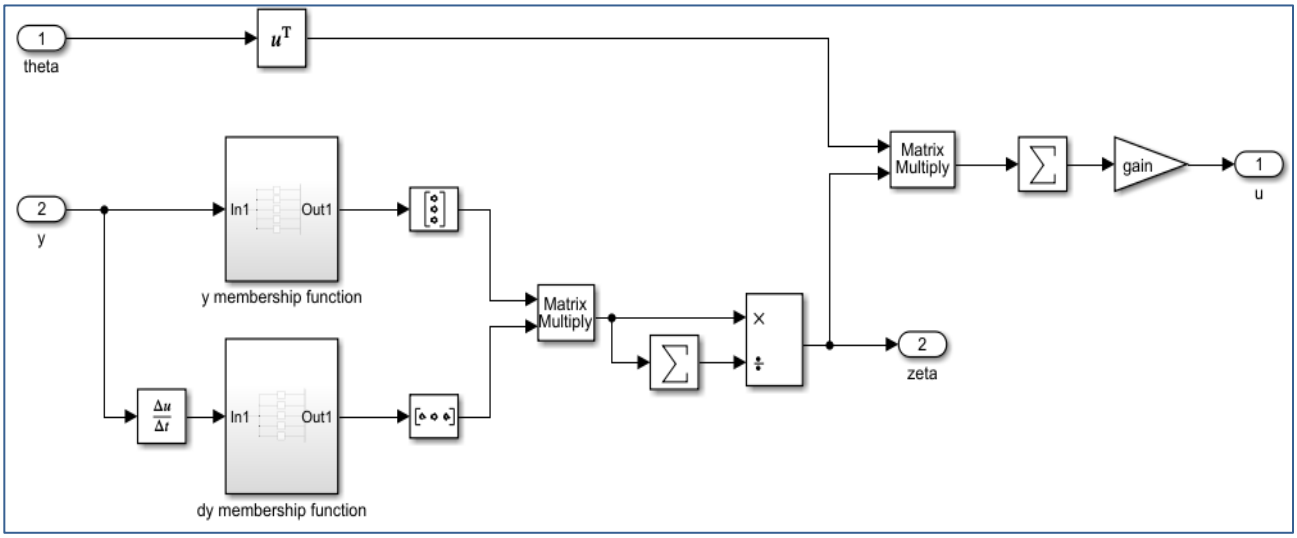


Figure 10. Fuzzy controller in simulink

- Function: Calculation to generate control signal u.

Where: theta = theta0 + dtheta; theta0 = $\begin{bmatrix} 10 & 10 & 10 & 10 & 10 \\ 20 & 20 & 20 & 20 & 20 \\ 30 & 30 & 30 & 30 & 30 \\ 40 & 40 & 40 & 40 & 40 \\ 50 & 50 & 50 & 50 & 50 \end{bmatrix}$

- Membership functions:

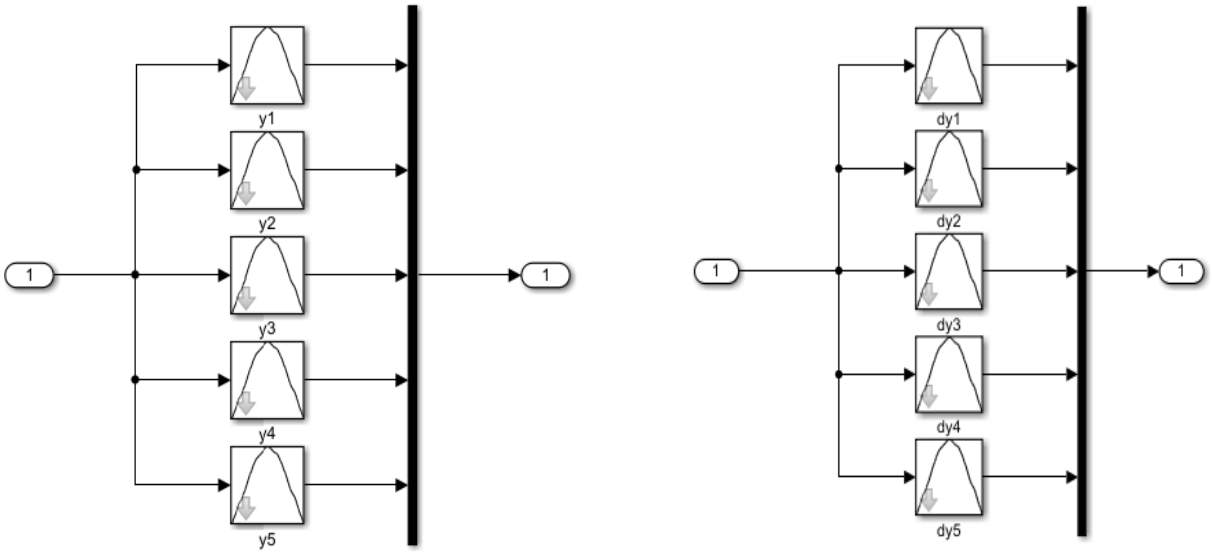


Figure 11. Membership function in simulink

  - Function: Identify (approximate) object model (Motor) with 2 fuzzy sets of y and dy.

- DC motor:

$$\frac{K}{(J*L)s^2+((R*J)+(b*L))s+(b*R+K*K)}$$

Motor

Figure 12. Motor in simulink

A DC motor has the continous-time transfer function $P(s) = \frac{K}{(J.s+b)(L.s+R)+K^2}\left[\frac{rad/sec}{V}\right]$ with parameters: K = 0.01 represent both motor torque constant K_t and back emf constant K_e, J =13/3000 (kg.m^2) is moment of inertia of rotor, b = 1/12 (N.m.s) is motor viscous friction constant, R = 1 (Ohm) is electric resistance, L = 1/5 (H) is electric inductance. (1)

- Scope:
  - Function: Display results of the desired speed (RPM) and the measured speed (RPM).
- PID controllers:
  - P = 10
  - I = 20
  - D = 0.5

4.3. Simulating result and discussion:

4.3.1. Simulating result of direct adaptive fuzzy controller:

The motor speed is controlled to follow the desired speed, which varies according to the sine function with different periods of 30s (Figure 13), 60s (Figure 14), 120s (Figure 15) and variation according to the step function (Figure 16).

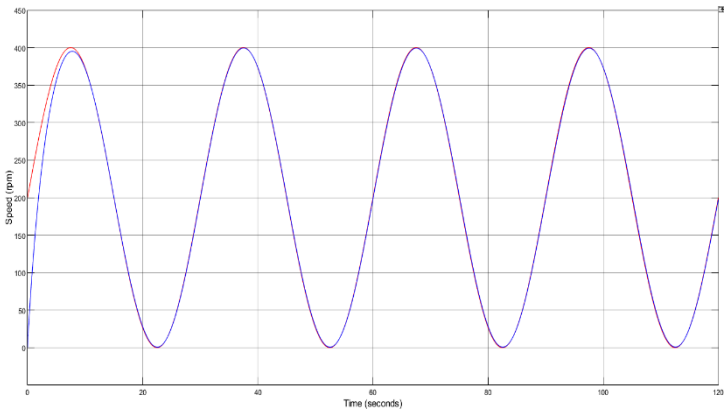


Figure 13. The response to y_m is a sine function of period 30s

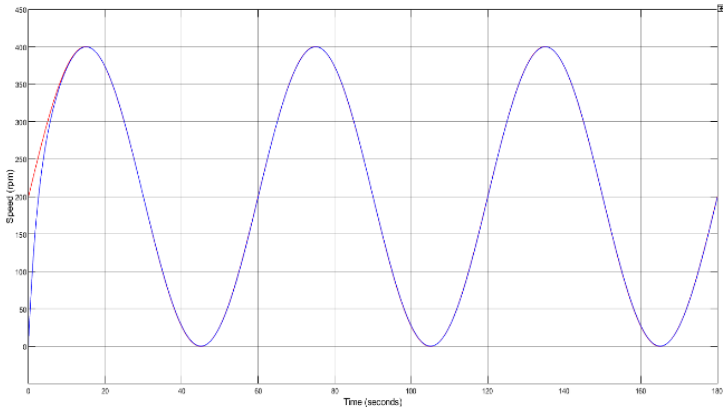


Figure 14. The response to y_m is a sine function of period 60s

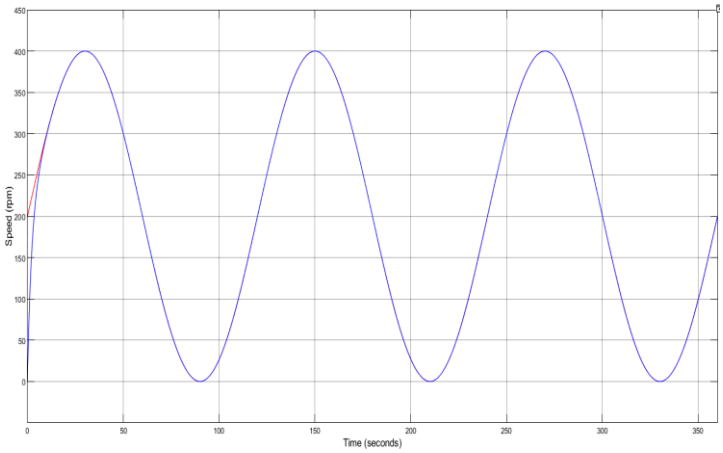


Figure 15. The response to y_m is a sine function of period 120s

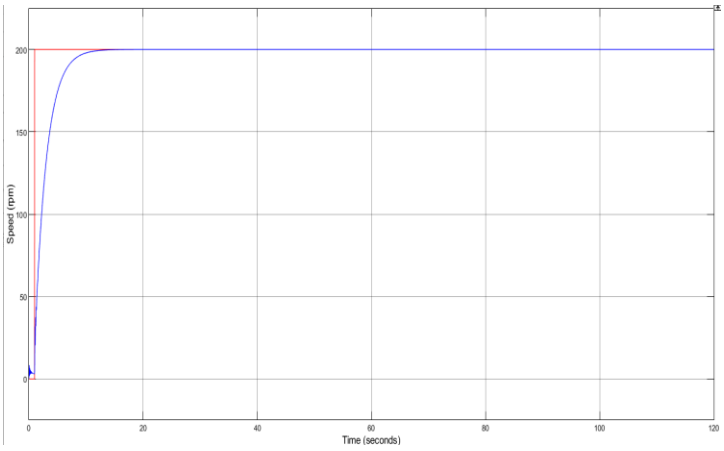


Figure 16. The response to y_m is a step function

4.3.2. Compare simulating result of direct adaptive fuzzy controller with PID controller:

Compare Direct Adaptive Fuzzy controller with PID controller when changing parameters (Input parameters, Motor's parameters, Adding noise) without changing controller parameters:

- Input parameters:
  - The response to y_m is a sine function with period 60s:

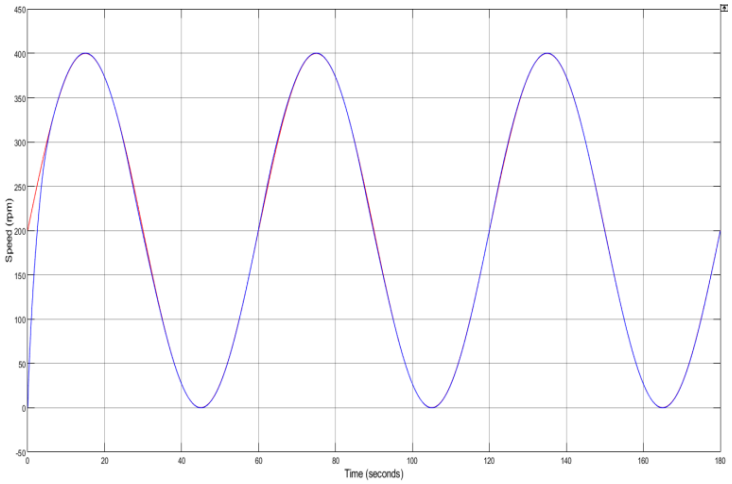


Figure 17. Direct Adaptive Fuzzy controller

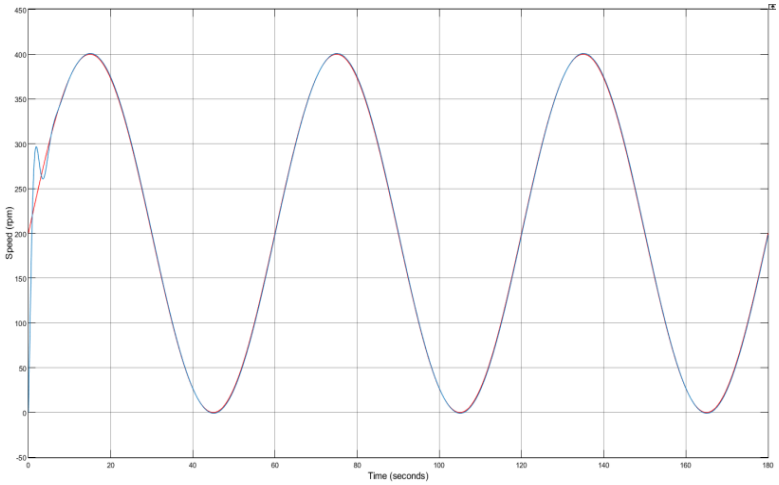


Figure 18. PID controller

➢ When decreasing period to 30s:

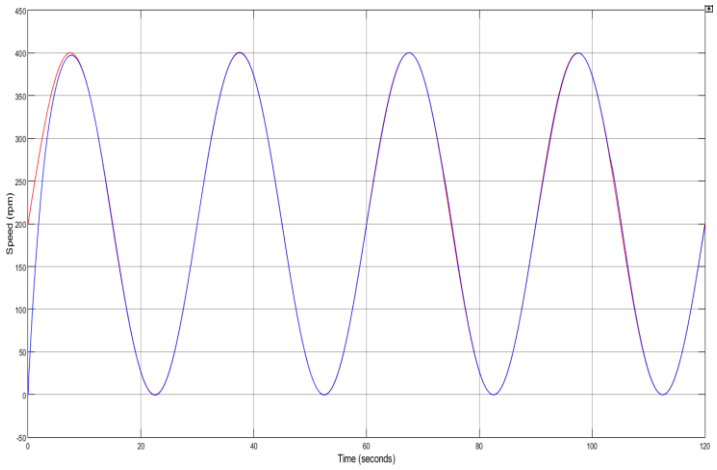


Figure 19. Direct Adaptive Fuzzy controller

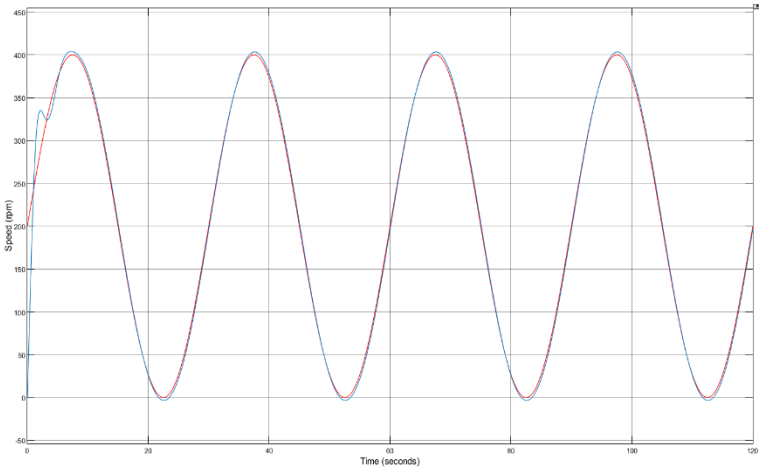


Figure 20. PID controller

➢ When decreasing period to 15s:

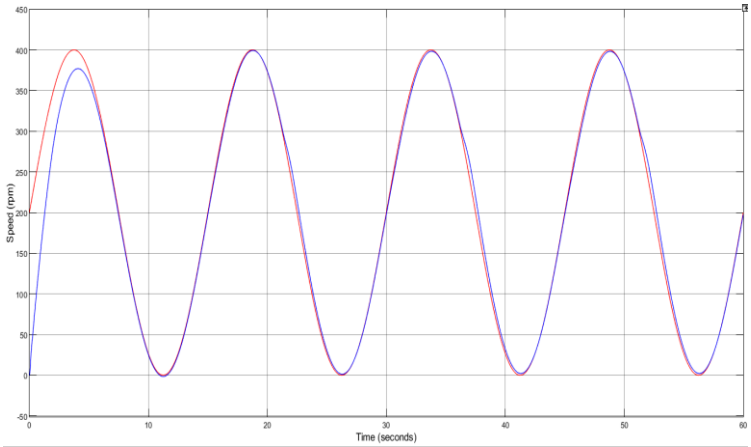


Figure 21. Direct Adaptive Fuzzy controller

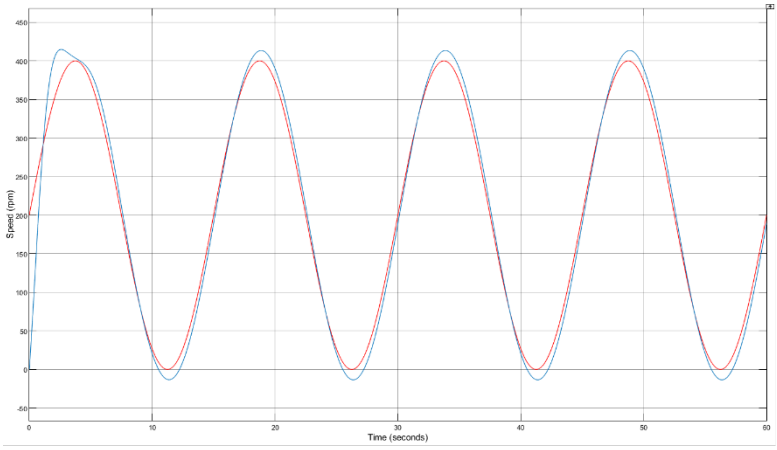


Figure 22. PID controller

Observing above result, with the input function as a sine function with a period of 60s, both the Direct adaptive fuzzy controller and the PID controller give very good control results with stable sticking to y_m. But when gradually reducing the period to 30s and 15s without changing the parameters of the two controllers, with the direct adaptive fuzzy controller, the signal y is still stable following the desired speed y_m, but with the PID controller, the measured speed y gradually loses its stability, deviating from the desired speed ym. -> No matter how with the input parameter change, the direct adaptive fuzzy controller still produces a speed y that closely matches the desired speed y_m, but the pid controller does not.

- Motor's parameters:
  - The motor's parameters has the transfer functions with parameters (1):

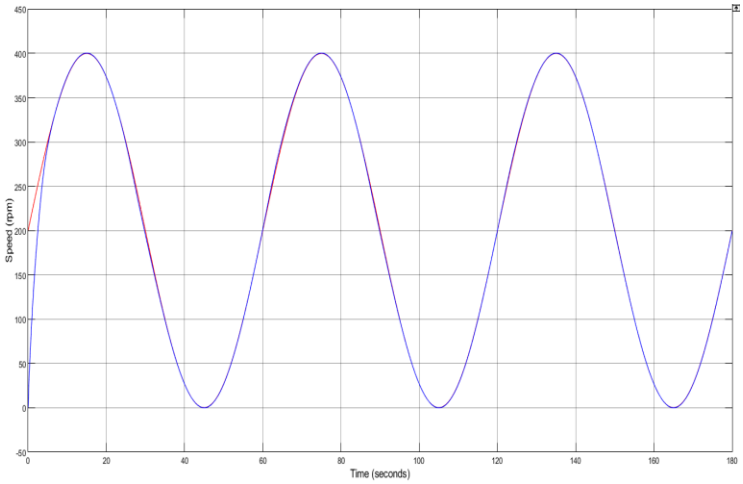


Figure 23. Direct Adaptive Fuzzy controller

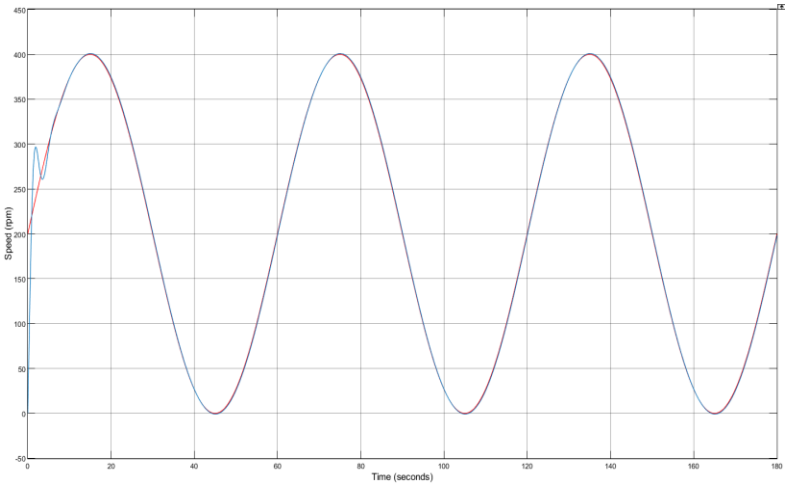


Figure 24. PID controller

  - When changing the motor's parameters by increasing J, b and L to double (J = 26/13000 $kg.m^2$, b = 1/6 N.m.s and L = 2/5 H).

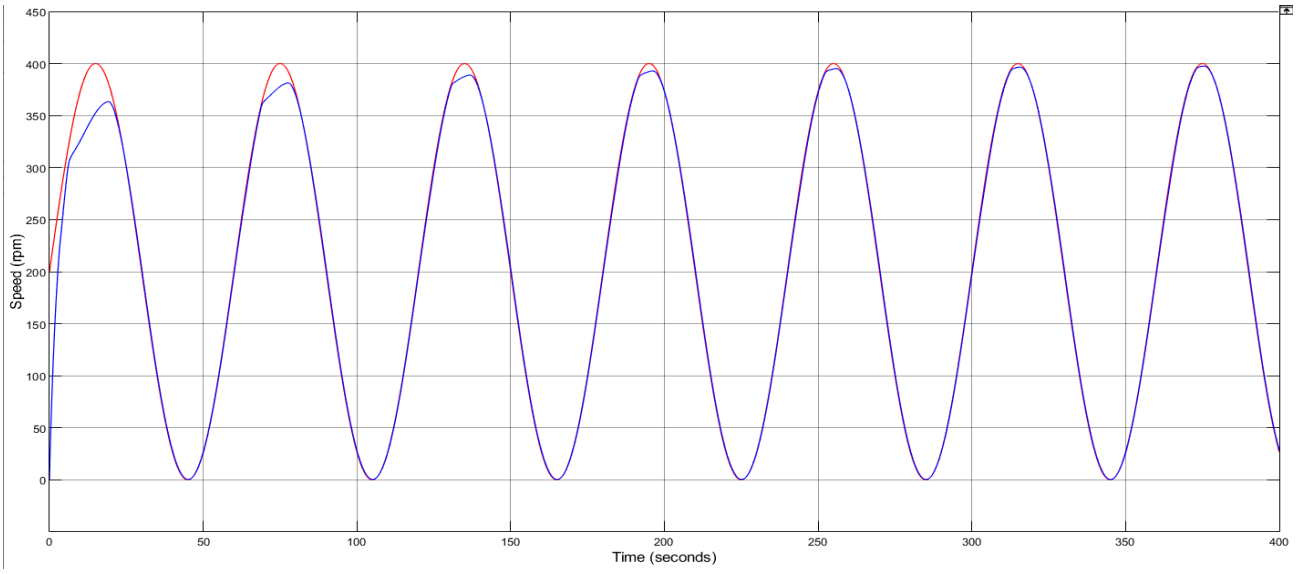


Figure 25. Direct Adaptive Fuzzy controller

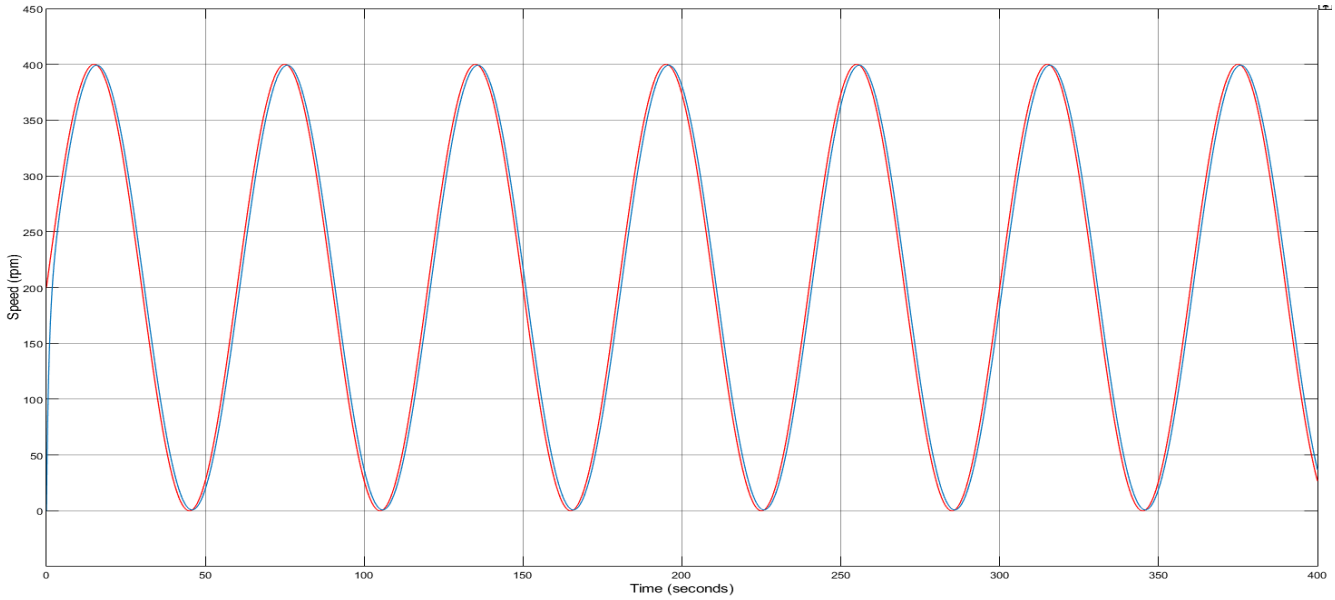


Figure 26. PID controller

Observing above result, when changing the motor's parameters, the direct adaptive fuzzy controller has self-adjusted (adapted) to produce a control signal that follows the desired speed y_m over time and PID controller can't do that, the measured speed y didn't match with y_m signal when changing motor parameter. -> For each type of motor (different parameters), it is necessary to find an appropriate set of P, I, D parameters, and a direct adaptive fuzzy controller is not needed.

- Adding noise: Using a random number to represent the influence of noise on the controller in simulink.

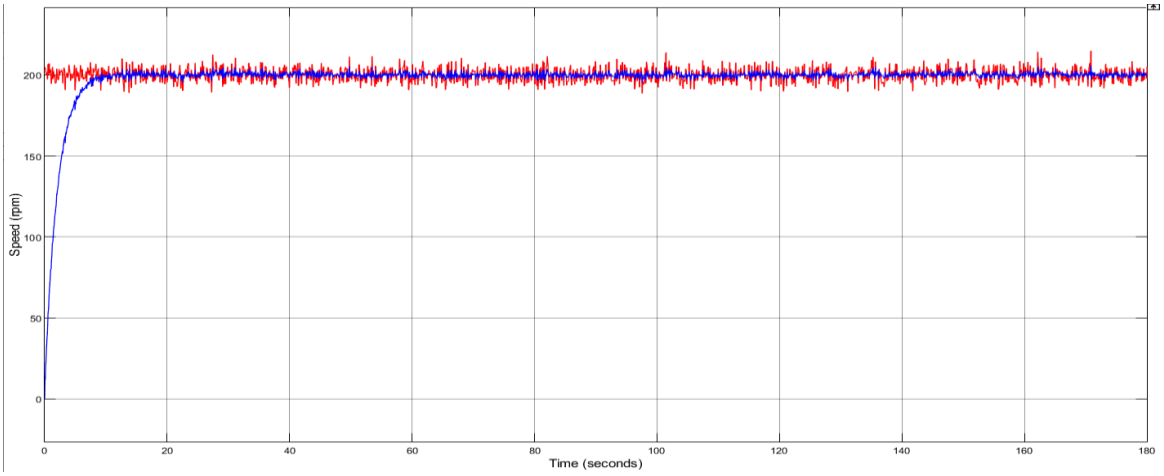


Figure 27. Direct Adaptive Fuzzy controller

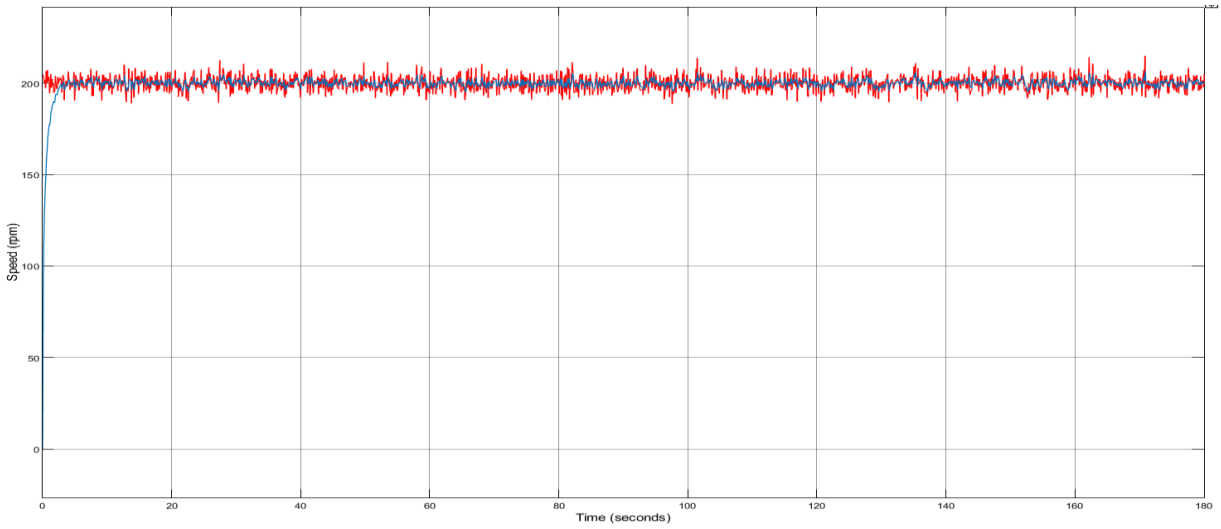


Figure 28. PID controller

Observing above result, the direct adaptive fuzzy controller is less affected by noise than the PID controller -> The output speed y is better.

4.3.3. Discussion:

The control results show that with only one controller, the motor speed is controlled according to many different desired speed patterns with short transient time, small error.

A direct adaptive fuzzy controller is a special form of nonlinear feedback control with response cannot be determined in advance. During the process of reaching steady state or transitioning from one steady state to another, the system may temporarily fall into an unstable state (Figure 16). In addition, only those parameters that are under the dominant fuzzy rules will be heavily adjusted, and will approach the ideal parameter. Therefore, at a time, only a few parameters converge to the ideal parameter. In other words, all parameters will not converge to the ideal set of parameters at the same time. Therefore, the system always has a delay (due to the loss of parameter convergence time) and may fall into a temporary unstable state (caused by the parameter not having converged).

4.4. App designer:

Create an user interface to display tools for users to manipulate.

Implement the code for the action buttons on the App designer in Matlab so that they can perform the tasks.

Import and export data to Simulink simulation.

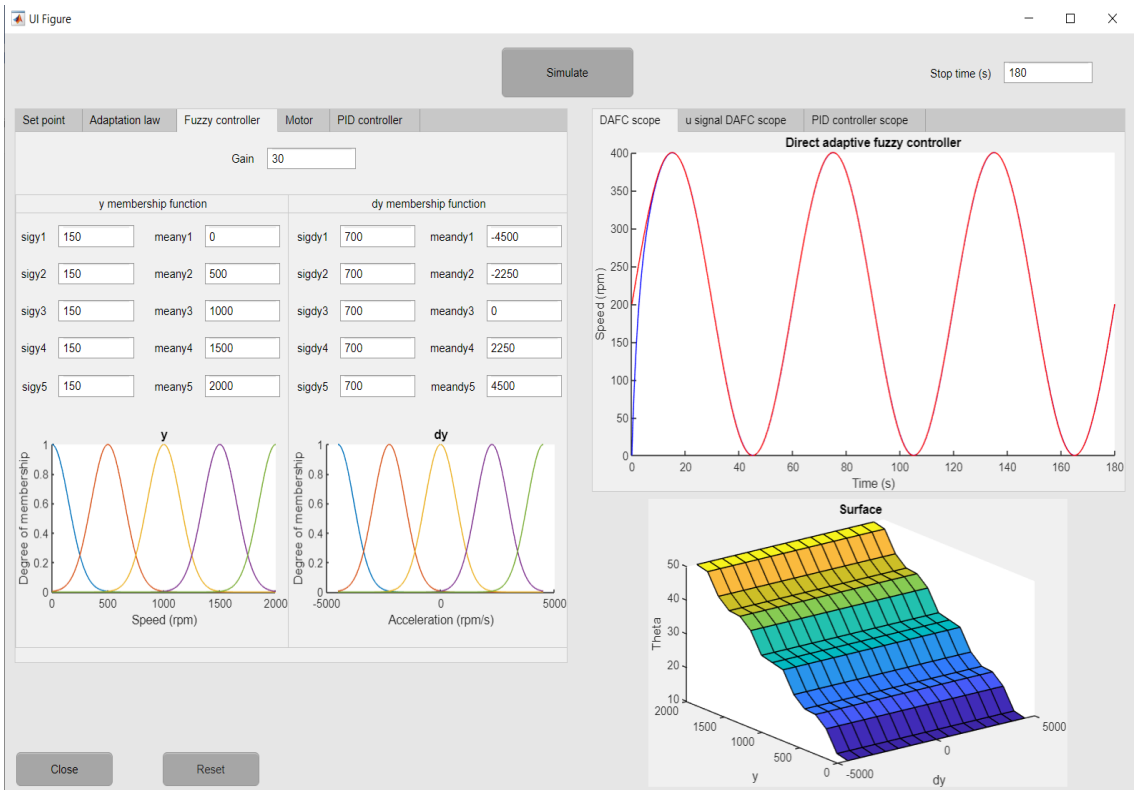


Figure 29. App designer to create user interface

The layout structure:

- Text boxes to input parameters (set point, adaption law, fuzzy controller, motor, PID controller) and to export parameters to simulink simulation.
- Axes to sketch the diagram shows the result (Control signal from the scope, surface of fuzzy rules, membership functions) after receiving the data from text boxes.
- Close button to close App Designer and Reset button to reset the diagram.

V. CONCLUSION:

The DC motor object controlled by the PWM method is a nonlinear object. A properly designed direct adaptive fuzzy controller controls the speed of the motor according to various desired speed patterns.

The direct adaptive fuzzy controller with the parameter adjustment rule is built on the Lyapunov stability criterion for good control quality, the structure is much simpler than other adaptive controllers. The design process is simple, without having to deal with complex modeling or identification problems. The control quality is good and the system achieves high stability even though the object parameters are not clearly defined, the object parameters are changed during the working process and the noise affects the system.

The experience and known information about the object will be very helpful in finding the optimal adaptive controller.

The parameters that determine the quality of the system are: coefficient γ, initial value θ , desired signal y_m, ... With each parameter having its own effect, finding the optimal set of parameters should be based on experience and knowledge of control systems.

However, this system also has some weakness:

- The parameter tuning law built on Lyapunov criteria ensures high system convergence and stability, but has not yet solved the optimal control problem. In addition, the unpredictable

excess characteristics and the error characteristics converge to zero, but the parameters do not converge to the ideal parameter, making the excess quality sometimes unsatisfactory. Therefore, there is a need for parameter monitors to improve excess quality.

- To improve the control quality, it is necessary to have a solution to adjust the update coefficient γ for each working area of the system.

VI. APPENDIX:

Set up user interface in App designer (Code):

```
switch app.SetpointDropDown.Value
        case 'Step'
            assignin('base',"st",str2num(app.SteptimeEditField.Value));
            assignin('base',"iv",str2num(app.InitialvalueEditField.Value));
            assignin('base',"fv",str2num(app.FinalvalueEditField.Value));
            assignin('base',"satSt",str2num(app.SampletimeStepEditField.Value));
        case 'Sine wave'
            assignin('base',"amp",str2num(app.AmplitudeEditField.Value));
            assignin('base',"bias",str2num(app.BiasEditField.Value));
            assignin('base',"fre",str2num(app.FrequencyradsEditField.Value));
            assignin('base',"pha",str2num(app.PhaseradEditField.Value));
            assignin('base',"satSi",str2num(app.SampletimeSineEditField.Value));
        case 'Noise'
            assignin('base',"m",str2num(app.MeanEditField.Value));
            assignin('base',"vari",str2num(app.VarianceEditField.Value));
            assignin('base',"seed",str2num(app.SeedEditField.Value));
            assignin('base',"Satrn",str2num(app.SampletimenoiseEditField.Value));
    end
    assignin('base',"g",str2num(app.GammaEditField.Value));
    assignin('base',"K",str2num(app.KEditField.Value));
    assignin('base',"b",str2num(app.MotorviscousfrictionconstantbNmsEditField.Value));
    assignin('base',"J",str2num(app.MomentofinertiaoftherotorJkgm2EditField.Value));
    assignin('base',"R",str2num(app.ElectricresistanceROhmEditField.Value));
    assignin('base',"L",str2num(app.ElectricInductanceLHEditField.Value));
    assignin('base',"s1",str2num(app.Switch1EditField.Value));
    assignin('base',"s2",str2num(app.Switch2EditField.Value));
    assignin('base',"gain",str2num(app.GainEditField.Value));
    assignin('base',"sigy1",str2num(app.sigy1EditField.Value));
    assignin('base',"meany1",str2num(app.meany1EditField.Value));
    assignin('base',"sigy2",str2num(app.sigy2EditField.Value));
    assignin('base',"meany2",str2num(app.meany2EditField.Value));
    assignin('base',"sigy3",str2num(app.sigy3EditField.Value));
    assignin('base',"meany3",str2num(app.meany3EditField.Value));
    assignin('base',"sigy4",str2num(app.sigy4EditField.Value));
    assignin('base',"meany4",str2num(app.meany4EditField.Value));
    assignin('base',"sigy5",str2num(app.sigy5EditField.Value));
    assignin('base',"meany5",str2num(app.meany5EditField.Value));
    assignin('base',"sigdy1",str2num(app.sigdy1EditField.Value));
    assignin('base',"meandy1",str2num(app.meandy1EditField.Value));
```

```
        assignin('base',"sigdy2",str2num(app.sigdy2EditField.Value));
        assignin('base',"meandy2",str2num(app.meandy2EditField.Value));
        assignin('base',"sigdy3",str2num(app.sigdy3EditField.Value));
        assignin('base',"meandy3",str2num(app.meandy3EditField.Value));
        assignin('base',"sigdy4",str2num(app.sigdy4EditField.Value));
        assignin('base',"meandy4",str2num(app.meandy4EditField.Value));
        assignin('base',"sigdy5",str2num(app.sigdy5EditField.Value));
        assignin('base',"meandy5",str2num(app.meandy5EditField.Value));
        assignin('base',"Pro",str2num(app.ProportionalPEditField.Value));
        assignin('base',"Int",str2num(app.IntegralIEditField.Value));
        assignin('base',"Der",str2num(app.DerivativeDEditField.Value));
        assignin('base',"N",str2num(app.FiltercoefficientNEditField.Value));
           simout = sim('testmotor','Stoptime',num2str(app.StoptimesEditField.Value));
        data = simout.y.Data;
        data = data(:,:)';
        data1 = simout.u.Data;
        data1 = data1(:,:)';
        hold(app.UIAxes,"on");
        plot(app.UIAxes,simout.y.Time,data,"b");
        plot(app.UIAxes,simout.ym.Time,simout.ym.Data,"r");
        plot(app.UIAxes5,simout.u.Time,data1,"y");
        data2 = simout.y2.Data;
        data2 = data2(:,:)';
        hold(app.UIAxes4,"on");
        plot(app.UIAxes4,simout.y.Time,data2,"b");
        plot(app.UIAxes4,simout.ym2.Time,simout.ym2.Data,"r");
        fis = mamfis;
fis = addInput(fis,[str2num(app.meandy1EditField.Value)
str2num(app.meandy5EditField.Value)],"Name","dy");
fis = addMF(fis,"dy","gaussmf",[str2num(app.sigdy1EditField.Value)
str2num(app.meandy1EditField.Value)],'Name',"dy1");
fis = addMF(fis,"dy","gaussmf",[str2num(app.sigdy2EditField.Value)
str2num(app.meandy2EditField.Value)],'Name',"dy2");
fis = addMF(fis,"dy","gaussmf",[str2num(app.sigdy3EditField.Value)
str2num(app.meandy3EditField.Value)],'Name',"dy3");
fis = addMF(fis,"dy","gaussmf",[str2num(app.sigdy4EditField.Value)
str2num(app.meandy4EditField.Value)],'Name',"dy4");
fis = addMF(fis,"dy","gaussmf",[str2num(app.sigdy5EditField.Value)
str2num(app.meandy5EditField.Value)],'Name',"dy5");
[X,Y] = plotmf(fis,"input",1);
plot(app.UIAxes3_2,X,Y);
        fis = mamfis;
fis = addInput(fis,[str2num(app.meany1EditField.Value)
str2num(app.meany5EditField.Value)],"Name","y");
fis = addMF(fis,"y","gaussmf",[str2num(app.sigy1EditField.Value)
str2num(app.meany1EditField.Value)],'Name',"y1");
fis = addMF(fis,"y","gaussmf",[str2num(app.sigy2EditField.Value)
str2num(app.meany2EditField.Value)],'Name',"y2");
```

```
fis = addMF(fis,"y","gaussmf",[str2num(app.sigy3EditField.Value)
str2num(app.meany3EditField.Value)],'Name',"y3");
fis = addMF(fis,"y","gaussmf",[str2num(app.sigy4EditField.Value)
str2num(app.meany4EditField.Value)],'Name',"y4");
fis = addMF(fis,"y","gaussmf",[str2num(app.sigy5EditField.Value)
str2num(app.meany5EditField.Value)],'Name',"y5");
[X,Y] = plotmf(fis,"input",1);
plot(app.UIAxes3,X,Y);
        fis = sugfis('Name','testmotor');
fis = addInput(fis,[str2num(app.meandy1EditField.Value)
str2num(app.meandy5EditField.Value)],"Name","dy");
fis = addMF(fis,"dy","gaussmf",[str2num(app.sigdy1EditField.Value)
str2num(app.meandy1EditField.Value)],'Name',"dy1");
fis = addMF(fis,"dy","gaussmf",[str2num(app.sigdy2EditField.Value)
str2num(app.meandy2EditField.Value)],'Name',"dy2");
fis = addMF(fis,"dy","gaussmf",[str2num(app.sigdy3EditField.Value)
str2num(app.meandy3EditField.Value)],'Name',"dy3");
fis = addMF(fis,"dy","gaussmf",[str2num(app.sigdy4EditField.Value)
str2num(app.meandy4EditField.Value)],'Name',"dy4");
fis = addMF(fis,"dy","gaussmf",[str2num(app.sigdy5EditField.Value)
str2num(app.meandy5EditField.Value)],'Name',"dy5");
fis = addInput(fis,[str2num(app.meany1EditField.Value)
str2num(app.meany5EditField.Value)],"Name","y");
fis = addMF(fis,"y","gaussmf",[str2num(app.sigy1EditField.Value)
str2num(app.meany1EditField.Value)],'Name',"y1");
fis = addMF(fis,"y","gaussmf",[str2num(app.sigy2EditField.Value)
str2num(app.meany2EditField.Value)],'Name',"y2");
fis = addMF(fis,"y","gaussmf",[str2num(app.sigy3EditField.Value)
str2num(app.meany3EditField.Value)],'Name',"y3");
fis = addMF(fis,"y","gaussmf",[str2num(app.sigy4EditField.Value)
str2num(app.meany4EditField.Value)],'Name',"y4");
fis = addMF(fis,"y","gaussmf",[str2num(app.sigy5EditField.Value)
str2num(app.meany5EditField.Value)],'Name',"y5");
fis = addOutput(fis,"Name","theta");
fis = addMF(fis,"theta","constant",10,'Name',"theta11");
fis = addMF(fis,"theta","constant",10,'Name',"theta12");
fis = addMF(fis,"theta","constant",10,'Name',"theta13");
fis = addMF(fis,"theta","constant",10,'Name',"theta14");
fis = addMF(fis,"theta","constant",10,'Name',"theta15");
fis = addMF(fis,"theta","constant",20,'Name',"theta21");
fis = addMF(fis,"theta","constant",20,'Name',"theta22");
fis = addMF(fis,"theta","constant",20,'Name',"theta23");
fis = addMF(fis,"theta","constant",20,'Name',"theta24");
fis = addMF(fis,"theta","constant",20,'Name',"theta25");
fis = addMF(fis,"theta","constant",30,'Name',"theta31");
fis = addMF(fis,"theta","constant",30,'Name',"theta32");
fis = addMF(fis,"theta","constant",30,'Name',"theta33");
fis = addMF(fis,"theta","constant",30,'Name',"theta34");
```

```
fis = addMF(fis,"theta","constant",30,'Name',"theta35");
fis = addMF(fis,"theta","constant",40,'Name',"theta41");
fis = addMF(fis,"theta","constant",40,'Name',"theta42");
fis = addMF(fis,"theta","constant",40,'Name',"theta43");
fis = addMF(fis,"theta","constant",40,'Name',"theta44");
fis = addMF(fis,"theta","constant",40,'Name',"theta45");
fis = addMF(fis,"theta","constant",50,'Name',"theta51");
fis = addMF(fis,"theta","constant",50,'Name',"theta52");
fis = addMF(fis,"theta","constant",50,'Name',"theta53");
fis = addMF(fis,"theta","constant",50,'Name',"theta54");
fis = addMF(fis,"theta","constant",50,'Name',"theta55");
ruleList = [1 1 1 1 1
            1 2 6 1 1
            1 3 11 1 1
            1 4 16 1 1
            1 5 21 1 1
            2 1 2 1 1
            2 2 7 1 1
            2 3 12 1 1
            2 4 17 1 1
            2 5 22 1 1
            3 1 3 1 1
            3 2 8 1 1
            3 3 13 1 1
            3 4 18 1 1
            3 5 23 1 1
            4 1 4 1 1
            4 2 9 1 1
            4 3 14 1 1
            4 4 19 1 1
            4 5 24 1 1
            5 1 5 1 1
            5 2 10 1 1
            5 3 15 1 1
            5 4 20 1 1
            5 5 25 1 1];
fis = addrule(fis,ruleList);
showrule(fis);
[X,Y,Z] = gensurf(fis);
app.UIAxes2.ZLabel.String = 'Theta';
surf(app.UIAxes2,X,Y,Z);
```

VII. REFERENCE:

[1] JOHN YEN & REZA LANGARI, "Fuzzy logic: Intelligence control and information", Prentice-Hall, 1997.

[2] LI-XIN WANG, "A Course in Fuzzy System and Control", Prentice-Hall International, 1997.

[3] NGUYỄN PHÙNG QUANG, "Matlab & Simulink dành cho kỹ sư điều khiển tự động", NXB khoa học và kỹ thuật, 2006.

[4] NGUYỄN DOÃN PHƯỚC, "Lý thuyết điều khiển tuyến tính", NXB Khoa học và kỹ thuật, 2002

[5] PHAN XUÂN MINH & NGUYỄN DOÃN PHƯỚC, "Lý thuyết điều khiển mờ", NXB khoa học và kỹ thuật, 2006.