TELKOMNIKA, Vol.9, No.2, August 2011, pp. 245~256
ISSN: 1693-6930
accredited by DGHE (DIKTI), Decree No: 51/Dikti/Kep/2010

Optimization of an Intelligent Controller for an Unmanned Underwater Vehicle

Amrul Faruq*[1], Shahrum Shah Bin Abdullah[2], M. Fauzi Nor Shah[2]
[1]University of Muhammadiyah Malang, Malang 65144, Indonesia
[2]Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia
e-mail: falux18@gmail.com

Abstrak

LIngkungan bawah laut merupakan tantangan yang sulit untuk navigasi otonom di bawah air. Sebuah masalah standar kendaraan bawah air adalah untuk mempertahankan posisinya pada kedalaman tertentu dalam rangka untuk melakukan operasi yang diinginkan. Sebuah pengendali yang efektif diperlukan untuk tujuan ini dan karenanya disain kendali kedalaman untuk kendaraan bawah laut tak berawak dijelaskan dalam tulisan ini. Algoritma kendali disimulasikan dengan menggunakan panduan navigasi kelautan dan kontrol simulator. Hasil pada penelitian ini menunjukkan bahwa metamodel fungsi radial basis dapat digunakan untuk mengatur faktor skala dari sebuah pengendali logika fuzi. Dengan menggunakan pendekatan optimasi offline, perbandingan antara algoritma genetika dan metamodeling telah dilakukan untuk meminimalkan kesalahan antara masukan awal dan tingkat kedalaman kendaraan bawah air. Hasil penelitian menunjukkan bahwa metode ini memungkinkan untuk mendapatkan kesalahan yang dapat diterima menggunakan pendekatan metamodeling di banyak waktu yang lebih singkat dibandingkan dengan pendekatan algoritma genetika.

***Kata kunci**: algoritma genetika, kendaraan bawah air, logika fuzi, metamodel, optimasi*

Abstract

Underwater environment poses a difficult challenge for autonomous underwater navigation. A standard problem of underwater vehicles is to maintain it position at a certain depth in order to perform desired operations. An effective controller is required for this purpose and hence the design of a depth controller for an unmanned underwater vehicle is described in this paper. The control algorithm is simulated by using the marine guidance navigation and control simulator. The project shows a radial basis function metamodel can be used to tune the scaling factors of a fuzzy logic controller. By using offline optimization approach, a comparison between genetic algorithm and metamodeling has been done to minimize the integral square error between the set point and the measured depth of the underwater vehicle. The results showed that it is possible to obtain a reasonably good error using metamodeling approach in much a shorter time compared to the genetic algorithm approach.

***Keywords**: fuzzy logic, genetic algorithm, metamodel, optimization, underwater vehicle*

1. Introduction

Underwater vehicles are important tools for undersea operations [1]. It is rapidly increasing as they can operate in deeper and riskier areas where divers cannot reach. Underwater vehicles of varying types have been designed and developed as an alternative for various tasks like inspection, repairs and retrieval that would be impractical with a manned mission.The first use of such devices was purely military, but typical applications today include: survey and research, surveillance, mine neutralization, inspection of man-made systems, recovery, repair and maintenance, construction, cleaning, and cable burial and repair [2].

Since underwater vehicles development require a high degree of operator skill for effective operation, the development of vehicles having greater hydrodynamic model becomes highly desirable. One of the critical parts of the vehicle is the control system that would affect the vehicles motion while descending into water. Autonomous underwater vehicles (AUVs) are unmanned, tether-free, powered by onboard energy sources such as batteries or fuel cells, equipped with various navigation sensors such as inertial measurement unit (IMU), sonar sensors, laser ranger and pressure sensor, and controlled by onboard devices, generally computers with preprogrammed mission.

Received March 30th, 2011; Revised June 12th, 2011; Accepted June 17th, 2011

P. Ridao *et al*. have explored an identification method of non-linear models for UUVs [2]. For the off-line identification, the integral method which is based on the minimization of the velocity one step prediction error gave better results compared to the direct method which is based on minimizing the acceleration prediction error. Budiono *et al.* described the coefficient diagram method (CDM) controller that can achieve a satisfactory performance with relatively simple design process [3].

There have been various efforts of the conventional and more than modern control schemes to develop the controller for the AUV which include unmanned underwater vehicle. Simple control techniques such as PID control have been more commonly used because of the relative ease of implementation [4]. Two differents control schemes which included continuous input smoother (CIS) block, which smoothes the PID reference input and discrete fuzzy smoother (DFS) have been proposed by Zanoli *et al.* to reduce potentially dangerous overshoots for depth control of the UUVs [5]. In Kashif a single input fuzzy logic controller (SIFLC) was designed and shown to give identical response with conventional fuzzy logic controller (CFLC) [6]. The SIFLC requires very minimum tuning effort and its execution time is in the orders of two magnitudes less than CFLC. Another application was described by Smith *et al.* to control heading, pitch, and depth by three separates fuzzy logic controllers. The fuzzy controllers were tested using a nonlinear simulation model of the ocean voyager and show good performances over a range of velocities [7].

A model based on fuzzy modeling and control for AUV was used to describe the nonlinear AUV system in [8], by applying a linear matrix inequality (LMI) method to design a stability condition for non linear FLC Takagi–Sugeno (T-S) type fuzzy model. A multivariable sliding mode autopilot have been designed by Haeley *et al.* based on state feedback, decoupled modeling of a slow speed for combining, steering and diving response of the AUV [9]. Intelligent techniques included genetic algorithm and neural network approaches have been proposed and implemented with success on AUVs in several cases [10, 11] for constructing controllers has the advantage that the dynamics of the controlled system need not be completely known.

In the traditional and modern control schemes, controller design requires an accurate model of the system to be controlled. In this study, the design is based on fuzzy logic which requires only an understanding on the relation between the input and output of the system and thereby can be derived to control the system. The focuses here are on optimizing the controller's scaling factors such that it minimizes the integral square error (ISE) between the set point and the measured depth of an unmanned underwater vehicle (UUV).

2. Unmanned Underwater Vehicle Model

Underwater vehicles can be classified into two basic categories; manned underwater vehicles and unmanned underwater vehicles (UUVs) [12]. Unmanned Underwater Vehicles (UUVs) is the term referring to remotely operated underwater vehicles (ROV) and autonomous underwater vehicles (AUVs). These two types of UUVs contribute to the same control problems. These vehicles have been used for over 100 years and have been known to be an interesting research area for universities and industries.

Using the *Society of Naval Architects and Marine Engineers* (SNAME) 1950 [13] notation, the Deep Submergence Rescue Vehicle (DSRV) modeling will be more discussed on that suggested by Fossen, 1994 [14]. For marine vehicles moving in six degrees of freedom (DOF), six independent coordinates are necessary to determine the position and orientation of a rigid body in three dimensions. The first three coordinates and their time derivatives correspond to the position and translational motion along the $x-, y-$ and $z-$ axes respectively, while the last three coordinates and their time derivatives are used to describe orientation and rotational motions. The six motion components are conveniently defined as surge, sway, heave, roll, pitch and yaw [14].

The general motion of an underwater vehicle in six DOF is modeled by using the notation of Fossen [14]. The velocity of the vehicle is described as a vector v:

$$v = [u \;\; v \;\; w \;\; p \;\; q \;\; r]^T \qquad (1)$$

Where u, v, w are translational along, and p, q, r are rotations around the three axes [X –axis, Y –axis, Z –axis] respectively. Using Euler angles the position and orientation of the vehicle may be described as a vector η relative to the global reference frame:

$$\eta = [x \ \ y \ \ z \ \ \phi \ \ \theta \ \ \psi]^T \quad (2)$$

And the nonlinear vehicle dynamics can be expressed in a compact form as:

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = B(v)u \quad (3)$$

Where,
M is the 6×6 inertia matrix including hydrodynamic added mass.
$C(v)$ is the Matrix of the Coriolis and centripetal forces.
$D(v)$ is the Hydrodynamic damping matrix.
$g(\eta)$ is the Vector of restoring forces and moments.
$B(v)$ is the 6×3 control matrix.

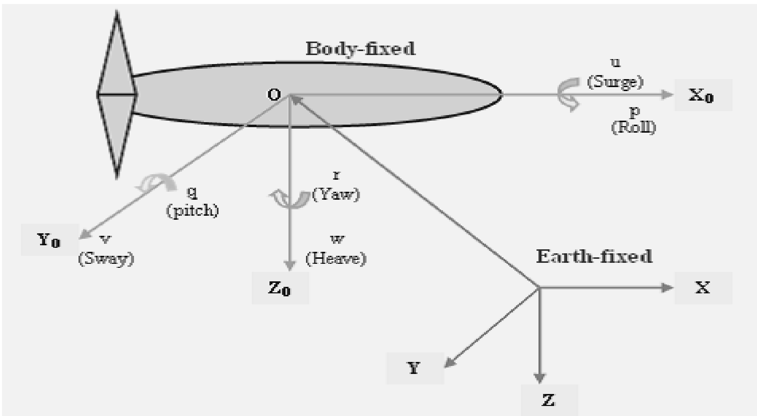

Figure 1. Body-fixed and earth-fixed reference frames

The 6 DOF rigid-body equations of motion are usually written in component form according to SNAME notation [13] and the parameters, hydrodynamic derivatives and main dimension of the vehicle [9] is given in Table 1. The diving equations of motion should include the heave velocity w, the angular velocity in pitch q, the pitch angle θ, the depth z, and the stern plane deflection δ_s. Forward speeds of the vehicle are assumed constant and the sway *v* and yaw *r* modes can be neglected. Then the simplified rigid-body equations of motion in heave and pitch can be expressed as:

$$m(\dot{w} - u_0 q) = Z$$
$$I_y \dot{q} = M \quad (4)$$

For a vehicle operating in the vertical plane the following assumptions can be made; speed is constant, nonlinear terms can be ignored, the roll angle is zero and the pitch angle is small. ($\theta_0 = q_0 = \phi_0 = 0$). Thus suggests the following relations:

$$\dot{\theta} = q$$
$$\dot{z} = -u_0 \sin\theta + w\cos\theta \approx -\theta u_0 + w \quad (5)$$

The external forces the external and moments are described by hydrodynamic added mass, linear damping, and the effects of the stern plane deflection. In addition, the moment caused by the vertical distance between the centre of gravity and the centre of buoyancy, $\overline{BG_z} = z_G - z_B$ must be modelled.

$$Z = Z_{\dot{w}}\dot{w} + Z_{\dot{q}}\dot{q} + Z_w w + Z_q q + Z_\delta \delta_s$$
$$M = M_{\dot{w}}\dot{w} + M_{\dot{q}}\dot{q} + M_w w + M_q q - mg(z_G - z_B)sin\theta + M_\delta \delta_s$$
$$M \approx M_{\dot{w}}\dot{w} + M_{\dot{q}}\dot{q} + M_w w + M_q q - W\overline{BG_z}\theta + M_\delta \delta_s \qquad (6)$$

This together with simplified heave and pitch equations (4), (5), and (6) can be expressed in matrix form as:

$$\begin{bmatrix} m-Z_{\dot{w}} & -Z_{\dot{q}} & 0 & 0 \\ -M_{\dot{w}} & I_y - M_{\dot{q}} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} -Z_w & mu_0 - Z_q & 0 & 0 \\ -M_w & -M_q & \overline{BGz}W & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & u_0 & 0 \end{bmatrix}\begin{bmatrix} w \\ q \\ \theta \\ z \end{bmatrix} = \begin{bmatrix} Z_\delta \\ M_\delta \\ 0 \\ 0 \end{bmatrix}\delta_s \qquad (7)$$

This implies a state space model;

$$\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & 0 \\ c_{21} & c_{22} & c_{23} & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -u_0 & 0 \end{bmatrix}\begin{bmatrix} w \\ q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \\ 0 \\ 0 \end{bmatrix}\delta_s \qquad (8)$$

The above model can further be reduced by considering the heave velocity during diving is small and that $X_G = 0$ and this is quite true because in real situation most small underwater vehicles move slowly in the vertical direction. This assumption implies that the linear model (7) reduces to:

$$\begin{bmatrix} \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \frac{M_q}{I_y - M_{\dot{q}}} & -\frac{\overline{BG_z}W}{I_y - M_{\dot{q}}} & 0 \\ 1 & 0 & 0 \\ 0 & -u_0 & 0 \end{bmatrix}\begin{bmatrix} q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} \frac{M_\delta}{I_y - M_{\dot{q}}} \\ 0 \\ 0 \end{bmatrix}\delta_s \qquad (9)$$

Table 1. Parameters, hydrodynamic derivatives and main dimensions

$I_y = I_z$ = 0.001925	M_δ = -0.012797
m = 0.036391	$Z_{\dot{q}}$ = -0.000130
$M_{\dot{q}}$ = -0.001573	$Z_{\dot{w}}$ = -0.031545
$M_{\dot{w}}$ = -0.000146	Zq = -0.017455
M_q = -0.01131	Zw = -0.043938
M_w = 0.011175	Z_δ = 0.027695
M_θ = -0.156276/U^2	

Transfer function for the above system related to Depth and Stern Plane is

$$\frac{z}{\delta_s}(s) = \frac{b_1 s^2 + (b_2 a_{12} - b_1 a_{22} - b_2 u_0)s + (b_2 u_0 a_{11} - b_1 a_{21} u_0 - b_1 a_{23} + b_2 a_{13})}{s[s^3 - (a_{11} + a_{22})s^2 + (a_{11}a_{22} - a_{23} - a_{21}a_{12})s + (a_{11}a_{23} - a_{21}a_{13})]} \qquad (10)$$

This model in (10) is well suited for control design. The transfer functions $\theta(s)/\delta_s(s)$ and $z(s)/\delta_s(s)$ are obtained as follows:

$$\frac{\theta(s)}{\delta_s(s)} = \frac{K_\theta}{s^2+2\zeta_\theta w_\theta s+w_\theta^2} \quad (11)$$

$$\frac{z(s)}{\delta_s(s)} = -\frac{u_0}{s}\frac{\theta(s)}{\delta_s(s)} \quad (12)$$

Where, the gain constant is (K_θ), the natural frequency (w_θ) and relative damping ratio (ζ_θ) are defined as;

$$K_\theta = \frac{M_\delta}{I_y-M_{\dot{q}}} \quad (13)$$

$$w_\theta = \sqrt{\frac{\overline{BG_z}W}{I_y-M_{\dot{q}}}} \quad (14)$$

$$\zeta_\theta = \frac{-M_q}{2\sqrt{\overline{BG_z}W(I_y-M_{\dot{q}})}} \quad (15)$$

Where,

q = pitch rate [rad/s]
θ = pitch angle [rad]
z = depth [m]
u_0 = vehicle's speed [m/s]
I_y =moment inertia around the vehicle's y-axes
w = heave speed [m/s]
W = mg = vehicle's weight [N]
m = vehicle's mass [kg]
M = mass and inertia
δ_s = stern plane deflection [rad]

Based on this model, we can observe that the system is complicated, since its dynamics are described by highly nonlinear high-order differential equations with uncertainties and disturbances that are difficult to model or measure. Thus designing and optimizing a controller for the system will not be easy. This makes it virtually impossible to apply linear control techniques since there are no clearly defined operating points to linearize about.

3. Research Method

This study has been done by combining of modeling, controller design and simulation. Complete design and procedures of this study are explained more detail. The optimization methods used here is to optimize the input and output gains of the fuzzy logic controller, also known as scaling factors (see Figure 3). Two optimization approaches are used as which are the Genetic Algorithm approach and the Radial Basis Function Artificial Neural Network metamodeling technique. Figure 3 shows the three (3) scaling factors for the fuzzy logic controller *(k, k_1 and k_2).* The performance measure that was used in this case is the ISE and also the time taken to complete both approaches. The ISE is defined by:

$$ISE = \int\left(y_d(t) - y(t)\right)^2 dt \quad (16)$$

where y_d is the desired output (set point, depth in this case) while y is the actual output. This criterion, although is not very selective, has been used because of the ease of computing the integral both analytically and experimentally [17].

3.1. Fuzzy Logic Scaling Factor

The proposed design for this study is described in figure 2. The basic idea behind fuzzy logic control is to incorporate the “expert experience” of a human operator in the design of a controller in controlling a process whose input-output relationship is described by a collection of fuzzy control rules (e.g. IF-THEN rules) involving linguistic variables. This utilization of linguistic variables, fuzzy control rules, and approximate reasoning provides a means to incorporate human expert experience in designing the controller.

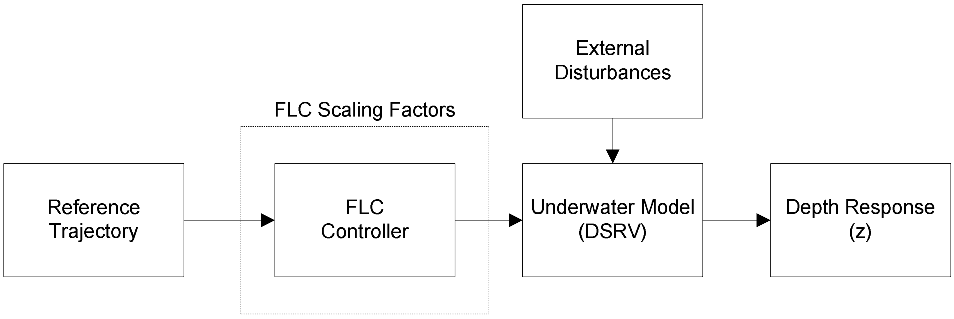

Figure 2. Block diagram of the proposed system

Basically, fuzzy controller comprises of four main components; fuzzification interface, knowledge base, inference engine and defuzzification interface. The fuzzification interface transforms input crisp values into fuzzy values, whereas the knowledge base contains knowledge of the application domain and the control goals. The inference mechanism consists of decision-making logic that performs inference for fuzzy control actions and the defuzzification interface changes back the fuzzy values into the crisp values.

For the fuzzification process, the triangular membership functions are used for both input and output with the universe of discourse as follows:

$$Z_e = [-1, +1]; W = [-1, +1];\ \delta_s = [-1, +1] \quad (17)$$

These values were obtained by observing the corresponding values of Z_e, W and δ_s in the original system using the classical controller that was available in MATLAB®. Table 2. shows the rules table used in this study for the DSRV model.

A scaling factor describes the particular normalization and output denormalization. This is necessary to map the physical values of the linguistic variables into a normalized domain. This plays a similar role to that of the gain coefficients of a conventional controller. From the scaling factors, the controller input and output values are mapped onto the universe of discourse of the fuzzy set definitions. The set definitions are often set at a normalized universe from -1.0 to $+1.0$. The relationship between scaling factor and the limits of linguistic variables is given by:

$$Limits = \pm \frac{1}{SF} \quad (18)$$

The effect of altering a Scaling Factor (SF) is shown in Figure 3. The values for an input variable may range, for example, from -0.1 to $+0.1$ and consequently need to be scaled. If the input value is multiplied by a scaling factor of 10, the input is mapped onto the universe of discourse, as shown by the middle scale in Figure 3. As the limits are $[-1/10, 1/10]$, the full scale is used. In this case, an input value of 0.05 is classified as "positive medium”.

For example, with a scaling factor of 5, the limits become $[-1/5, 1/5]$ and only the central part of the scale is available since the variable ranges from -0.1 to $+0.1$. As a consequence, an input value of 0.05 is now classified as "positive small," as shown by the bottom scale. This example clearly shows that altering the scaling factor causes a change in the classification of the input value. In this example, with a scaling factor of 5, the sensitivity of the controller to an input is reduced, and as in conventional control, the controller gain is reduced.

Table 2. Rules Table for DSRV model

W \ Ze	PL	PM	PS	Z	NS	NM	NL
NL	0	-0.33	-0.66	-1	-1	-1	**-1**
NM	0.33	0	-0.33	-0.66	-1	-1	**-1**
NS	0.66	0.33	0	-0.33	-0.66	-1	**-1**
Z	1	0.66	0.33	0	-0.33	-0.66	**-1**
PS	1	1	0.66	0.33	0	-0.33	-0.66
PM	1	1	1	0.66	0.33	0	-0.33
PL	**1**	**1**	**1**	**1**	0.66	0.33	0

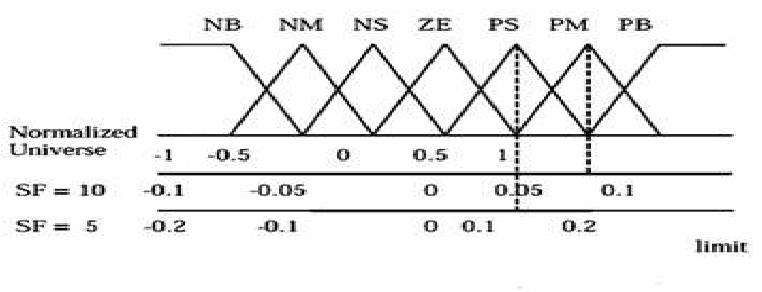

Figure 3. The effect of altering a scaling factor

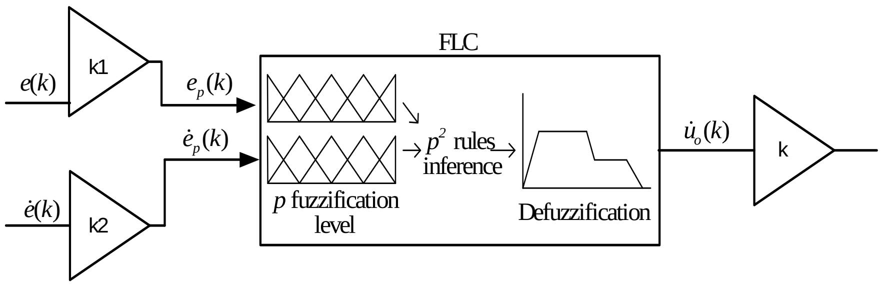

Figure 4. Fuzzy logic controller scaling factor

Due to the utmost importance of the scaling factors with respect to the fuzzy controller performance, this work will present results on optimizing these values using soft computing techniques, namely artificial neural networks and genetic algorithm.

3.2. Genetic Algorithm

Searching is one of ways to solve problems for a lot of problems we are not able to construct an algorithm by definition method of searching step by step, but very often we can specify a set of potential solutions. Goal of strategy of searching is to analyze elements of set in order to fix the best one. It is easy for small sets but if the set increases it becomes more and more complicate and impossible. One of the most advanced and modern searching method are genetic algorithm.

First, a number of individuals (the population) are randomly initialized. Replication starts from base point again and the best individuals are chosen. The selection of chromosomes is a random process, but it is very strongly directed for choosing the best individuals for reproduction. The objective function is then evaluated for these individuals, producing the first generation of genomes. If the optimization criteria are not met, the creation of a new generation starts. Individuals are selected according to their fitness for the production of offspring. Parents are recombined (crossover) to produce offspring. All off spring will be mutated with a certain probability. The fitness of the offspring is then computed. The offspring are inserted into the population replacing the parents, producing a new generation. This cycle is performed until the optimization criteria are reached, or until a pre-set maximum number of generations have been generated. The different settings that were used are shown in Table 3. The best response will then be selected.

3.3. Radial Basis Function Metamodel

For many years, Metamodels or Surrogate Models have been used in simulation to provide approximations to the input-output functions provided by a simulation model. Metamodeling techniques have been widely used in engineering design to improve efficiency in the simulation and optimization of design systems that involve computationally expensive simulation programs [19]. Many existing applications are restricted to deterministic optimization.

When using computationally expensive simulation programs in engineering design, it becomes impractical to rely exclusively on simulation codes for the purpose of design optimization. A preferable strategy is to utilize approximation models that are often referred to as Metamodel as they provide a “model of the model” to replace the expensive simulation model [15]. A comprehensive review of Metamodeling applications in mechanical and aerospace systems has been written by Simpson in 1997 [16], the figures (including letterings and numbers) are large enough to be clearly seen after reduction. If photographs are to be used, only black and white ones are acceptable.

A Radial Basis Function Artificial Neural Network (RBF ANN) was used in this case as the metamodel to approximate the parameters of the fuzzy logic scaling factors. The network consists of three layers: an input layer, a hidden layer and an output layer [17]. If the number of output, $\phi = 1$, the output of the RBF ANN η is calculated according to:

$$\eta(x, w) = \sum_{k=1}^{S1} w_{1k}\, \phi_k(\|x - c_k\|_2) \qquad (19)$$

Where $x \in \Re^{Rx1}$ is an input vector, $\phi_k(.)$ is a basis function, $\|.\|_2$ denotes an Euclidean norm, w_{1k} are the weights in the output layer, S_1 is the number of neurons (and centers) in the hidden layer and $c_k \in \Re^{Rx1}$ are the RBF centers in the input vector space. Equation (16) can also be written as:

$$\eta\,(x, w) = \phi^T(x) w \qquad (20)$$

where

$$\phi^T(x) = [\phi_1(\|x - c_1\|)\ \ \phi_2(\|x - c_2\|) \ldots\ \ \phi_{S1}(\|x - c_{S1}\|)] \qquad (21)$$

And

$$w^T = [w_1\, w_2 \ldots\, w_{1S1}\,] \qquad (21)$$

Even though an intelligent controller can be applied to control the non-linear system, the membership function and the scaling factors have to be tuned in order to reduce the error. Using trial and error approach, this can take a long time in order to achieve the best performance. Here, metamodeling approach is proposed to optimize the scaling factors of the fuzzy logic controller. Metamodeling requires simple computational algorithm to provide best controller parameters [17]. The output of the neuron in a hidden layer is a nonlinear function of the distance between its input and the center c_k. Some typical choices for the functional from of $\phi_k(.)$ are as follows in [18].

Before proceeding with the findings of the controller parameters, the stability of the system needs to be determined. It was found out that the system is indeed stable [6,14] and hence the control of the system should be possible.

The approach to optimize the controller parameters is summarized as follows:
1. Define the input design space, *D*, which consists of a set of initial values of the controller parameters.
2. Obtain the ISE for the Heave for all the design space defined in 1.
3. Create the target data set, *T*, which are consists of the ISE for Heave.
4. Choose design of ANN, which are consists of spread and centre.
5. Train the RBF NN using *D* and *T* (training I/O data).
6. Evaluate (simulate) the RBF NN on a larger input space, *D'*.
7. Find the minimum of the RBF NN output (estimated E). The corresponding controller gains that minimized the RBF output will be the gains to be verified in actual model simulation.
8. Repeat step 1 to 7 should the controller parameter gains are not satisfactory.

In this study, *D* and *D'* are the sets of discrete values given in Table 3. The parameters for the RBF NN that are used to fit the data *D* are summarized below:

Table 3. Initial and large data sets for FLSF

Training Sets (D)	
k	{0.4, 0.65, …, 2.9}
k_1	{0.01, 0.022, …, 0.09}
k_2	{0.001, 0.001, …, 0.007}
Total number of data configurations	539
Test Sets (D')	
k	{0.2, 0.35, …, 3.35}
k_1	{0.01, 0.022, …, 0.1}
k_2	{0.001, 0.001, …, 0.012}
Total number of data configurations	2112

The initial data sets need to be properly identified to achieve best approximation by training the Radial Basis Neural Network. If the initial data sets do not cover the maximum and minimum value of the large data sets, the ANN will try to extrapolate which will produce unacceptable results. The initial data sets should not be too small for proper training and should not be too large to minimize the training time.

4. Results and Discussion

The first approach used in this case is the genetic algorithm toolbox which is available in MATLAB™ toolbox. This tool was used to tune the best parameters of Fuzzy Logic Scaling Factor (FLSF), k, k_1 and k_2.

The results of the implemented Genetic Algorithm (GA) of the scaling factors will be analyzed in this section. The GA designed scaling factors was initially initialized with population size of 50 and 100 (refer Table 4). The response of the GA designed scaling factors will then be analyzed for the ISE value and time evaluation. By using the genetic algorithm toolbox (gatool) which is embedded in MATLAB® it is possible to conveniently select different GA parameters before running the algorithm. The results for all the settings used can be observed from figure 5.

Table 4. Parameters used in GA

all parameters	Population size	Maximum generations	Cross over fraction	Elite Count	Selection	Migration
setting i	50	100	0.8	4	Stochastic uniform	forward
setting ii	50	100	1	2	Stochastic uniform	forward
setting iii	100	200	0.8	4	Stochastic uniform	forward
setting IV	100	200	1	2	Stochastic uniform	forward

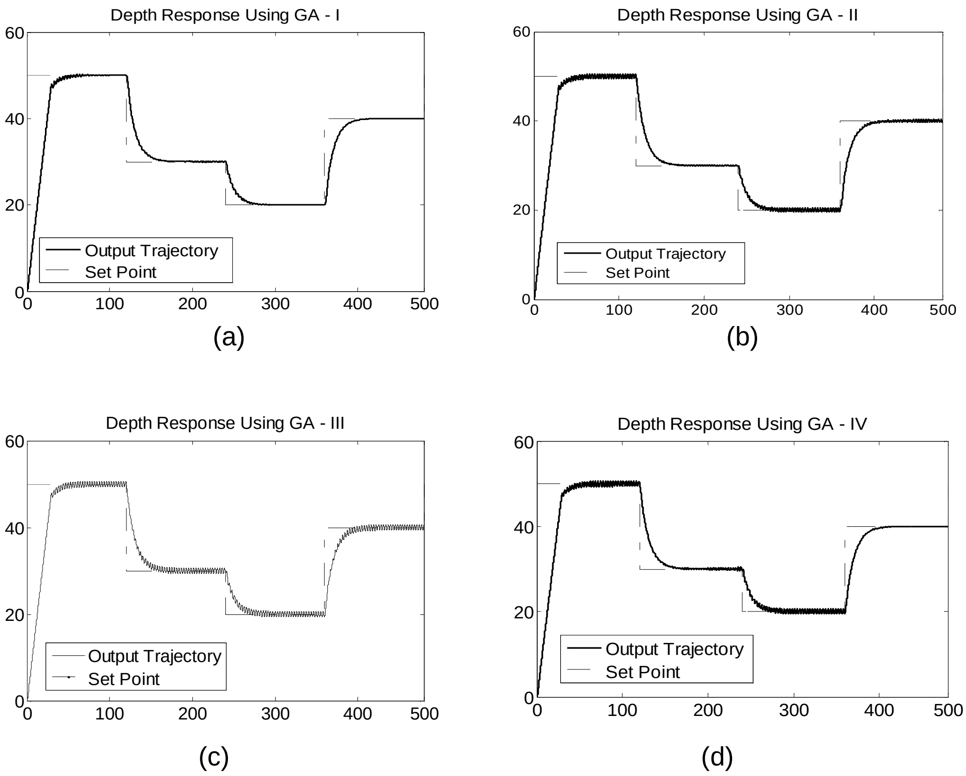

Figure 5. Depth response for variable setting respectively using GA

The other approach was by using Radial Basis Function Neural Network in the metamodeling approach. RBF ANN was used as the metamodel to approximate the mapping of the fuzzy logic scaling factors. The simulation was done by doing a few steps in order to get the optimized parameters. The range of values for *(k, k_1 and k_2)* is as shown in Table 3. These data was used as the input training data of the RBF metamodel. The target training data of the RBF ANN comes from the actual simulation of the DSRV model.

The trained RBF-ANN which will then be used as the metamodel of the DSRV to evaluate the ISE for the corresponding test data sets (D') of the scaling factors parameters. The RBF-ANN was used to evaluate 2112 data sets. The spread value of 20 and 50 were used in the training process and using different number of centers (see Table 5). The results for all the settings used can be observed from figure 6.

Here, there are 3 parameters need to be tuned in order to obtain the best performance. These are *(k, k_1 and k_2)* for the scaling factors of the fuzzy logic controller. By changing the scaling factors, the universe of discourse for the error, change of error and the stern plane angle will be changed. The performance measure that was used in this study was the ISE value. The initial data sets are used to obtain the ISE by simulation.

Table 5.Parameters used in RBF-ANN

All parameters	Total number of input train data configurations	Total number of test data configurations	Centers	Spread
setting i	539	2112	10	20
setting ii	539	2112	50	50
setting iii	539	2112	100	50
setting iv	539	2112	200	20

From the simulation results in Figure 5 and Figure 6, the results obtained by using metamodel approach are almost equal to the result evaluated by using genetic algorithm. However, there is a difference in the simulation time and ISE value (see Table 6). Using Genetic

Algorithm, we managed to get better ISE but the time taken is too long. Using metamodeling, we managed to obtain a reasonably good ISE in a much shorter time, i.e. 9 hours compared to 34 hours. In this case, the data sets D was created simply by choosing the input values in a grid like fashion, based on background knowledge of the problem.

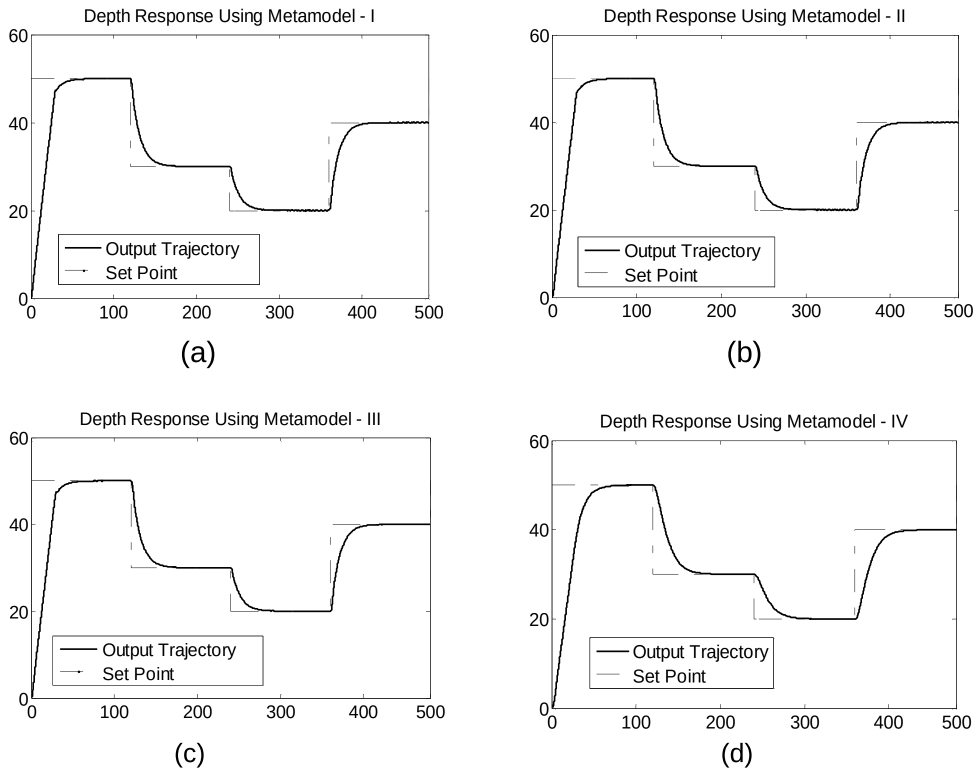

Figure 6. Depth Response using Metamodeling

Table 6. Summary of Results

GA settings	k	k_1	k_2	Time taken (minutes)	ISE (bold lowest)
GA Setting I	0.9200	0.7151	0.5755	183.3311	**3.74e+04**
GA Setting II	0.9582	0.9227	0.2103	509.4022	3.75e+04
GA Setting III	3.3247	1.4181	0.0009	389.7644	3.81e+04
GA Setting IV	0.9244	0.9828	0.2224	1015.3841	3.76e+04
Total time (mnts)				2097.8818	
Total time (hrs)				**34.9647**	
RBF-MM settings	k	k_1	k_2	Time taken (minutes)	ISE (bold lowest)
RBF Setting I	3.35	0.094	0.004	173.0332	**3.88e+04**
RBF Setting II	2.75	0.094	0.012	175.7698	3.94e+04
RBF Setting III	3.35	0.094	0.001	99.2958	3.89e+04
RBF Setting IV	0.5	0.082	0.008	97.2954	4.38E+04
Total time (mnts)				545.3942	
Total time (hrs)				**9.08990**	

5. Conclusion

RBF-ANN has proven its effectiveness as a method of controller optimization in this case. It is able to give a good estimate of the controller parameters in a short time. As the Deep Submergence Rescue Vehicle (DSRV) is a nonlinear system, a non-linear controller can be designed to handle the non-linearities of the DSRV. However, in this work only the depth of the vehicle is considered which makes the system single input and single output. An actual marine vessel model is actually Multi Input Multi Output (MIMO) system, and can be used for further investigation of the study.

A PID controller and a more intelligent controller can also be used in the future work. As an example a neuro-fuzzy controller can be adopted to overcome the highly nonlinear, coupled, and time-varying vehicle. The modified fuzzy membership function-based neural networks can be used to combine advantages of fuzzy logics and neural networks, such as inference capability and adoption of human operators experience with fuzzy logics, and universal approximation and learning capability with neural networks. The parameters of the neural-fuzzy controller can be tuned using the metamodeling approach presented in this paper.

References

[1]. J. Yuh, R. L., An Intelligent Control System for Remotely Operated Vehicles. *IEEE Journal of Oceanic Engineering.* 1993. 18(1) : 55-62.
[2]. Ridao, P., Tiano, A., El-Fakdi, Carreras, Zirilli. On the Identification of non-linear models of Unmanned Underwater Vehicle. *Control Engineering Practice*. 2004. 12 : 1483-1499.
[3]. Budiono, A., Kartidjo, M., Sugama, A., Coefficient Diagram Method for the Control of An Unmanned Underwater Vehicle. *Indian Journal of Marine Science.* 2009. 38(3): 316-323.
[4]. Santhakumar, M., Asokan, T., A Self-Tuning Proportional-Integral-Derivative Controller for An Autonomous Underwater Vehicle, Based on Taguchi Method. *Journal of Computer Science.* 2010. 6 (8): 862-871.
[5]. Zanoli, S. M., Conte, G., Remotely Operated Vehicle Depth Control. *Control Engineering Practice.* 2003. 11 : 453-459.
[6]. Kashif, S. S. Abdullah., Single Input Fuzzy Logic Controller for Unmanned Underwater Vehicle. *Journal of Intelligent and Robotic Systems*. 2010. 59(1): 87-100.
[7]. Smith, S. M., Rae, G.J.S., Anderson, D.T., *Application of Fuzzy Logic to the Control of an Autonomous Underwater Vehicle.* IEEE International Conference. 1993. 2 : 1099 – 1106.
[8]. Chang, W. J., Chang, W., Liu, H., Model-Based Fuzzy Modelling and Control For Autonomous Vehicle in the Horizontal Plane. *Journal of Marine Science and Technology.* 2003. 11 (3) : 155-163.
[9]. Healey, J., Lienard, D., Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE J. Oceanic Eng*. 1993. 18 (3).
[10]. Euan, W. M., Murray, D.J., Li, Y., Fossen, T.I., Ship Steering Control System Optimisation Using Genetic Algorithm. *Control Engineering Practice.* 1996. 8(2000) : 429-443.
[11]. Kodogiannis, V. S., Lisboa, P.J.G., Lucas, J., Neural Network Modelling and Control for Underwater Vehicles. *Artificial Intelligent and Engineering*. 1996. 1 : 203-212.
[12]. Chuhran, C. D., Obstacle Avoidance Control For The Remus Autonomous Underwater Vehicle. PhD theses. California, Naval Postgraduate School Monterey. 2003.
[13]. Society of Naval Architects and Marine Engineers (SNAME), *Nomenclature for treating the motion of a submerged body through a fluid*. 1950. Bull. 1–5.
[14]. Fossen, T. I., Guidance and Control of Ocean Vehicles. England, John Wiley and Sons Ltd. 1994.
[15]. Kleijnen, J. P. C., Statistical tools for simulation practitioners. New York, Marcel Dekker. 1987.
[16]. Simpson, T.W., Peplinski, J., Koch, P.N., Allen, J.K., *On the use of statistics in design and the implications for deterministic computer experiments*. In: Proc. Design Theory and Methodology (DTM '97) Sacramento, ASME-DETC97/DTM-3881. Sacramento. 1997.
[17]. Mohamed Ali, M. S., S. S. Abdullah., Controller Optimization for a Fluid Mixing System Using Metamodeling Approach, *Int J Simul Model*. 2009; 8 (1) : 48-59.
[18]. Ham, F. M., and I. Kostanic, Principles of Neurocomputing for Science and Engineering. Singapore, McGraw-Hill. 2001.
[19]. Mullur, A. M., Extended Radial Basis Functions: More Flexible and Effective Metamodeling. *AIAA Journal*. 2005; 43(6): 1306-1315.