Improved Fuzzy Neural Modeling Based on Differential Evolution for Underwater Vehicles

O. Hassanein*, Sreenatha G. Anavatti2* and Tapabrata Ray3*
1SEIT, UNSW@ADFA, Canberra, Australia
1o.hassan@adfa.edu.au
2s.anavatti@adfa.edu.au
3t.ray@adfa.edu.au

Abstract— Autonomous Underwater Vehicles (AUVs) have gained importance over the years as specialized tools for performing various underwater missions in military and civilian operations. This study presents the on-line system identification of AUV dynamics to obtain the coupled nonlinear dynamic model of AUV. This proposed model has an input-output relationship based upon neural fuzzy network (NFN) model technique to overcome the uncertain external disturbance and the difficulties of modelling the hydrodynamic forces of the AUVs instead of using the mathematical model with hydrodynamic parameters estimation. Initially the models’ parameters are generated randomly and tuned by Differential Evolution algorithm (DE) with a set of real input-output data. Secondly, the back propagation algorithm based upon the error between the identified model and the actual outputs of the plant are used to adopt the model’s parameters. The proposed NFN model adopts a functional link neural network (FLNN) as the consequent part of the fuzzy rules. Thus, the consequent part of the NFN model is a nonlinear combination of input variables. Simulation results show the superiority of the proposed adaptive neural fuzzy network (ANFN) model in tracking of the behavior of the AUV.

Keywords: AUV dynamic model, fuzzy modeling, adaptive fuzzy system, system identification, neural fuzzy model, FLNN.

1 Introduction

Autonomous Underwater Vehicles (AUVs) have gained importance over the years as specialized tools for performing various underwater missions in military and civilian operations. The autonomous control of underwater vehicles poses serious challenges due to the AUVs’ dynamics. AUVs dynamics are highly nonlinear and time varying and the hydrodynamic coefficients of vehicles are difficult to estimate accurately because of the variations of these coefficients with different navigation conditions and external disturbances.

According to [1] the two most significant technological challenges in AUV’s design are power and autonomy. Power sources limit the mission time of the vehicle and autonomy limits the degree to which an AUV can be left unattended by human operators.

Reference [2] presented a simple model identification method for UUV and applied this method to the underwater robot GARBI. The system identification was aimed at decoupling the different degrees of freedom in low speed vehicles. Least Square techniques were used to estimate the UUV dynamics. Experiments in lab and real underwater environments were carried out. A neural fuzzy system based on modified differential evolution for nonlinear system control is discussed in [3]. The proposed controller adopts a nonlinear combination of input variables to the consequent part of fuzzy rules and uses a differential evolution to optimize the system parameters. This controller is applied to the planetary-train-type inverted pendulum system and the magnetic levitation system in the VisSim solving nonlinear control problems.

The FLNN is a single-layer neural structure capable of forming arbitrarily complex decision regions by generating nonlinear decision boundaries with nonlinear functional expansion. The FLNN [4] was conveniently used for function approximation and pattern classification with faster convergence rate and less computational loading than a multilayer neural network. Recently, genetic fuzzy systems [5,6] have received increasing attention mainly because they combine the approximate reasoning method of fuzzy systems with the learning capabilities of evolutionary algorithms. However, the search is extremely time-consuming, which is one of the basic disadvantages of all genetic algorithms (GAs). Similar to GAs, differential evolution (DE) has emerged as a robust numerical optimization algorithm and has been successfully applied to solve various difficult optimization problems [7, 8]. Basically, DE is fast, easy to use, and not only astonishingly simple, but also performs extremely well in a wide variety of test problems. The basic strategy employs the difference of two randomly selected individuals as the source of random variations for a third individual. However, DE usually explores too many search points before locating the global optimum. In addition,
although DE is particularly simple to work with, having only a few control parameters, the choice of these parameters is often critical for the performance of DE [8, 9].

This study proposes an online system identification of AUV dynamics as a black box which has an input-output relationship instead of using the mathematical model with hydrodynamic parameters to obtain an accurate dynamic model to overcome the uncertainity, nonlinearity and the difficulties of modelling the AUVs. The only information required for generating and tuning the neural fuzzy systems is the input-output data without any prior knowledge of the physical relationship inside the system and it offers a ‘black box’ modelling tool. ADFA AUV has been developed and built in UNSW@ADFA. The hydrodynamic coefficients of this AUV are calculated under certain conditions using CFD method to derive the exact mathematical model. Simulation program is built up to simulate the dynamic behaviour of the AUV based upon the calculated mathematical model of the AUV.

The general dynamic equation describing the mathematical model of the ADFA AUV is provided in Section 2. The design details of identification of AUV using adaptive neural fuzzy network (ANFN) techniques are presented in section 3. The optimization and tuning techniques for the system identification are provided in section 4. Moreover, Numerical simulation results are presented in section 5. Finally, the paper is concluded in Section 7.

2 AUV’s mathematical model

Fig. 1 shows a typical UNSW@ADFA AUV during one of the experiments. One electrical thruster powers the vehicle for forward motion. Two electrical pumps are used for maneuvering in the horizontal plane. In addition, two electrical pumps help the AUV to navigate in the vertical plane. The middle box is used for carrying the sensors, battery and the electronic accessories.

The hydrodynamic forces per unit mass acting on each axis will be denoted by the uppercase letters X, Y and Z. \( u \), \( v \) and \( w \) represent the forward, lateral and vertical velocities along x, y and z axes respectively. Similarly, the hydrodynamic moments on AUV will be denoted by L, M and N acting around x, y and z axis respectively. The angular rates will be denoted by \( p \), \( q \) and \( r \) along x, y and z axes respectively.

Dynamics of AUVs, including hydrodynamic parameters uncertainties, are highly nonlinear, coupled and time varying. According to [10], the six degrees-of-freedom nonlinear equations of motion of the vehicle are defined with respect to two coordinate systems as shown in Fig. 2. The dynamic models of thrusters and pumps have been included in the present study. The AUV model is simulated by a mathematical model based on physical laws and design data of the ADFA AUV.

3 NFN System Identification of AUV

The NFN uses a nonlinear combination of input variables (FLNN) [11] with the fuzzy system. Each fuzzy rule corresponding to the FLNN comprises a functional expansion of input variables. The FLNN, initially proposed by [4], is a single-layer ANN structure capable of forming complex decision regions by generating nonlinear decision boundaries. In a FLNN, the need of hidden layer is removed.

In this study, the functional expansion block comprises of a subset of orthogonal polynomials bases function. The FLNN has been inserted to the consequent part of the fuzzy rules. The local properties of the consequent part in the NFN model enable a nonlinear combination of input variables to be approximated more effectively.

3.1 Functional link neural network structure (FLNN)

The FLNN is a single-layer network while the input variables generated by the linear links of neural networks are linearly weighted, the functional link acts on an element of input variables by generating a set of linearly independent functions, orthogonal polynomials for a functional expansion, and then evaluating these functions with the variables as the arguments. Therefore, the FLNN structure considers trigonometric functions. In the FLNN structure as shown in Fig. 3, a set of basis functions \( \Phi \) and a fixed number of
weight parameters $W$ represent $f_{W(k)}$. The theory behind the FLNN for multidimensional function approximation has been discussed in [12]. Consider a set of basis functions $B = \{\varphi_k \in \Phi(A)\}_{k \in K}$, $K = \{1, 2, \ldots\}$. Let $B = \{\varphi\}_{k=1}^M$ be a set of basis functions. The FLNN comprises $M$ basis functions $\{\varphi_1, \varphi_2, \ldots, \varphi_M\} \in B_M$. The linear sum of the $j^{th}$ node is given by

$$\hat{y}_j = \sum_{k=1}^M w_{kj} \varphi_k(X)$$  \hspace{1cm} (2)

where $X \in A \subset \mathbb{R}^N$, $X = [x_1, \ldots, x_N]^T$ is the input vector and $W_j = [w_{j1}, \ldots, w_{jM}]^T$ is the weight vector associated with the $j^{th}$ output of the FLNN. $\hat{y}_j$ denotes the local output of the FLNN structure and the consequent part of the $j^{th}$ fuzzy rule in the NFN model. The $m$-dimensional linear output may be given by $\hat{y} = W \Phi$, where $\hat{y} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m]^T$, $m$ denotes the number of functional link bases, which equals the number of fuzzy rules in the NFN model, and $W$ is an $(m \times M)$-dimensional weight matrix of the FLNN given by $W = [w_{11}, w_{22}, \ldots, w_{MM}]^T$.

In this study, each degree of freedom of underwater vehicle dynamics is represented by one NFN model. Therefore, there are six SISO NFN models to represent six degrees of freedom of AUV. In the precondition part, the input is represented by five Gaussian membership functions (mean five means and variance). In the consequent part, the output is generated by FLNN. The function expansion in FLNN uses trigonometric functions, given by $[1, \hat{x}_1, \sin(\pi \hat{x}_1), \cos(\pi \hat{x}_1)]$ for one input variable.
each parent vector from the current population (target vector), a mutant vector (donor vector) is obtained. Finally, an offspring is formed by combining the donor with the target vector. A tournament is then held between each parent and its offspring with the better being copied to the next generation [13, 14]. The DE learning algorithm consists of four major steps—the initialization step, the evaluation step, the reproduction step, and the mutation step.

A. Initialization step: The first step in DE is the coding of the neural fuzzy network model parameters into an individual. Equation (4) shows the way of the individual coding of NFN parameters, where \( i \) and \( j \) represent the \( i^{th} \) input variable and the \( j^{th} \) rule, respectively. In this study, a Gaussian membership function is used with variables that represent the mean and variance of the membership function. \( m_{ij} \) and \( \sigma_{ij} \) are the mean and variance of a Gaussian membership function, respectively, and \( w_{Mj} \) represents the corresponding link weight of the consequent part that is connected to the \( j^{th} \) rule node.

\[
\text{Individual} = m_{1j}, m_{2j}, \ldots, m_{nj}, w_{1j}, \ldots, w_{Mj}
\]  

(4)

B. Evaluation step: The objective function is used to provide a measure of how individuals have performed in the problem domain. In the minimization problem, the fit individuals will have the lowest numerical value of the associated problem objective function. This raw measure of fitness is usually only used as an intermediate stage in determining the relative performance of individuals in a DE. Another function, the fitness function, is normally used to transform the objective function value into a measure of relative fitness.

\[
F(x) = g(f(x))
\]  

(5)

where \( f(x) \) is the objective function, \( g \) transforms the value of the objective function to a non-negative number and \( F \) is the resulting relative fitness. The cost function that attempts to optimize the whole NFN model parameters keeping the interaction between the different degree of freedom is ITSE (Integral Time of Square Error) over the total simulation time.

\[
ITSE = \int_{t_1}^{t_2} e^2 \, dt
\]  

(6)

\[
J_{E1} = \text{Max}(ITSE)
\]  

(7)

\[
F_{E1} = 1/J_{E1}
\]  

(8)

where \( J_{E1} \) is the objective function of each degree of freedom separately for all population. \( F_{E1} \) is the overall fitness function of that degree of freedom. The objective function using fitness produces distribution in the range \((0, 1)\).  

C. Mutation and crossover step: This operation enables DE to explore the search space and maintain diversity. The simplest form of this operation is that a mutant vector is generated by multiplying an amplification factor, \( F \), by the difference between two random vectors and the result is added to a third random vector (DE/rand/1) [9] as:

\[
\bar{V}_{z,t} = \bar{x}_{r_1,t} + F \times (\bar{x}_{r_2,t} - \bar{x}_{r_3,t})
\]  

(9)

where \( r_1, r_2, r_3 \) are random numbers \((1, 2, \ldots, PS)\), \( r_1 \neq r_2 \neq r_3 \neq G \), \( x \) is a decision vector, \( PS \) is the population size, \( F \) is a positive control parameter for scaling the DE and \( G \) the current generation. For more details, readers are referred to [8].

D. Reproduction and selection step: To keep the population size constant over subsequent generations, the next step of the algorithm calls for selection to determine whether the target or the trial vector survives to the next generation, i.e., at \( G = G + 1 \). The selection operation is described as

\[
\bar{x}_{i,G+1} = \bar{U}_{iG} \quad \text{if} \quad f(\bar{U}_{iG}) \leq f(\bar{x}_{i,G})
\]

\[
= \bar{x}_{i,G} \quad \text{if} \quad f(\bar{U}_{iG}) > f(\bar{x}_{i,G})
\]  

(11)

where \( f(\alpha) \) is the objective function to be minimized.

4.2 ON-LINE procedure

The learning algorithm is using the back propagation technique. The back-propagation algorithm minimizes a given cost function, Eq (13), by adjusting the parameter of the membership in the antecedent part of the NFN system as mentioned in Eq (15, 16) and the link weight parameters of the FLNN in the consequent part of the NFN system as mentioned in Eq (18, 19).

4.2.1 Parameter learning phase

After the NFN parameters have been tuned in the optimization phase according to a certain input-output training data set, the network enters the parameter-learning phase to adjust the parameters of the network based on a different real input-output data, mathematical model simulation or real system. The final output of the NFN model is given by equation (12) with \( c^T_i \) and \( \sigma^T_i \) representing the centre and width of Gaussian memberships for input variable \( x_i \) for the rule \( i, j \) and \( n \) being the number of rules of fuzzy model and number of inputs respectively.

\[
y_m(k + 1) = \frac{\sum_{i=1}^{R} \prod_{l=1}^{n} \exp\left(-0.5 \frac{(x_i - c^T_i)^2}{\sigma^T_i}ight)}{\sum_{i=1}^{R} \prod_{l=1}^{n} \exp\left(-0.5 \frac{(x_i - c^T_i)^2}{\sigma^T_i}ight)}
\]  

(12)

The adaptive neural fuzzy network model is a neural fuzzy network model with a training algorithm where the model is synthesized from a bundle of fuzzy If-Then rules. As
shown in Figure 5, the fuzzy model is placed in parallel with the process to be identified. This process could be a real system, set of input-output data or mathematical model. It aims to identify the model of the process online by using the input-output measurements of the process based on a training algorithm. The learning process involves determining the minimum of a given cost function (13). The gradient method is used to adapt the model parameters based on the following objective function

\[ E(k) = \frac{1}{2}(y_m(k + 1) - y_f(k + 1))^2 \]  

where \( E(k) \) is error between the ANFN model and the actual plant outputs. If \( Z(k) \) represents the parameter to be adapted at iteration \( k \) in the ANFN model, the back propagation algorithm seeks to minimize the value of the objective function by [15].

\[ z(k + 1) = z(k) - \alpha \frac{\partial E}{\partial z} \]  

To train \( c_i \):

\[ c_i^{(k + 1)} = c_i^{(k)} - \alpha \frac{y_m - y_f}{b} (y_f - y_m)z \frac{2(x_i - k)(k)}{\sigma_i^2(k)} \]  

To train \( \sigma_i \):

\[ \sigma_i^{(k + 1)} = \sigma_i^{(k)} - \alpha \frac{y_m - y_f}{b} (y_f - y_m)z \frac{2(x_i - k)(k)}{\sigma_i^2(k)} \]  

where

\[ b = \sum_{i=1}^{n} z_i; \quad Z = \prod_{i=1}^{n} \exp \left( -0.5 \left( \frac{k - c_i}{\sigma_i} \right)^2 \right) \]

By following the same sequence, the equation for adapting parameter \( w \) is derived as shown in the following equations.

\[ w_{ij}(k + 1) = w_{ij} - \alpha_w \frac{\partial E}{\partial w_{ij}} \]  

where \( \alpha_w \) is the learning rate parameter of the FLNN weight. The learning rate \( \alpha \) in equation (15, 16 and 18) [10] has a significant effect on the stability and convergence of the system. A higher learning rate may enhance the convergence rate but can reduce the stability of the system. A smaller value of the learning rate guarantees the stability of the system but slows the convergence. The proper choice of the learning rate is therefore very important.

In this technique, the fuzzy network is adapted through two ways. The back propagation algorithm is applied to tune the membership function parameters in antecedent part. The consequent part of the fuzzy rules is adapted through the FLNN. In addition, the back propagation algorithm is used for tuning the parameters of FLNN as well.

4.3 Convergence analysis of ANFN system

Each learning rate parameter of the weight, the mean, and the variance, \( \alpha \), has a significant effect on the convergence. To ensure a quick and stable convergence of fuzzy controller parameters a convergence analysis of the learning rate \( \alpha \) will be considered according to the following theorem.

Theorem [16]: Let \( \alpha \) be the learning rate for the parameters of fuzzy controller and \( g_{\text{max}} \) be defined as \( g_{\text{max}} = \max_k ||g(k)|| \) where \( g(k) = \delta y(k)/\delta z(k) \) and ||.|| is the usual Euclidean norm in \( \mathbb{R}^n \) and \( S = \delta y_m/\delta u \). Then the convergence is guaranteed if \( \alpha \) is chosen as

\[ 0 < \alpha < \frac{2}{S^2 g_{\text{max}}} \]  

5 Simulation Results

This study demonstrated the performance of the NFN model for nonlinear system modeling. This section simulates the AUV system behavior and compares the performance of the NFN model with the mathematical model of the AUV.

Fig. 6 shows the motion of the AUV in square trajectory in XY plane, which means the yaw angle control. It is obviously seen that DE has a good job to tune the model’s parameter from initial values and get them close enough to the original behavior of the vehicle and it makes the job easier for the online tuning to get the model track successfully the behavior of the vehicle. The motion of the AUV with NFN model with online tuning algorithm does a better job.
compared to the performance of the same model without adaptive algorithm in terms of accuracy as well as the speed.

Fig. 7, Fig. 8 and Fig. 9 provide an example of the performance of NFN model and its tracking capability of the mathematical model. These figures show the angular velocity in pitch motion, \( q \), the error between the mathematical and NFN model in pitch motion, \( q \), and the error between the mathematical and NFN model in yaw motion, \( r \), respectively. It is clearly seen that the behavior of the NFN model has same as the mathematical model. The NFN system identification has the capability to track the plant successfully whatever the change in the operation condition.

The learning rates, for all models were initially set to 1 to train the model parameters. Then the convergence conditions were verified at every sampling time. In the present work, the learning rate belongs to the range \([0.009, 0.236]\) was always within the limit of convergence for parameters in NFN model.

### 5.1 Open loop accuracy evaluation of AUV model

The most widely used method for measuring performance and the accuracy indicators of the system is the root mean square error (RMSE). Table 1 shows the RMSE values of the modeling error in velocity in each degree of freedom. Those errors are the error between the actual velocities generated from the mathematical model and velocities generated from NFN model with and without online adaptive.

<table>
<thead>
<tr>
<th>Velocity Error</th>
<th>NFN Model without online adaptive</th>
<th>NFN Model with online adaptive</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u )</td>
<td>0.0539</td>
<td>3.4493e-004</td>
</tr>
<tr>
<td>( v )</td>
<td>0.1072</td>
<td>0.0048</td>
</tr>
<tr>
<td>( w )</td>
<td>0.0214</td>
<td>1.5662e-006</td>
</tr>
<tr>
<td>( q )</td>
<td>0.0361</td>
<td>4.3173e-006</td>
</tr>
<tr>
<td>( r )</td>
<td>0.1288</td>
<td>9.8547e-007</td>
</tr>
</tbody>
</table>
6 Concluding Remarks

The paper presents the numerical simulation results of the online adaptive NFN modeling with and without online adaptive as system identification of the mathematical modeling of the AUV. System identification with the NFN with FLNN structure is found to be quite effective for the coupled nonlinear, six degree of freedom dynamic model. It is shown that the overall performance of a suitably chosen NFN structure is superior to an adaptive fuzzy structure for identification of nonlinear dynamic systems.

NFN modeling is used to identify the model of the AUV using input-output data. The back propagation as a training algorithm for the system proves the fast convergence of the NFN model identifier successfully achieved a similar performance of the process.

7 References


