Study of the Effect in the Output Membership Function When Tuning a Fuzzy Logic Controller

Mohd Shahrieel bin Mohd Aras
Faculty of Electrical Engineering
Universiti Teknikal Malaysia Melaka,
Melaka, Malaysia
shahrieel@utem.edu.my

Fadilah binti Abdul Azis
Faculty of Electrical Engineering
Universiti Teknikal Malaysia Melaka,
Melaka, Malaysia
fadilah@utem.edu.my

Syed Mohamad Shazali b Syed Abdul Hamid
Faculty of Electrical Engineering
Universiti Teknikal Malaysia Melaka,
Melaka, Malaysia
syedmohamad@utem.edu.my

Fara Ashikin binti Ali
Faculty of Electrical Engineering
Universiti Teknikal Malaysia Melaka,
Melaka, Malaysia
fara@utem.edu.my

Dr Shahrum Shah b Abdullah
Malaysia-Japan International Institute of Technology
Universiti Teknologi Malaysia,
Kuala Lumpur, Malaysia
shahrum@ic.utm.my

Abstract— **This paper describes a study of tuning process for fuzzy logic controller (FLC) design. In fuzzy logic controller design, there is no systematic procedure to tune fuzzy logic controller to follow a desired set point. The tuning process of Fuzzy Logic Controllers (FLCs) using trial-and-error approach is commonly done until satisfactory results are obtained. This is usually a tedious and time-consuming task but it has been widely employed and has been used in many successful industrial applications. The performance of the system can be analyzed. If the results are not as desired, changes are made either to the number of the fuzzy partitions or the mapping of the membership function and then the system can be tested again. This paper demonstrates a faster tuning process by adjusting the mapping of the membership function to get desired output. Through identifying and analyzing what will be done on adjusting mapping of membership functions by utilizing knowledge from the experts, it will be demonstrated in this paper that the tuning process of FLCs can be easily simplified.**

Keywords- Fuzzy Logic Controller, membership function partitions, Deep Submergence Rescue Vehicle

I. INTRODUCTION

A fuzzy controller is designed to emulate human deductive thinking, that is, the process people use to infer conclusions (decision making) from what they know. Fuzzy control has been primarily applied to the control of processes through fuzzy linguistic descriptions. Fuzzy logic is widely used in machine control applications [1].

MATLAB Fuzzy logic Toolbox is use to design fuzzy logic controller. Basically, the Fuzzy Logic controller consists of four basic components: fuzzification, a knowledge base, inference engine, and a defuzzification interface as shown in Figure 1. Each component affects the effectiveness of the fuzzy controller and the behavior of the controlled system. In the fuzzification interface, a measurement of inputs and a transformation, which converts input data into suitable linguistic variables, are performed which mimic human decision making [1]. The results obtained by fuzzy logic depend on fuzzy inference rules and fuzzy implication operators. The knowledge base provides necessary information for linguistic control rules and the information for fuzzification and defuzzification. In the defuzzification interface, an actual control action is obtained from the results of fuzzy inference engine [1].
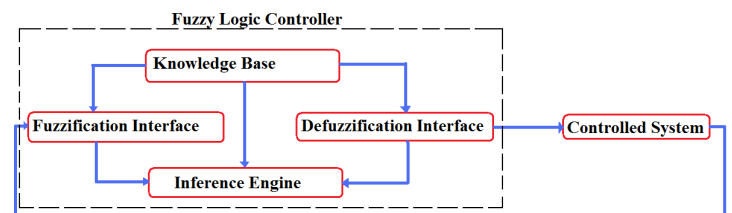

Figure 1: Basic Configuration of Fuzzy Logic Controller

II. THEORY ON TUNING PROCESS

The tuning process of Fuzzy Logic Controllers (FLCs) through trial-and-error approach will continue until satisfactory results are obtained. Definitely it is usually a tedious and time-consuming task. The derivation of the rule based basically follows these four approaches:

(i) Expert experiences and control engineering knowledge
(ii) Based on operator's control actions
(iii) Based on fuzzy model of a process
(iv) Based on learning

From these four methods, only two will be used in this case which is based on learning and from the expert experiences and control engineering knowledge. An expert experience is the least structured of the four methods and yet it is one of the most widely used today. It is based on the experiment done from the experience based knowledge of the process control. Currently many research efforts are focused on emulating human learning, mainly on the ability to create fuzzy control rules and to modify them or tune based on experience. There are now many examples of fuzzy controllers which have the capability to learn and to tune the controllers involving neural networks and genetic algorithm. However, this approach requires one to understand neural networks and genetic algorithm before it can be used. It will take more time and the controller may become complicated. So, the aim of this paper is to demonstrate several techniques and method to tune controller to desired response as simple as possible. When designing the controller, we should try to keep the controller as simple as possible with good and acceptable performances. FLCs are reasonably easy to design but the tuning process of FLCs through trial-and-error approach will continue until satisfactory results are obtained; definitely it is usually a tedious and time-consuming task. As was mentioned earlier, there is no systematic procedure to tune FLC. If the results are not as desired, changes are made either to the number of the fuzzy partitions or the mapping of the membership function and then the system can be tested again. Figure 2 is the flow chart to design FLC.
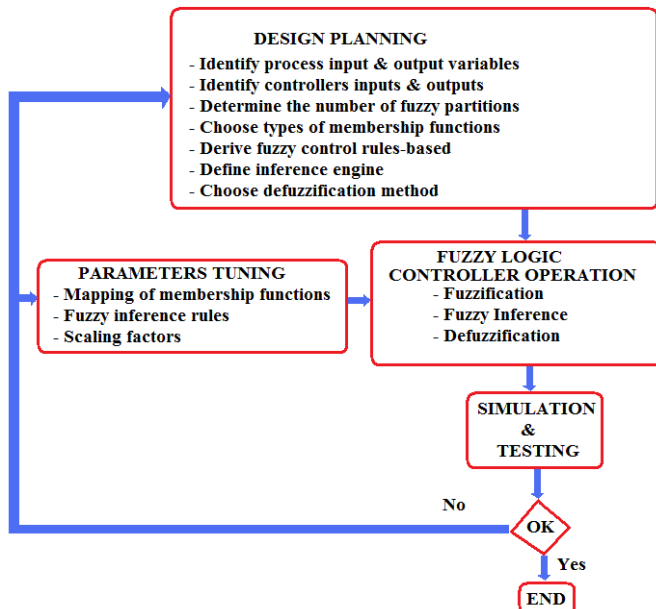

Figure 2. Flow chart of FLC design

III. EXPERIMENTAL SET-UP

The first step is to setup the experiment. The example used in this case is based on the paper entitled "*A simplified approach to design fuzzy logic controller for an underwater vehicle*" [2]. Figure 3, Figure 4 and Figure 5 is the membership function for inputs and output. Two inputs to the controller are depth error, Ze, and W the heave velocity in z-axis. The output that was used in this design is the voltage, δs for stern plane. Based on DSRV model of FIS [2], we set 7 X 7 membership functions partitions for inputs and output. 7 x 7 membership functions partitions are more complicated to tune compared with 3x3 partitions. More membership function partitions imply better performances but are harder to tune. For MATLAB, the maximum partitions we set up are 9x9 partitions. Constructing rules by using the graphical Rule Editor interface is fairly straight forward. Based on the descriptions of the input and output variables defined with the FIS Editor, the Rule Editor allows us to construct the rule statements automatically as shown in Table 1. In theory, the number of rules that can be constructed is based on the number of membership function partitions. For this study we develop 49 rules from 7x7 membership function partitions.
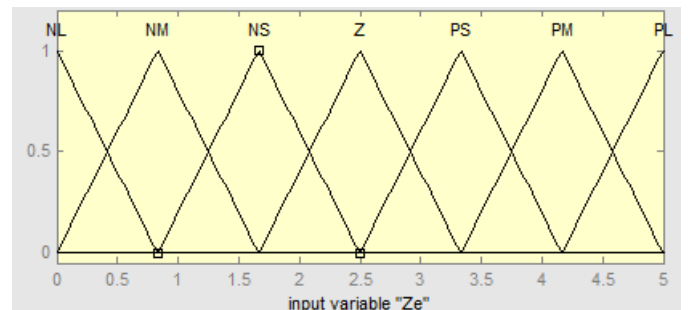

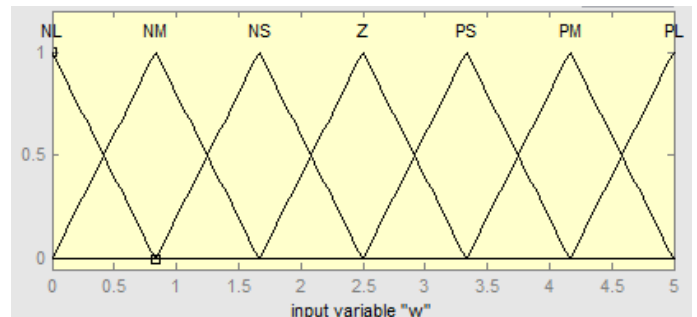Figure 3. The membership function input 'Ze'
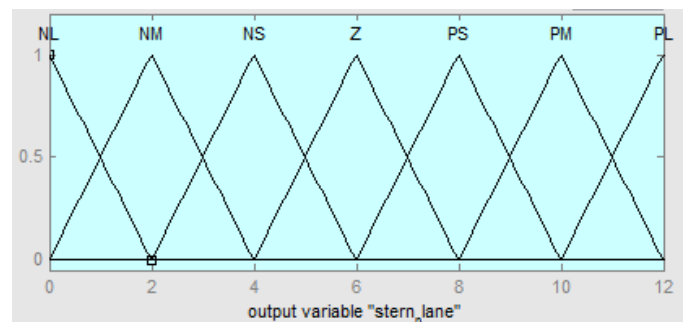

Figure 4. The membership function input 'W'

Figure 5. The membership function for output 'δs'

TABLE 1. RULE TABLE FOR DSRV

Ze W	PL	PM	PS	Z	NS	NM	NL
NL	Z	NS	NM	NL	NL	NL	NL
NM	PS	Z	NS	NM	NL	NL	NL
NS	PM	PS	Z	NS	NM	NL	NL
Z	PL	PM	PS	Z	NS	NM	NL
PS	PL	PL	PM	PS	Z	NS	NM
PM	PL	PL	PL	PM	PS	Z	NS
PL	PL	PL	PL	PL	PM	PS	Z

IV. EXPERIMENTAL RESULTS

The Fuzzy Logic Controller block implements a fuzzy inference system (FIS) in simulink. We can take fuzzy systems directly into simulink and test them out in a simulation environment. A simulink block diagram for DSRV system is shown in the Figure 6. Figure 7 shows the output response of the DSRV model. From this response we can see that no overshoot occurred but the response does follow the set point in the beginning. So we have to tune our fuzzy system. We have three basic methods to tune our fuzzy system such as scale of membership functions, mapping of membership function and adjusting rules. In this study we focused on only adjusting the mapping of membership functions for output membership function. In this study we try to analyze the effect when we tune mapping of membership function until a satisfactory response is obtained. We figure several ways to tune our mapping of membership functions which will be described in this paper.
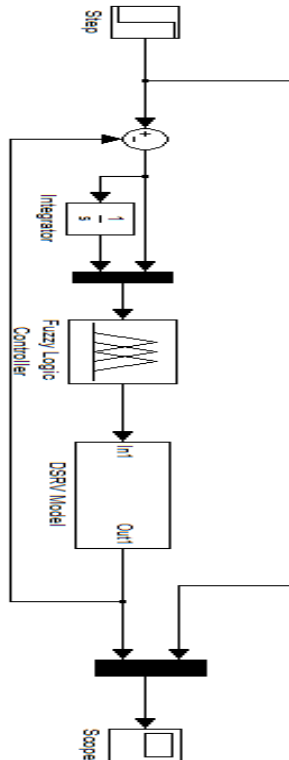

Figure 6. Fuzzy logic controller design

Figure 7 shows the system response for DSRV system. It looks like some noise is present in the beginning of the response. Figure 8 is the output of the fuzzy logic controller. From the output fuzzy have little disturbances that will affect the stability of the DSRV.

Inputs for fuzzy logic controller are shown in Figure 9. The yellow line is the input for 'Ze' while the purple line is the input for 'W'. The input 'Ze' is not too smooth at early stage and we can tune the input 'Ze' to get better response but in this project we focus on the output of fuzzy logic controller.
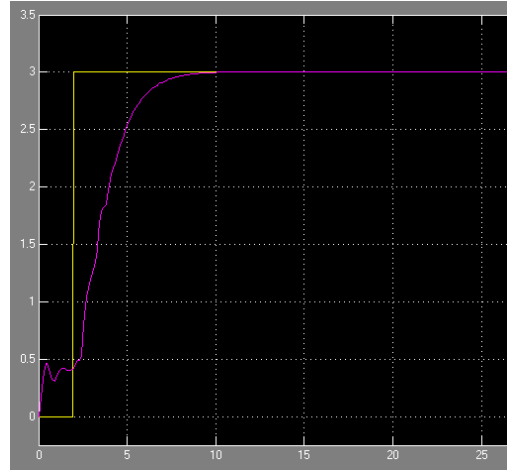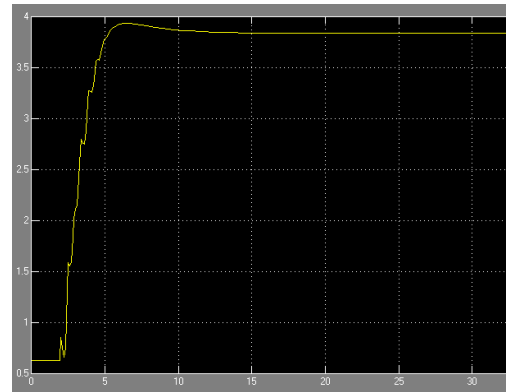

Figure 7: system response

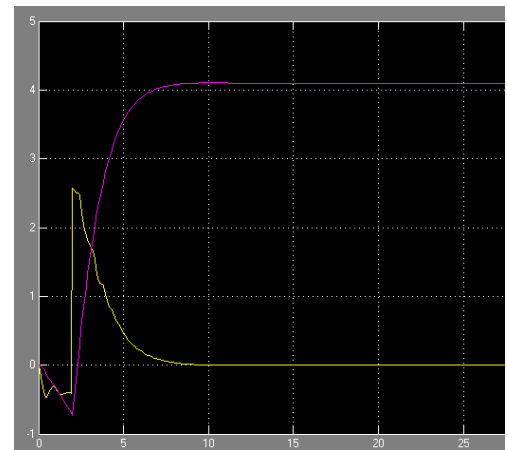Figure 8: Output of fuzzy controller

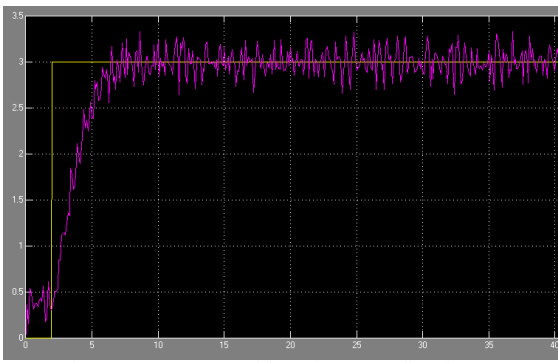
Figure 9: Inputs of Fuzzy controller

Figure 10. Response with environment disturbances

Figure 10 is the result of system response with environment disturbances. The type of environmental disturbances common to underwater vehicles are waves (wind generated), wind and ocean currents. In general these disturbances will be both additive and multiplicative to the dynamic equations of motion. In this project we add both to the dynamic equation of motion. Based on Fossen [5], the wind moment parameters are about 0.0127.

V. TUNING PROCESS

Figure 11 shows the rule viewer of rules that we set. From this rule viewer it can be seen that at rule no 43 only have an output at the both inputs is zero as shown below;

43. If (Ze is NL) and (w is NL) then (stern_plane is NL) (1)

The process tuning will be focusing on NL output since the NL output is the starting process of system response. This is the way how to start the process of tuning. Based on rule viewer we can know which one to adjust. Figure 12 is shows the membership function for output after tuned and Figure 13 shows it the responses. The disturbances in Figure 13 will be better and follows the set point.
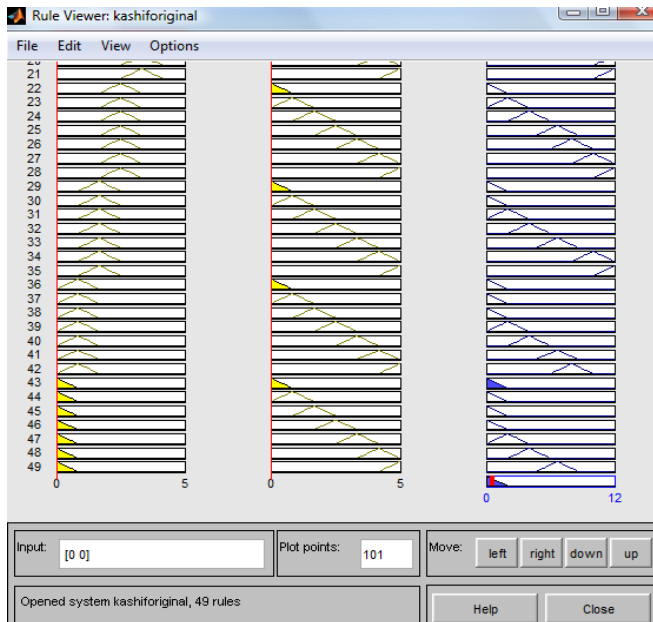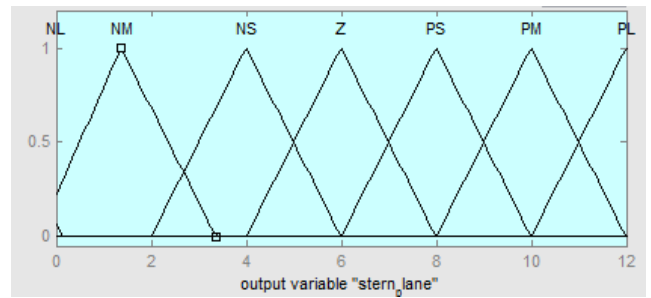

Figure 11. The rule viewer

Figure 12. Output variable "stern_plane" after tuned

Here only NL partitions will be adjusting to left (decrease of values of NL membership partitions). Adjust NL until responses follow the desired response as shown in Figure 14. In Figure 15 occurred two unwanted signal (dropped). To eliminate this unwanted dropped signal, we adjusted NM and NS membership functions as shown in Figure 16 and Figure 17. Reducing the scaling factor of NM and move it to the left and also move the NS to left also. So the dropped signal will be reduced to only one dropped signal as shown in Figure 18. The rise time, settling time, overshoot and steady-state error is almost the same after several tunes of output membership function partitions. The values of all the parameters are shown in Table 4.
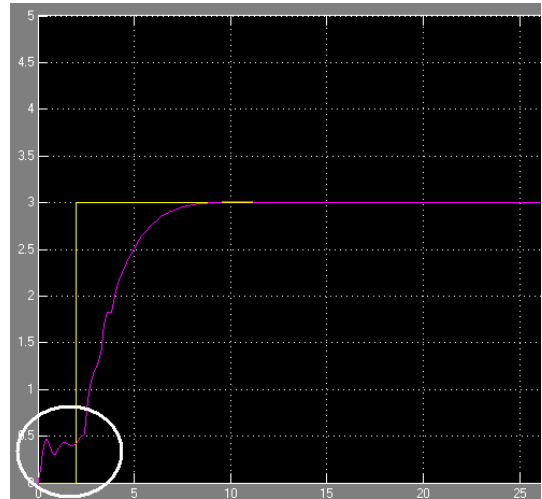

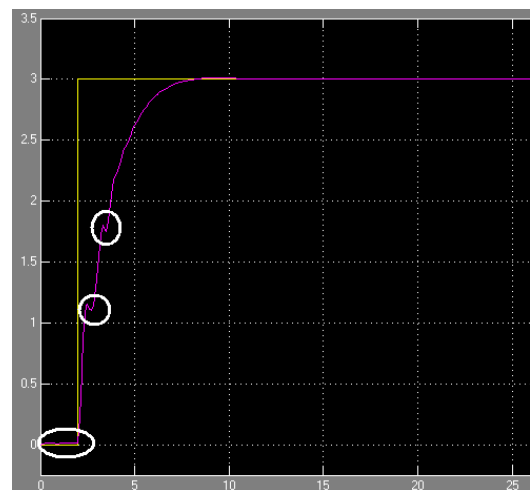Figure 13. The response of fuzzy logic controller
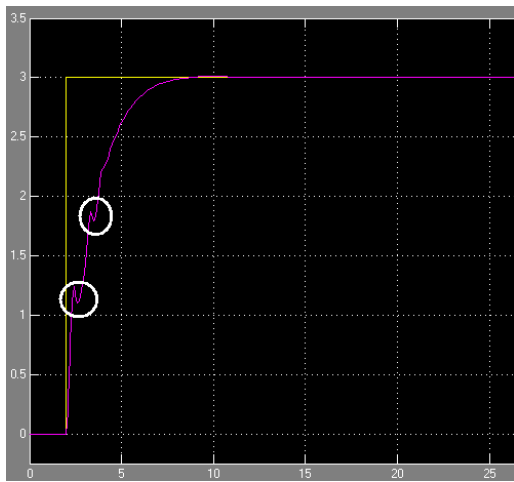

Figure 14. Response after tuning

Figure 15. Two dropped signal occurred

To eliminate the dropped signal, the output membership function will move to left. PS, PM, and PL will move to left and it will becomes the smaller values of the output responses as wanted as shown in Figure 19.Figure 20 shows the results but Figure 21 is the best response that we got. The output membership function partitions will be adjusted to the left until satisfactory responses are obtained. Based on Table 3 we can tune to get better results in term of rise time and steady-state error. But this result is good enough compared with conventional controller as shown in Table 2.

TABLE 2. PERFORMANCES OF CONTROLLER

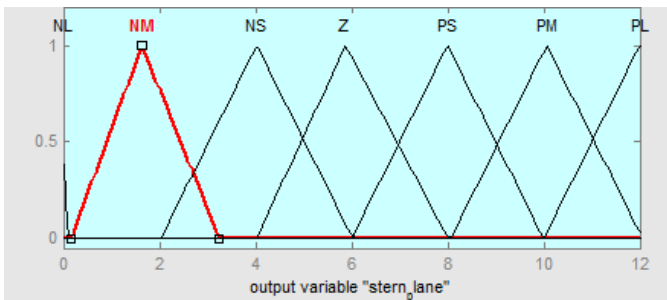| Parameters	PI	I	PID
Rise time (s)	8	2.4	2.4
Overshoot (m)	0.1485	0.04	0.04
Settling time (s)	10	7.34	7.34
Steady-state time (s)	12	12	12

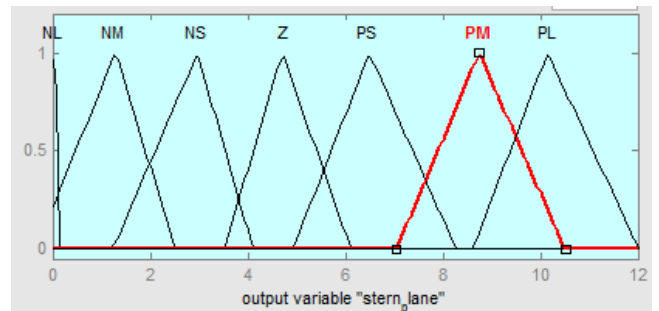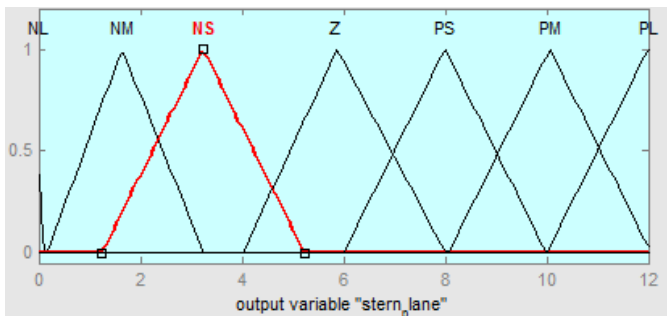Figure 16. Scalling factor and move to left

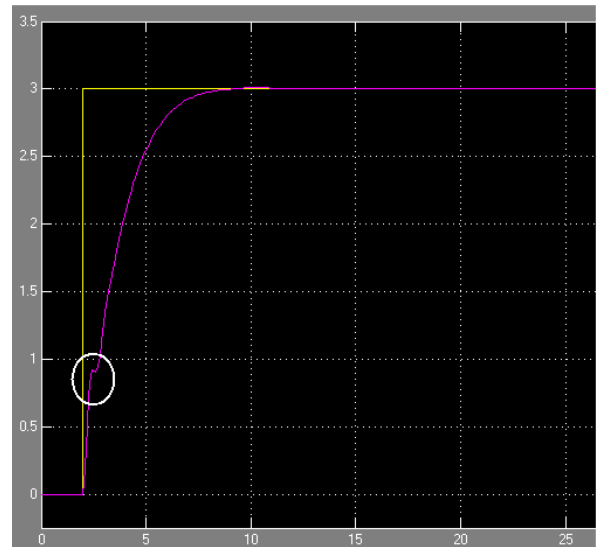Figure 19. Move to left

Figure 17. Move to left

Figure 20. Very small dropped signal
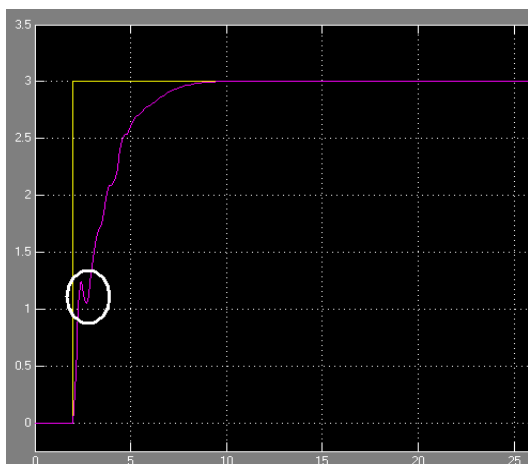

Figure 18. Only one dropped signal
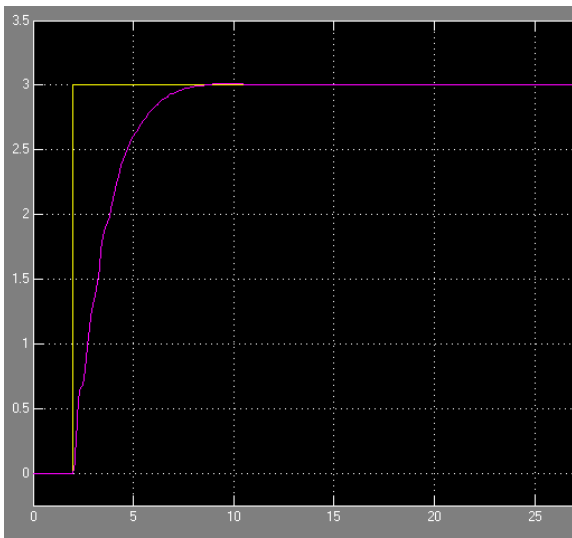
Figure 21. The best response obtained
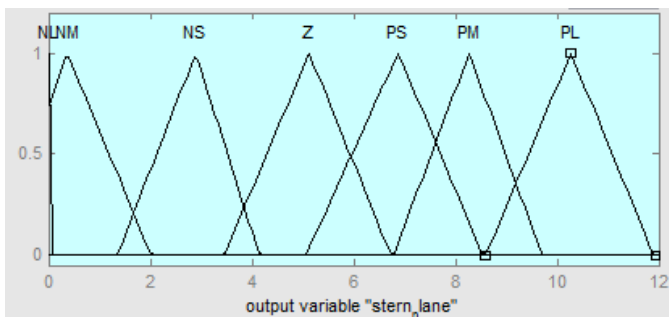
Figure 22. The last tuned for output membership functions

As was mentioned before, when designing the controller we must keep the controller as simple as possible. In terms of steady state error and overshoot, fuzzy controller gave much better response than PI but in terms of rise time, the PI controller gave much faster response than fuzzy. Actually we can tune fuzzy logic controller more until we get the best result. For this moment this one is the better result after several tuning process of fuzzy controller as shown in Table 2 and Table 4. Fuzzy logic controller requires someone that are familiar and have a lot of experience in designing fuzzy controller.

VI   CONCLUSION

As a conclusion, the tuning of FLC will be easier and faster with several tips as explained in this paper. For users with limited knowledge of tuning of FLC or in designing FLC itself, this paper can be used as a guideline to design FLC with good performances compared with conventional controller. The tuning process of Fuzzy Logic Controllers (FLCs) through trial-and-error approach may still be required but certainly it is not the best method. After describing several rules of thumb to tune the FLC, no trial and error approach is needed.

VI   DISCUSSION

Table 3 is summary what we have done in this experiment. So, this will be guide lines for others when designing FLC. This is only a few techniques or a way to tuning FLC. Every response will give other way to solve. It is not fixed but it will be helpful for others to tune FLC with fastest and gives a best response as good as possible.

TABLE 3. SUMMARY OF TUNING PROCESS

Problem occurred	MF partition at Output	Tuning
More ripples	Z	Reduced scale
Overshoot	PL or PM	Reduced scale
Noise at early response	NL	Reduced scale and shift to left
Noise at rise time	PL or PM	Reduced scale and shift to left
Steady state time	Z or NS or PS	Reduced scale

TABLE 4. PERFORMANCES OF FL CONTROLLER

Parameters	FLC
Rise time (s)	5
Overshoot (m)	0
Settling time (s)	5
Steady-state time (s)	8

REFERENCES

[1] Mohd Shahrieel Mohd Aras, Eric Chee Sai Hoo, Syed Najib Bin Syed Salim, Intan Azmira binti Wan Abd Razak, Mohd Hendra bin Hairi, *Comparison of Fuzzy Control Rules using MATLAB Toolbox and Simulink for DC Induction Motor-Speed Control*, International Conference of Soft Computing and Pattern Recognition, 2009.

[2] K. Ishaque n, S.S. Abdullah, S.M. Ayob, Z. Salam, *A simplified approach to design fuzzy logic controller for an underwater vehicle*, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 2010.

[3] Zhijie Tang, Luojun, and Qingbo He, *A Fuzzy-PID Depth Control Method with Overshoot Suppression for Underwater Vehicle*, School of Mechatronics Engineering and Automation, Shanghai University, Shanghai, 2010.

[4] G.V.Nagesh Kumar,K.A.Gopala Rao, P.V.S Sobhan, D.Deepak Chowdary, *Robustness of Fuzzy Logic based Controller for Unmanned Autonomous Underwater Vehicle*, College of Engineering, Andhra University,2008

[5] T.I. Fossen, *Guidance and control of ocean vehicles*, Wiley, New York, 1994.

[6] T.J. Ross, *Fuzzy Logic with Engineering Applications*, 2nd Ed, Hoboken, NJ: John Wiley & Sons, 2004.