A Graphic Simulator for Autonomous Underwater Vehicle Docking Operations

Xiangming Feng, Bob Qiu and David Y. Y. Yun
Department of Electrical Engineering, University of Hawaii at Manoa
Holmes 492, 2540 Dole Street, Honolulu, Hawaii 96822

Abstract - The performance of the control module in an autonomous underwater vehicle (AUV) is critical to the success of vehicle missions. However, testing such a control unit in-situ is usually very costly and risky. A graphic simulator based on X-window for AUV docking process has been developed. Based on a 3-dimensional physical model of low-speed underwater marine vehicle and a radial basis function (RBF) neural network (NN) as the control unit, the simulator provides an experimental platform for developing, training and testing of AUV controllers in a safe, economic and reliable way. Through its use, the feasibility and effectiveness of a **NN-based** control module (for the vehicle borehole docking project funded by NSF and in cooperation with IFERMER) have been successfully demonstrated.

1. INTRODUCTION

An autonomous underwater vehicle (AUV) deals with various environmental conditions by its control module. Since the situations in the undersea space are not predictable, the control unit must be able to adaptively adjust its output. However, the complexity of the hydrodynamic behavior of the AUV makes the conventional control method difficult because of the necessity of precise but complicated physical model and intricate mathematical methods.

To achieve the adaptability of the control unit, a neural network (NN) based control module is developed. Without knowing the knowledge about the physical behavior of the AUV, a neural network can learning the control strategies from successful docking processes by human operator. As a "model free" approach, it has its own advantages over conventional control method.

However, once such a control unit is developed, it is hard to test it immediately in real operation simply because of the high risk and cost of such a test. Any control methods including NN based approach need an effective and affordable test bed.

This motivates our research work on developing a graphic simulator of the AUV docking process to train and test NN based control module. The simulator does not require an extremely precise physical model and control strategy; however, the ability of the neural network can be demonstrated in a simplified physical world.

In the following, we will first briefly describe the project of AUV borehole reentry project and then introduce the physical model and neural network we use in the simulator. Finally we introduce the structure and function of our simulator.

2. PROJECT OF THE AUV FOR DEEP-SEA ROREHOLF REENTRY

The Autonomous Underwater Vehicle for Deep-sea Borehole Reentry Project is a coordinated undertaking of researchers through the France-USA Cooperative Programs in oceanography. The project integrates several research topics in borehole reentry problem. The boreholes are located at the bottom of the sea-bed, some as deep as 5,000 meters. The AUV is released from the surface vessel and free-falls in a slightly negatively buoyant state until it reaches the vicinity (about several hundred meters away) of the borehole., Then, the control system of the AUV is responsible for the following docking process.

The control system is the central part during AUV's autonomous navigation phase. It must be able to adaptively adjust its control command when it reaches the vicinity of the borehole. Before full autonomous operation phase, a communication line is linked to the AUV and control commands are given through the line from the human operator. In full autonomous phase, the AUV only receives additional information by radio wave and is controlled by the control module on board.

V. :k • -k It • It: •

The dynamics of AUV is fundamentally nonlinear in nature due to the rigid body coupling and the hydrodynamic forces on the vehicle. This nonlinear behavior is similar to the motion of aircraft and conventional submarines with following important differences: 1) AUVs may have comparable velocities along all three directions; therefore, conventional control techniques developed for aircraft and submarine can not be used effectively since most of these techniques depend on linearization of the differential equations along the direction of the motion. 2) The high density of the sea water makes the hydrodynamic model of AUV different from that for aircraft's. Thus, a special dynamic physical model for AUV has been developed.

This work is supported in part by the National Science Foundation under Grant No. BCS-9240421

0-7803-2056-5 © 1994 IEEE

3.1. Coordinate Systems

The dynamic model uses two orthogonal coordinate systems [1][2]: global coordinate system which remains fixed at the top of the borehole with origin 0, Z pointing upward normal to seabed and X and f chosen in two convenient mutually perpendicular horizontal directions forming a right-handed system; and the local coordinate system, (o,x,y,z), which is fixed on vehicle with origin o at the gravitational center of AUV, x' pointing through the nose of the vehicle, z pointing through the belly of the vehicle and y' completing the right-handed system (Figure 1).

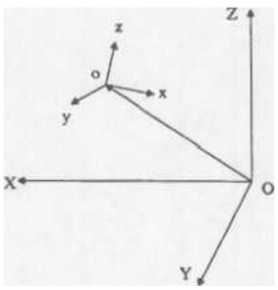

Figure 1. Two coordinate systems of AUV mi

The position and orientation of the vehicle in global coordinates can be specified by R_o, the vector from o to 0 and the Euler angles t4r,0,0. The development of the dynamic model is carried out in the local coordinate system.

3.2.Rioid Body Six-Degree-of-Freedom Dynamic&

The motion of AUV can be described by a combination of translational motion and rotational motion:

Translational Motion: $F = m[U + f2 \times R\sim + S2 \times (f2 \times R\cdot)]$

Rotational Motion: $N = (lit) + m(R_R \times U)$

where $F=(F_,,F_y,F_=)T$ and $N=(N_,,N_y,N_,)T$ are the external force and the external moment, $U=(U.,U_y,U_l)^T$ is the velocity of the origin, $f2=(w_,,w_y,(^0{}_z)^T$ is the angular velocity about the origin, $j?, =(x''_y,,l\mathbf{f}$ is the position of the center of the mass in local coordinates, and

$$l = \begin{matrix} 1, & -1,, & -I\sim \\ -I,_y & l_y & -I_) \\ -1_a & -I_{yr} & It \end{matrix}$$ is the inertia tensor.

We often choose the origin of the local coordinate to be the center of the mass of the vehicle so that $R. = 0$. Also, we assume that the inertia tensor does change with time. Thus the equation is simplified to:

Translational Motion: $F = mU$

Rotational Motion: $N = I\frac{d}{dt}f2$

33.Hydrodynamic Forces and Moment

Assume that the current velocity is U_1 in global coordinate, the relative velocity of the vehicle relative to the fluid is $O_r = U - T^T U_I$.

a) Added Mass: During the motion of the vehicle, some fluid surrounding the vehicle is also accelerated. The amount of effective added mass is an additional inertia term to the above physical equation. The coefficients can be defined as the proportionality with each of the linear and angular accelerations with each of the hydrodynamic forces and moments they generate. For example, the hydrodynamic force along the x-axis due to acceleration in the x-direction is $F_A = -X_{Uj}U$, where $X_{U,} = aU_s$. In a similar manner, other coefficients can be defined and form an added mass matrix.

b) Fluid Motion: This effect describes the forces and moments exerted on the vehicle by fluid motion. For AUV traveling in ocean, these effects are significant and must be included in the dynamic model: $F_F = m_I U_f$ and $N_f = m_I(R_B \times U_I)$, where m_I is the mass of the fluid displaced by the vehicle and fi $_g$ is the position of the center of buoyancy in local coordinates.

c) The Drag: The drag force is largely dependent on the relative velocity and the surface of the vehicle. Since the AUV we consider moves at low speed, we assume that the drag force is proportional to its speed. For example, the drag force along x-axis due to relative velocity in that direction is expressed as k lu, |.

d) Weight and Buoyancy: The gravitational force and buoyant force should be transformed into local coordinates:

$$F_w = mg(\text{-sine cosh sin4 cosh cosq})^T$$

$$F_B = \text{-pgV(-sine cose sin4 coso cosq)}^r$$

e) Thrusters: The total forces and moments of all the thrusters are the vector sum of each individual thruster:

$$FT\text{-}\sim \overset{N}{F7} \quad \text{and} \quad GT=\sim \overset{N}{G7} + \sim \overset{N}{RTi} \quad \bar{T}$$

where R_T is the position of the ith thruster in local coordinates.

Since we assume that the center of the local coordinate is at the gravitational center of the vehicle, we can use a simpler form of equation. Moving the constant coefficient from the added-mass effect to the left and leave all external forces and moments at right, we have

$$MQ - (NF), +(N)F +CN)w +(N)8 + `N/T$$

where vector $Q=(v\sim,v_y, v\sim, \text{lil}_n(J_{Y}, t\ddot{u}_z)^T$ and subscript D, F, H, G, B and T denote the amount of drag, fluid, drag, gravity, buoyancy and thruster, respectively, and

$$M= \begin{matrix} m+F(+'). & F,(v,) & l'(v,) & F(\sim,) & F,(\sim,) & F^{(o},) \\ F('',) & m+F,(v,) & F,(v.) & F,(to,) & F,(to,) & F,(\sim) \\ F,(J,) & F,(v,) & m+F,(v,) & F.(\&.) & F,(d,) & F(6) \end{matrix}$$

3.4 Control Strategy

We assume that the ideal cruising of the AUV is to move at an appropriate and constant velocity towards the center of the borehole and all roll, yaw and pitch rotations are zero. Thus, according to the current position and orientation of the AUV in global coordinate, we can obtain an ideal status of the AUV under current situation. The control strategy is to minimize the difference of the current AUV status with the ideal status.

The dynamics of the AUV now can be described as

$$Q=M^{-1}(N)_D + M^{-1}(N)_F + M^{-1}(F)_W + M1(F)_B + M^{-1}(F)_T$$

On the right side, the term with subscript of W, J, B and T arc constants and the first term can be written as

$$\begin{cases} F & -k_1 V \\ N_{JD} = \sim\text{-}k_2W = d,ag(\text{-}k_i, \end{cases} \quad k_2,\text{-}k_2 \text{ -}k2) \times Q = KQ$$

so the equation can be written as $Q=M'KQ+M\text{-}'u+M\text{-}'c$ or $Q=AQ+Bu+C$ where A, B and C are constant matrix and Q is the status of the vehicle and u is input thruster. Consider time interval from t to t+l, the above equation can be approximated by

$$\frac{Q(t+1) - Q(t)}{At} = AQ(t) + Bu + C$$

If the ideal status of the current situation is $Q(t)$, the control strategy requires that:

$$Q(t+1)\text{-}Q(t) = \frac{}{2}(Q(t)\text{-}Q(t))$$

By substituting the expression of $Q(t+1)$ into it, we have our desired control signal:

$$u=B\text{-}1 \frac{Q(t) \text{-}2AtC\text{-}(2AAt+I)Q(t)}{2\&t}$$

3.5. Neural Network Traininn Sample Generation

To reduce the number of training samples, we select the global coordinate so that the AUV is on the x-z plane. Now there are 7 inputs to the neural network, *i.e.* the angle from the AUV to the borehole, the velocity of the AUV in x, y and z-axis and the three angular velocities of the AUV. Output of the neural network includes 3 thrusters and 3 moment thrusters. Each input argument will take 3 or 4 values spread evenly among the possible values. The total number of training samples is over 9,000. The training results are given in the following section.

4. DYNAMIC CONTROL OF AUV USING NEURAL NETWORKS

Autonomous underwater vehicle (AUV) brings unique control problems because of the complexity of its dynamic behavior, uncompleted knowledge of physical model and disturbances from the surrounding ocean environment. Conventional linear control techniques cannot provide the desired high performance due to many wide variations in AUV system dynamics. Also, conventionally, a theoretical model representing the physical behavior of the system has to be selected before the control techniques can be applied to implement the dynamics of system. However, such a precise model does not exist and the development and implementation of control strategies will be difficult because of the mathematical complexity. On the other hand, the learning ability of neural network (NN) provides another way to obtain control strategies without establishing precise physical model and determining exact parameters of the system. As a non-parametric adaptive control system, a neural network can learn the dynamic navigation control of AUV from the examples of operator's successful control behaviors, including both linear and non-linear dynamics of system performed by human operators. Our current objective is to investigate possible neural network techniques and select the best one for future operation.

From a systematic point of view, the multilayer network represents a versatile nonlinear mapping. The adjustment of weights of a neural network when it is used as a compouznt in a dynamic system can be considered as the adaptive part in the control process. The major drawback of the popular neural networks such as back-propagation (BP) is the computational cost for convergence. Our previous simulation results show that, with a simplified underwater navigation control model and a small set of training data (only 360 examples with 6 input and 3 outputs), it takes days of computing time in workstation (Sun 4) before convergence. Thus, a fast alternative algorithm is necessary for realistic AUV control system.

In this research work, the radial basis function (RBF) neural network has been studied and implemented for dynamic AUV control system. The RBF network offers a fast alternative to the three layer NN. Comparing with traditional BP networks, RBF greatly reduces training time at the expense of requiring a few times as many connection weights. RBF has been applied to function approximation, predictions, and classification. However, few have addressed the adaptive control problem using RBF networks.

4.1 Radial Basis Function Network

RBF network is a simple, idealized network model based upon abstract processing units with locally-tuned response functions. There are three layers in RBF: input layer, hidden layer, and output layer. Each node in the input layer is fully connected to the hidden layer, and each hidden node is fully connected to output layer. The hidden units have radically symmetric activation functions (kernel function) that is maximum when the input is near the centroid of the node. The output of hidden unit is

$$R_j(x)=R(\|x-x_j\|/\sigma_j)$$

Here, x is a real-valued vector in the input space, R_j is the response function of the j-th locally-tuned hidden unit, R is a radically symmetric function with a single maximum at the origin and which drops rapidly to zero at large radii, x_j and σ_j are the centroid and width of the j-th unit in the input space. These parameters are adjustable during the learning phase. Typically, Gaussian response functions with unit normalization are chosen as the radial basis function. $R_j(x) = \exp\left[-\frac{\|x-x_j\|^2}{\sigma_j^2}\right]$ After appropriate training, each kernel node covers a certain range of the input space. We say that each hidden unit has its own receptive field in the input space, a region centered on . with size proportional to σ_j.

The output of the network is $f(X) = \sum_{j=1}^{M} w_j R_j(X)$

where w_j is the weight or amplitude associated with each unit. Since each kernel node has a "locally-tuned" kernel function, those nodes whose centroids are close to the current input contribute more to the output of the network. Weights from kernel nodes to output nodes are determined using the least mean square (LMS) algorithm.

4.2 Learning Algorithm in RBF Network

The processing units' centers and widths in RBF are determined in a bottom-up or unsupervised learning manner. It dramatically reduces the amount of computational time required by supervised learning.

The standard K-means clustering algorithm is used to find the processing unit centers which represent a local minimum of the total squared Euclidean distance E between the N examples (training vectors) X_t and the nearest of the centers x_i:

$$E=\sum_{t=1}^{N}\sum_{i=1}^{K} M_{ti}(X_t-x_i)^2$$

where M_{ji} is the cluster membership function, a K x N matrix of 0's and l's with exactly one 1 per column which identifies the processing unit to which a given exemplar belongs.

The widths of these units are determined using "P-nearest-neighbor' heuristics in order to achieve a certain amount of response overlap between each hidden unit and its neighbors so that they form a smooth and contiguous interpolation over those regions of the input space which they represent.

The weights from the hidden layer to output layer are found in a top-down manner using the supervised LMS rule. When the locally-tuned network above is applied to adaptive control task, the set of hidden (kernel) units consists of a set of functions which are assumed to constitute an arbitrary "basis" for the inputs if we expand them into hidden unit space. It may be shown that expansion of input vectors into a hidden-unit space of relative high dimension (i.e., by defining many radial basis functions) will result in a greater likelihood of the classification problem becoming linearly separable [3]. Thus, the hidden-to-output layer can be a linear network trained by least mean square rule. The total squared error

for an entire training set is $E=\frac{1}{2}\sum_{i,l}(f_i'(x_l)-f_i(x_l))^2$,

where X, is the j-th training pattern, f_i is the i-th output of the network, and f_i' is the desired result. The delta learning rule can be obtained to optimize the cost function. That is, at each time step, a random exemplar x_l is presented and the nearest cluster centroid X; is moved by the amount: $\Delta x_i = \eta(x_l - x_i)$, where 't] is the learning rate.

RBF allows a layer-by-layer type of training, that is, the input-to-hidden weights may be trained first, followed by the hidden-to-output weights. This allows the incorporation of domain-specific knowledge in the initial conditions of the network and offers faster training times than conventional backpropagation networks -- over two orders of magnitude faster in the classification problems reported [4].

4.3 Simulation RrvOts~

The training data generated by the physical model described in the previous section are used. During the training, the number of hidden units is found to vary from 800 to 1200. For the input to hidden layer, the initial weights are obtained by randomly choosing the sample feature vectors. For variance by P-nearest neighbor heuristic, P is set as 2. For the hidden to output layer, the weights are initialized as small random values. The LMS learning rate is set as T] = 0.2. The mean squared error (MSE) over all training samples is used as the criteria for network convergence. The algorithm converges after about 30 iterations with MSE less than 0.1 %. The CPU time in Sun 4 is about one hour.

Before we use NN to control the AUV in our simulator, a group of test samples is generated and the control results on these test samples from neural network and control models are compared. The test samples are chosen at the center of grids of training samples. Figure 2 shows a two-dimensional case of the test data selection These test samples are far away from the training samples so that they can represent, in some sense, the possible worst cases of all possible AUV states. There are 3,000 test samples and the MSE for the test samples is 2.8%. The simulation results show that this error is small enough for the NN to give correct control commands.

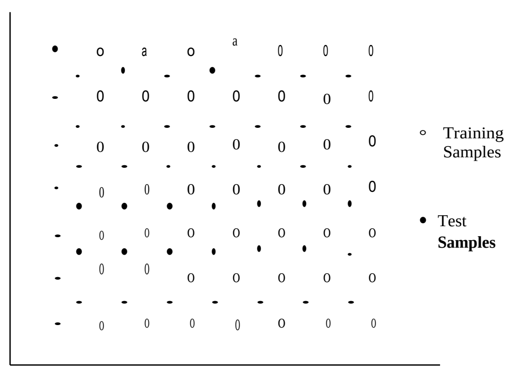

Figure 2. **Training Samples and Test Samples**

V. It till t • V. . J . •t

Our simulator for AUV docking process includes 3 major modules, i.e., the physical model of the AUV, the control unit and the graphic display (Fig. 3). The simulator can perform following tasks:

1) It can simultaneously display in two views (top-view and side-view) the physical status of the vehicle including its position, orientation and other physical parameters such as speed, acceleration and thruster value;

2) In learning phase, it accepts control signals from human operator by means of a pre-stored file or on-time mouse-

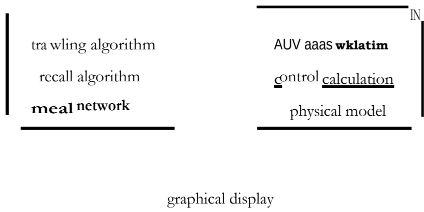

Figure 3. Three Modules in the Simulator

controlled thruster value and then triggers the neural network learning algorithm (Figure 4.a);

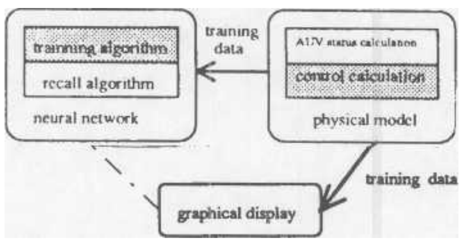

Figure 4.a. Learning phase in the simulator

3) In control phase, it can release the vehicle at any starting point given by the user and begins the neural network recall algorithm as the control unit; a trajectory along with all other information about the AUV status will be displayed (Figure 4.b).

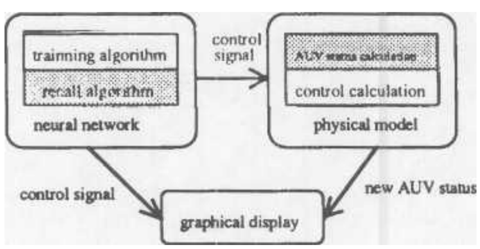

Figure 4.b. Recall (control) phase of the simulator

In our simulator, two pop-up windows display the top-view and side-view of the area between the AUV and the borehole. A set of slide bars shows the control output. The physical status of the AUV including its position, speed and acceleration is numerically displayed. A pull-down menu allows the user to select the initial vehicle position and set the simulator in either learning mode or control mode.

When learning mode is picked, the user can either feed the neural network with a set of training examples in a file or he can control the thruster value on the slider bar by mouse. The control value will be recorded every second and stored temporarily. After the docking, a confirmation whether to use these temporary data is asked. In control mode, a set of weight is loaded and the recall algorithm gives out the control signal, which, along with the AUV status, is shown in the slide bar. In both modes, the top-view and side-view is updated every second.

A prototype of the simulator has also been developed on Macintosh using Micromind Director 3.1 simulation language and has been put on display at the Bishop Museum, Honolulu, HI for 4 months. The simulator was executing smoothly during the entire exhibition. Users of the simulator, who are visitors to the Museum included people from all walks of life, were asked to guide the vehicle to dock through mouse control, thereby also to "teach" the NN. Total number of serious users is about 1,400, among which 10% were successful. These data are used to train the RBF NN and a MSE of about 4% is obtained. This further demonstrates the capability of our NN in learning from *a broad range* of control strategies from human operators.

6. CONCLUSION

A graphic simulator for AUV borehole docking is developed. To achieve adaptive control capability, a neural network based on RBF is used. The simulator provides a safe and affordable, way to test the neural network based control unit before it is used in real operations.

A physical model suitable for low-speed underwater vehicle is developed to generate training data and test the control result of the neural network. Although it is not essential to have an extremely precise model for the purpose of this simulator, most major hydrodynamic effects have been included in the physical model. Neural network learning of control strategies, then, takes place first in this artificial "world" and, then, from in-situ real data.

REFERENCES

[1] N.D.P. Barltrop, "Dynamics of fixed marine structures", London, Butterworth Heinemom, 1991.
[2] J. P. Hooft, "Advanced dynamics of marine structures", New York, J. Wiley, 1982.
[3] T. M. Cover, "Geometric and statistical properties of systems of linear inequalities with applications in pattern recognition", *IEEE Trans. on Electronic Computers,* EC-14, pp.326-334, 1990.
[4] S. Renals, R. Rohwer, "Phoneme classification experiments using radial basis functions," *Proc. of IJCNN,* 1990,1.461.