J Mar Sci Technol (2015) 20:559–578
DOI 10.1007/s00773-015-0312-7

ORIGINAL ARTICLE

Modeling and control of autonomous underwater vehicle (AUV) in heading and depth attitude via self-adaptive fuzzy PID controller

Mohammad Hedayati Khodayari · Saeed Balochian

Received: 21 November 2014 / Accepted: 25 February 2015 / Published online: 18 March 2015

Abstract This paper focuses on design of a new self-adaptive fuzzy PID controller based on nonlinear MIMO structure for an AUV. Complexity and highly coupled dynamics, time-variance, and difficulty in hydrodynamic modeling and simulation, complicates the AUV modeling process and the design of proper and acceptable controller. In this work, the comprehensive nonlinear model of AUV is derived through kinematics and dynamic equations and then its treatment in open-loop is verified. In proposed controller, the PID parameters are adjusted by Mamdani fuzzy rules. Combined adaptive methods and dual PID controllers can improve solving of the uncertainty challenge in the PID parameters and AUV modeling uncertainty. The simulation results indicate that developed control system is stable, competent, and efficient enough to control the AUV in tracking the two channels of heading and depth with stabilized speed. Obtained results show that the proposed controller is not only robust, but also gives excellent dynamic, stunning steady-state characteristics and robust stability compared with a classically tuned PID controller.

Keywords AUV · PID (Proportional Integral Derivative) · IAE (Integrated Absolute Error) · AFPIDC (Adaptive Fuzzy PID Controller) · MIMO systems

M. H. Khodayari · S. Balochian (✉)
Department of Electrical Engineering, Gonabad Branch, Islamic Azad University, 96916-29 Gonabad, Khorasan-e-Razavi, Iran
e-mail: saeed.balochian@gmail.com

M. H. Khodayari
e-mail: m.hedayati.kh@gmail.com

1 Introduction

Autonomous underwater vehicles (AUV) is an unmanned submersible in different sizes. It is intended to provide scientists and researchers with simple, low-cost, medium and long-range, appropriate time response capability to collect environmental data. There are a lot of applications for AUVs, including oil industry, survey on underwater animals and plants, operations in dangerous waters, photometric survey, pipeline route survey, seabed mapping, environmental monitoring, chemical plume tracing [1], salvage and rescue requests, and so on [2]. Today the significance of AUVs can easily be understood if the Unmanned Underwater Vehicles (UUV) program for the US Navy is studied [3].

Derivation of AUV parameters is a difficult process and finally its validation demands to analyze practical test like two tank test [4] or telemetry under sea [5, 6]. However, at last, estimation of parameters has uncertainty and variation; therefore, the controller must be self-tuning and robust in counter to variation of AUV parameters and also unpredictable environmental disturbances.

Inherently, nonlinear dynamics of AUVs make it more difficult to exert commonly used linear control. The dynamic characteristics of an AUV are quite complex due to its high nonlinearity, time-varying dynamic behavior, uncertainties in hydrodynamic coefficients, and disturbances caused by sea currents and waves.

A good control system must be regarded for two reasons: the first one for being robust and its ability to account for parameter changes and the second one for having a self-adaptive capability to account for variations in control performance during operation due to environmental disturbances, sensor noise, and changes in AUV dynamics.

Throughout the years various models of control techniques have been proposed. This includes linear controllers

[1, 2, 7, 8, 10, 28], which have performed satisfactorily; SMC controllers [11, 13], adaptive control [12, 13, 22], FLC (Fuzzy Logic Control) [14], predictive control [18–21], static feedback control [30], and neural-network-based control [15–17] have also shown good robustness and tuning ability. Since almost all control methods have some pros and cons, a proper controller can be achieved by the combination of classical and modern intelligent method.

One of the most important disadvantages of linear controllers like LQR and LQG is that they are unable to account for the nonlinearities of the system, thus they can result in suitable performance and even instability in high maneuver treatments.

In brief, the adaptive control is a type of nonlinear control using a system with uncertainty or time-varying parameters. It is implemented on plants with a definite structure with unspecified fixed or slowly varying parameters. Adaptive method is useful for AUVs because of variation of real model parameters. The controller can adapt itself according to the level or characteristics of waves and currents or to the changing weight of AUV.

Also, neural network has some weak points that bind its improvement. It converges to a precise model with long training time and slow rate, which is not acceptable by many systems. Also, classical neural network does not qualify the main requirements such as fast response, less overshoot–undershoot.

SMC is an earlier method that is a good solution for nonlinear system but it can cause chattering on actuators, waste energy, and make fault on fins. However, there are some methods like combination with fuzzy or changing the sign function by saturation function to reduce chattering.

The FLC is easy to use in industrial process because of its simple control structure, easy and cost-effective design [29]. However, FLC with fixed scaling factors and fuzzy rules may not give complete performance if the controlled plant has uncertainty and high nonlinearity [29]. Traditional FLC can have errors in steady state if the system does not have an inherent integrating property. Modern controllers are more robust to dynamic variations and can offer better performance index than classical controllers; however, they may require neat to exact models.

The main aim of this paper is to develop an attitude control system of an AUV based on model REMUS100 via using a self-adaptive fuzzy PID controller. The goal is to

1. Understand the general dynamics of AUVs and especially on the model of REMUS100 by MATLAB2014/SIMULINK.
2. Achieve the existing PID controller employed in REMUS100 model as an initial controller in three channels of depth, heading, and velocity.
3. Develop a detailed understanding of control systems available to AUVs, concentrating on self-adaptive fuzzy PID controller, and its simulation results.
4. Compare and analyze the performance of both controllers in the presence of environmental disturbances, sensor noise, and parameter variations of AUV dynamics.

Controlling AUV is considered to be an important problem due to aforementioned reasons. It is not easy to model these characteristics easily and accurately, hence it is suitable to have a self- adaptive controller to be able to handle parameter variations.

The advantage of using a PID controller is it is simple to implement and maintain, however, it is primarily applicable for linear time-invariant systems, though many extensions to nonlinear systems have been made such as [23] and references therein. As a matter of fact the performance of this type of controller is variable, and the employed methods are essentially a trial and error procedure and do not guarantee suitable and enough acceptable results. In addition, this kind of controller has the disadvantage of the difficulty of adjusting parameter on-line. Hence, the combination of traditional and modern or intelligent methods such as fuzzy controller is useful [24–26].

This paper is organized as the following. Section 1 deals with a brief discussion on the dynamic equations, coordinate systems, and modeling. Section 2 deals with classical controller. In Sect. 3, the proposed controller, i.e. self-tuning PID controller is described. In Sect. 4, comparative simulation results and analysis of the robustness of the controllers are presented.

2 Model description

2.1 Coordinate systems and kinematic and dynamic equations of motion

Generally, the motion of an AUV can be introduced by six degrees of freedom (6-DOF) differential equations of motion [4], [27]. These equations are developed using two coordinate frames shown in Fig. 1. Six velocity components [*u, v, w, p, q, r*] (surge, sway, heave velocity, roll, pitch rate, yaw rate) are defined in the body-fixed frame, while the earth-fixed frame defines the corresponding attitudes and positions [*x, y, z,* φ, θ, and ψ]. It is listed in Table 1. The axis is right-handed. The origin of the body-fixed coordinate system is center of mass, it means [$X_{cg} = 0.00\,\text{m}$ $Y_{cg} = 0.00\,\text{m}$ $Z_{cg} = 1.96e-002$].

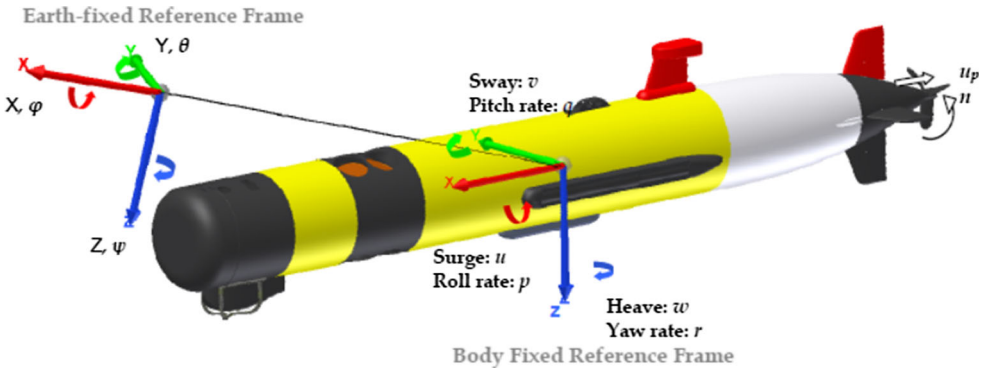


Fig. 1 Reference frame of AUV

Table 1 Symbols used to describe 6-DOF

DOF	Motion	Forces and moments	Linear and angular velocity	Positions and Euler angles
1	Surge	X (N)	u (m/s)	x (m)
2	Sway	Y (N)	v (m/s)	y (m)
3	Heave	Z (N)	w (m/s)	z (m)
4	Roll	K (N m)	p (rad/s)	ϕ (rad)
5	Pitch	M (N m)	q (rad/s)	θ (rad)
6	Yaw	N (N m)	r (rad/s)	ψ (rad)

The AUV motion is described by these vectors:

$$\eta_1 = [x \quad y \quad z]^T \text{position vector}, \qquad \eta_2 = [\phi \quad \theta \quad \psi]^T \text{Euler angles vector},$$
$$v_1 = [u \quad v \quad w]^T \text{linear velocity vector}, \qquad v_2 = [p \quad q \quad r]^T \text{angular velocity vector},$$
$$\tau_1 = [X \quad Y \quad Z]^T \text{forces vector}, \qquad \tau_2 = [K \quad M \quad N]^T \text{moments vector}.$$

It should be considered that in order to avoid singularity in calculations and transformation, Euler angles should be in this boundary:

Transformation between these two coordinate systems is as follows:

$$J_1(\eta_2) = C^T_{z,\psi}.C^T_{y,\theta}.C^T_{x,\varphi} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \quad (1)$$

$$-\pi < \phi \leq \pi, \qquad -\frac{\pi}{2} < \theta < \frac{\pi}{2}, \qquad 0 \leq \psi < 2\pi.$$

For transformation of linear velocities, by the following matrix equation time rate of the displacements described with respect to world coordinate (earth-fixed) frame rates can be obtained as follows:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} = J_1(\eta_2)\begin{bmatrix}u\\ v\\ w\end{bmatrix} \tag{2}$$

Inversely, body coordinate frame velocities can be determined from world coordinate frame velocities in a similar fashion:

$$\begin{bmatrix}u\\ v\\ w\end{bmatrix} = \left[J_1^{-1}(\eta_2)\right]\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} \tag{3}$$

$$J_1(\eta_2) = \begin{bmatrix}\cos\theta\cdot\cos\psi & \sin\varphi\cdot\sin\theta\cdot\cos\psi - \cos\varphi\cdot\sin\psi & \cos\varphi\cdot\sin\theta\cdot\cos\psi + \sin\varphi\cdot\sin\psi\\ \cos\theta\cdot\sin\psi & \sin\varphi\cdot\sin\theta\cdot\sin\psi + \cos\varphi\cdot\cos\psi & \cos\varphi\cdot\sin\theta\cdot\sin\psi - \sin\varphi\cdot\cos\psi\\ -\sin\theta & \sin\varphi\cdot\cos\theta & \cos\varphi\cdot\cos\theta\end{bmatrix} \tag{4}$$

Since the body to world coordination rotation matrix $[J_1]$ is an orthogonal matrix, it follows that inverse of $[J_1]$ equals to transpose of $[J_1]$.

$$J_1^{-1}(\eta_2) = J_1^T(\eta_2) = C_{x,\varphi}\cdot C_{y,\varphi}\cdot C_{z,\varphi} \tag{5}$$

Therefore:

$$\begin{bmatrix}u\\ v\\ w\end{bmatrix} = [J_1(\eta_2)]^T\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{bmatrix} \tag{6}$$

Angular rates described with respect to body-fixed frame are transformed into the time rate of Euler angles by following non-orthogonal transformation matrix.

$$\dot{\varphi} = p + q\,\sin(\varphi)\,\tan(\theta) + r\,\cos(\varphi)\tan(\theta) \tag{7}$$

$$\dot{\theta} = q\,\cos(\varphi) - r\,\sin(\varphi) \tag{8}$$

$$\dot{\psi} = \frac{q\,\sin(\varphi) + r\,\cos(\varphi)}{\cos(\theta)} \tag{9}$$

These three equation forms are represented in matrix notation:

$$\begin{bmatrix}\dot{\varphi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = J_2(\eta_2)\begin{bmatrix}p\\ q\\ r\end{bmatrix} \tag{10}$$

$$J_2(\eta_2) = \begin{bmatrix}1 & \sin(\varphi)\tan(\theta) & \cos(\varphi)\tan(\theta)\\ 0 & \cos(\varphi) & -\sin(\varphi)\\ 0 & \sin(\varphi)\sec(\theta) & \cos(\varphi)\sec(\theta)\end{bmatrix} \tag{11}$$

Angular velocity from Euler angular rates is as follows:

$$\begin{bmatrix}p\\ q\\ r\end{bmatrix} = J_2^{-1}(\eta_2)\begin{bmatrix}\dot{\varphi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} \tag{12}$$

$$J_2^{-1}(\eta_2) = \begin{bmatrix}1 & 0 & -\sin(\theta)\\ 0 & \cos(\varphi) & \sin(\varphi)\cos(\theta)\\ 0 & -\sin(\varphi) & \cos(\varphi)\cos(\theta)\end{bmatrix} \tag{13}$$

Combined speed matrix definitions are as follows in matrix notation:

$$\dot{\eta} = J(\eta_2).v \Leftrightarrow \begin{bmatrix}\dot{\eta}_1\\ \dot{\eta}_2\end{bmatrix} = \begin{bmatrix}J_1(\eta_2) & 0_{3\times3}\\ 0_{3\times3} & J_2(\eta_2)\end{bmatrix}\begin{bmatrix}v_1\\ v_2\end{bmatrix} \tag{14}$$

$$[V]_{world} = [J(\eta_2)][V]_{body} = \begin{bmatrix}J_1(\eta_2) & 0\\ 0 & J_2(\eta_2)\end{bmatrix}[V]_{body} \tag{15}$$

$$\begin{aligned}[V]_{body} &= [J(\eta_2)]^{-1}[V]_{world}\\ &= \begin{bmatrix}[J_1(\eta_2)]^T & 0\\ 0 & [J_2(\eta_2)]^{-1}\end{bmatrix}[V]_{world}\end{aligned} \tag{16}$$

2.2 Dynamics

2.2.1 Forces and moments equations

Centers of buoyancy and gravity are introduced; $r_B = [x_B, y_B, z_B]^T$., $r_G = [x_G, y_G, z_G]^T$

External forces and moments are obtained according to [4]. The external forces acting on the rigid body of AUV are combined of hydrostatic forces, hydrodynamic forces, and forces due to the control surfaces and propeller; that is

$$\sum F_{\text{ext}} = F_{\text{hydrostatic}} + F_{\text{lift}} + F_{\text{drag}} + F_{\text{control}} + F_{\text{disturbances}}. \tag{17}$$

According to this subject of REMUS100, the centers of buoyancy and gravity are according to [27], if the selections of origin in body coordinate are in this way that the inertial moment matrix $I_o = \text{diag}\{I_{xx}, I_{yy}, I_{zz}\}$ is orthogonal, in another word $\{I_{xy}, I_{xz}, I_{yz},\}$ are negligible compared with $\{I_{xx}, I_{yy}, I_{zz}\}$. It means

$$I_o = \begin{bmatrix}I_{xx} & 0 & 0\\ 0 & I_{yy} & 0\\ 0 & 0 & I_{zz}\end{bmatrix} \tag{18}$$

After the simplification of the motion equations for quantities driven by external forces and moments, we have:

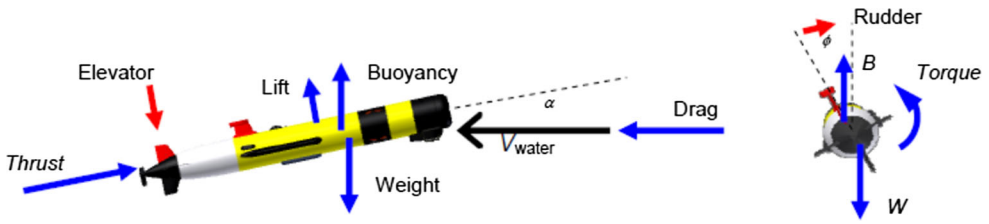


Fig. 2 External forces and moments

$$m\left[\dot{u} - vr + wq - x_G\left(q^2 + r^2\right) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})\right] = X \tag{19}$$

$$m\left[\dot{v} - wp + ur - y_G\left(r^2 + p^2\right) + z_G(pr - \dot{p}) + x_G(qp + \dot{r})\right] = Y \tag{20}$$

$$m\left[w - uq + vp - z_G\left(p^2 + q^2\right) + x_G(rp - q) + y_G(rq + \dot{p})\right] = Z \tag{21}$$

$$I_{xx}\dot{p} + \left(I_{zz} - I_{yy}\right)qr + m[y_G(\dot{w} - uq + vp) - z_G(\dot{v} - wp + ur)] = K \tag{22}$$

$$I_{yy}\dot{q} + (I_{xx} - I_{zz})rp + m[z_G(\dot{u} - vr + wq) - x_G(\dot{w} - uq + vp)] = M \tag{23}$$

$$I_{zz}\dot{r} + \left(I_{yy} - I_{xx}\right)pq + m[x_G(\dot{v} - wq + ur) - y_G(\dot{u} - vr + wq)] = N. \tag{24}$$

The first three equations are related to external forces for translational motion and the second three equations are related to rotational motion. Generally, Schematic of forces and moments are illustrated in Fig. 2.

2.2.2 *Hydrostatic forces and moments*

The buoyancy and weight force vectors do not change with AUV attitude for bodies that are submerged. The buoyant and weight components are acting in the global vertical direction and they must be transformed into components in the body coordinates in order to be added into the equations of motion. By exerting the transformation from earth to body, the vertical force components can be obtained as:

$$f_G(\eta_2) = J_1^{-1}\begin{bmatrix} 0 \\ 0 \\ W \end{bmatrix} \tag{25}$$

$$f_B(\eta_2) = J_1^{-1}\begin{bmatrix} 0 \\ 0 \\ B \end{bmatrix} \tag{26}$$

The hydrostatic forces and moments on the vehicle can be explained as:

$$F_{HS} = f_G - f_B. \tag{27}$$

$$M_{HS} = r_G \times f_G - r_B \times f_B,\ r_G = [x_G, y_G, z_G]^T,\ r_B = [x_B, y_B, z_B]^T \tag{28}$$

These equations can be expanded to yield the nonlinear equations for hydrostatic forces and moments:

$$\begin{bmatrix} F_{\text{hydrostatic}} \\ M_{\text{hydrostatic}} \end{bmatrix} = -\begin{bmatrix} (W-B)s\theta \\ -(W-B)c\theta s\phi \\ -(W-B)c\theta s\varphi \\ -(y_GW - y_BB)c\theta c\varphi + (z_GW - z_BB)c\theta s\phi \\ (z_GW - z_BB)s\theta + (x_GW - x_BB)c\theta c\phi \\ -(x_GW - x_BB)c\theta s\phi - (y_GW - y_BB)s\theta \end{bmatrix} \tag{29}$$

This equation can be added to the right hand side of the equations of motion in Eqs. 19–24. After the combination of other force and moment components such as axial and lateral of body, roll drag, axial and lateral add mass effect, roll added mass, body lift and moment, actuator lifts, and propeller forces according to [4], [27, 28], the final equation obtained is

where right-hand side of Eq. 30 are

$$\begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m - Y_{\dot{v}} & 0 & 0 & 0 & mx_g - Y_{\dot{r}} \\ 0 & 0 & m - Z_{\dot{w}} & my_g & -mx_g - Z_{\dot{q}} & 0 \\ 0 & -mz_g & my_g & I_{xx} - K_{\dot{p}} & 0 & 0 \\ mz_g & 0 & -mx_g - M_{\dot{w}} & 0 & I_{yy} - M_{\dot{q}} & 0 \\ -my_g & mx_g - N_{\dot{v}} & 0 & 0 & 0 & I_{zz} - N_{\dot{r}} \end{bmatrix}\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \sum X \\ \sum Y \\ \sum Z \\ \sum K \\ \sum M \\ \sum N \end{bmatrix} \tag{30}$$

$$\sum X_{ext} = X_{HS} + X_{u|u|}u|u| + X_{\dot{u}}\dot{u} + X_{wq}wq + X_{qq}qq + X_{vr}vr + X_{rr}rr + X_{\text{prop}} \quad (31)$$

$$\sum Y_{ext} = Y_{HS} + Y_{v|v|}v|v| + Y_{r|r|}r|r| + Y_{\dot{r}}\dot{r} + Y_{\dot{v}}\dot{v} + Y_{ur}ur + Y_{wp}wp + Y_{pq}pq + Y_{uv}uv + Y_{uu\delta_r}u^2\delta_r \quad (32)$$

$$\sum Z_{ext} = Z_{HS} + Z_{w|w|}w|w| + Z_{q|q|}q|q| + Z_{\dot{w}}\dot{w} + Z_{\dot{q}}\dot{q} + Z_{uq}uq + Z_{vp}vp + Z_{rp}rp + Z_{uw}uw + Z_{uu\delta_s}u^2\delta_s \quad (33)$$

Table 2 Some of the main used parameters in equations

Parameters	Values	Used equations
Initial surge velocity (u)	1.54 m/s	Eqs. 20–23
Xprop	3.861 N	Eqs. 39
Kprop	0 N m	Eqs. 34, 40
Weight	2.99×10^2 N	Eqs. 19–25, 29, 40
Buoyancy	2.99×10^2 N	Eqs. 26, 29, 40
z_B	0 m	Eqs. 29, 41
x_B	0 m	Eqs. 29, 40
y_B	0 m	Eqs. 29, 40
z_G	0.0196 m	Eqs. 29, 30
x_G	0 m	Eqs. 29, 30
y_G	0 m	Eqs. 29, 30
I_{xx}	0.177 kg m^2	Eqs. 18, 22–24
I_{yy}	3.45 kg m^2	Eqs. 18, 22–24
I_{zz}	3.45 kg m^2	Eqs. 18, 23, 24

$$\sum K_{ext} = K_{HS} + K_{p|p|}p|p| + K_{\dot{p}}\dot{p} + K_{prop} \quad (34)$$

$$\sum M_{ext} = M_{HS} + M_{w|w|}w|w| + M_{q|q|}q|q| + M_{\dot{w}}\dot{w} + M_{\dot{q}}\dot{q} + M_{uq}uq + M_{vp}vp + M_{rp}rp + M_{uw}uw + M_{uu\delta_s}u^2\delta_s \quad (35)$$

$$\sum N_{ext} = N_{HS} + N_{v|v|}v|v| + N_{r|r|}r|r| + N_{\dot{r}}\dot{r} + N_{\dot{v}}\dot{v} + N_{ur}ur + N_{wp}wp + N_{pq}pq + N_{uv}uv + N_{uu\delta_r}u^2\delta_r\,. \quad (36)$$

3 Control methodology

AUV motion in the water is created by propeller system and control fin surfaces. The REMUS AUV control system uses two rudder fins and two elevator (stern) fins. Through the control of propeller and fins deflection, control is achievable. For completeness, this paper presents two control schemes and implements them in AUV control system. These are tuned PID control system and self-adaptive fuzzy PID control system. In this section, the specific designs are presented.

As a matter of fact, there are three means that we want to maximize the maneuverability of the AUV:

1. Depth control
2. Steering (heading) control
3. Forward velocity [surge speed (u)] control.

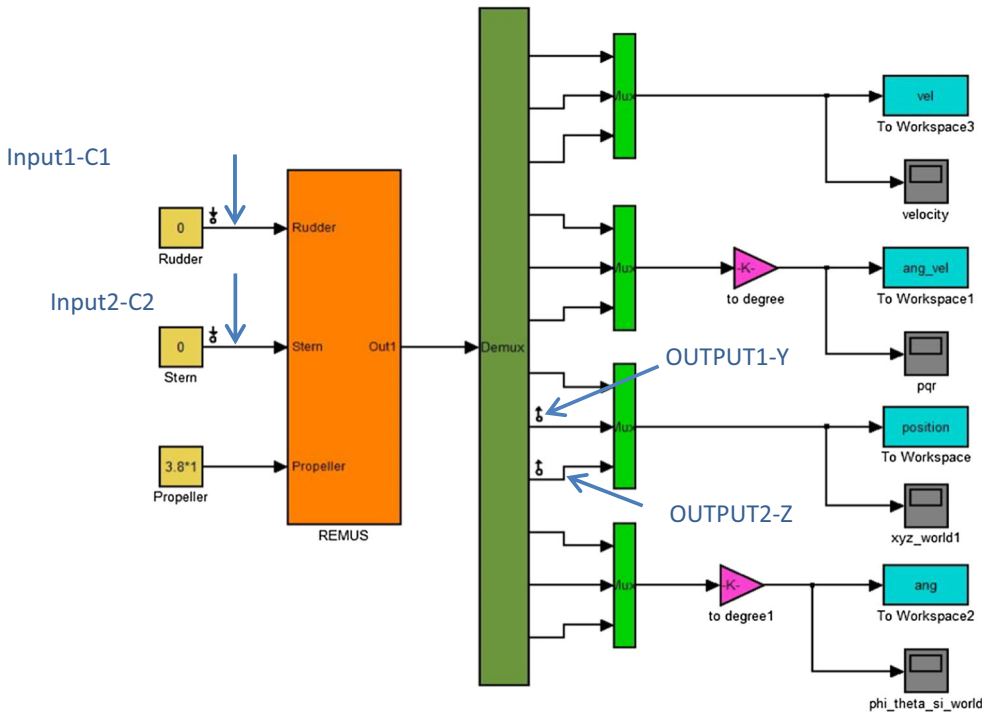


Fig. 3 SIMULINK model of AUV with specified inputs/outputs for linearization and output categorizing

3.1 Modeling in SIMULINK

In this model, initial states are as follows and can be changed to the desired values according to Table 2.

This model is completely nonlinear and has three inputs and three desired outputs. The system has 12 states ($\chi = [u \quad v \quad w \quad p \quad q \quad r \quad x \quad y \quad z \quad \phi \quad \theta \quad \varphi]$). In our model, inputs are $[X_{prop} \quad \delta_s \quad \delta_r]$ and outputs are $[x \quad y \quad z]$ (MIMO structure). This model is shown in Fig. 3. The model includes one sub-block in which nonlinear equations with an integral function (with initial condition $u = 1.54$ *m/s*; $[1.54\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$) has been used in it. In this sub-block, all velocities and acceleration are calculated and obtained by nonlinear equations.

The trim calculation consists of estimating the control parameters—elevator angle, rudder angle and propeller torque—which result in vehicle orientation in which all the forces and moments acting on the vehicle are balanced. For example, two of this transfer function for $\frac{y}{\delta_r}$ and $\frac{z}{\delta_s}$ are in Eqs. 37, 38 through SIMULINK linearization mode. In this linearization, trim point defines in this way: $[u = 1.54$ $v = 0\ w = 0\ p = 0\ q = 0\ r = 0\ \ x\ y\ z\ \phi = 0\ \theta = 0\ \varphi = 0]$. $[x\ y\ z]$ is free and can get any desired values depending on desired trajectory. Note that trim point is a type of equilibrium point and could be defined by our desires and requirements.

related to initial condition and system inputs that are not completely clear in [4].

3.2 PID controller

As mentioned before, the control of *z*-axis and *y*-axis are related to stern and rudder actuators, respectively, but interaction between two types of actuator should not be forgotten. In this literature, the control strategy of both channels is almost identical.

The individual PID controller was tuned using constraint optimization. The constraints are formulated as a feasibility problem, thus the optimization algorithm finds parameter values that satisfy all constraints within specified tolerances but almost does not minimize any objective or cost function.

3.2.1 Speed controller

In this section, in order to simplify the design of the speed controller, only the surge is considered. As a rational assumption, it is assumed that the interactions with other parameters such as heave, sway, pitch, roll, and yaw to swage are minor. Moreover, the system with regardless of forward disturbances is stable in *x*-axis; therefore, it seems that a proportional controller for propeller is enough.

$$G1 = \frac{Y}{C_1} = \frac{0.2721S^8 + 100.5S^7 - 1241S^6 - 1.911e005S^5 - 1.948e006S^4 - 7.429e007s^3 - 4.617e006s^2 - 7.904e008s - 0.0001404}{s^{10} + 531.3s^9 + 6.028e004S^8 + 8.625e005S^7 + 3.701e007S^6 + 3.088e008S^5 + 3.989e008S^4 + 6.467e007s^3 + 5.578e007s^2} \tag{37}$$

$$G2 = \frac{Z}{C_2} = \frac{-0.06271S^7 - 25.08S^6 - 604.3S^5 - 2.266e004S^4 + 4.691e004S^3 + 4.524e007s^2 + 4.162e006s + 5.096e008}{s^9 + 531.3s^8 + 6.028e004S^7 + 8.625e005S^6 + 3.701e007S^5 + 3.088e008S^4 + 3.989e008S^3 + 6.467e007s^2 + 5.578e007s}. \tag{38}$$

After modeling of an AUV dynamics, the validation of open-loop results has been compared with good standing references [4], [28]. It shows that the modeling is very near to these references. Both of the comparing results are in Figs. 4 and 5. In this comparison, in some cases, there are some small and minor differences that most of them are

$$X_{prop} = -X_{u|u|}u|u| = -2.28X_{u|u|}, \tag{39}$$

where $X_{u|u|}$ is axial drag coefficient–resisting forward motion (-1.62 kg m^{-1}) [27].

Moreover, K_{prop} that describes the torque of motor is achievable in model. The main equation of K_{prop} is

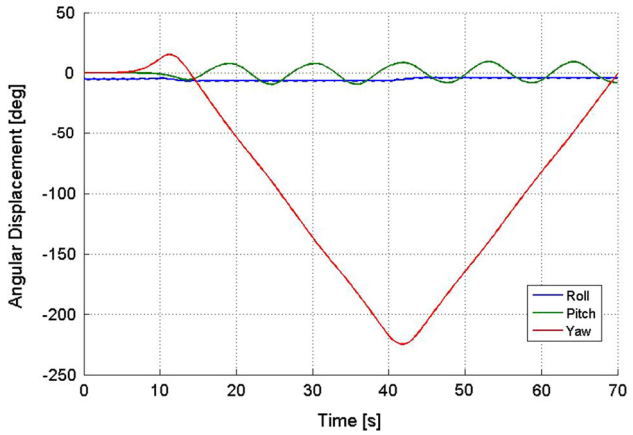


Fig. 4 Euler angles in our open-loop model

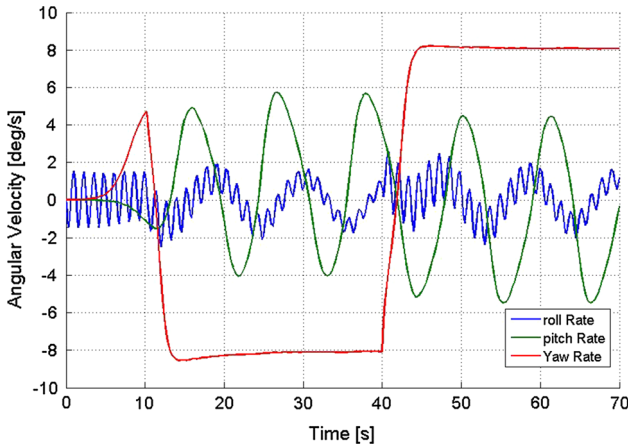


Fig. 5 Angular speed in our open-loop model

$$K_{\text{prop}} = K_{\text{HS}} = (y_g W - y_b B)\cos\theta\cos\phi + (z_g W - z_b B)\cos\theta\sin\phi = 0.995(y_g W - y_b B) - 0.093(z_g W - z_b B). \quad (40)$$

Simulations have been carried out using MATLAB2014/SIMULINK programs to determine the possible values of k_p (for more details about MATLAB2014/SIMULINK programs refer to [31]). The value of k_p is chosen to achieve an acceptable level of performance. The suitable choice for the gain is $k_p = 10.08$ (according to SIMULINK tuning) and the desired speed is 1.54 m/s. Figure 6 shows that the velocity of AUV decreases to 1.27 m/s with the desired speed 1.54 m/s and the steady speed 1.27 m/s. For more accurate speed it can be changed to PI controller. It should be noticed that, it is also seen that the nonlinearity is much more severe at higher speeds than at the lower values.

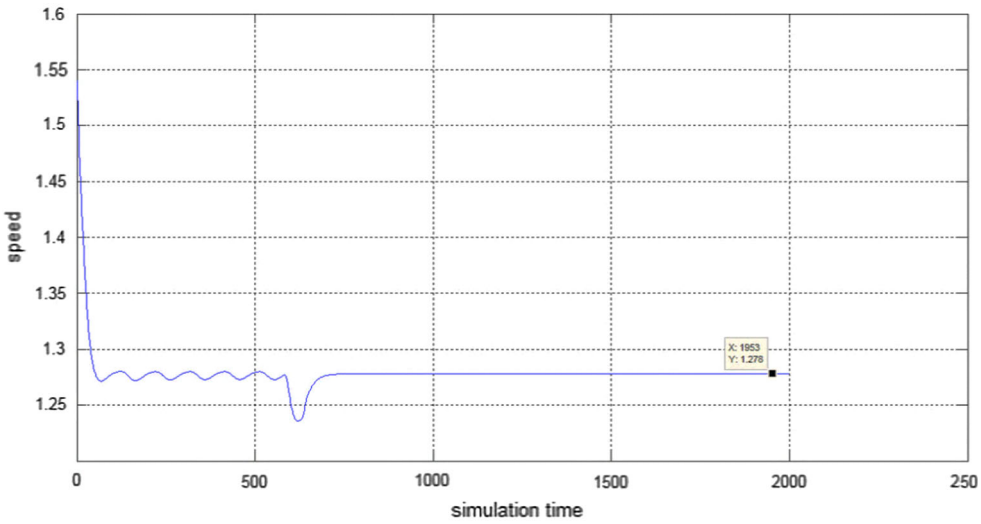
3.2.2 *Depth and heading control by PID controller*

For depth control, the four variables involved are heave velocity $w(t)$, pitch rate $q(t)$, pitch angle $\theta(t)$, and depth $z(t)$. The control variable is the deflection angle of stern planes $\delta_s(t)$. For heading control, the four variables involved are sway velocity $v(t)$, yaw rate $r(t)$, yaw angle $\psi(t)$, and $y(t)$. The control variable is the deflection angle of rudder planes $\delta_r(t)$.

In this work, according to estimated AUV equations and by ignoring minor terms, the control of depth limits to control of θ and z and the control of heading limits to control of φ and y, respectively. In this way, MIMO structure is in this shape by three inputs $[X_{\text{prop}} \quad \delta_s \quad \delta_r]$, and control of two outputs z and y is achievable.

In other words, the control can be divided in the two independent planes. Each plane has inner and outer loop.

Fig. 6 Surge velocity of tuning with $k_p = 10.08$



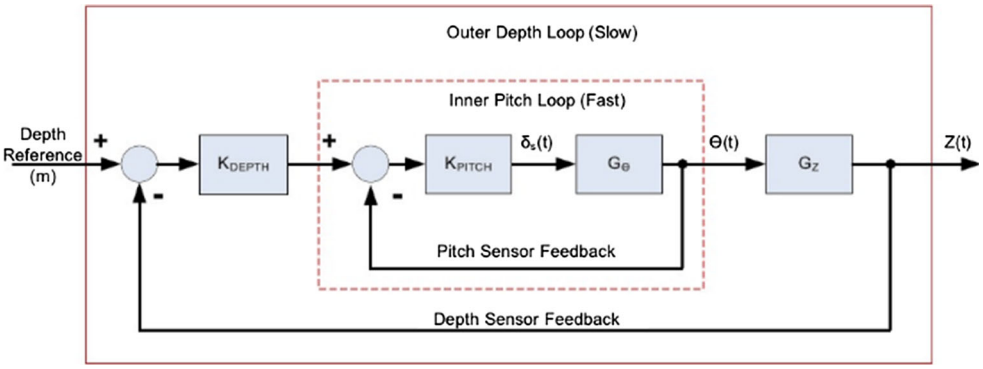


Fig. 7 Main block diagram of depth controller in PID strategy

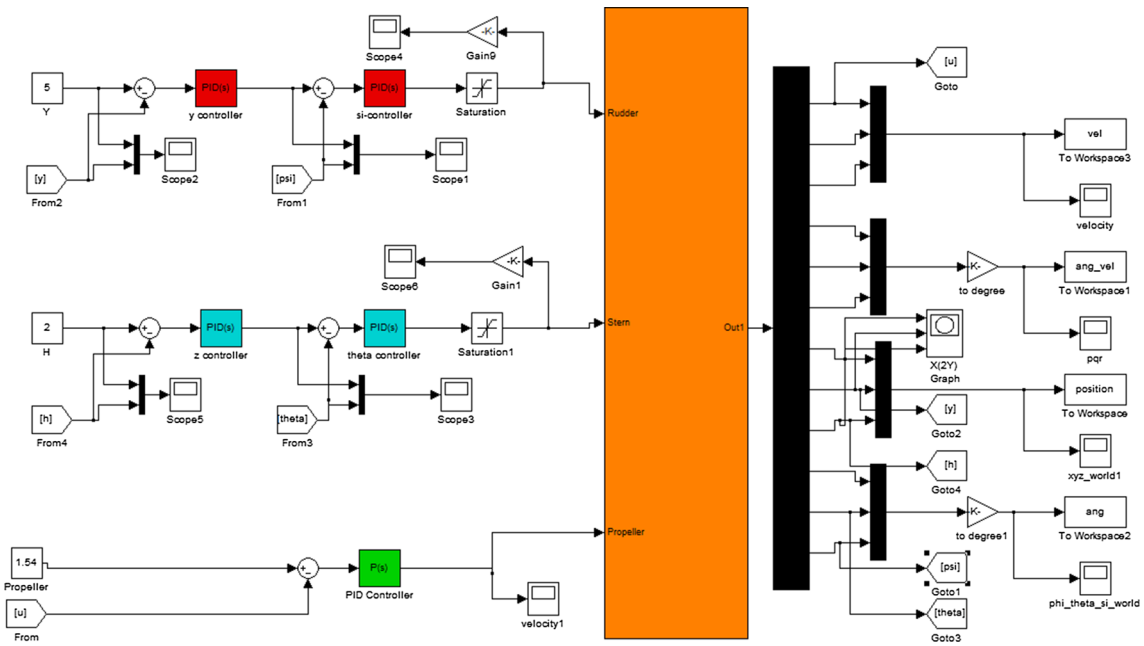


Fig. 8 Total dual PID controller model in SIMULINK

For example in depth we adopt dual loop control methodology by means of an inner pitch control loop and an outer depth control loop. In the dual loop methodology, the depth controller makes a desired pitch angle which becomes the input to the pitch controller. The pitch controller then handles the elevator deflection δ_s, based on the proper pitch angle. This idea for depth is illustrated in Fig. 7.

The inner plane should be stable and faster than outer plane. Therefore, stability and optimal accurate tracking are enough for inner PID loop. This can be done by classical methods or by changing the tuning parameter by increasing the loop speed. Therefore, further improvement was not required. Finally, each plane can be tuned by tuning of outer loop from classical Zeigler–Nichols rules.

It should be mentioned that after the investigation on inner loop by obtaining transfer functions, it is clear that both channels (inner loop of depth and heading channels) have a zero on right side of imaginary axis, and therefore, we have two non-minimum phase system, so proper strategy should be regarded (Fig. 8).

In this model, the boundary of fins is limited between [−10° +10°] and is implemented by saturation function because of actuator inputs. To improve actuator treatments, a low pass filter can be added after saturation function. The results of simulation are shown as follows:

As we can see in Figs. 9, 10 and 11, this controller has a good situation in fins and can track a complex trajectory in a good way. There is no hard effort on fins and tracking without interaction has been met in both channels simultaneously. Tuning parameters have been brought in Table 3.

A disadvantage of the derivative term of the PID controller is that small amounts of noise measurement can

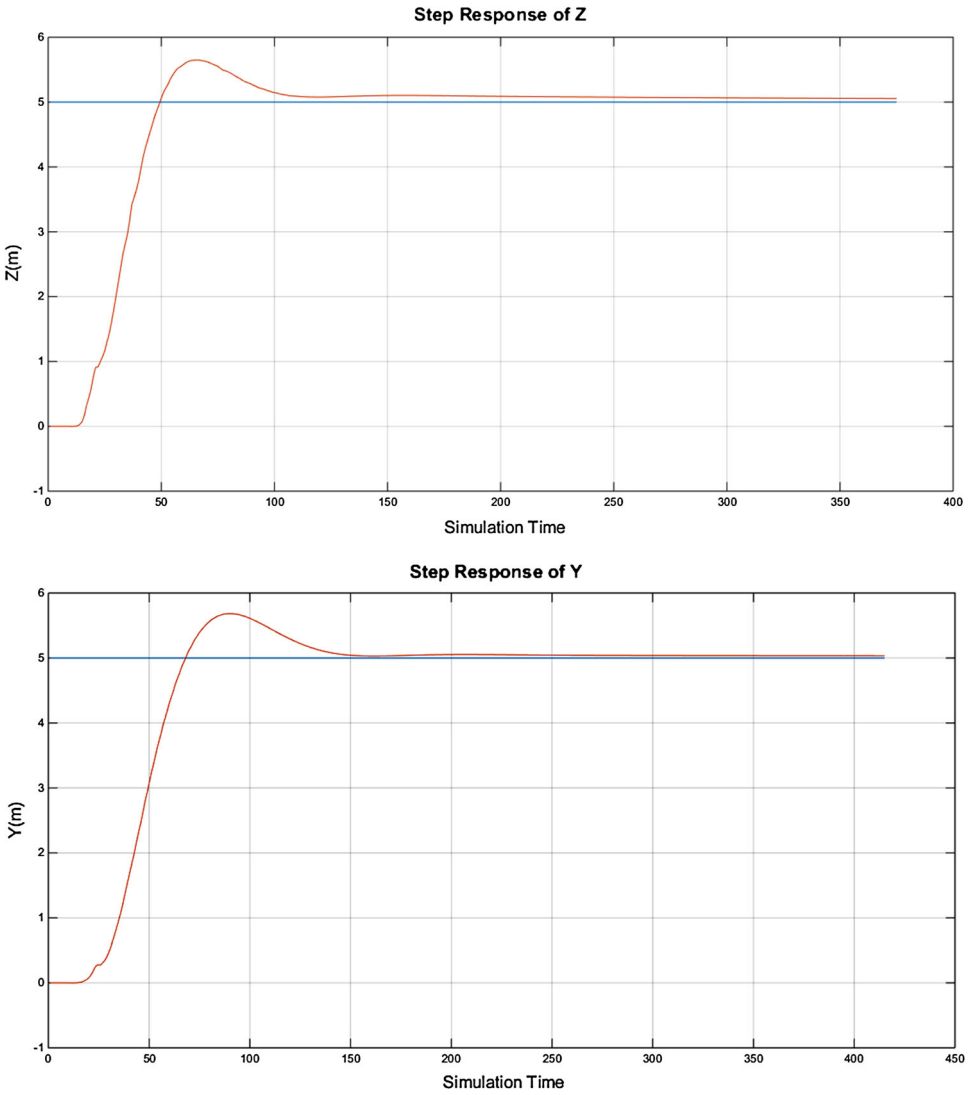


Fig. 9 Step response of Y and Z channel, respectively, from *bottom* to *top*

cause big changes in the output. Measurements are therefore changed with an LPF to remove high frequency noise.

According to physical structure of REMUS100 and its symmetrical weight distribution [4], its low speed and its environmental assumptions [4], the AUV is symmetric about the x–z plane and close to symmetric about the y–z plane. Although the AUV is not symmetric on x–y plane it is assumed that it is symmetric about this plane, so one is able to decouple the degrees of freedom. The AUV can be assumed to be symmetric about three planes since the vehicle operates at low speed. According to linearization in SIMULINK on model, this subject confirmed that the MIMO matrix is $\begin{bmatrix} \frac{y1}{u1} & \simeq 0 \\ \simeq 0 & \frac{y2}{u2} \end{bmatrix}$.

$$H_1 = \frac{y_1}{u_1} = \frac{0.3042S^4 - 1.455S^3 + 2.288S^2 - 34.46S - 116.5}{s^6 + 1.558s^5 + 22.29S^4 + 37.48S^3 - 57.42S^2} \quad (41)$$

$$H_2 = \frac{y_2}{u_2} = \frac{-0.297S^3 + 1.377S^2 + 4.79S + 0.7267}{s^5 + 1.7s^4 - 1.329S^3 + 0.395S^2 + 0.1008S} \quad (42)$$

In order to obtain the minimum proper period of desired trajectory in both channels, the response system in both channels should be investigated simultaneously. This shows that T = 40 s is minimum for having both proper channels at the same time in mixed trajectory.

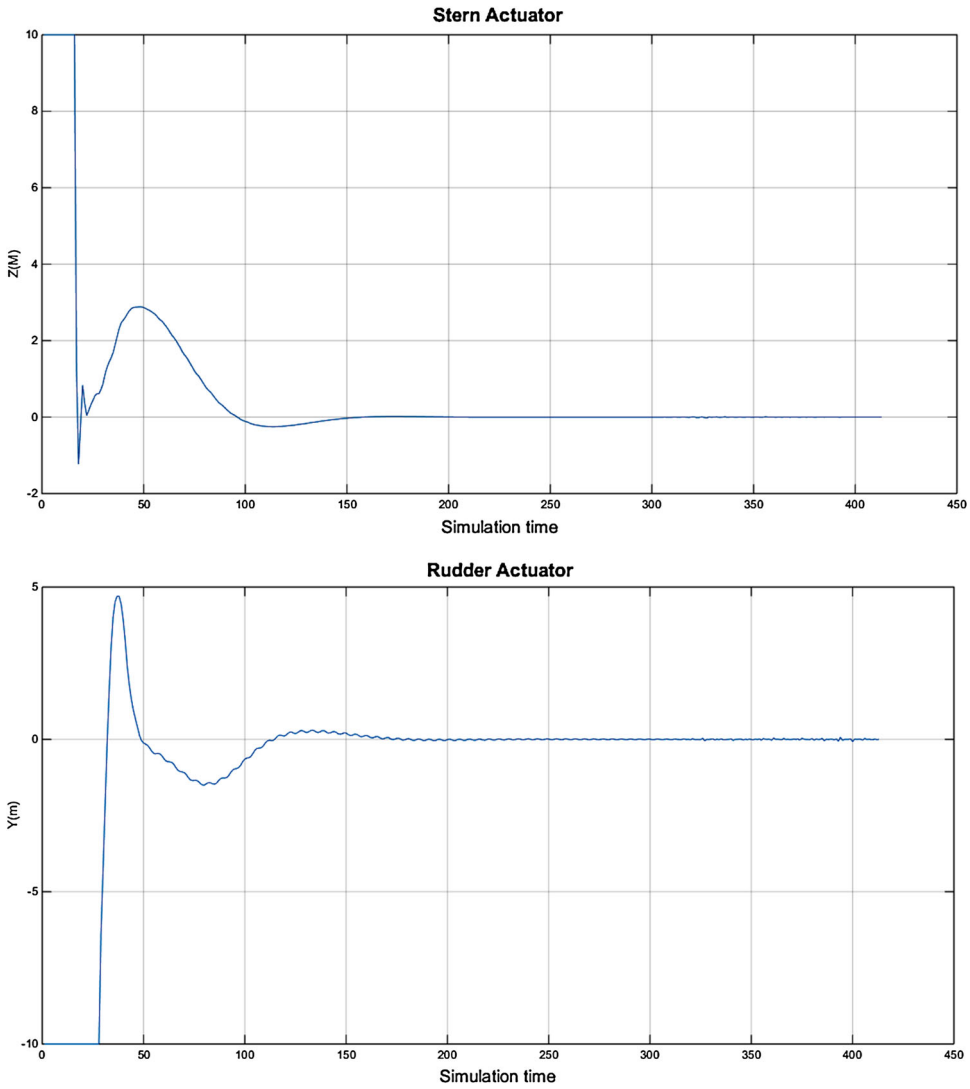


Fig. 10 Situation of rudder and stern, respectively

4 Self-adaptive fuzzy PID controllers

In this work, fuzzy control offers a proper degree of robustness. A variation of PID structure that was mentioned before is formulated as follows:

$$u(t) = \left[k_p(t)e(t)\right] + \int_0^t \left[k_I(\tau)e(\tau)\right] d\tau + \frac{d[k_D(t)e(t)]}{dt} \quad (43)$$

$$= \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \left[k_p(t)e(t)\right] \\ \int_0^t \left[k_I(\tau)e(\tau)\right] d\tau \\ \dfrac{d[k_D(t)e(t)]}{dt} \end{bmatrix}$$

$$= \Psi\Gamma(k_p(t)e(t), k_I(\tau)e(\tau), k_D(t)e(t)), \quad (44)$$

where

$$\Psi = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$\Gamma^T = \left[k_p(t)e(t) \int_0^t \left[k_I(\tau)e(\tau)\right] d\tau \frac{d[k_D(t)e(t)]}{dt} \right] \quad (45)$$

The PID controller is equivalently represented as:

$$u(t) = \left[k_p(t)e(t)\right] + \int_0^t \left[k_I(\tau)e(\tau)\right] d\tau + \frac{d[k_D(t)e(t)]}{dt}$$

$$= \left[k_p^0 + \Delta k_p(t)\right]e(t) + \int_o^t \left[k_I^0 + \Delta k_I(\tau)\right]e(\tau) + \frac{d[k_D^0 + \Delta k_D(t)e(t)]}{dt} \quad (46)$$

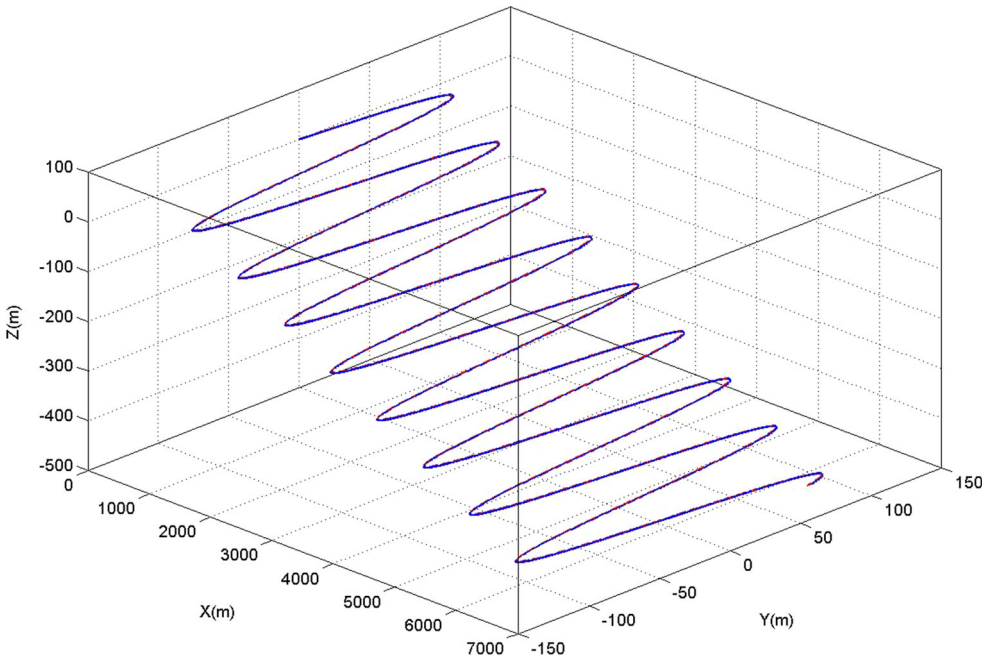


Fig. 11 Path following in 3 axes in dual PID

Table 3 Coefficients of PID controller

Coefficient/ channel	φ	y	θ	z
k_p	−4.9761	0.16742	−6	−0.2803285
k_I	−0.32442	0.0003153	−0.1	−0.000829
k_D	−4.934	−0.45	−9	0.47021093
LPF coefficient (N)	21.6765	0.30459	10.6636	0.47859

where

$$\begin{cases} k_p(t) = k_p^0 + \Delta k_p(t) \\ k_I(t) = k_I^0 + \Delta k_I(t) \\ k_D(t) = k_D^0 + \Delta k_D(t)e(t) \end{cases} \tag{47}$$

The parameters k_p^0·k_I^0, and k_D^0 have been achieved from PID tuning from pervious section and are time-invariant during simulations. But the parameters Δk_P, Δk_I, and Δk_D are adopted during simulation time. Using these time-varying parameters in control system, we have

$$\begin{aligned} u(t) &= k_p^0(t)e(t) + k_I^0 \int_0^t e(\tau)d\tau + k_D^0 \frac{de(t)}{dt} + \Delta k_P(t)e(t) \\ &\quad + \int_0^t \Delta k_I(\tau)e(\tau)d\tau + \frac{d[\Delta k_D(t)e(t)]}{dt} \\ &= u^0(t) + \Delta u(t), \end{aligned}$$

where

$$\begin{cases} u^0(t) = k_p^0(t)e(t) + k_I^0 \int_0^t e(\tau)d\tau + k_D^0 \dfrac{\mathrm{de(t)}}{\mathrm{dt}} \\ \Delta u(t) = \Delta k_\mathrm{P}(\mathrm{t})\mathrm{e}(\mathrm{t}) + \int_0^t \Delta k_\mathrm{I}(\tau)e(\tau)\mathrm{d}\tau + \dfrac{\mathrm{d}[\Delta k_\mathrm{D}(\mathrm{t})\mathrm{e}(\mathrm{t})]}{\mathrm{dt}} \end{cases} \tag{48}$$

In order to generate the signals of Δk_P, Δk_I, and Δk_D, an FLC is recommended in this paper. The FLC is the fuzzy linguistic variables NB, NM, NS, ZR, PS, PM, PB which represent Negative Big, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium, and Positive Big, respectively. The FLC has two inputs. One is the system error e(t) and the other is its change ec(t). To produce the three signals, the FLC needs three outputs. Consequently, the FLC in this paper has two inputs and three outputs that are shown in Figs. 12 and 13.

When the deviation |e| is large, in order to have fast-tracking performance, k_P should be greater, taking a smaller value of k_D prevents instantaneous value of |ec| too large, at the same time a larger system response in order to avoid the overshoot, the integral action should be limited, the $k_I$value should normally be very small.

When the deviation |e| is of medium size, in order to ensure fast system response and have small overshoot, k_P should be reduced, while larger k_D increase the impact of system response, k_I should be appropriate.

When |e| is small, to ensure that the system has the ideal static performance, should make k_P and k_I bigger, while to

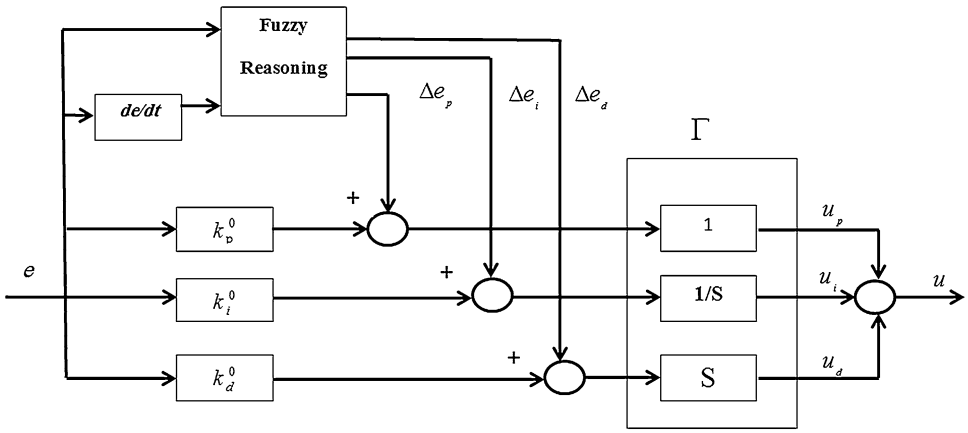


Fig. 12 Configuration of AFPIDC controller

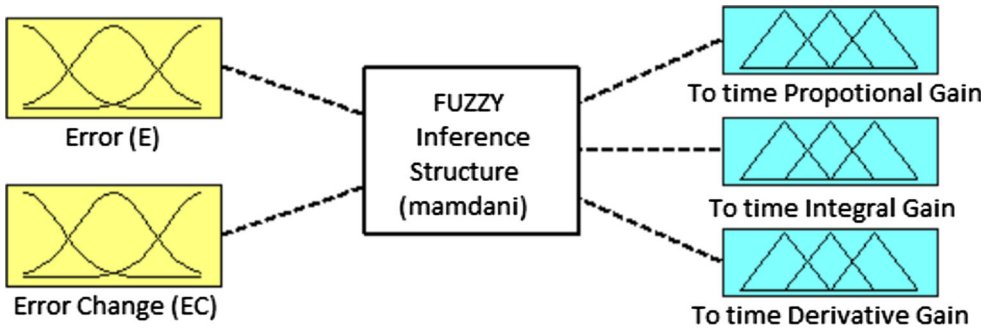


Fig. 13 Selection of inputs/ outputs for designing fuzzy inference structure for AFPIDC

avoid the vicinity of a shock at the system settings, k_D shall be chosen by the change of |e|.

Considering the experience to design fuzzy tuning rules, k_P, k_I and k_D of Fuzzy rule set as follows:

The membership functions for the inputs and the outputs are trimf and gbellmf, respectively. Here gbellmf and trimf represent generalized bell curve membership function and triangular curve member function respectively in fuzzy logic.

The fuzzy rules to compute Δk_P, Δk_I and Δk_D are listed in Tables 4, 5 and 6, in order.

In the following, the fuzzy PID scheme and one of its subsystems (purple blocks) are shown in Figs. 14 and 15. In SIMULINK model, for each channel, entering of disturbance, sensor noise and changing the nonlinear model are considered as input problems and the result has been investigated. According to next simulation, the situation of adaptive fuzzy controller is better than classical PID controller in all alternation of parameters, noise presence and disturbances. Solver of differential equation has been considered Bogacki–Shampine with fixed step size 0.1 s.

In the following, the simulation results of tracking the two inputs are shown in Fig. 16. The left desired trajectory that is more difficult to meet is related to a pulse with amplitude 25 and period 400 s and the right one is related to a sin trajectory with the near highest frequency that

Table 4 The fuzzy control rules for ΔkP

e / ec	NB	NM	NS	Z	PS	PM	PB
NB	PB	PB	PM	PM	PS	PS	Z
NM	PB	PB	PM	PM	PS	Z	Z
NS	PM	PM	PM	PS	Z	NS	NM
Z	PM	PS	PS	Z	NS	NM	NM
PS	PS	PS	Z	NS	NS	NM	NM
PM	Z	Z	NS	NM	NM	NM	NB
PB	Z	NS	NS	NM	NM	NB	NB

AUV can qualify is (T ≈ 40 s). In both trajectories the stability has been met. In this figure, intentionally the trajectory includes variation in three axis, to prove the good cancelation of interaction between axes.

5 Simulation results

In the proposed Adaptive Fuzzy PID Controller (AFPIDC), the parameters k_p^0, k_I^0, and k_D^0 need to be designated. The discourse universes for the *e, ec*, fuzzy logic outputs and fixed PID parameters are listed in Table 7.

In this model, the nonlinear equation of AUV plant is identical to first model in classical PID, but the subsystems include both conventional PID and self-adaptive Fuzzy PID that can switch between them easily and compare the results. At first for initial assessment of step response in both methods, a step with 20 amplitude is illustrated in Fig. 17. This shows that we improved the overshoot, undershoot, and rising time. Improvement in O.V (overshot) is 7.05 % and in undershoot is about 9.55 % (Figs. 18, 19, 20).

Table 5 The fuzzy control rules for ΔkI

e / ec	NB	NM	NS	Z	PS	PM
NB	NB	NB	NB	NM	NM	Z
NM	NB	NB	NM	NM	NS	Z
NS	NM	NM	NS	NS	Z	PS
Z	NM	NS	NS	Z	PS	PS
PS	NS	NS	Z	PS	PS	PM
PM	Z	Z	PS	PM	PM	PB
PB	Z	Z	PS	PM	PB	PB

Table 6 The fuzzy control rules for ΔkD

e / ec	NB	NM	NS	Z	PS	PM
NB	PS	PS	Z	Z	Z	PB
NM	NS	NS	NS	NS	Z	NS
NS	NB	NB	NM	NS	Z	PS
Z	Z	Z	Z	Z	Z	Z
PS	NB	NM	NS	NS	Z	PS
PM	NM	NS	NS	NS	Z	PS
PB	PS	Z	Z	Z	Z	PB

5.1 Encountering by uncertainty in parameters

The story of the problem is as follows the input of the system is considered in y channel (it is desirable, channel z can be selected) and the desired trajectory is a pulse with 20 amplitude, 300 s simulation time period, and pulse width is equal to 50 %. The results are stunning and are shown in Figs. 21, 22 and 23. Substituted parameters for assessment of uncertainty are listed in Tables 8 and 9.

These results show that the total error by IAE performance index during simulation time relatively decreases, and the situation of actuator is very different and better, as well the effort of control decreases. Some other tests in encountering with noise and disturbances have been done in this work; the results and their improvements are listed in Table 10. More details about the way of causing probable disturbance in the sea have been described in [9], but our assumptions on disturbance in this model have been exaggerated to ensure us good robustness of model.

In the following, other comparison of figures between AFPIDC and classical PID in parameter variations and different inputs are depicted and the stability is examined.

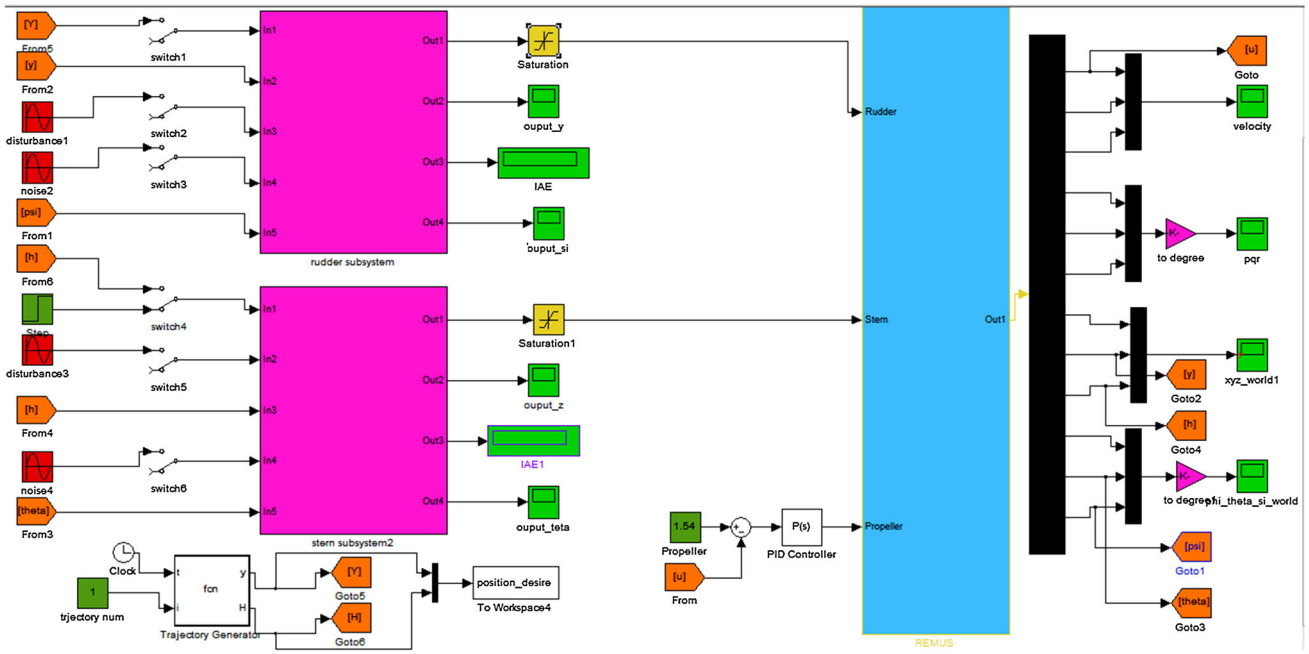


Fig. 14 Total scheme of AFPIDC SIMULINK by possibility of entering noise and disturbance and trajectory maker on *left corner*

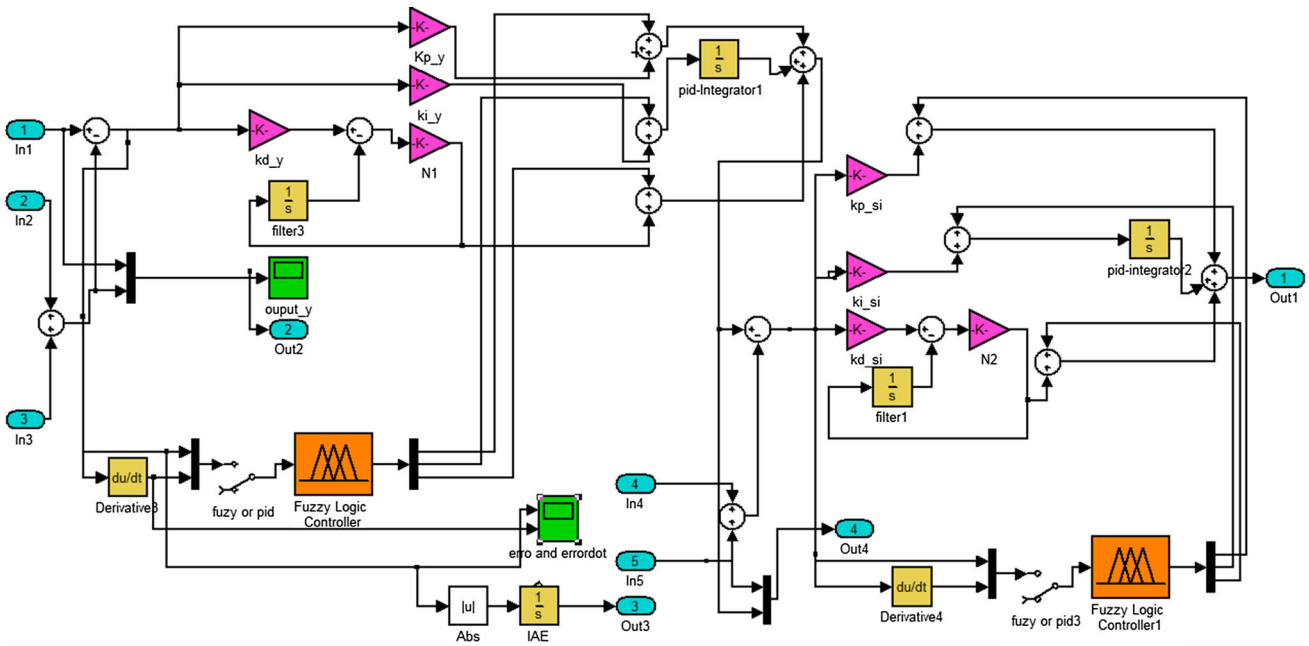


Fig. 15 Fuzzy PID subsystem (*upper purple block* in Fig. 14) in *y*-axis channel

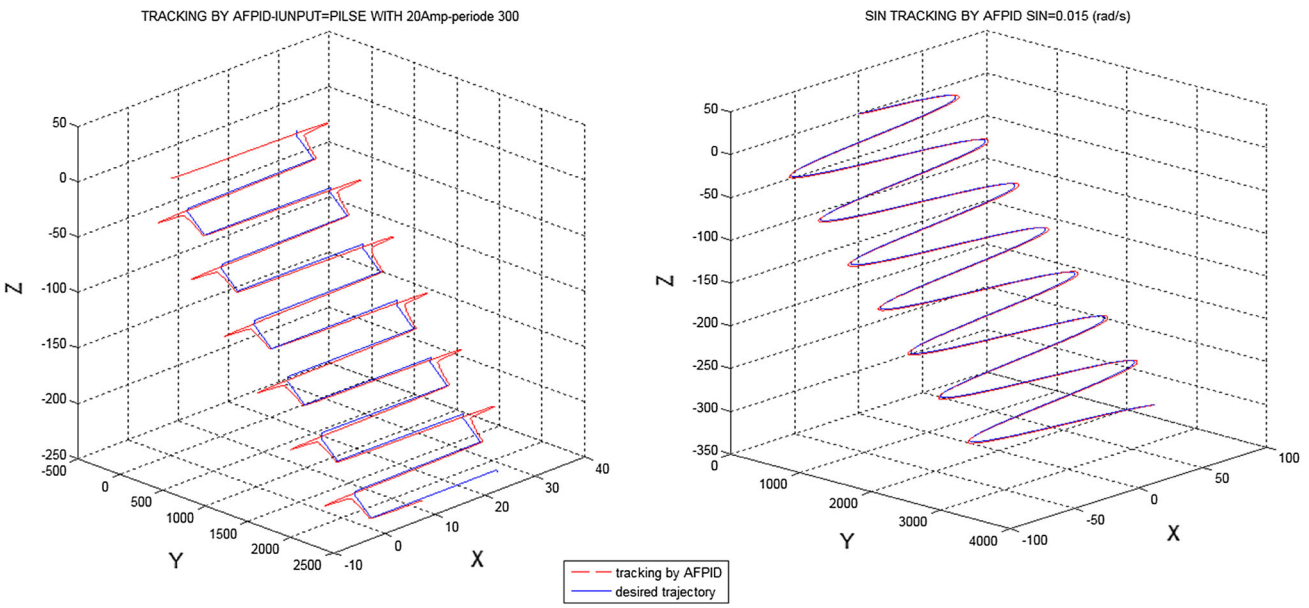


Fig. 16 3D Pulse and sin tracking by AFPIDC

Figures 21, 22, and 23 with variations in internal parameter show that classical PID cannot be stable and track the desired trajectory in the presence of uncertainty parameters. PID controller is completely dependent on nominal design point, however, though PID is almost a robust controller inherently but it cannot adapt itself vice versa as AFPIDC. The tracking of both controllers has been examined by input pulses and in other desired inputs like sin and so on. According to this simulation and these figures (Figs. 21, 22, 23), the situation of classical dual PID deteriorates as time elapses, which is vice versa of AFPIDC that keeps itself in a good manner.

Table 7 FLC domain of inputs/outputs

	e	ec	$\Delta k_p(t)$	$\Delta k_I(t)$	$\Delta k_D(t)$	k_p^0	k_I^0	k_D^0
θ	[−25 25]	[−5 5]	[−3 3]	[−0.06 0.06]	[−4 4]	−6	−0.1	−9
	trimf (triangle)	trimf	gbellmf	gbellmf	gbellmf			
z	[−20 20]	[−5 5]	[−0.885 0.885]	[−0.005 0.005]	[−1.7 1.7]	−0.2803285	−0.000829	0.47021093
	trimf	trimf	gbellmf	gbellmf	gbellmf			
φ	[−25 25]	[−5 5]	[−0.65 0.65]	[−9e−063 9e−063]	[−0.62 0.62]	−4.9761	−0.32442	−4.934
	trimf	trimf	gbellmf	gbellmf	gbellmf			
y	[−16 16]	[−5 5]	[−0.6 0.6]	[−0.0063 0.0063]	[−0.4 0.4]	0.16742	0.0003153	−0.45
	trimf	trimf	gbellmf	gbellmf	gbellmf			

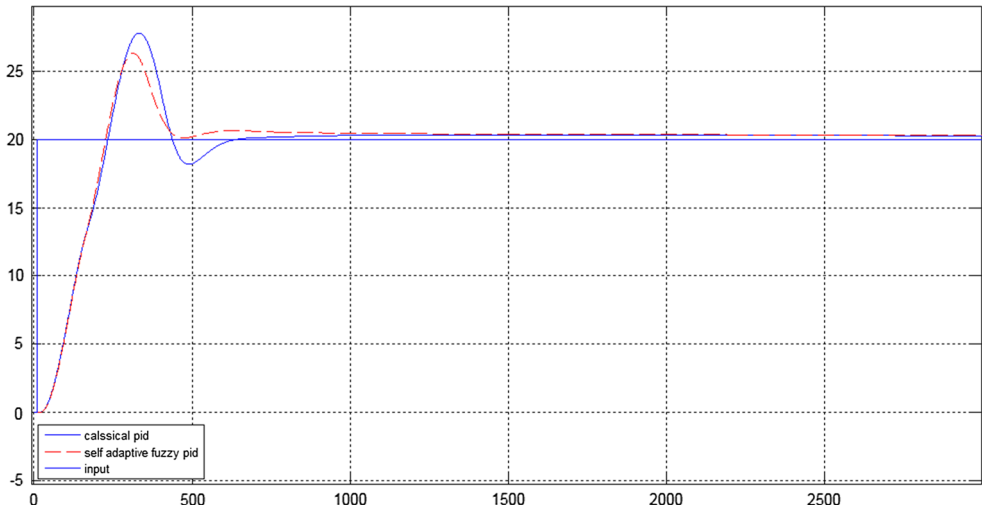


Fig. 17 Comparing of step response in PID by AFPIDC

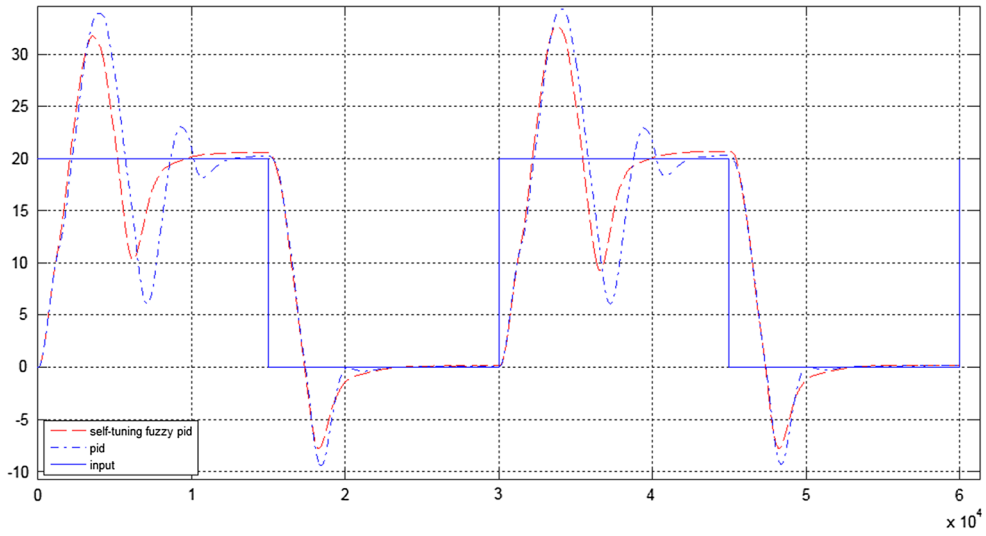


Fig. 18 Comparing response to input pulse entrance as a main trajectory

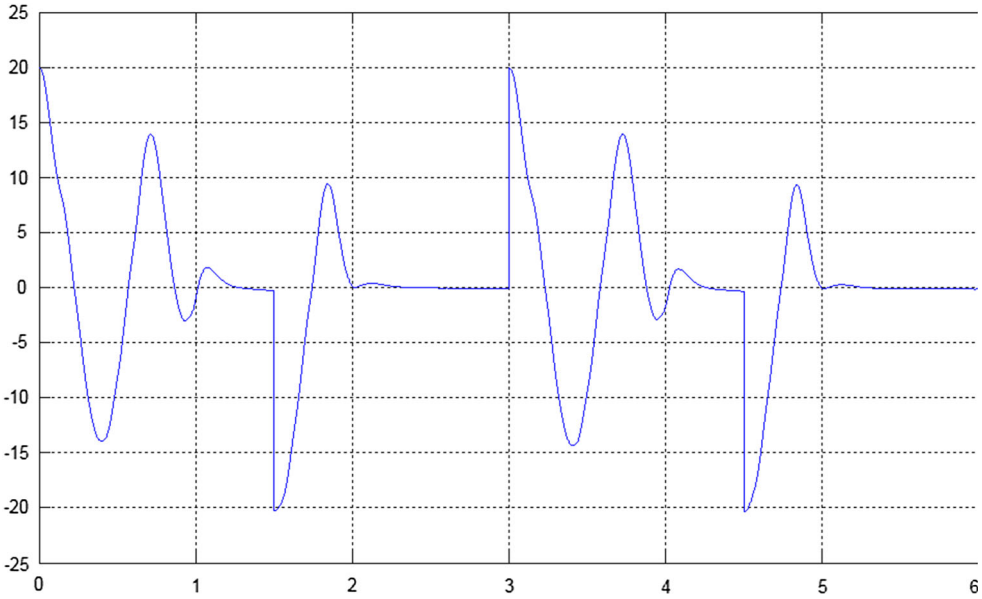


Fig. 19 Error of channel *y* via PID controller

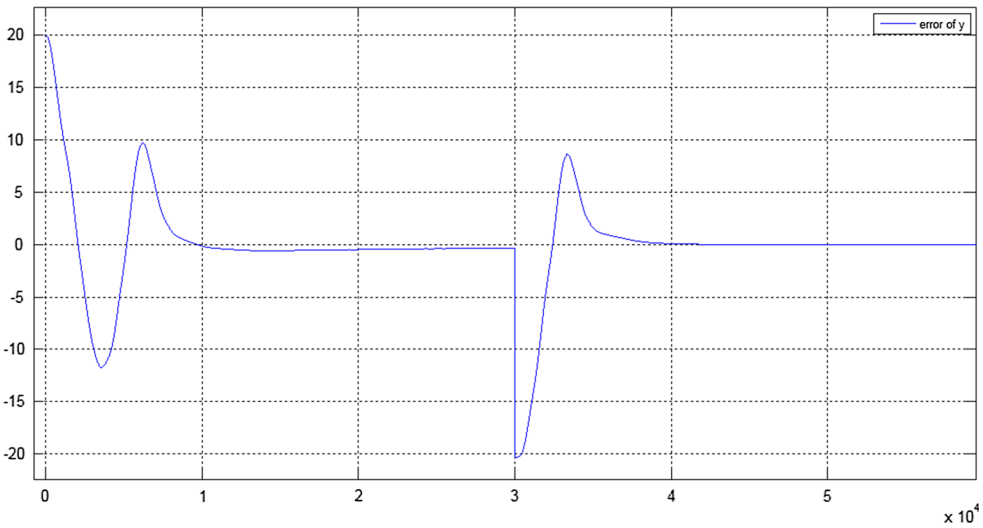


Fig. 20 Error of channel *y* via AFPIDC

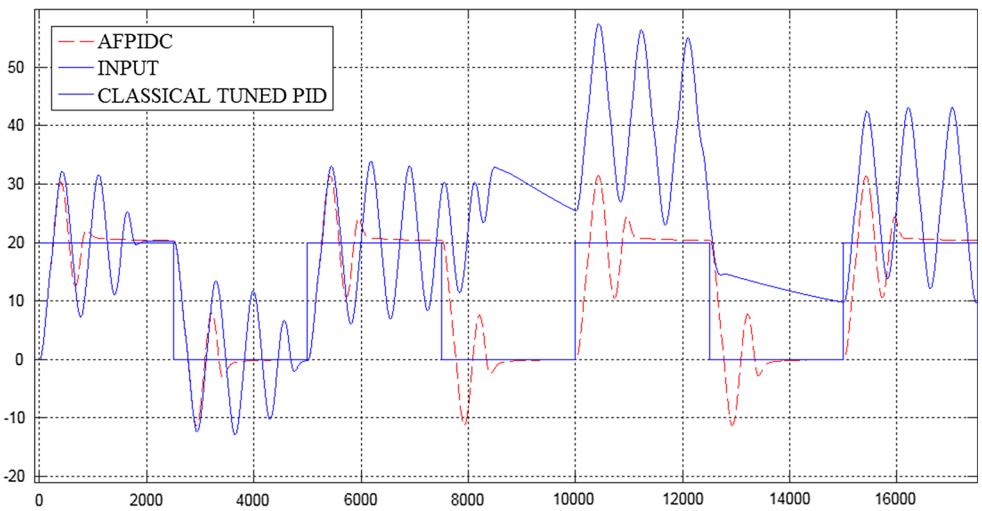


Fig. 21 Desired trajectory with input pulse with period 500 s in *y*-axis with instability in PID No. 7 in Table 10

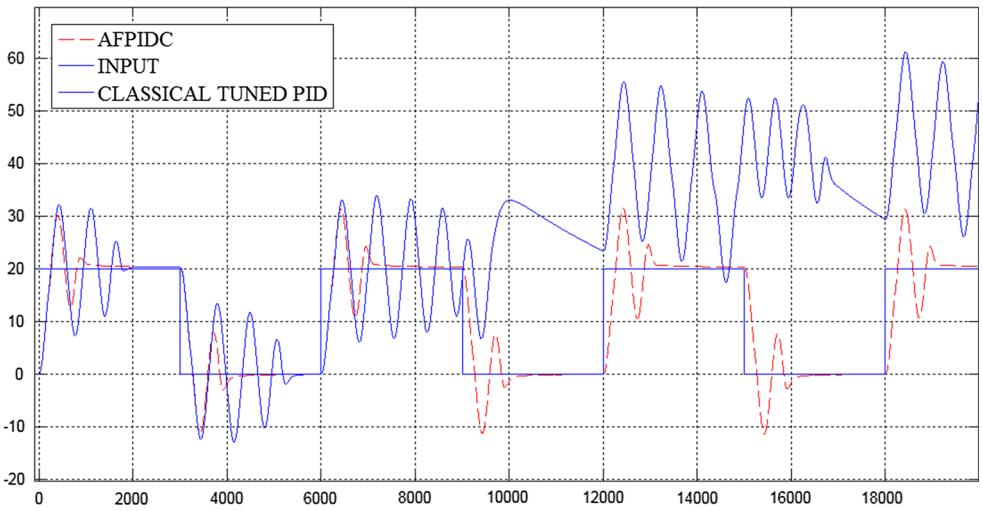


Fig. 22 Desired trajectory with input pulse with period 600 s in y-axis with instability in PID No. 8 in Table 10

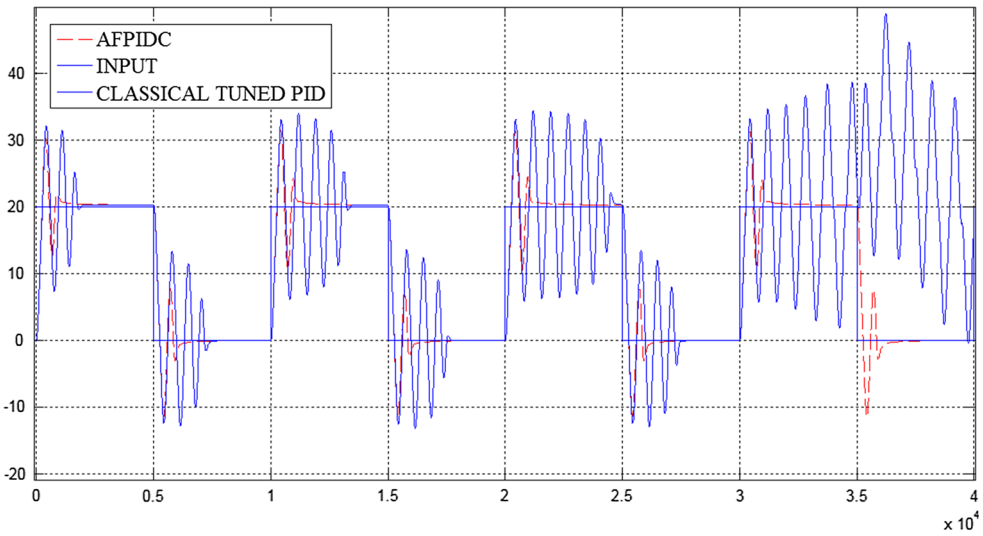


Fig. 23 Desired trajectory with input pulse with period 1000 s in y-axis with instability in PID No. 9 in Table 10

Table 8 Substituted parameters for assessment of uncertainty

Input	Initial parameters	Substituted parameters	Improvement results
Pulse type	Yuv = −2.86e1	Yuvl = −22.8e1	First O.V = 11 %
Amplitude:20	Zuw = −2.86e1	Zuwl = −22.86e1	First under shoot = 21.02 %
Priode:300 s	Muw = −4.47	Muwl = −0.47	Second O.V = 16.05 %
Pulse width = 50 %	Nuv = 4.47	Nuvl = 18.47	Error of Adaptive Fuzzy by IAE = 1192
	Zuq = 9.3e−1	Zuqa = 19.3e−1	
	Muq = 1.93	Muqa = 5.93	Error of PID by IAE index = 2583 (IAE in during of simulation time)
	Muw = 3.46e1	Muwa = 9.46e1	
	Weight = B	Weight = 2.99e2	
	Weight = 2.99e2	B = 3.06e2	
	Iyy = 3.45	Iyy = 0.25	
	Izz = 3.45	Izz = 0.25	
	zg = 0.0196	zg = 0.0196	
	xg = 0.0	xg = 0.005	
	yg = 0.0	yg = 0.007	

Table 9 Introduction of the parameters have been altered in Table 8

No.	Parameter	Description	Units
1	Yuv	Drag resisting sway due to forward and sway motion	kg m^{-1}
2	Zuw	Drag resisting heave due to forward and yaw motion	kg m^{-1}
3	Muw	Coefficient of moment resisting pitch due to forward and yaw motion	kg
4	Nuv	Coefficient of moment resisting yaw due to forward and sway motion	kg
5	Zuq	Heave coefficient for forward & pitching motion	kg rad^{-1}
6	Muq	Pitch mom Coefficient for forward & pitching motion	kg m rad^{-1}
7	Muw	Coefficient of moment resisting pitch due to forward and yaw motion	kg m^{-1}
8	Iyy	Vehicle moment of inertia around *y*-axis	kg m^2
9	Izz	Vehicle moment of inertia around *z*-axis	kg m^2
10	x_g	Center of gravity in x-axis	m
11	y_g	Center of gravity in y-axis	m

Table 10 Description in the presence of noise and disturbance via AFPIDC

No.	Input pulse [Amp-period(s)] (Desired trajectory)	Sensor noise (Amp-ω (rad/s))	Disturbance (Amp-ω (rad/s))	Improvement results
1	20–100	Amp = 0.1 ω = 0.011	None	O.V = 8.15 % Under shoot = 4.14 %
2	20–260	None	Amp = 13 ω = 0.01	O.V = 8.85 % Under shoot = 3.37 %
3	20–200	Amp = 0.19 ω = 0.035	Amp = 0.12 ω = 0.9	O.V = 7.7 % Under shoot = 2.15 %
4	20–260	Amp = 0.3 ω = 0.055	None	O.V = 4.6 % Under shoot = 3.35 %
5	20–260	None	Pulse 1, 2: Time:[0 60], Amp = 5 Time:[160 200], Amp = −5	O.V = 7.05 % Under shoot = 4.11 %
6	20–260	None	Pulse 1, 2: Time:[0 80], Amp = 10 Time:[160 200], Amp = −10	O.V = 6.95 % Under shoot = 4.1 %
7	20–450	None	None	Unstable PID Stable AFPIDC Figure 21
8	20–600	None	None	Unstable PID Stable AFPIDC Figure 22
9	20–1000	None	None	Unstable PID Stable AFPIDC Figure 23

6 Conclusion

In this paper, after the modeling of an AUV according to the first mechanical principles and verifying it by outstanding references, a fuzzy self-adapting PID controller has been presented. In the proposed approach, the AFPID controllers have two inputs. One input signal is a function of the system error which is proportional to the traditional PID parameters k_p^0, k_I^0, and k_D^0, respectively, and the other one is rate of system error. The parameters k_p^0, k_I^0, and k_D^0 are kept fixed during the PID controller working according

to initial optimizing. The performance of the proposed controller is investigated in simulation by SIMULINK. In this literature, the requirements for implementation in reality have been considered. The simulation results show that the AFPID controller's adaptive ability and robustness are very better than the dual PID controller. It also shows that the proposed AFPIDC can improve the robustness, effort control, overshoot, undershoot, and good stability compared to the conventionally tuned PID in different conditions. Today fuzzy logic and PID controller have been implemented in many industries easily by means of microcontroller or mini-PLC, therefore implementation of this controller would be easy.

References

1. Farrell JA, Pang S, Li W, Arrieta R (2004) Biologically inspired chemical plume tracing demonstrated on an autonomous underwater vehicle, Man, and Cybernetics Conference, September 2004, Hague, Netherlands
2. Yildiz O, Gokalp RB, Yilmaz AE (2009) A review on motion control of the Underwater Vehicles. In: Proceedings of electrical and electronics engineering, 2009. ELECO 2009, Bursa, 2009, pp 337–341
3. UUV programs (2007) http://ftp.fas.org/irp/program/collect/uuv.htm
4. Prestero T (2001) Verification of a six-degree-of-freedom simulation model for the REMUS autonomous underwater vehicle, MSc/ME Thesis, Massachusetts Institute of Technology
5. Geisbert JS (2007) Hydrodynamic modeling for autonomous underwater vehicles using computational and semi-empirical methods. Verginia Polytechnis Institute and State University
6. Yue C, Guo S, Li M (2012) ANSYS fluent-based modeling and hydrodynamic analysis for a spherical Underwater robot. In: Proceedings of 2012 IEEE international conference on mechatronics and automation, pp 1577–1581
7. Guo S, Mao S, Shi L, Li M (2012) Design and kinematic analysis of an amphibious spherical robot. In: Proceedings of 2012 IEEE international conference on mechatronics and automation, pp 2214–2219
8. Herman P (2009) Decoupled PD set-point controller for underwater vehicles. J Ocean Eng 36(6–7):529–534
9. Model based predictive control of AUVS for station keeping in a shallow water wave environment (2005) Naval Postgraduate School, Center for AUV search, Monterey, CA, 93943-5000
10. Wadoo S, Kachroo P (2010) Autonomous underwater vehicles: modeling, control design and simulation. CRC Press, edn 1
11. Buckham BJ, Podhorodeski RP, Soylu S (2008) A chattering-free sliding-mode controller for underwater vehicles with fault tolerant infinity-norm thrust allocation. J Ocean Eng, 35(16):1647–1659
12. Qi X (2014) Adaptive coordinated tracking control of multiple autonomous underwater vehicles. Ocean Eng 91:84–90
13. Zeinali M, Notash L (2010) Adaptive sliding mode control with uncertainty estimator for robot manipulators. Mech Mach Theory 45(1):80–90
14. Jun SW, Kim DW, Lee HJ (2011) Design of T-S fuzzy-model-based controller for depth control of autonomous underwater vehicles with parametric uncertainties. In: 2011 11th international conference on control, automation and systems, ICCAS 2011, Gyeonggi-do, Korea, Republic of, 2011, pp 1682–1684
15. Kumar N, Panwar V, Sukavanam N, Sharma SP, Borm JH (2011) Neural network-based nonlinear tracking control of kinematically redundant robot manipulators. Math Comput Model 53(9–10):1889–1901
16. Sun T, Pei H, Pan Y, Zhou H, Zhang C (2011) Neural network-based sliding mode adaptive control for robot manipulators. Neurocomputing 74(14–15):2377–2384
17. Xu B, Pandian SR, Sakagami N, Petry F (2012) Neuro-fuzzy control of underwater vehicle-manipulator systems. J Franklin Institute, 349(3):1125–1138
18. Medagoda L, Williams SB (2012) Model predictive control of an autonomous underwater vehicle in an in situ estimated water current profile. Oceans, Yeosu, pp 1–8
19. Steenson LV (2013) Experimentally varied model predictive control of a hover-capable AUV. PhD thesis, University of Southampton
20. Wang L (2010) Model predictive control system design and implementation using MATLAB, Springer
21. Steenson LV, Phillips AB, Turnock SR, Furlong ME, Rogers E (2012) Effect of measurement noise on the performance of a depth and pitch controller using the model predictive control method. Autonomous underwater vehicles (AUV), 2012 IEEE/OES, 1(8):24-27
22. Mohan S, Kim J (2012) Indirect adaptive control of an autonomous underwater vehicle-manipulator system for underwater manipulation tasks. Original Res Article Ocean Eng 54(1):233–243
23. Cooney LA (2009) Dynamic response and maneuvering strategies of a hybrid autonomous underwater vehicle in hovering. Thesis of master of science in ocean engineering, Massachusetts Institute of Technology
24. Yang Y, Yang W, Wu M, Yang Q, Xue Y (2010) A new type of intelligent control and automation, Jinan, Adaptive Fuzzy PID Controller. In: Proceedings of the 8th World Congress on China
25. Chang X, Jianhong L, Ming C et al (2007) Neural network PID adaptive control and its application. Control Eng China 14(3):284–286 (In Chinese)
26. Jiangjiang W, Chunfa Z, Youyin J (2008) Adaptive PID control with BP neural network self-tuning in exhaust temperature of micro gas turbine. In: Proceedings of the 2008 IEEE international conference on industrial electronics and applications, Piscataway, NJ, USA, pp 532–537
27. Sgarioto D (2008) Steady state trim and open loop stability analysis for the REMUS autonomous underwater vehicle. Defense Technology Agency, New Zealand Defense Force, DTA Report 254
28. Yang C (2007) Modular modeling and control for autonomous underwater vehicle (AUV). Thesis of master of engineering department of mechanical engineering national university of Singapore
29. Lin FC (2003) Adaptive fuzzy logic-based velocity observer for servo motor drives. Mechatronics 13:229–241
30. Subudhi B, Mukherjee K, Ghosh S (2013) A static output feedback control design for path following of autonomous underwater vehicle in vertical plane. Ocean Eng 63:72–76
31. http://www.mathworks.com/help.html