Adaptive Neural Network Control of an Underwater Remotely Operated Vehicle (ROV)

A. Bagheri, N. Amanifard, T. Karimi, M. H. Farahani, S. M. Besarati
Dept. of Mechanical Engineering
Guilan University
P.O. Box 3756, Rasht, Iran

Abstract: An adaptive neural network (NN) controller is developed for an underwater remotely operated vehicle (ROV). Radial basis neural network and multilayer neural network are used in the closed-loop to approximate the nonlinear dynamics of the vehicle. The low cost ROV was designed and fabricated by department Mechanical Engineering of Guilan University, is considered. This technique does not require the iterative off-line training process to identify the plant parameters. The stability of the presented control systems are guaranteed on the basis of the Lyapunov theory. Finally the performances of the vehicle with NN controllers are compared with a PD controller. The significant improvements are observed in tracking performance of the ROV in all controllable degrees of freedom.

Keywords: Remotely Operated Vehicle, Underwater Exploration, Adaptive Controller, Neural Network.

1 Introduction

Underwater vehicles have the increasingly applications in exploring the underwater environment. These applications are involved to good maneuvering capabilities and a high precision track-keeping along the specified path. However controlling the position of the underwater vehicle with suitable performance is so difficult due to the strong nonlinearity of the dynamic behavior and uncertainties. There has been a great deal of research in the area of control of the underwater vehicles. Several different control approaches have been studied for underwater vehicles including sliding mode control [1], robust control [2], adaptive control [3], fuzzy control [4], neural networks [5] and etc. The results of adaptive control approaches have been shown an acceptable tracking error and parameter adaptation. However, all conventional adaptive control techniques assume that the unknown dynamic model is presented as a linear in the unknown parameters and a regression matrix specific for the ROVs[3].

Neural network controllers have important features that overcome the typical difficulties in designing the control systems for underwater vehicles. Yuh proposed an on-line approach using direct learning technique and showed good control performance in the present of unpredictable changes in the dynamic of the ROV [5]. Ishii and Ura proposed an adaptive neural network controller system for an underwater vehicle with two parts, a control part and a forward part [6].

The remotely operated vehicle that is designed and fabricated by the department of Mechanical Engineering of Guilan University is considered. The main goal of development of this ROV is underwater exploring and studying fish behavior in the Caspian Sea.

Two adaptive neural network controller are developed for the ROV. The Radial basis neural network (RBFNN) and the multilayer neural network (MLNN) are used to approximate the nonlinear vehicle dynamics. Using this approach, there is no need to the prior off-line training and explicit knowledge of the structure of the plant. The control law and the stable on-line adaptation law are derived using the Lyapunov theory as the procedure is presented by [8] and [9]. Finally, the convergence of the tracking error to zero and the boundedness of signals are guaranteed.

The paper is structured as follows. In section 2 the specification of ROV is described. Dynamic model of ROV is evaluated in section 3. The proposed control approach is presented and simulated in section 4. The concluding remarks and future work are given in section 6.

2 ROV Description

This ROV could be conceived in waters up to 300 meters. With the aim of designing a low cost vehicle, the underwater ROV has been built using low cost materials. To solve the problem of resistance to underwater pressure, the vehicle is equipped with a compressed air cylinder like those to be used by submersible vehicle in diving mission. The vehicle has an umbilical cable used for energy supply, supervision and transmission of control signals. The vehicle is equipped with conventional analogue TV cameras for observing underwater and guiding the vehicle.
This vehicle has 5 thrusters, two for horizontal movements \((T_{x1}, T_{x2})\) and two for vertical movements \((T_{z1}, T_{z2})\). An additional fifth thruster is installed for helping swaying movement (Figure 1). Due to distribution of weight, the vehicle is completely stable. Pitch \((\theta)\) and roll \((\phi)\) angles are insignificant and no need to be controlled. For this reason, the vertical and horizontal movements are totally independent.

### 3 Dynamic Model

The dynamics of underwater vehicles are strongly nonlinear, coupled, time-varying, and uncertain in the parameters. Also, it is difficult to obtain the exact values of hydrodynamic coefficients. On the other hand, the disturbances from currents and waves are very difficult to model.

The dynamic equations of motion of underwater vehicles have been presented by several authors [1,7,10]. In this study, we consider a nonlinear six degree of freedom model based on Fossen outlines [7]. The rigid body underwater vehicle model in the body fixed reference frame can be represented as

\[
M(q) + C(q) \dot{q} + D(q) \dot{q} + g(q) = \tau
\]

\[\dot{x} = J(x) \dot{q}\]

Where

- \(\dot{q} = [u \ v \ w \ p \ q \ r]^T\): velocity vector
- \(\dot{x} = [x \ y \ z \ \phi \ \theta \ \psi]^T\): position and orientation vector
- \(M\): inertial matrix,
- \(C(q)\): Coriolis and centrifugal matrix,
- \(D(q)\): hydrodynamic damping matrix,
- \(g(x)\): gravitational vector,
- \(\tau\): control input forces and moments
- \(J\): transformation matrix

### 4 Application of Neural Network for Control of the Underwater ROV

The equation of motion (1) and (2) can be written in the earth fixed reference frame in terms of position and attitude through the kinematic transformations:

\[
M_x(x)\ddot{x} + C_x(x,\dot{x})\dot{x} + D_x(x,\dot{x})\dot{\dot{x}} + g_x(x) = \tau_x
\]

Where

\[
M_x(x) = J^{-T}(x)MJ^{-1}(x)
\]

\[
C_x(x,\dot{x}) = J^{-T}(x)[C(q)-MJ^{-1}(x)J(x)]J^{-1}(x)
\]

\[
D_x(x,\dot{x}) = J^{-T}(x)DJ^{-1}(x)
\]

\[
g_x(x) = J^{-T}(x)g
\]

\[
\tau_x = J^{-T}(x)\tau
\]

Tracking error \(e(t)\) and filtered tracking error \(r(t)\) define by

\[
e = x - x_d
\]

\[
r = \dot{e} + \lambda e
\]

where \(\lambda\) is a positive constant and the given desired position and orientation of vehicle denoted by \(x_d\) and the time derivative \(\dot{x}_d\), \(\dot{\dot{x}}_d\) are assumed to be bounded and generated slow varying. The dynamic equation (3) can be written in terms of filtered tracking error

\[
M_x(x)r = -C_x(x,\dot{x})r - D_x(x,\dot{x}) + f + \tau_x
\]

where the nonlinear function \(f\) is

\[
f(x) = M_x(x)(\dddot{x}_d + \lambda \dot{e}) + C_x(x,\dot{x})(\ddot{x}_d + \lambda e) + D_x(x,\dot{x})(\dot{x}_d + \lambda e) + g_x(x)
\]

The main problem of the adaptive control techniques is selecting and updating the estimation of \(f(x)\). It could be solved by use of the neural...
networks together with the conventional PD controller.

An adaptive controller is the requirement for the assumption of linearity for the unknown system parameters[3]:

$$ f(x) = \Phi(x) \theta $$

where \( f(x) \) is the nonlinear system function, \( \Phi(x) \) is regression matrix of known system function and \( \theta \) is a vector of unknown parameters.

Funahashi presented that a neural network with one hidden layer and a sufficient number of neurons can approximate any continuous and smooth mathematical function \( f(x) \)[11], so there is a neural network such that:

$$ f(x) = \Phi_{NN} + \epsilon $$

where \( \Phi_{NN} \) is a network output vector and \( \epsilon \) is the functional estimation error.

In this section an adaptive neural network based on multi-layer and radial basis function neural networks are evaluated for controlling the position of a remotely operated vehicle. The structure of adaptive neural network controller is shown by Figure 2.

4.1 Multi-Layer Neural Networks (MLNN)

The multi-layer network is composed of input, hidden, and output layer and the connections between the three layers are multiplied by weights of appropriate dimensions (Figure 3).

The output of a two-layer neural network with sigmoid active function on the hidden layer and linear active function on the output layer can be given by

$$ f(x, W, V) = W \sigma(Vx) $$

The sigmoid function is represented as

$$ \sigma_i(x) = \frac{1}{1 + \exp(-x)} \quad i = 1, \ldots, L $$

Where \( x = [I \ x_{NN1} \ldots x_{NNn}]^T \), \( V \) is the first layer weight matrix and \( W \) is the second layer weight matrix, \( n \) is number of input vector and \( L \) is the number of neurons of hidden layer. The input vector have been augmented by 1 to include the threshold. With considering the close loop equation (11), the input vector of neural network is:

$$ x = [1 \ x \ \dot{x} \ \dot{x}_d \ \ddot{x}_d]^T $$

It is assumed that there are a proper number of neurons in the hidden layer and weight matrix, \( W \) and \( V \), so that the neural network can approximate a nonlinear function with small error.

The procedure of design of neural network controller is based on the approach developed by Lewis et al. and applied to the adaptive controller for robot manipulators [8]. Based on this technique, adopting the multilayer neural network to approximate the nonlinear vehicle dynamics and present the control input as

$$ \tau = \dot{W} \sigma(\dot{V}x) + K_d r $$

Network weights update law is

$$ \dot{W} = F \sigma r^T - F \sigma \dot{V} x r^T - k_s F^T \| \dot{r} \| \dot{W} $$

$$ \dot{V} = G_x (\dot{\sigma}^T W \dot{r} - k_r G \| \dot{r} \| \dot{V} $$
and $G$ are positive definite matrices and $k_w > 0$ is a small design parameter.

The Lyapunov function is chosen as

$$L = \frac{1}{2} r^T M_x r + \frac{1}{2} \text{tr} \left[ \tilde{W} F^{-1} \tilde{W}^T \right] + \frac{1}{2} \text{tr} \left[ \tilde{G}^{-1} \tilde{V}^T \right].$$  \tag{21}$$

And as presented by [2] the boundedness of $r$ and $\dot{r}$ guarantees that $\dot{L}$ goes to zero with $t$, and hence that $r(t)$ vanishes.

## 4.2 Radial Basis Function Neural Networks (RBFNN)

The RBF neural networks consist of two layers: an input non-linear layer and an output linear layer (Figure 4). The output of the first layer for a RBF network is

$$\phi(x_{NN}) = \exp \left( - \frac{1}{2\sigma_i^2} \| x_{NN} - e_i \|^2 \right), \quad i = 1, 2, \ldots, L. \tag{22}$$

Where $x_{NN} \in \mathbb{R}^n$ is input vector and $L$ is the number of hidden neurons. Also parameters $e_i$ and $\sigma_i$ presents center and radius at node $i$ respectively. The output of the linear layer is

$$y_j = f(x) = \sum_{i=1}^{m} w_j \phi(x_{NN}) = W_j^T \phi(x_{NN}), \quad j = 1, 2, \ldots, m. \tag{23}$$

Where $m$ is the number of DOFs of the plant. $\phi = [\phi_1 \ldots \phi_L]$ is the output vector of hidden layer and $W_j = [w_{j1} \ldots w_{jL}]$ is the weights vector of network. It is assumed that there is a certain combination of optimal weights of the network and number of neurons that provide the satisfactory approximation of the nonlinear function.

The Lyapunov function is chosen as

$$L = \frac{1}{2} r^T M_x r + \frac{1}{2} \text{tr} \left[ \tilde{W} \gamma^{-1} \tilde{W}^T \right].$$  \tag{24}$$

Where $\gamma$ is the learning rate matrix of appropriate dimension and $W = \tilde{W} \cdot \tilde{W}$ is estimation error of the network output weights. Differentiating $L$ with respect to time yields

$$L = \frac{1}{2} \left( r^T M_x \dot{r} + r^T M_x \dot{r} + r^T \dot{M}_x r \right) + \frac{1}{2} \text{tr} \left[ \tilde{W} \gamma^{-1} \tilde{W}^T \right]. \tag{25}$$

With considering that $r^T M_x \dot{r} = \dot{r}^T M_x r$, the closed loop system equation (6) and with using $r^T (M_x - 2C_x) r = 0$, we can rewrite (25) as

$$\dot{L} = r^T (M_x \dot{r} + C_x r) + \text{tr} \left[ \tilde{W} \gamma^{-1} \tilde{W}^T \right],$$

$$= -r^T D_x r + r^T \left[ \tau - f(x, \dot{x}, \ddot{x}) \right] + \text{tr} \left[ \tilde{W} \gamma^{-1} \tilde{W}^T \right]. \tag{26}$$

The RBFNN is used as an approximation of the dynamics of the system as

$$f(x, \dot{x}, \ddot{x}, \dddot{x}) = W \phi(x_{NN}) + \epsilon(x_{NN}) \tag{27}$$

Where $\epsilon(x_{NN})$ denotes the approximation error and it is assumed to be bounded as $|\epsilon(x_{NN})| \leq \epsilon_0$ for $x_{NN} \in \Omega$ where $\Omega$ is the domain of approximation. The bound $\epsilon_0$ on the approximation error can be neglected by using enough number of neurons. The control input is chosen as

$$\tau = \tilde{W} \gamma \phi(x_{NN}) + K_x r \tag{28}$$

Now equation (26) becomes
The equation (29) can be shown to be negative semi-definite, which implies the convergence of $r$ to zero [9].

$$L = -r^T (D_s + K_s) r + r^T \dot{\hat{W}} \varphi(x_{NN}) + tr \{ \dot{\hat{W}} \hat{W}^T \}$$

$$\leq -r^T (D_s + K_s) r + tr \{ \dot{\hat{W}} \varphi(x_{NN}) r^T + \gamma^{-1} \dot{\hat{W}}^T \}$$

choosing the adaptive law of the network weight as

$$\dot{\hat{W}}^T = \gamma \varphi(x_{NN}) r^T$$  \hspace{1cm} (30)

4.3 Simulation

The results of simulation for an underwater vehicle model of the ROV are presented in this section. The equation of motion for four DOFs of underwater vehicle was considered, and it was controlled in all four controllable degrees of freedom. The desired paths are obtained from a trajectory planner. The reference model is chosen as considered by [12]. The desired parameters are computed from
\[ x_d^{\prime} + (2A + I)\Omega x_d + (2A + I)^2 x_d \]
\[ + \Omega^3 x_d = \Omega^3 t^n \]  

(31)

Where \( r_n \) is a command input and \( \Omega = \Omega^r \) and \( \Omega^0 \) describing the preferred damping and stiffness of the system.

The unknown current \( \dot{x}_f = 0.8 \text{ m/s} \), \( \dot{y}_f = -0.7 \text{ m/s} \) and \( \dot{z}_f = 0.0 \text{ m/s} \) was injected after 5 seconds.

The results of simulation of MLNN and RBFNN adaptive controllers are shown in Figure 5. It is shown that there is a relatively small increasing in the tracking error of NN Adaptive controllers at \( t = 5s \) when the current disturbances are injected.

It can be seen that using MLNN and RBFNN give fairly good tracking performance in all controllable degrees of freedom. It should be noted that the neural network weights are initialized at zero. The weight tuning algorithms of MLNN and RBFNN require no preliminary offline learning and weight tuning occurs online in real time.

The values of the design parameters of each controller used in the simulation are listed in Table 1. In general, higher update gain gave better tracking performance, but when the gain was too high, oscillatory behavior was observed.

### Table 1 - The PD and NN adaptive controllers’ parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PD</strong></td>
<td></td>
</tr>
<tr>
<td>PD Gain (Kd)</td>
<td>diag{150,...,150}</td>
</tr>
<tr>
<td>PD Gain (Kd)</td>
<td>diag{50,...,50}</td>
</tr>
<tr>
<td>No. Inputs (n)</td>
<td>21</td>
</tr>
<tr>
<td>No. Outputs(m)</td>
<td>4</td>
</tr>
<tr>
<td><strong>MLNN</strong></td>
<td></td>
</tr>
<tr>
<td>No. Hidden Neurons (L)</td>
<td>21</td>
</tr>
<tr>
<td>Adaptive Gain(F)</td>
<td>diag{180,...,180}</td>
</tr>
<tr>
<td>Adaptive Gain(G)</td>
<td>diag{15,...,15}</td>
</tr>
<tr>
<td>No. Inputs (n)</td>
<td>20</td>
</tr>
<tr>
<td>No. Outputs(m)</td>
<td>4</td>
</tr>
<tr>
<td><strong>RBFNN</strong></td>
<td></td>
</tr>
<tr>
<td>No. Hidden Neurons (L)</td>
<td>350</td>
</tr>
<tr>
<td>Adaptive Gain(γ)</td>
<td>diag{10,...,10}</td>
</tr>
</tbody>
</table>

### 5 Conclusion

The application of the neural network control schemes by using MLNN and RBFNN for an underwater vehicle in four degrees of freedom, is developed. The proposed control techniques could achieve improved tracking performance, without explicit prior knowledge of the vehicle dynamics. Finally, the control laws guarantee more robust performance to the unmodeled disturbances than the conventional PD controller. Adaptive neural network controllers could ensure good position tracking which is necessary during the specified underwater ROVs’ maneuvers.

### 6 References


