OPTIMIZED FUZZY CONTROL DESIGN OF AN AUTONOMOUS UNDERWATER VEHICLE

B. RAEISY, A. A. SAFAVI AND A. R. KHAYATIAN

Abstract. In this study, the roll, yaw and depth fuzzy control of an Autonomous Underwater Vehicle (AUV) are addressed. Yaw and roll angles are regulated only using their errors and rates, but due to the complexity of depth dynamic channel, additional pitch rate quantity is used to improve the depth loop performance. The discussed AUV has four flaps at the rear of the vehicle as actuators. Two rule bases and membership functions based on Mamdani type and Sugeno type fuzzy rule have been chosen in each loop. By invoking the normalized steepest descent optimization method, the optimum values for the membership function parameters are found. Though the AUV is a highly nonlinear system, the simulation of the designed fuzzy logic control system based on the equations of motion shows desirable behavior of the AUV specially when the parameters of the fuzzy membership functions are optimized.

1. Introduction

An AUV system is known as a highly nonlinear dynamic system. The interaction between the system and its environment is very abstruse, therefore the control system design and simulation of AUVs are very difficult. The AUV is a special class of the general category of Underwater Vehicles (UV). UV's can be divided into three types:

- Remotely Underwater Vehicles (ROV): This type is threaded and the control commands and even sometimes power are transferred to the vehicle by a cable.
- Unmanned Underwater Vehicles (UUV): Although this type is unthreaded and it has no cable, but, the control command is transferred to the system by an acoustic modem.
- Autonomous Underwater Vehicles (AUV): There is no communication channel for an AUV and it has to complete its mission without any human help. It is programmed autonomously for the whole mission. With respect to unthreaded specification of an AUV, such system has more maneuverability than other UV systems and its operation range can be extended because it needs no communication with the mother ship.

Received: June 2010; Revised: April 2011 and May 2011; Accepted: July 2011

Key words and phrases: Fuzzy optimized control, Autonomous underwater vehicle, Normalized steepest descent, Neural network.

The preceding reasons allow the use of AUVs in a variety of applications, from marine industries to military operations. The widespread use and the highly non-linear dynamics of AUVs have motivated rigorous researches on choosing different control design strategies but the control of such systems is fully case dependent. The important specifications of AUVs like the shape, actuator types and propellant are very different and these diversities can raise new problems when a new AUV is selected. Conventional control method of a AUV based on a linearized model around a point is presented in [7]. This method has been applied to AUV under our study in [3]. The lack of perfect performance of linear control due to the highly non-linear dynamics has led researchers to invoke other advance control methods. Sliding mode control has been addressed in [5, 19]. Fuzzy logic is another useful tool for AUV control. Balasuriya and Cong combined sliding method and fuzzy logic to control the ODIN AUV [2]. This AUV uses truster as actuators. Wang and Lee introduced a self-adaptive recurrent neuro-fuzzy control as a feedforward controller and a proportional-plus-derivative (PD) control as a feedback controller to control an AUV [23]. They trained recurrent neuro-fuzzy system to model the inverse dynamics of the AUV for feedforward and PD control of the AUV. Some other advance control methods which are commonly used for AUVs are:

- Adaptive control [1, 12].
- Nonlinear control [6, 14].
- Neural network control [13, 10].

In this paper, simple optimized fuzzy controllers are developed for yaw, pitch and depth control of an AUV with time varying mass. Although the method is intuitively simple, but it is practical and implementable and it works quite well. Optimization of fuzzy variable values is an important problem [21] which it has been addresses in this paper too.

The AUV has three yaw, pitch and depth loops. In the yaw and depth loops the system must follow the input commands and the control goal is roll stabilization in the roll loop. For investigating such controllers, a motion simulator has been designed for the AUV. Considering the fact that the simulation process must be repeated many times for the optimization study, a faster simulation method with the aid of neural network model is proposed. Normalized steepest descent method is used for optimization of fuzzy logic membership parameters and it is shown that this algorithm with a variable step has very good performance.

The paper has been arranged as follows. In section 2 the actuators of the system are introduced. The control objectives and control loops are discussed in section 3. Fuzzy logic parameters, membership functions and fuzzy rule bases are introduced in section 4. Numerical method for the optimization of the controller parameters are proposed in section 5. The simulation results are presented in section 6.

2. **Actuators for the AUVs**

Two methods are commonly used to the motion control of an AUV. In the first method, thrusters correct the direction of AUV and in the second method, flaps

do the job. ODIN vehicle [23] uses thrusters and REMUS [15], MUST [5] , and MARIUS [19] use flaps.

In our special case, there are four flaps at the rear of the vehicle which are used for control and navigation as shown in Figure 1. These vertical and horizontal flaps can provide AUV proper means to move in different directions.

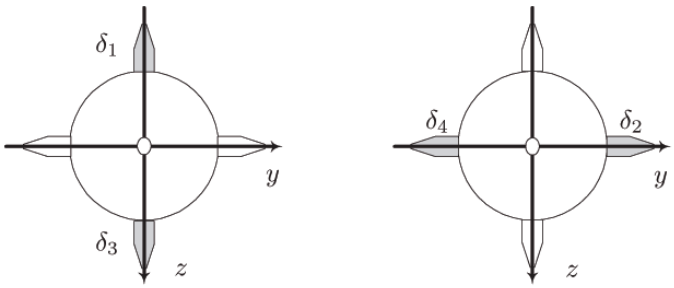


FIGURE 1. Rudder and Elevator Flaps

Angles of the vertical flaps are denoted by δ_1 and δ_3 symbols and δ_2 and δ_4 represent the angles of horizontal flaps. Positive value of each flap causes the body to rotate in positive direction around the longitude axis. In this AUV, δ_1 and δ_3 are free to move, therefore their actuations can control the roll and direction of the vehicle but δ_2 and δ_4 are mechanically coupled, so that $\delta_2 = -\delta_4$ and they have no effect on the body rotation, and with this arrangement, only the elevator command (D_{ec}) is applied to these flaps. In order to move the vehicle up and down, the command applied to the horizontal flaps should be changed with opposite values. As it was discussed before, the first and third flaps are used to control the rotation and direction of the body simultaneously. A drift of vertical flaps with opposite values causes the body to divert to right or left, and a drift with the same values causes the body to rotate around the longitude axis. Denoting the rotation signal (aileron) by D_{ac} and the direction command by D_{ar} the following relations between these commands and angle of flaps exist:

$$D_{ac} = \frac{\delta_1 + \delta_2 + \delta_3 + \delta_4}{2} = \frac{\delta_1 + \delta_2 + \delta_3 - \delta_2}{2} = \frac{\delta_1 + \delta_3}{2} \tag{1}$$

$$D_{rc} = \frac{\delta_3 - \delta_1}{2} \tag{2}$$

$$\delta_1 = D_{ac} - D_{rc} \tag{3}$$

$$\delta_3 = D_{ac} + D_{rc} \tag{4}$$

3. **Fast Neural Network Aided Simulation**

Body motion simulation is a useful tool that can reduce the need for large amount of real experiments on the object. Such experiments usually require a lot of resources and are timely. Body motion equations express mathematical relationships between body state parameters and by solving these equations, the motion of the object to be simulated. In general, for a rigid body, such equations consist of six relation which include three torque and three force equations. If the control system is also considered, then the equations of the control system must be solved with the equations of motion simultaneously. Equations of motion extraction are based on Newton's law. For constant mass AUV such equations are derived in many papers [15] but for the case where the mass is time varying, these equations are extracted around bouncy center in [16]. For a complete solution of the equations, all forces and torques are needed. The effective underwater forces can be classified as added mass, hydrostatic and hydrodynamic forces. The first two categories are not within the scope of this paper but hydrodynamic parts will be discussed here. Hydrodynamic forces and torques are very important because by changing the actuators (flaps), important vehicle coefficients are varied and the object can be controlled.

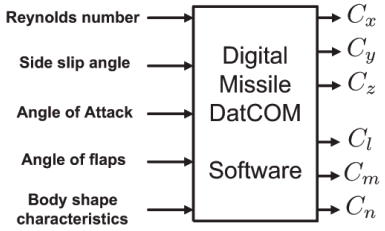


FIGURE 2. The Input Required for Calculating the Hydrodynamic Coefficients

Hydrodynamic coefficients are six dimensionless coefficients which named C_x, C_y, C_z, C_l, C_m, C_n. The first three quantities are used for calculating the forces in three X, Y and Z axes. With C_l, C_m and C_n, the torques around the axes can be computed. The values of these coefficients depend on body shape, flap angles, Reynolds number, angle of attack and side slip angle. Digital Missile DATCOM (MD) is a known program for calculating these parameters. This tool has been originally used for missiles but it also can be used for underwater media with similar Reynolds number [20]. MD needs an input file to get input parameters and its output is written in a file too [22]. MD inputs and outputs are shown in Figure 2. MD program should be executed at each iteration of motion equation solving to find the required hydrodynamic forces and torques. These operations are inherently very slow and must be repeated many times for optimization phase which is very time consuming. In order to solve this problem a neural network model is substituted with MD to reduce the total time considerably [16]. Three flap angles (δ_1, δ_2

and δ_3), angle of attack, side slip angle, and velocity are the inputs of the neural networks model and six hydrodynamic coefficients are the outputs of that model. For such substitution, the following steps has been taken:

A: The system was simulated with proper commands in each loop with the simulator software (This version of the simulator uses the MD for coefficients finding during the simulation).

B: The statistical parameters of the parameters to be used by the neural network were found by the simulation results.

C: 3000 Random input sets with normal distributions for all inputs were generated with the found statistical parameters.

D: All the coefficients (C_x, C_y, C_z, C_l, C_m, C_n) for the found input sets were calculated using MD software.

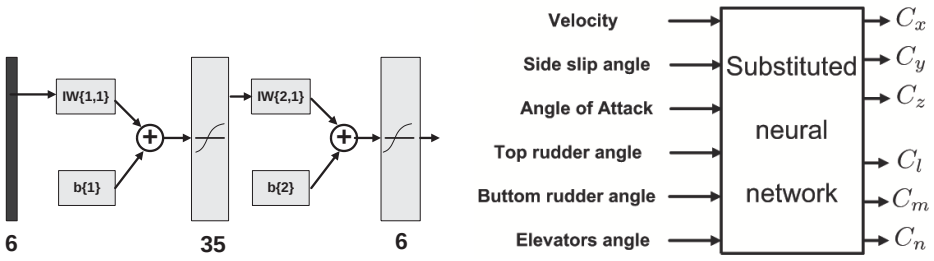
E: A neural network with these input and output sets were trained. Multilayer Perceptron Neural Network with a single intermediate layer is considered. Gradual increase to 35 nodes get a good approximation.

F: New data were generated and the trained network was tested with them.

G: The trained and tested neural network was substituted with MD based conventional coefficients finder.

H: The results of the new version were compared with the original simulator results and the accuracy of the results was proved.

The complete procedure of substituting MD with neural network is discussed in [16] and also it has been shown that the neural version of the simulator has 95 to 96 percent improvement in simulation time. The block diagram and inputs/outputs of the neural network are shown in Figure 3.



(a) Block diagram (b) Inputs and outputs

FIGURE 3. The Substituted Neural Network Model

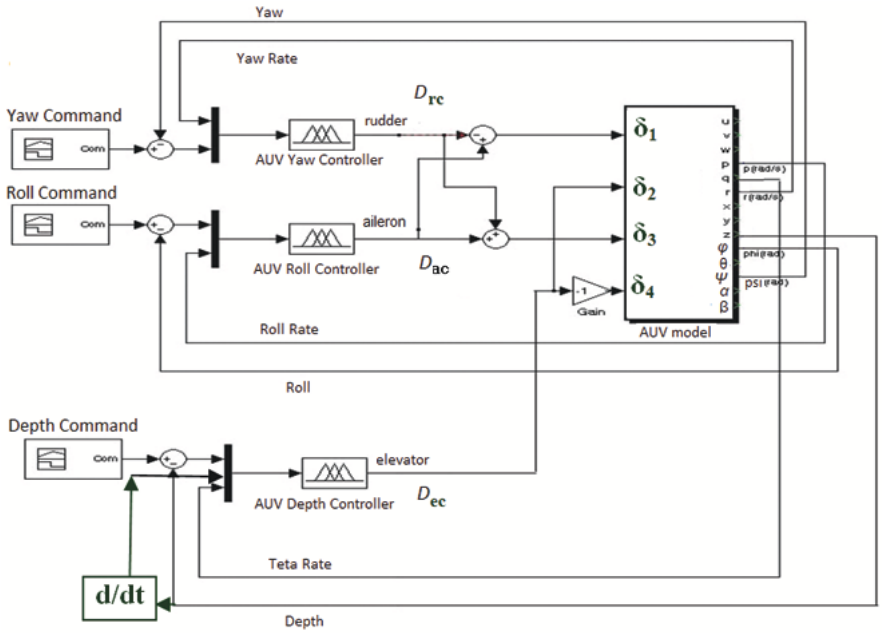


FIGURE 4. Fuzzy Logic Control Block Diagram

4. **Fuzzy Control**

Three separate control loops must be designed for this AUV. The rotation of the system is dumped by the roll loop and the direction angle and depth of the system are forced to follow their commands by the yaw and depth loops. A block diagram of fuzzy logic controller which is used in this research is shown in Figure 4.

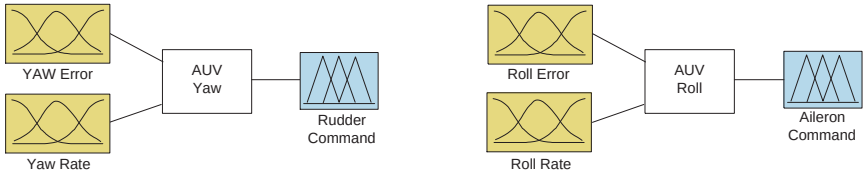
Dynamic of the roll and yaw channels of the AUV are very similar and in addition they use the same flaps as actuators. The control logic of these channels are the same. Depth loop is different and it will be discussed in a separate subsection.

4.1. **Roll and Yaw Loops.**

4.1.1. **Fuzzy Variables.** Outputs of fuzzy blocks are rudder and aileron commands (D_{rc} and D_{ac}) that must be converted to command of flaps (δ_1 and δ_3) with help of equations (3) and (4). As explained in the previous sections, these commands can control roll and yaw angles of the vehicle simultaneously. The AUV equations of motion are completely implemented in 'AUV model' sub-block. Three fuzzy variables are introduced in these two loops to control the vehicle [18]. One of them is an output type and two others are input types. Yaw fuzzy variables are shown in Figure 5(a) and the roll variables are shown in Figure 5(b)

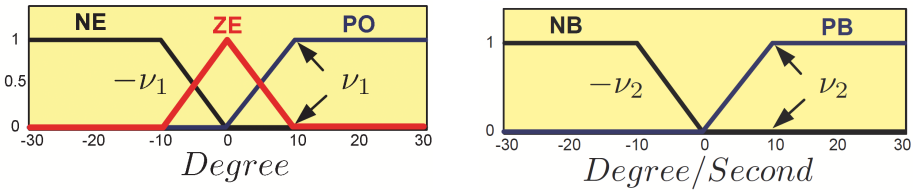
Two fuzzy rule bases are investigated in this research. The first is based on Mamdany and the second is based on Sugeno type.

4.1.2. **Mamdani Type Fuzzy Membership Functions and Rules.** Yaw and roll variables are similar, therefore only the yaw loop is discussed in the following.

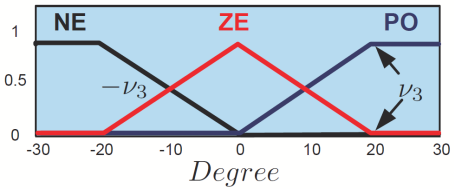


(a) Yaw loop (b) Roll loop

FIGURE 5. Fuzzy Variables



(a) Input variable:yaw error (b) Input variable:yaw error rate



(c) Output variable:Rudder command

FIGURE 6. Fuzzy Membership Function of Yaw Loop

The membership functions of variables of this loop for Mamdani type are shown in Figure 6(a) to (c) and some details are summarized in Table 1. In the first method, to achieve the control goals only five rules in each loop are used. The first to third laws control the system with respect to the error signal and the forth and fifth laws prevent system from unstability. The rules in the yaw loop are as follows:

(1) If error of yaw angle(yawError) is negative(NE)
Then rudder command (RudCom) is negative(NE)

(2) If error of yaw angle(yawError) is zero(ZE)
Then rudder command (RudCom) is zero(ZE)

(3) If error of yaw angle(yawError) is positive(PO)
Then rudder command (RudCom) is positive(PO)

(4) If the rate of yaw angle(yawRate) is positive big (PB)
Then the rudder command (RudCom) is negative(NE).

(5) If the rate of yaw angle (yawRate) is negative big(NB)
Then the rudder command (RudCom) is positive(PO).

These rules in the roll loop can be shown simply by the following expressions:

$$\begin{array}{lllllllll} \text{If} & \text{(rollError)} & \text{is} & \text{(NE)} & \Longrightarrow & \text{(ileCom)} & = & \text{(NE)} \\ \text{If} & \text{(rollError)} & \text{is} & \text{(ZE)} & \Longrightarrow & \text{(ileCom)} & = & \text{(ZE)} \\ \text{If} & \text{(rollError)} & \text{is} & \text{(PO)} & \Longrightarrow & \text{(ileCom)} & = & \text{(PO)} \\ \text{If} & \text{(rollRate)} & \text{is} & \text{(PB)} & \Longrightarrow & \text{(ileCom)} & = & \text{(NE)} \\ \text{If} & \text{(rollRate)} & \text{is} & \text{(NB)} & \Longrightarrow & \text{(ileCom)} & = & \text{(PO)} \end{array}$$

Variable Name	In/Out type	Name	Abbreviation	Fuzzy Membership type
Yaw Error	In	Negative Zero Positive	NE ZE PO	Trampf Trimf Trampf
yaw Rate	In	Negative Big Positive Big	NB PB	Trampf Trampf
Rud Command	Out	Negative Zero Positive	NE ZE PO	Trampf Trimf Trampf

TABLE 1. Parameters of Yaw Loop Fuzzy Variables

4.1.3. **Sugeno Type Fuzzy Rules.** In this method, the number of variables are similar to Mamdani type and two input variables and one output variable exist at each loop. Although the number of variables are the same but the error rate membership functions have been changed. As it is shown in Figure 7, error rate variables in yaw and roll loops have been assigned to positive(PO), negative(NE) and zero(ZE) membership functions . The following rules have been introduced to cover all of the input variable ranges and control the system:

(1) If (yawError) is (NE) AND (yawRate) is (NE) $\Longrightarrow$ (RudCom) = (-C_1)
(2) If (yawError) is (NE) AND (yawRate) is (ZE) $\Longrightarrow$ (RudCom) = (-C_2)
(3) If (yawError) is (NE) AND (yawRate) is (PO) $\Longrightarrow$ (RudCom) = (-C_3)
(4) If (yawError) is (ZE) AND (yawRate) is (NE) $\Longrightarrow$ (RudCom) = (-C_4)
(5) If (yawError) is (ZE) AND (yawRate) is (ZE) $\Longrightarrow$ (RudCom) = 0
(6) If (yawError) is (ZE) AND (yawRate) is (PO) $\Longrightarrow$ (RudCom) = (+C_4)
(7) If (yawError) is (PO) AND (yawRate) is (NE) $\Longrightarrow$ (RudCom) = (+C_3)
(8) If (yawError) is (PO) AND (yawRate) is (ZE) $\Longrightarrow$ (RudCom) = (+C_2)

(9) If (yawError) is (PO) AND (yawRate) is (PO) $\Longrightarrow$ (RudCom) = $(+C_1)$

C_1 to C_4 are constants that they must be tunned to reach proper performance. Four pairs of laws are symmetric and then the value of outputs are chosen negative of each other respectively.

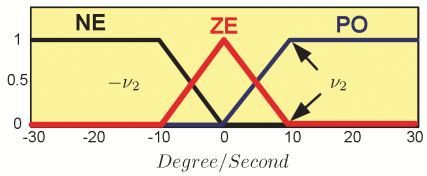


FIGURE 7. Sugeno Method Yaw Error Rate Membership Function

4.2. **Depth Loop.**

Depth dynamic of such system is usually more complicated than roll and yaw dynamic. Therefore it is more difficult to get good depth control performance. Before any changes appear in the depth, the pitch angle (rotation angle around y axis in Figure 1) begins to change. Therefore by controlling the pitch angle, one can in fact control the depth and this results in the stabilization of the depth variables [4, 3]. Based on this observation, three fuzzy variables are selected to control the depth loop in Table 2. The depth loop variables and membership functions of the fuzzy control system are shown in Figure 8(a) and Figure 8(b).

Variables names	Description	Variables types	Membership Functions name	Membership Functions description
zErr	Depth error	Input	NE ZE PO	Negative Zero Positive
zDot	Depth error rate	Input	NE ZE PO	Negative Zero Positive
q	Pitch rate	Input	NE ZE PO	Negative Zero Positive

TABLE 2. Depth Loop Fuzzy Variables and Membership Function Names

Several rule bases are introduced for fuzzy depth loop control and their effectiveness are compared [17]. The following rules are chosen to have a good performance after the optimization process:

$$
\begin{array}{lllllll}
\text{If} & \text{zErr} & \text{is} & \text{(NE)} & \Longrightarrow & f(u) & = & +C_5 \\
\text{If} & \text{zErr} & \text{is} & \text{(ZE)} & \Longrightarrow & f(u) & = & 0 \\
\text{If} & \text{zErr} & \text{is} & \text{(PO)} & \Longrightarrow & f(u) & = & -C_5 \\
\text{If} & \text{zDot} & \text{is} & \text{(NE)} & \Longrightarrow & f(u) & = & +C_6 \\
\text{If} & \text{zDot} & \text{is} & \text{(ZE)} & \Longrightarrow & f(u) & = & 0 \\
\text{If} & \text{zDot} & \text{is} & \text{(PO)} & \Longrightarrow & f(u) & = & -C_6 \\
\text{If} & \text{q} & \text{is} & \text{(NE)} & \Longrightarrow & f(u) & = & +C_7 \\
\text{If} & \text{q} & \text{is} & \text{(ZE)} & \Longrightarrow & f(u) & = & 0 \\
\text{If} & \text{q} & \text{is} & \text{(PO)} & \Longrightarrow & f(u) & = & -C_7
\end{array}
$$

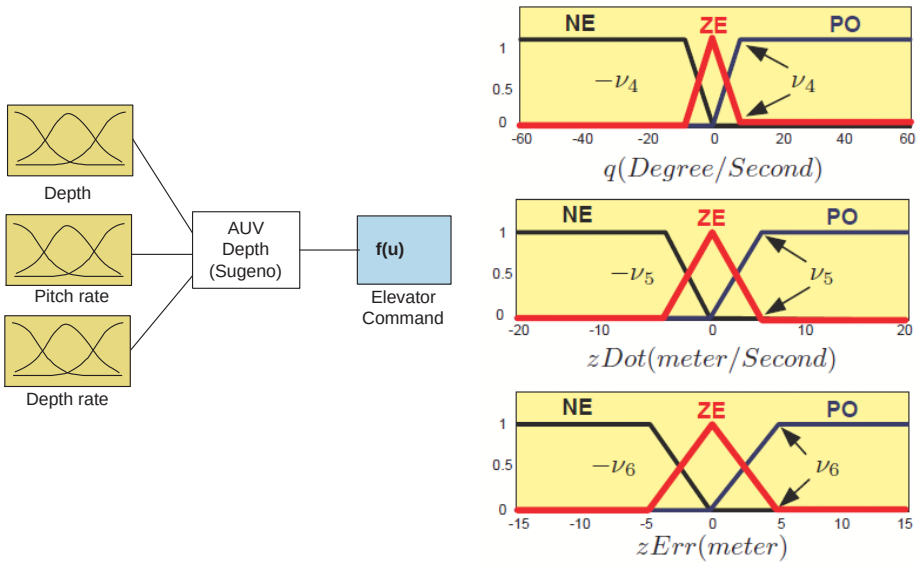
C_5 to C_7 are constants which are fixed in the optimization process.

5. **Optimization**

At this stage, by changing the fuzzy parameters, the output response of each loop is changed and the best parameters are found using a proper numerical method. For each optimization problem a variable vector $\vec{\Theta}$ and a positive scaler cost function $\{E(\vec{\Theta})\}$ must be defined. One of the best candidate for the cost function is root mean square error between the input and the output which can be defined as:

$$E(\vec{\Theta}) \;=\; RMSE \;=\; \sqrt{\frac{\sum_i (out_i - in_i)^2}{n}} \tag{5}$$

This cost function is very simple and effective . If the minimization of the steady state error is the goal of the control system, then a time parameter must be included



(a) Depth loop fuzzy control block diagram (b) Membership functions

FIGURE 8. Depth Loop Variables and Membership Functions

in the cost and this suggests the Integral Time Square Error(ITSE) for a better performance:

$$E(\vec{\Theta}) \;=\; ITSE \;=\; \sum_i t_i \times (out_i - in_i)^2 \tag{6}$$

In the depth and yaw loops, input commands are defined as step and all initial conditions are set to zero. In the roll loop, input command must be set to zero but the initial condition is set to a nonzero value. In this research the normalized steepest descent method [8] was used as the optimization method.

The following notations are used to describe the optimization procedure:

$\vec{\Theta}$	n dimensional vector of the parameters.	n	Number of parameters.
θ_i	The ith parameters.	m	Step number.
$E(\vec{\Theta})$	Cost function.	$g(\vec{\Theta})$	Gradient vector.
$\vec{\Theta}_m$	Parameter vector in the mth iteration.	$\kappa(m)$	Variable step

$$\vec{\Theta} \;=\; [\theta_1\ \theta_2\ ...\ \theta_n]^T \tag{7}$$

$$\vec{g}(\vec{\Theta}) \;=\; \nabla(E(\vec{\Theta})) = \frac{dE(\vec{\Theta})}{d\vec{\Theta}} \tag{8}$$

$$\frac{dE(\vec{\Theta})}{d\vec{\Theta}} \;=\; [\frac{\partial E(\vec{\Theta})}{\partial\theta_1}\ \frac{\partial E(\vec{\Theta})}{\partial\theta_2}\ ...\frac{\partial(\vec{\Theta})}{\partial\theta_n}]^T \tag{9}$$

$$\vec{\Theta}_{m+1} \;=\; \vec{\Theta}_m - \kappa(m)\frac{\vec{g}(\vec{\Theta})}{\|\vec{g}(\vec{\Theta})\|} \tag{10}$$

Although $E(\vec{\Theta})$ is not usually an analytic function, but the gradient operator in equation 8 can be defined for non analytic functions too [9, 10]. For such functions the gradient must be estimated numerically.

Optimizing method begins by selecting an initial vector of parameters. In the roll and yaw loops, three methods are considered. In the first method, the optimization algorithm is applied to Mamdani fuzzy rule base and only the ν_1 is allowed to be changed (The knee of the PB and negative of the knee of the NB membership function as shown in Figure 6(a)). In the second method ν_1, ν_2 and ν_3 are allowed to vary (These variables are the knees of the membership function as shown in Figure 6(a)-(c). In the third method, optimization algorithm is applied to the Sugeno rule base parameters ν_1, ν_2 and C_1, C_2, C_3 and C_4, which were introduced in section 4.1.3. The parameters of three optimization methods are summarized in Table 3.

In the depth loop, only one set of parameters are selected for optimization procedure. These parameters are ν_4, ν_5 and ν_6 in Figure 8(b) and C_5, C_6 and C_7 which were introduced in section 4.2:

$$\vec{\Theta} = \quad [\nu_4 \quad \nu_5 \quad \nu_6 \quad C_5 \quad C_6 \quad C_7]^T \tag{11}$$

First Method	Mamdani with one variable $\vec{\Theta} = [\nu_1]^T$
Second Method	Mamdani with three variables $\vec{\Theta} = [\nu_1 \quad \nu_2 \quad \nu_3]^T$
Third Method	Sugeno with six variables $\vec{\Theta} = [\nu_1 \quad \nu_2 \quad C_1 \quad C_2 \quad C_3 \quad C_4]^T$

TABLE 3. Vector of Variables in Three Optimization Methods

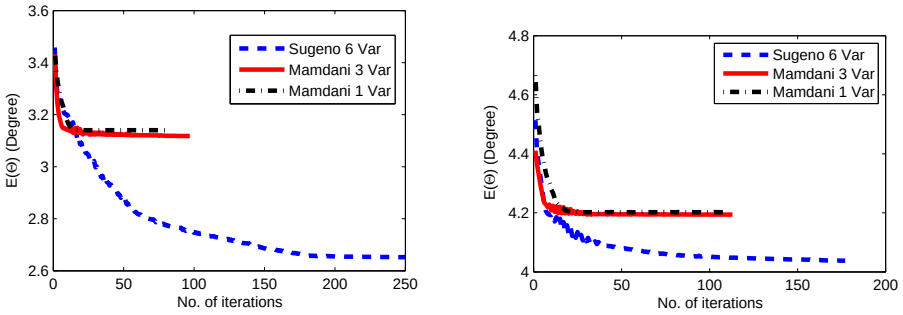
Vector of parameters must be updated at each iteration. The recursive equation (10) expresses the updating algorithm. Gradient vector in equation (9) is calculated from partial derivative $E(\vec{\Theta})$ versus θ_i in n step at each iteration. In order to calculate the gradient vector, the input command is applied to the system simulator and the response of the system is obtained. Then, the defined cost function $E(\vec{\Theta})$ is calculated. With the replacement of $\theta_i + \Delta\theta_i$ with θ_i in the parameter vector($\vec{\Theta}$) the output is found again and the cost function is recalculated. Variation of the cost function versus the variation of θ_i shows the desired partial derivative. To achieve a better performance, $\kappa(m)$ is varied . At the first step, $\kappa(1)$ is set to 1. If $E(\vec{\Theta})$ value decreases in two successive iterations $\kappa(m)$ will be increased by 10%, otherwise if $E(\vec{\Theta})$ value increases, the program decreases $\kappa(m)$ by 10% for the next step to get finer pacing.

6. **Simulation Results**

In this section, the results of simulations are discussed. The trends of the cost function variation during the optimization procedure in all three methods in the yaw and roll loop are shown in Figures 9(b) and 9(a). These curves show that the Sugeno method has better performance than the two other methods in both loops. Figure 10 shows how the optimization algorithm improves the yaw loop transient response. Figures 11(a) and 11(b) compares the final time response of the system after optimization process for different three methods. In both loops, optimized Sugeno method has a better performance not only in the peak overshoot but also for the rise time characteristics.

Depth loop simulation results are different from the two other loops. Figure 12(a) shows the output of the depth loop before and after optimization process with the RMSE criteria(equation (5)). As it can be seen there are some oscillations at the end of simulation time. Changing the cost function to ITSE criterion(equation (6)), the control performance at the end of simulation time is improved as it is shown in Figure 12(a). Figure 13 shows the behavior of the cost function and κ values during the optimization process in the depth loop.

The results of optimized fuzzy method and linear approach [3] are compared in Figure 14. Figures 14(a) to 14(c) show the output of yaw, roll and depth loops for both methods and all cases, the optimized fuzzy method has a better performance than the linear method.



(a) Yaw loop (b) Roll Loop

FIGURE 9. Cost Function Trends in Optimization for the Three Methods

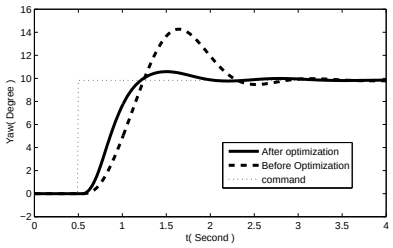
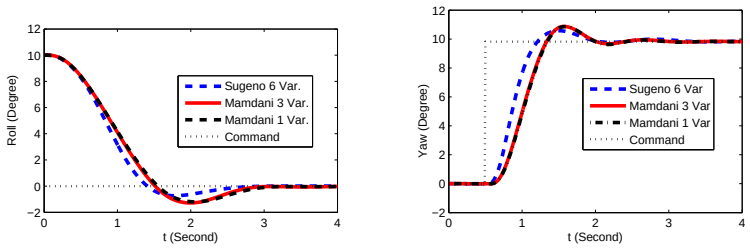


FIGURE 10. Time Response of the Selected Method in Yaw Before and After the Optimization

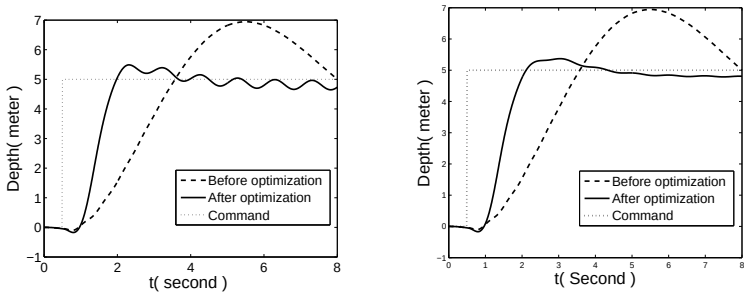
7. **Conclusions**

This study presented a fuzzy logic controller with a few understandable rules which could properly control the AUV system in the yaw, roll and depth loops. For investigating such controllers based on derived motion equations of an underwater variable mass vehicle, a motion simulator was designed for the AUV. Considering the fact that the simulation process must be repeated many times for the parameter optimization, a faster method with aid of a neural network model was proposed.



(a) Roll Loop (b) Yaw loop

FIGURE 11. The Final Time Response of the System After Optimization in Three Methods



(a) Optimization with RMSE cost function (b) Optimization with ITSE cost function

FIGURE 12. Depth Output Before and After Optimization

Besides, it was shown that the normalized steepest descent algorithm with a variable step is a good choice for this special optimization procedure and it can improve the system performance.

Acknowledgments. The authors would like to thank the members of the Mechanic Institute of Iranian Space Center, in particular M. kharati and M. shafiee for providing the equations of motion of the specified AUV.

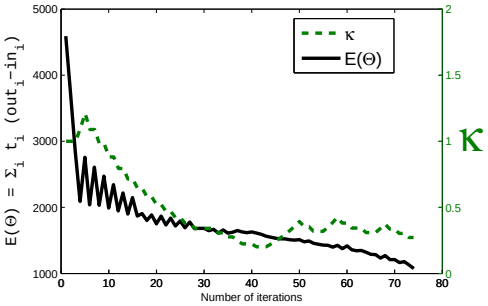
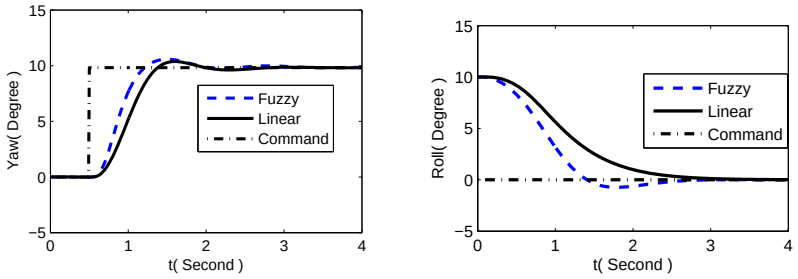
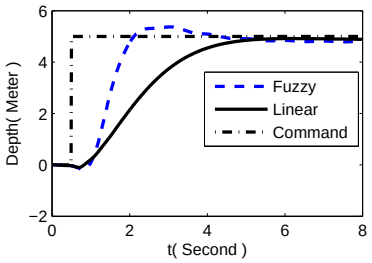


FIGURE 13. Change of Cost Function Value and κ During Optimization



(a) Yaw loop

(b) Roll loop



(c) Depth loop

FIGURE 14. Results Comparison of Linear and Optimize Fuzzy Approach

REFERENCES

[1] G. Antonelli, S. Chiaverini, N. Sarkar and M. West, *Adaptive control of an autonomous underwater vehicle. experimental results on ODIN*, IEEE Proceedings, International Symposium on Computational Intelligence in Robotics and Automation, 1999.

[2] A. Balasuriya and L. Cong, *Adaptive fuzzy sliding mode controller for underwater vehicles*, IEEE Proceedings, The 4th international conference on control and automations (ICCA'03), Canada, June 2003.

[3] T. BinaZadeh, A. R. Khayatian and P. KarimAghaee, *Identification and control of 6 DOF underwater variable mass object*, 13th iranian conference on electrical engineering (ICEE2005), Zanjan, Iran, 2005.

[4] J. Blakelock, *Automatic control of aircraft and missiles*, 2nd edition, Willy, February 1991.

[5] F. Dougherty, T. Sherman, G. Woolweaver and G. Lovell, *An autonomous underwater vehicle (AUV) flight control system using sliding mode control*, Proceedings, OCEANS '88. Baltimore, MD USA, Oct. 1988.

[6] T. Fossen and M. Blanke, *Nonlinear output feedback control of underwater vehicle propellers using feedback from estimated axial flow velocity*, IEEE Journal of Oceanic Engineering, Apr 2000.

[7] T. Fossen, *Guidance and control of ocean vehicles*, John Wiley & Sons, 1994.

[8] J. S. Han, H. S. Kim and J. Neggers, *Actions, norms, subactions and kernels of (fuzzy) norms*, Iranian Journal of Fuzzy Systems, **7(2)** (2010), 141-147.

[9] A. Hasankhani, A. Nazari and M. Sahelis, *Some properties of fuzzy hilbert spaces and norm of operators*, Iranian Journal of Fuzzy Systems, **7(3)** (2010), 129-157.

[10] K. Ishii and T. Ura, *An adaptive neural-net controller system for an underwater vehicle*, Control Engineering Practice, Elsevierm, **8(2)** (2000), 177-184.

[11] J. S. R. Jang, C. T. Sun and E. Mizutani, *Neuro-fuzzy and soft computing*, Prentic Hall, 1997.

[12] N. E. Leonard and P. S. Krishnaprasad, *Motion control of an autonomous underwater vehicle with an adaptive feature*, IEEE Proceedings of Autonomous Underwater Vehicle Technology, AUV '94, Cambridge, MA, USA, Jul 1994.

[13] J. H. Li, P. M. Lee and S. J. Lee, *Neural net based nonlinear daptive control for autonomous underwater vehicles*, IEEE international Conference on Robotics and Automation, May 2002.

[14] Y. Nakamura and S. Savant, *Nonlinear tracking control of autonomous underwater vehicles*, IEEE Proceedings on Robotics and Automation, May 1992.

[15] T. Prestero, *Development of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle*, MTS/IEEE Conference and Exhibition, OCEANS, 2001.

[16] B. Raeisy, M. Kharati, A. A. Safavi and A. R. Khayatian, *Equation of motion derivation of variable mass underwater vehicle and 6DOF simulation with helping of neural network*, 17th Anual International Conference on Mechanical Engineering, Tehran, Iran, May 2009.

[17] B. Raeisy, A. A. Safavi and A. R. Khayatian, *Fuzzy logic depth control of an autonomous underwater vehicle and optimization of it with normalize steepened descent method*, 17th Anual International Conference on Mechanical Engineering, Tehran, Iran, May 2009.

[18] B. Raeisy, A. A. Safavi and A. R. Khayatian, *Optimized fuzzy logic yaw and roll control of an autonomous underwater vehicle*, 8th Iranian Conference on Fuzzy System, Tehran, Iran, October 2008.

[19] L. Rodrigues and P.Tavares, *Sliding mode control of an AUV in the diving and steering planes*, MTS/IEEE Conference Proceedings, MG de Sousa Prado- OCEANS'96, Sep 1996.

[20] N. Sey'edi and M. A. Mirjalili, *Hydrodynamic stability coefficients calculation of submarine and its weapons using added mass and misile DATCOM*, 4th Conference of Underwater Science and Technology (fcoust), Isfahan, May 2007.

[21] E. Shivanian and E. Khoram, *Optimization of linear objective function subject to fuzzy relation inequalities constraints with max-product compozition*, Iranian Journal of Fuzzy Systems, **7(3)** (2010), 51-71.

[22] S. R.Vukelich, S. L. Stoy and M. E. Moore, *Missile DATCOM user's manual*, Dought Aircraft Company Inc., 1988.

[23] J. Wang and G. Lee, *Self-adaptive recurrent neuro-fuzzy control of an autonomous underwater vehicle*, IEEE Transactions on Robotics and Automation, **19(2)** (2003).

Behrooz Raeisy *, School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran and Iranian Space Agency, Iranian Space Center, Mechanic Institute, Shiraz, Iran, P.O. Box: 71555-414
E-mail address: raeisy@shirazu.ac.ir

Ali Akbar Safavi, School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran
E-mail address: safavi@shirazu.ac.ir

Ali Reza Khayatian, School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran
E-mail address: khayatia@shirazu.ac.ir

*Corresponding author