International Journal of Advanced Robotic Systems

OPEN ACCESS ARTICLE

Design of Neural Network Control System for Controlling Trajectory of Autonomous Underwater Vehicles

Regular Paper

İkbal Eski[1,*] and Şahin Yıldırım[2]

1 Department of Mechatronics Engineering, University of Erciyes, Kayseri Turkey
2 Erciyes University, Faculty of Engineering, Mechatronics Engineering Department, Kayseri, Turkey
* Corresponding author E-mail: ikbal@erciyes.edu.tr

Received 19 Mar 2012; Accepted 10 Jun 2013

DOI: 10.5772/56740

Abstract A neural network based robust control system design for the trajectory of Autonomous Underwater Vehicles (AUVs) is presented in this paper. Two types of control structure were used to control prescribed trajectories of an AUV. The vehicle was tested with random disturbances while taxiing under water. The results of the simulation showed that the proposed neural network based robust control system has superior performance in adapting to large random disturbances such as underwater flow.
It is proved that this kind of neural predictor could be used in real-time AUV applications.

Keywords Neural Network Control, Robust Control, Autonomous Underwater Vehicles, Trajectory Control

Nomenclature

C	Matrix of Coriolis and centrifugal terms
D	Matrix of the hydrodynamic damping terms
g	Vector of gravity and buoyant forces
I_x	Mass moment of inertia coefficient about body longitudinal axis
I_y	Mass moment of inertia coefficient about body lateral axis
I_z	Mass moment of inertia coefficient about body vertical axis
m	Mass of the underwater vehicle body
M	Matrix of inertia and added inertia
M_δ	Drag moment about lateral body axis due to stern and rudder deflection
M_q	Drag moment about lateral body axis due to existing pitch q corresponding to laminar flow
$M_{\dot{q}}$	Fluid inertia moment about lateral body axis due to time rate of change of pitch rate ($\dot{q}$)
N_r	Drag moment about vertical body axis due to existing yaw r corresponding to laminar flow
$N_{\dot{r}}$	Fluid inertia moment about vertical body axis due to time rate of change of yaw ($\dot{r}$)
N_v	Fluid inertia moment about vertical body axis due to existing sway v corresponding to laminar flow

N_δ	Drag moment about vertical body axis due to stern and rudder deflection
n_I	Number of neurons in the input layer
n_H	Number of neurons in the hidden layer
n_O	Number of neurons in the output layer
P	Training patterns
RPROP	Resilient back-propagation algorithm
RNF	Robust neural feedback
t	Time
u_0	Constant forward speed of underwater vehicles
AUV	Autonomous underwater vehicle
u(t)	Force of the RNF control system
$u_{NN}(t)$	Force of the neural controller
$u_R(t)$	Force of robust feedback controller
W	Submerged weight of vehicle
w_{ij}	Weights matrices from *i*th layer to *j*th layer
w_{jk}	Weights matrices from *j*th layer to k*th* layer
X Y Z	Body-fixed coordinate system
X_0 Y_0 Z_0	Earth-fixed coordinate system
x_G, y_G, z_G	Position of the centre of gravity in the body-fixed frame
y(t)	System output signal
$y_r(t)$	Desired input signal
$y_{NN}(t)$	Neural network model signal
$Y_{\dot{v}}$	Fluid inertia in the lateral y direction due to time rate of change of sway velocity ($\dot{v}$)
$Y_{\dot{r}}$	Drag contribution in the lateral y direction due to time rate of change of sway velocity ($\dot{v}$)
Y_δ	Sway force due to deflection angle of rudder and stern plane
z	Depth
ϕ	Roll angle of AUV
θ	Pitch angle of AUV
ψ	Yaw angle of AUV
δ_r	Rudder deflection of AUV
δ_s	Stern plane deflection of AUV
η	Vector of position and attitude of the AUV in the inertial frame
τ	Resultant vector of thruster forces and moments
$\Delta_{ij}(t)$	Update value for each weight
α	Weight-decay parameter
λ	Update parameter

1. Introduction

Nowadays, AUVs are widely used for underwater investigations. Son and Kim [1] investigated manoeuvrable control of an underwater vehicle using a combined discrete-event and discrete-time system simulation. The proposed simulation model was established on the basis of discrete-event system specification formalism, representative of a discrete-event system simulation. The proposed approach made it possible to build a simulation-based expert system to support decision-making in the acquisition of an underwater vehicle.

Santhakumar and Asokan analysed dynamic station keeping of an under-actuated flat-fish-type AUV, and proposed a new method of station keeping with the addition of dedicated thrusters [2]. The effect of the additional thrusters on tracking performance was analysed and a modular configuration (using retractable thrusters) was developed. The effects of underwater current magnitudes and angle of incidences on the station keeping performance were also investigated. A comparative analysis of power consumption during station keeping proved the effectiveness of the proposed modular configuration.

An adaptive neuro-fuzzy sliding-mode-based genetic algorithm control system for a remotely operated vehicle with four degrees of freedom was presented by Moghaddam and Bagheri [3]. A set-point controller for autonomous underwater vehicles was proposed by Herman [4]. The controller was expressed in transformed equations of motion with a diagonal inertia matrix. The stability of the proposed control law was proven and the performance of the developed controller was verified via simulation on the underwater vehicle.

Kumar et al. presented a new control scheme for robust trajectory control of underwater vehicles. The effectiveness of the controller was verified through simulations and execution issues were discussed. Adaptive control of low-speed bio-robotic autonomous underwater vehicles in the dive plane using dorsal fins was also considered [5].

Narasimhan and Singh developed an indirect adaptive control system for depth control using dorsal fins. According to the simulation results, the adaptive control system accomplished precise depth control of the bio-robotic autonomous underwater vehicle using dorsal fins, in spite of large uncertainties in the system parameters [6]. Autonomous underwater vehicle control architectures were reviewed and sensor data bus based control architecture was investigated by Kim and Yuh [7].

A wave drift force that severely affects the underwater vehicle in shallow water was examined by Luo et al. [8]. On the basis of wave force analysis, three-dimensional disturbances caused by wavy surge water were measured, and a control system was proposed using least-squares multi-order data fitting polynomial prediction and fuzzy compensation, combined with a PID controller. The experimental results showed that the control system for disturbance of surge and wave was feasible and effective.

A chattering-free sliding-mode controller was developed by Soylu et al. for the trajectory control of remotely operated vehicles. A new approach for thrust allocation was also proposed based on minimizing the largest individual component of the thrust manifold [9]. Bessa et al. [10] developed an adaptive fuzzy sliding mode controller for remotely operated underwater vehicles. Their study was carried out based on the sliding mode control strategy and enhanced by an adaptive fuzzy algorithm for uncertainty/disturbance compensation. The performance of the proposed control structure was also appraised using numerical simulations.

Naik and Singh [11] investigated the problem of suboptimal dive plane control of autonomous underwater vehicles using the state-dependent Riccati equation technique. The simulation results showed that effective depth control was accomplished in spite of the uncertainties in the system parameters and control fin deflection constraints.

In other research, a neuro-fuzzy controller for autonomous underwater vehicles was proposed by Kim and Yuh [12]. Simulation results showed effectiveness of the neuro-fuzzy controller for autonomous underwater vehicles. Akkizidis et al. [13] used a fuzzy-like PD controller for an underwater vehicle, and analysed and presented experimental results. A switched control law for stabilizing an under-actuated underwater vehicle was proposed by Sankaranarayanan et al. [14], and simulation results were presented to validate this. Lapierre [15] designed and verified a diving-control method based on Lyapunov theory and back-stepping techniques. The results of the control system and subsequent simulations demonstrated the performance of the solutions proposed.

The organization of the present paper is as follows. The following section describes some of the theory of AUVs. A proposed robust neural feedback control system is outlined in Section 3. The simulation results are given in Section 4. Finally, conclusions are presented.

2. Dynamics equations of autonomous underwater vehicle

The motion for an underwater vehicle's generalized six-degree of freedom equations is derived under the following assumptions:

- The vehicle behaves as a rigid body;
- The earth's rotation is negligible as far as acceleration of the centre of mass is concerned;
- The vehicle moves at low speed;
- The hydrodynamics parameters are constant;

The equations of motion for an underwater vehicle can be represented as in [16, 17]:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau \tag{1}$$

where $\boldsymbol{M}$ is the matrix of inertia and added inertia, $\boldsymbol{C}$ is the matrix of Coriolis and centrifugal terms, $\boldsymbol{D}$ is the matrix of the hydrodynamic damping terms, $g(\eta)$ is the vector of gravity and buoyant forces, and τ is the resultant vector of thruster forces and moments.

$$\dot{\eta} = J(\eta)\nu \tag{2}$$

where $J(\eta)$ is the kinematics transformation matrix and $\eta = (x, y, z, \varphi, \theta, \psi)^T$. The linear steering equations of motion are:

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} v \\ r \\ \phi \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \end{bmatrix} \delta_r \tag{3}$$

where the components of the matrix are as follows:

$$a_{11} = \frac{u_0\left[Y_v(m - Y_{\dot{v}})^2(I_{zz} - N_{\dot{r}}) - (mx_G - Y_{\dot{r}})N_v\right]}{(m - Y_{\dot{v}})(I_{zz} - N_{\dot{r}})}$$

$$a_{12} = \frac{u_0[(m - Y_{\dot{v}})^2(Y_r - m)(I_{zz} - N_{\dot{r}}) - (mx_G - Y_{\dot{r}})(N_r - mx_G)]}{(m - Y_{\dot{v}})(I_{zz} - N_{\dot{r}})}$$

$$a_{13} = 0$$

$$a_{21} = \frac{u_0\left[N_v(m - Y_{\dot{v}})(I_{zz} - N_{\dot{r}})^2 - (mx_G - N_{\dot{v}})Y_v\right]}{(m - Y_{\dot{v}})(I_{zz} - N_{\dot{r}})}$$

$$a_{22} = \frac{(I_{zz} - N_{\dot{r}})^2(N_r - mx_G)(m - Y_{\dot{v}}) - (mx_G - N_{\dot{v}})(Y_r - m)u_0}{(m - Y_{\dot{v}})(I_{zz} - N_{\dot{r}})}$$

$$a_{23} = 0$$

$$a_{31} = 0$$

$$a_{32} = 1 - \frac{(mx_G - N_{\dot{v}})(mx_G - Y_{\dot{r}})}{(m - Y_{\dot{v}})(I_{zz} - N_{\dot{r}})}$$

$$a_{33} = 0$$

$$b_{11} = \frac{(I_{zz} - N_{\dot{r}})(m - Y_{\dot{v}})^2 Y_\delta - (mx_G - Y_{\dot{r}})N_\delta}{(m - Y_{\dot{v}})(I_{zz} - N_{\dot{r}})}$$

$$b_{12} = \frac{(I_{zz} - N_{\dot{r}})^2(m - Y_{\dot{v}})N_\delta - (mx_G - N_{\dot{v}})Y_\delta}{(m - Y_{\dot{v}})(I_{zz} - N_{\dot{r}})}$$

$$b_{13} = 0$$

The linearized forms for equations of the AUV motion containing heave and pitch are as follows:

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ 1 & 0 & 0 \\ 0 & a_{32} & 0 \end{bmatrix} \begin{bmatrix} q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} b_{11} \\ 0 \\ 0 \end{bmatrix} \delta_s \qquad (4)$$

where the components of the matrix are as follows:

$$a_{11} = \frac{M_q - mx_G u_0}{I_{yy} - M_{\dot{q}}}$$

$$a_{12} = \frac{-W(z_G - z_B)}{I_{yy} - M_{\dot{q}}}$$

$$a_{32} = -u_0$$

$$b_{11} = \frac{M_\delta}{I_{yy} - M_{\dot{q}}}$$

Schematic representation of the AUV system with coordinates is shown in Figure 1. The hydrodynamics parameters and the AUV parameters are given in Table 1.

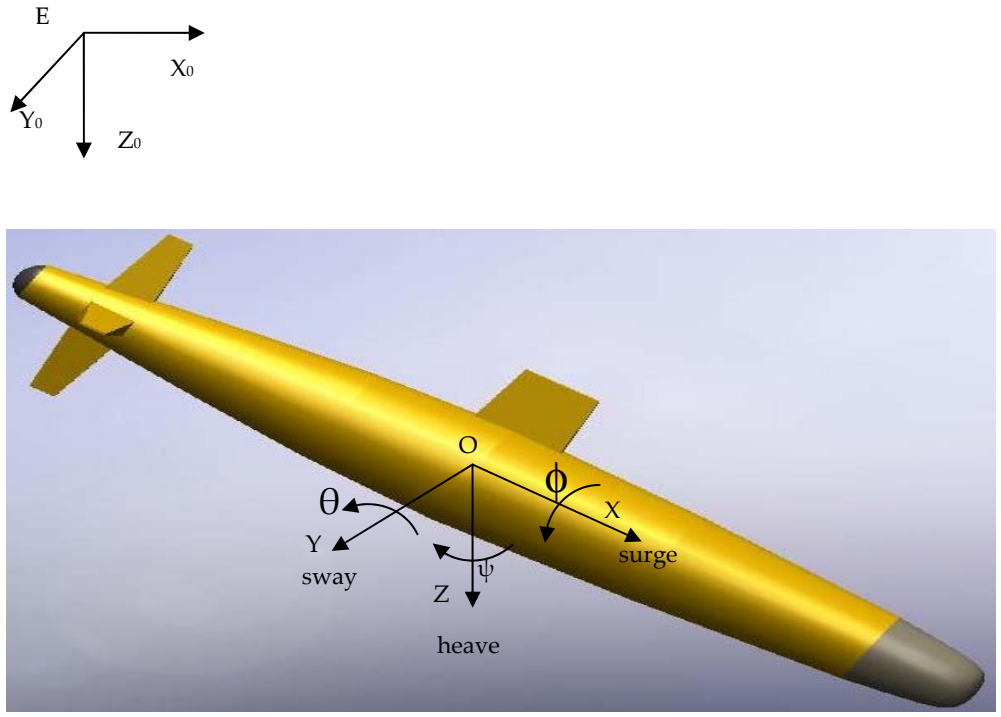

Figure 1. Schematic representation of the AUV system with coordinates

m	250 kg	**Mq**	300 N.m
g	9.81 m/s^2	$\mathbf{M_{\dot{q}}}$	-30 kgm^2
$\mathbf{u_0}$	2 m/s	**Nv**	300 N.m
$\mathbf{x_G}$	-0.15 m	$\mathbf{N_{\dot{v}}}$	10 kg.m
$\mathbf{z_G}$	0.03 m	$\mathbf{Y_\delta}$	8047 N
$\mathbf{z_B}$	0	$\mathbf{N_\delta}$	-76 N.m
Izz	140 kg.m^2	**Yr**	100
Iyy	150 kg.m^2	$\mathbf{Y_{\dot{r}}}$	10 kg.m
Nr	300 N.m	**Yv**	100 N
$\mathbf{N_{\dot{r}}}$	-30 kg.m^2	$\mathbf{Y_{\dot{v}}}$	-250 kg

Table 1. The hydrodynamics parameters and the AUV parameters

3. Robust Neural Feedback Control System (RNFCS)

A designed control system is employed to control the trajectory of the AUV. The purpose of this proposed control system is to provide the appropriate control action. The mathematical expression of the force of the RNF control system is given by:

$$u(t) = u_R(t) + u_{NN}(t) \quad (5)$$

where $u_R(t)$ is the force of the robust controller and $u_{NN}(t)$ is the force of the neural controller. The sum of these two forces gives the control force signal $u(t)$. The first part of the control input for the robust controller can be described as follows:

$$u_R(t) = \left(K_P e(t) + K_D \frac{de(t)}{dt} \right) * e^{(-Rt)} \quad (6)$$

where K_P, K_D and R are the proposed control system parameters and are empirically set to K_P = 10, K_D=7 and R = 0.0001. In the following equation, e(t) is the control error:

$$e(t) = y_r(t) - y(t) \quad (7)$$

where $y_r(t)$ is the reference input signal and $y(t)$ is the system output signal. Neural network structure is shown in Figure 2. The second part of the control input for the proposed control system is explained in the following subsection.

3.1 Neural controller

A neural controller with Resilient Back-propagation Algorithm is one popular neural network structure for control and prediction. Fundamentally, two steps are involved when using this control: system identification and control design. The identification stage of this control is to train a neural network to present the forward dynamics of the plant. The neural network model of the plant that needs to be controlled is developed using two sub-networks for the model approximation. The neural model is as follows:

$$y(t+d) = N[y(t),..., y(t-m+1), u(t-1),..., u(t-n+1)] \quad (8)$$

where $y(t)$ is the system output, $u(t)$ is the system input and d is the relative degree (d≥2). Multilayer neural networks can be used to identify the function F. The identification model has the form:

$$\hat{y}(t+d) = f[y(t),..., y(t-m+1), u(t-1),..., u(t-n+1)] + g[y(t),..., y(t-m+1), u(t-1),..., u(t-n+1)].u_{NN}(t) \quad (9)$$

where $\hat{y}(t+d)$ is the estimate of $y(t+d)$. Identification is carried out at every instant t by adjusting the parameters of the neural network using the error $e(t) = \hat{y}(t) - y(t)$. For a system output, y(t+d) is used, and for a reference trajectory y_r(t+d).

$$y(t+d) = f[y(t),..., y(t-m+1), u(t-1),..., u(t-n+1)] + g[y(t),..., y(t-m+1), u(t-1),..., u(t-n+1)].u_{NN}(t) \quad (10)$$

f and g are activation functions of the hidden layer in the first and second sub-networks, respectively, as follows:

$$f(t) = g(t) = \frac{1}{e^{-t}+1} \quad (11)$$

For each sub-network, the linear activation function uses the output layer. The controller output will have the form:

$$u(t) = \frac{y_r(t+d) - f[y(t),..., y(t-m+1), u(t-1),..., u(t-n+1)]}{g[y(t),..., y(t-m+1), u(t-1),..., u(t-n+1)]} \quad (12)$$

Using the equation directly causes a realization problem based on the output at the same time, y(t). So, instead:

$$y(t+d) = f[y(t),..., y(t-m+1), u(t-1),..., u(t-n+1)] + g[y(t),..., y(t-m+1), u(t-1),..., u(t-n+1)].u_{NN}(t+1) \quad (13)$$

Using Eq. (13):

$$u(t+1) = \frac{y_r(t+d) - f[y(t),..., y(t-m+1), u(t-1),..., u(t-n+1)]}{g[y(t),..., y(t-m+1), u(t-1),..., u(t-n+1)]} \quad (14)$$

Figures 3 and 4 represent the neural plant model identification and the neural controller, respectively. The proposed RNF control system architecture is shown in Figure 5. The neural network training parameters are given in Table 2. The Resilient Back-propagation algorithm is used to adjust the weights of the neural network.

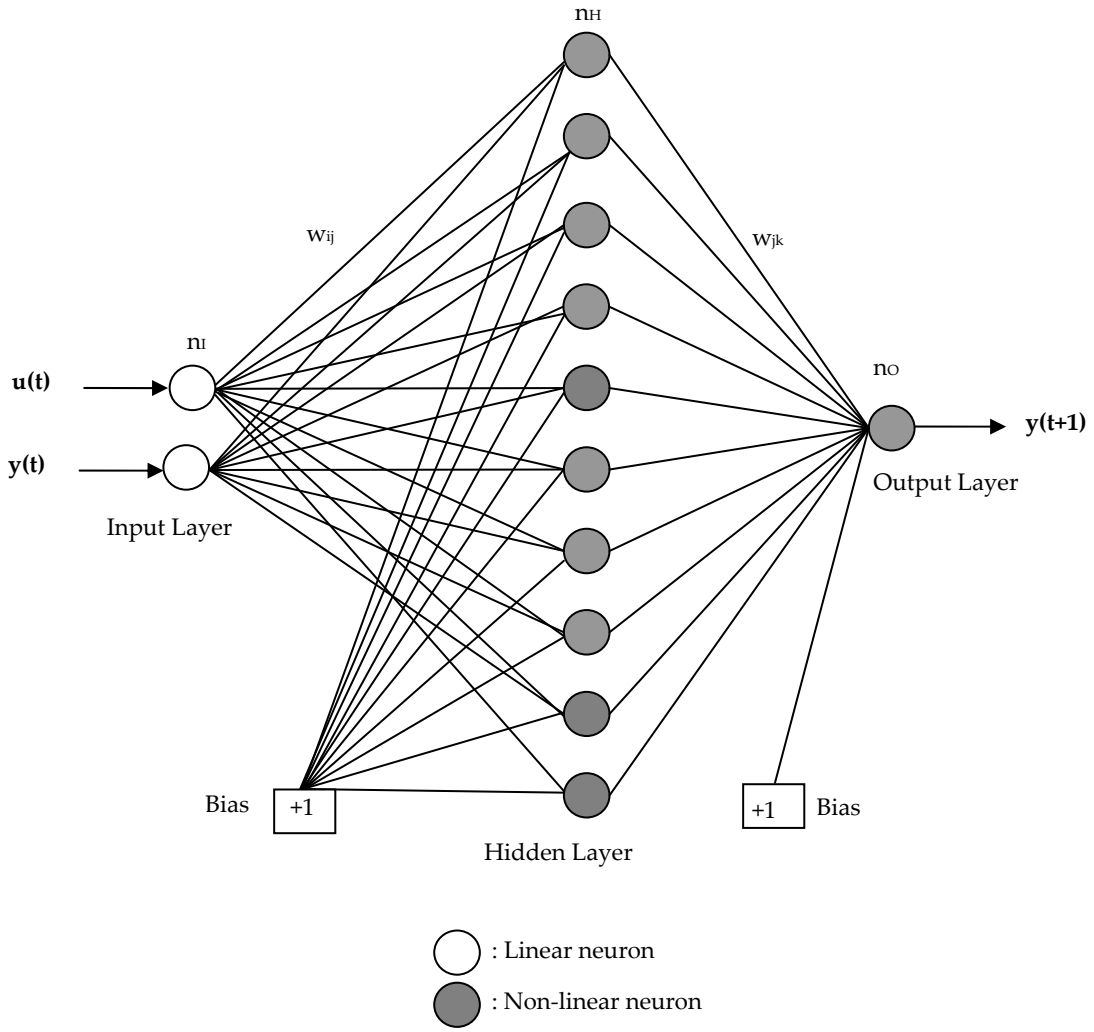

Figure 2. Schematic representation of the neural network model

Learning parameter η	0.5	**Hidden layer neurons** n_H	10
Momentum term α	0.3	**Output layer neurons** n_O	1
Iteration number N	5500000	**Activation Function** g(.)	Sigmoid
Input layer neurons n_I	2	**Errors** RMSEs	0.03

Table 2. The neural network training parameters

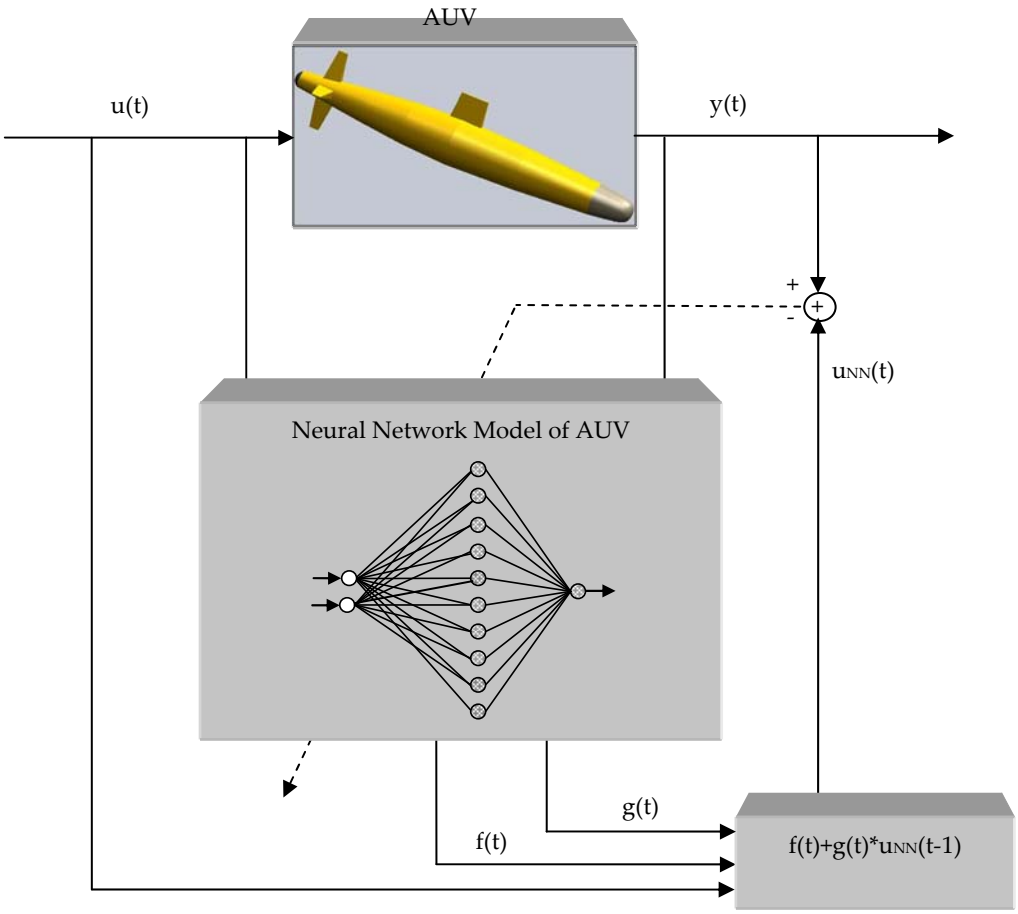

Figure 3. Neural controller plant model identification

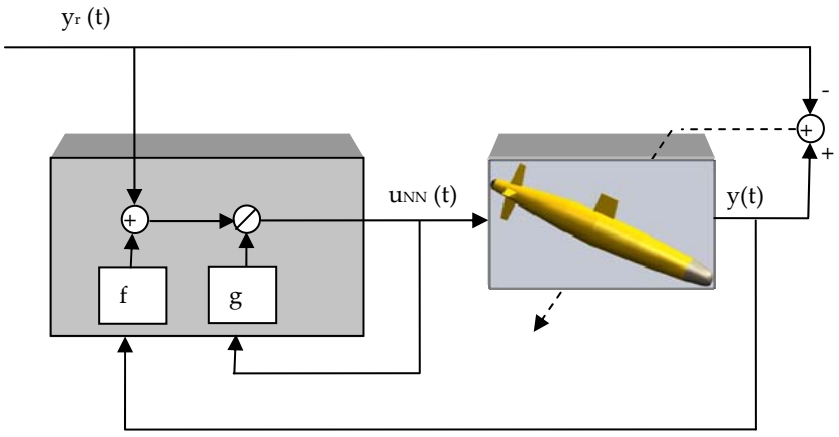

Figure 4. Neural controller

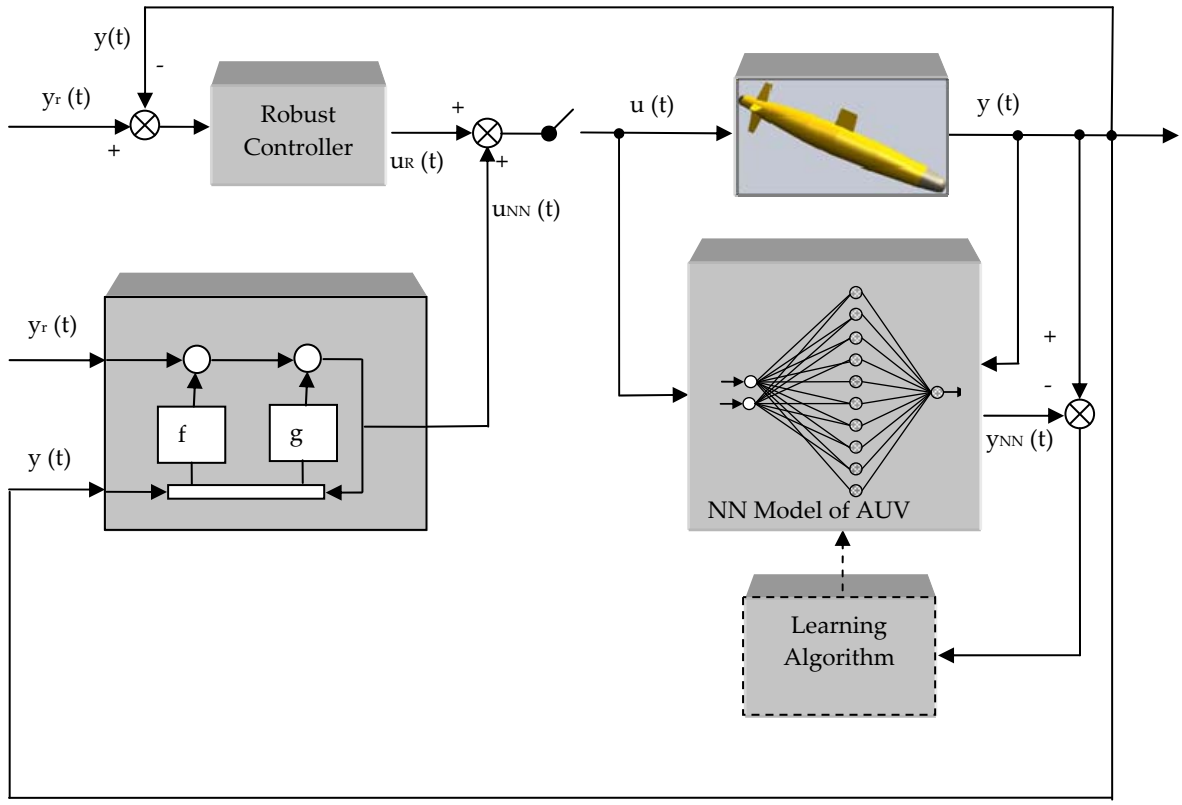

Figure 5. Proposed RNF control system architecture

3.1.1 Resilient Back-propagation Algorithm (RPROP)

The Resilient Back-propagation Algorithm is a local adaptive learning scheme performing supervised batch learning in feed-forward neural networks. The basic principle of this algorithm is to eliminate the harmful influence of the partial derivative's size on the weight step. As a consequence, only the sign of the derivative is considered to indicate the direction of the weight update. This algorithm typically uses a sigmoid function in the hidden layers and a linear function in the output layer. Here, w_{ij} is the weight matrix, and $\Delta_{ij}(t)$ is the update value for each weight. A second learning rule is introduced which determines the evolution of the update value $\Delta_{ij}(t)$. This estimation is based on the observed behaviour of the partial derivative during two successive weight-steps:

$$\Delta_{ij}(t)=\begin{cases}\lambda^{+}\Delta_{ij}(t-1), & \text{if } \frac{\partial E}{\partial w_{ij}}(t)\frac{\partial E}{\partial w_{ij}}(t-1)>0\\ \lambda^{-}\Delta_{ij}(t-1), & \text{if } \frac{\partial E}{\partial w_{ij}}(t)\frac{\partial E}{\partial w_{ij}}(t-1)<0\\ \Delta_{ij}(t-1), & \text{else}\end{cases} \quad (15)$$

(16)

where $0<\lambda^{-}<1<\lambda^{+}$.

The adaptation rule works as follows. Every time the partial derivative of the corresponding weight w_{ij} changes its sign, which indicates that the last update was too big and the algorithm has jumped over a local minimum, the update value $\Delta_{ij}(t)$ is decreased by the factor λ^{-}. If the derivative retains its sign, the update value is slightly increased in order to accelerate convergence in shallow regions. Once the update value for each weight is adapted, the weight-update itself follows a very simple rule: if the derivative is positive (increasing error), the weight is decreased by its update-value; if the derivative is negative, the update value is added. This is formulated as follows:

$$\Delta w_{ij}(t)=\begin{cases}-\Delta_{ij}(t), & \text{if } \frac{\partial E}{\partial w_{ij}}(t)>0\\ \Delta_{ij}(t), & \text{if } \frac{\partial E}{\partial w_{ij}}(t)<0\\ 0, & \text{else}\end{cases} \quad (17)$$

$$w_{ij}(t+1)=w_{ij}(t)+\Delta w_{ij}(t). \quad (18)$$

	K_P	K_I	K_D
Rudder deflection	13	4.8	8
Yaw angle	1.2	0.5	0.75
Theta angle	5.4	4.3	1.68
Depth change	36	32	9.9

Table 3. The PID gain parameters with Zeigler-Nichols method

However, there is one exception. If the partial derivative changes sign, i.e., the previous step is too large and the minimum is missed, the previous weight-update is reverted to:

$$\Delta w_{ij}(t) = -\Delta w_{ij}(t-1), \quad \text{if} \quad \frac{\partial E}{\partial w_{ij}}(t) \cdot \frac{\partial E}{\partial w_{ij}}(t-1) < 0 \tag{19}$$

Due to this 'backtracking' weight-step, the derivative is supposed to change its sign once again in the following step. In order to avoid a double punishment of the update value, there should be no adaptation of the update value in the succeeding step. In practice, this can be achieved by setting $\frac{\partial E}{\partial w_{ij}}(t-1) = 0$ in the Δ_{ij} update rule above. The partial derivative of the total error is given by:

$$\frac{\partial E}{\partial w_{ij}}(t) = \frac{1}{2}\sum_{p=1}^{P}\frac{\partial E_p}{\partial w_{ij}}(t) \tag{20}$$

Hence, the partial derivatives of the errors must be accumulated for all *P* training patterns. This means that the weights are updated only after the presentation of all training patterns. The α (weight-decay) parameter determines the relationship of two goals, namely to reduce the output error (the standard goal) and to reduce the size of weights (to improve generalization). The composite error function is:

$$E = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{N_O}(d_{pj} - e_{pj})^2 + \frac{1}{10^{\alpha}}\sum_{i,j} w_{ij}^2 \tag{21}$$

For comparison purposes, the classical PID controller was used for trajectory control of the AUV. The PID controller was initially tuned using the Ziegler-Nichols method; the PID parameters are given in Table 3.

4. Simulation results

This section presents simulation results of the AUV system with neural network based controller for rudder deflection, yaw angle, theta angle and depth change. Figures 6(a)-(c) show the responses of these parameters without any controller and in the presence of a PID controller and the proposed RNF control system, respectively.

Figure 6(a) shows the rudder deflection of the AUV for sinusoidal input signal. The response of the AUV unstable behaviour (shown with dashed lines) is also seen in Figure 6(a). Figure 6(b) depicts the response of rudder deflection of the AUV for sinusoidal input signal using the standard PID controller. As seen in the Figure , the system response does not show unstable behaviour, but the desired sinusoidal input signal does not follow. The results show that the proposed RNF control system has better performance in terms of adapting a sinusoidal input signal. Figure 7(a) shows rudder deflection of the AUV for a random input signal; in Figure 6(a) the response of the AUV is unstable behaviour. The result of the PID controller for rudder deflection of the AUV is shown in Figure 7(b). As seen in Figure 7(b), the rudder deflection response of the AUV with the PID controller does not track the desired random input signal. Figure 7(c) shows the result of the RNF control system. The graph shows a small overshoot error between the desired random input signal and the proposed control system.

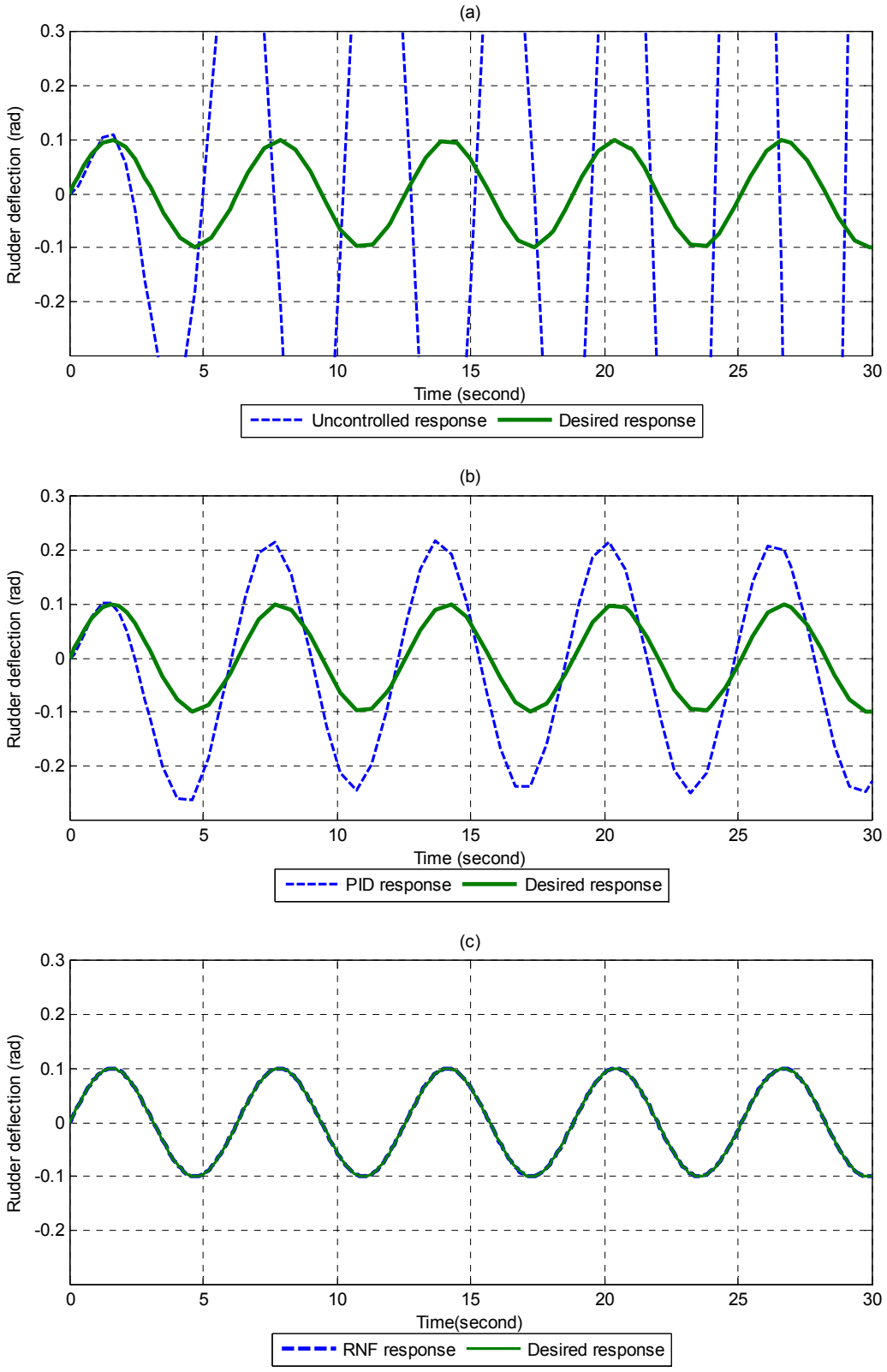

Figure 6. Rudder deflection of AUV for sinusoidal input signal (a) Uncontrolled response (b) PID controller response, and (c) RNF control system response

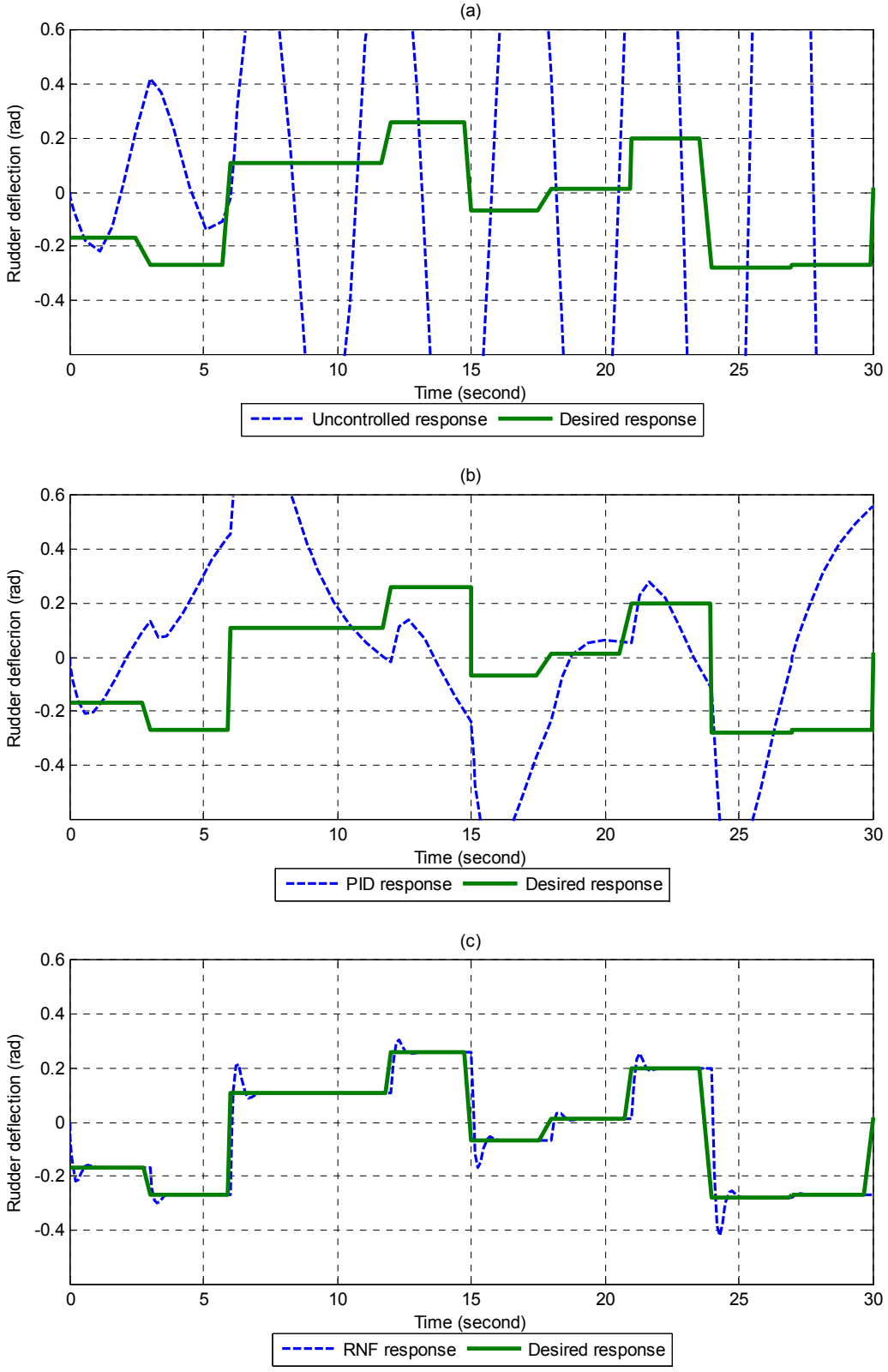

Figure 7. Rudder deflection of AUV for random input signal (a) Uncontrolled response (b) PID controller response, and (c) RNF control system response

Figures 8(a), (b) and (c) indicate the yaw angle result without any controller and with the PID controller and the proposed RNF control system for a sinusoidal input signal. As seen in the Figure s, the yaw angle response of the AUV does not show unstable behaviour, whereas the proposed RNF control system exactly follows the desired sinusoidal input signal.

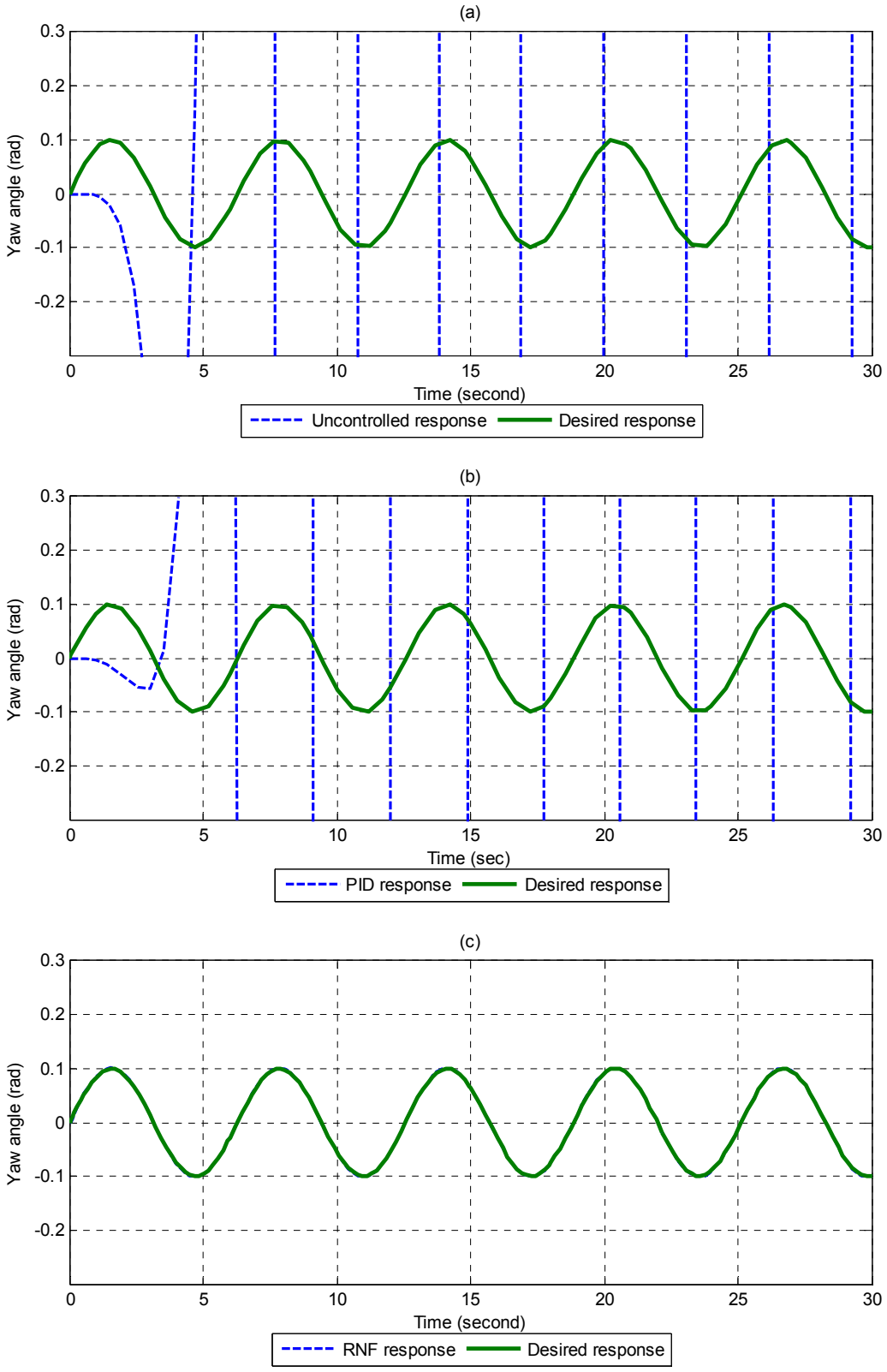

Figure 8. Yaw angle of AUV for sinusoidal input signal (a) Uncontrolled response (b) PID controller response, and (c) RNF control system response

Figures 9(a), (b) and (c) show the yaw angle without any controller, and with the standard PID controller and the proposed RNF control system in the case of a random input signal. As depicted in Figure 9(c), some differences can be seen between the proposed control system response and the desired random input signal.

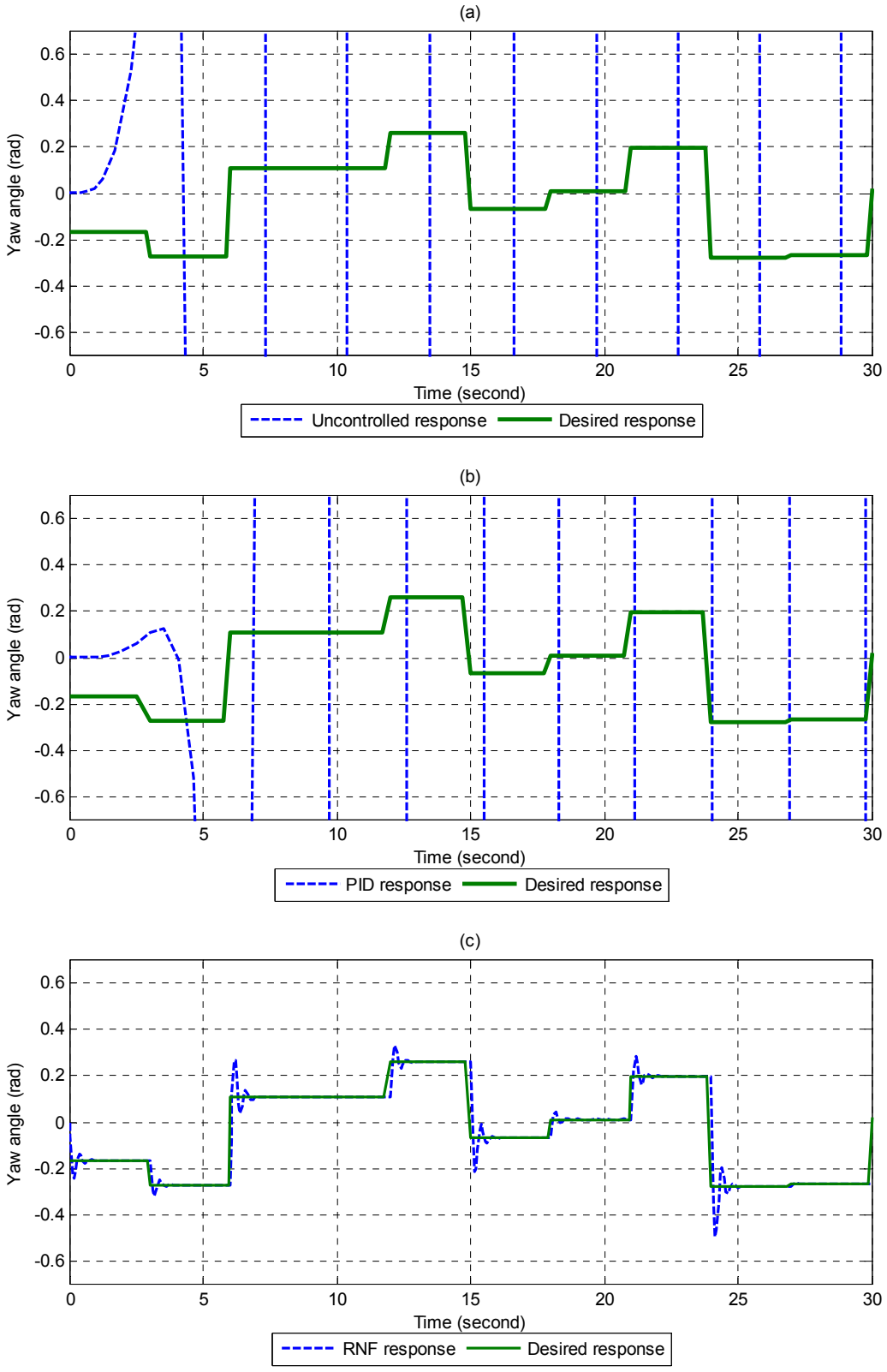

Figure 9. Yaw angle of AUV for random input signal (a) Uncontrolled response (b) PID controller response, and (c) RNF control system response

The response of the theta angle of the AUV for a sinusoidal input signal is given in Figures 10(a), (b) and (c). As shown in the Figure s, the proposed control system has the best performance in terms of adapting to the sinusoidal input signal. Figures 11(a), (b) and (c) present the response of the theta angle for a random input signal. The results prove that the proposed RNF control system is suitable for controlling the theta angle.

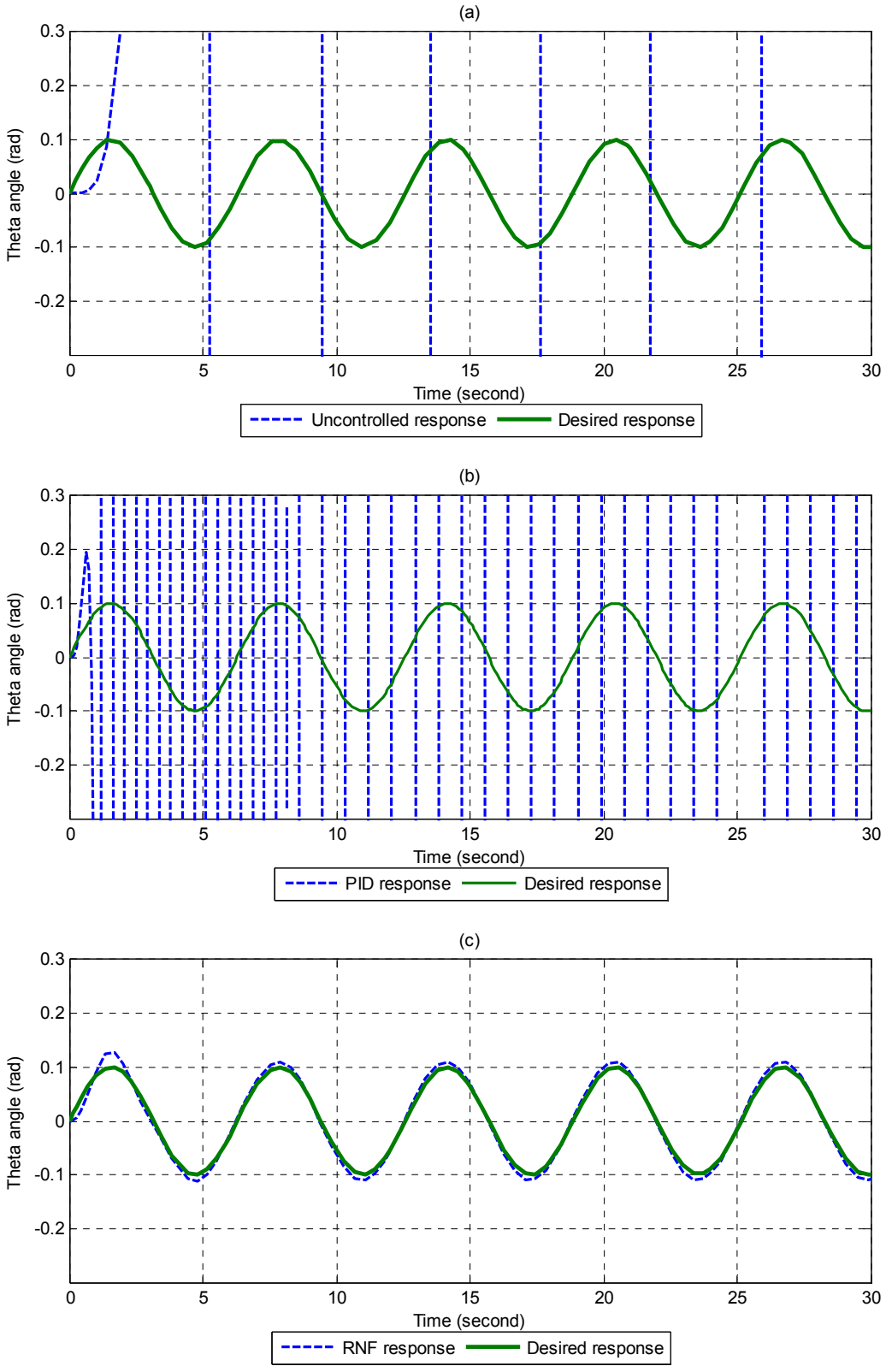

Figure 10. Theta angle of AUV for sinusoidal input signal (a) Uncontrolled response (b) PID controller response, and (c) RNF control system response

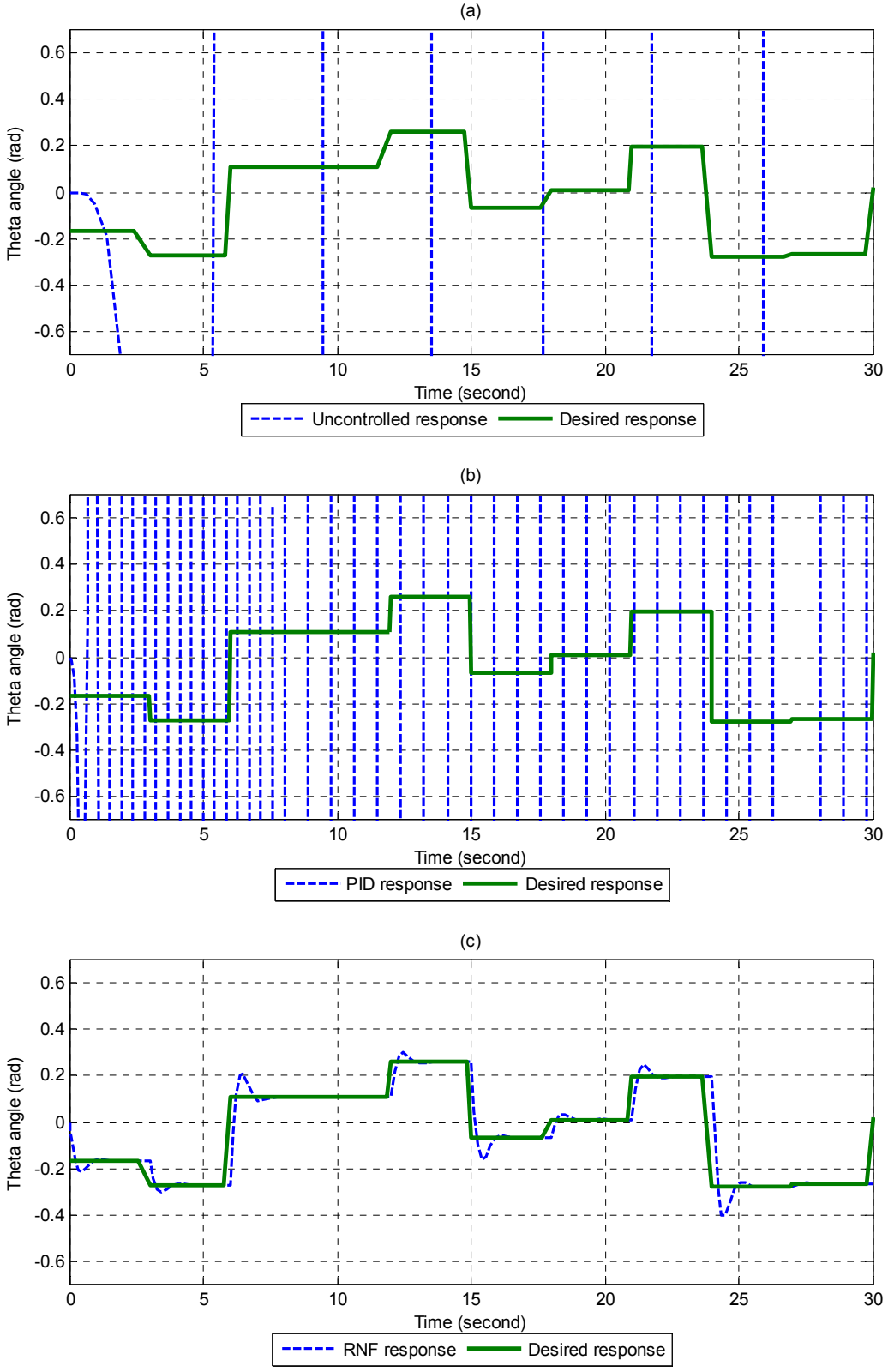

Figure 11. Theta angle of AUV for random input signal (a) Uncontrolled response (b) PID controller response, and (c) RNF control system response

Finally, Figures 12(a), (b) and (c) show the results of the AUV's depth change. According to simulation results, the proposed control system has excellent performance for controlling the AUV parameters.

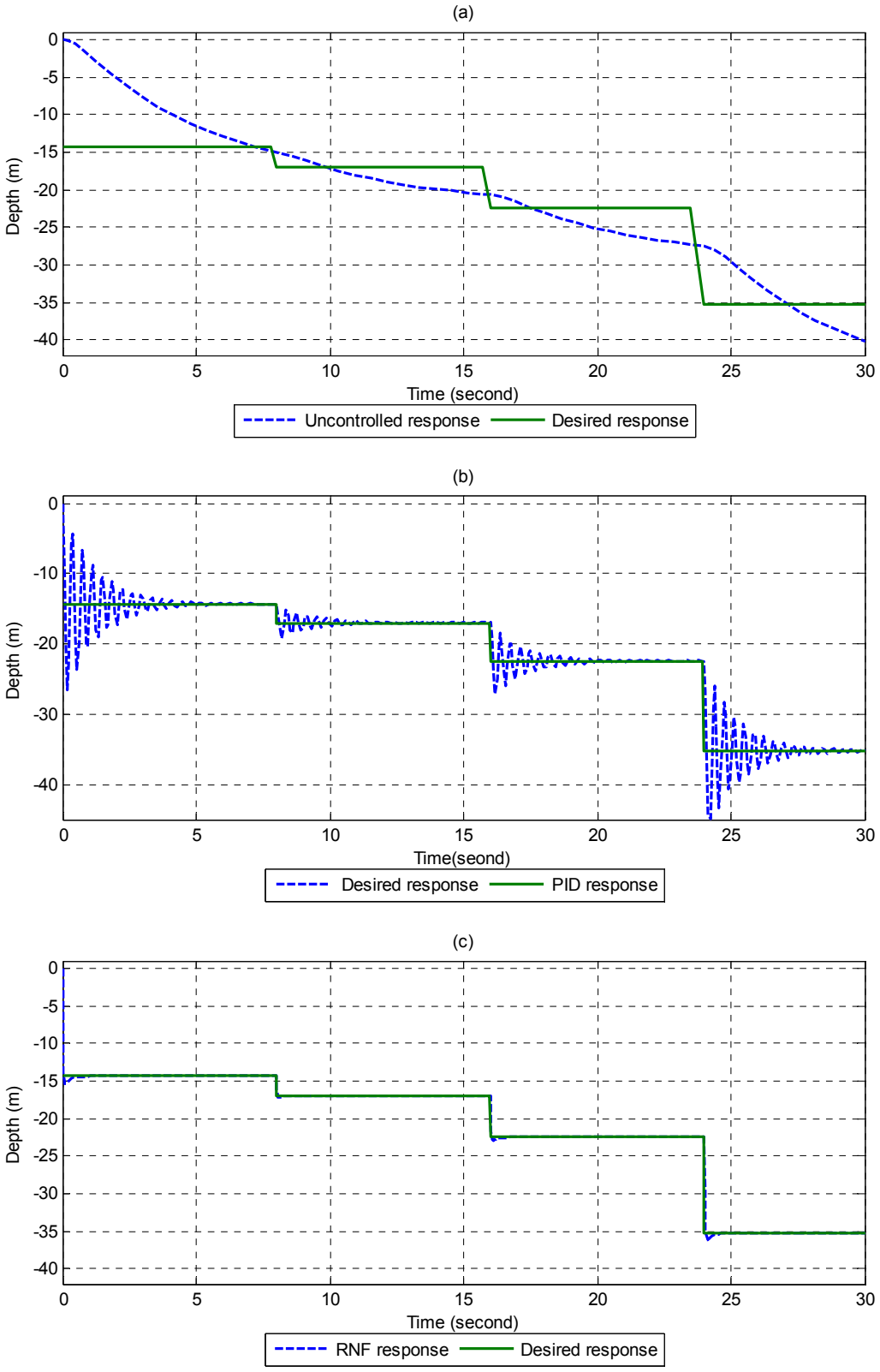

Figure 12. Depth change of AUV (a) Uncontrolled response (b) PID controller response, and (c) RNF control system response

5. Conclusions

A robust control system with a neural network was designed for trajectory controlling of an AUV system. Four parameters of the system were analysed and controlled by the proposed control system structure.

For comparison, the standard PID controller, tuned using the Ziegler-Nichols method, was also employed to control the AUV. The results of both control systems showed that the use of the proposed robust neural feedback control system was effective in controlling the AUV and more robust than the PID controller. The strong performance of the proposed RNF control system was due to the inclusion of both linear and non-linear neurons in the network. As shown by the simulation results, the proposed control system can effectively track a given trajectory for experimental applications.

6. References

[1] Son M.J., Kim T., Maneuvering control simulation of underwater vehicle based on combined discrete-event and discrete-time modeling, Expert Systems with Applications 39 (2012) 12992–13008.

[2] Santhakumar M., Asokan T., Power efficient dynamic station keeping control of a flat-fish type autonomous underwater vehicle through design modifications of thruster configuration, Ocean Engineering 58 (2012) 11–21.

[3] Moghaddam J., Bagheri A., An adaptive neuro-fuzzy sliding mode based genetic algorithm control system for under water remotely operated vehicle, Expert Systems with Applications 37 (2010) 647–660.

[4] Herman P., Decoupled PD set-point controller for underwater vehicles, Ocean Engineering 36 (2009) 529–534.

[5] Kumar R.P., Dasgupta A., Kumar C.S., Robust trajectory control of underwater vehicles using time delay control law, Ocean Engineering 34 (2007) 842–849.

[6] Narasimhan M., Singh S.N., Adaptive optimal control of an autonomous underwater vehicle in the dive plane using dorsal fins, Ocean Engineering 33 (2006) 404–416.

[7] Kim T.W., Yuh J., Development of real-time control architecture for a semiautonomous underwater vehicle for intervention missions, Control Engineering Practice 12 (2004) 1521–1530.

[8] Luo J., Tang Z., Peng Y., Xie. S., Cheng T., Li H., Anti-disturbance control for an underwater vehicle in shallow wavy water, Advanced in Control Engineering and Information Science, Procedia Engineering 15 (2011) 915–921.

[9] Soylu S., Buckham B.J., Podhorodeski R.P., A chattering-free sliding-mode controller for underwater vehicles with fault tolerant infinity-norm thrust allocation, Ocean Engineering 35 (2008) 1647–1659.

[10] Bessa W.M., Dutra M.S., Kreuzer E., An adaptive fuzzy sliding mode controller for remotely operated underwater vehicles, Robotics and Autonomous Systems 58 (2010) 16–26.

[11] Naik M.S., Singh S. N., State-dependent Riccati equation-based robust dive plane control of AUV with control constraints, Ocean Engineering 334 (2007) 1711–1723.

[12] Kim T.W., Yuh J., Application of on-line neuro-fuzzy controller to AUVs, Information Sciences, 145 (2002) 169–182.

[13] Akkizidis I.S., Roberts G.N., Ridao P., Batlle J., Designing a fuzzy-like PD controller for an underwater robot, Control Engineering Practice 11 (2003) 471–480.

[14] Sankaranarayanan V., Mahindrakar A.D., Banavar R. N., A switched controller for an under actuated underwater vehicle, Communication in Nonlinear Science and Numerical Simulation 13 (2008) 2266–2278.

[15] Lapierre L., Robust diving control of an AUV, Ocean Engineering 36 (2009) 92–104.

[16] Fossen T.I., Guidance and control of ocean vehicles, 1994, John Wiley, Chichester.

[17] Aranda J., Armada M.A., Cruz J.M., Automation for the maritime industries, 2004 PGM, Spain.