IFAC Publications
www.elsevier.com/locate/ifac

A SURVEY OF AI TECHNIQUES FOR CONTROL OF UNDERWATER VEHICLES

Pepijn van de Ven, Colin Flanagan and Daniel Toal

Control and Instrumentation Systems Group
Department of Electronic and Computer Engineering
University of Limerick, Ireland

Abstract:
This paper presents a review of recent research efforts in the field of the application of neural and fuzzy networks in the control of unmanned underwater vehicles (UUV). A classification for UUV control architectures using AI techniques is presented and consecutively used to categorise the approaches found in the literature. Several projects are discussed in detail and each control strategy is categorized as per the presented framework. Based on practical results from those projects, as reported in the literature, a qualitative assessment regarding the performance of the control strategies is given. Their advantages and disadvantages are identified and discussed. Based on the authors' observations, possible future trends are identified.

Keywords: artificial intelligence, attitude control, autonomous control, autonomous vehicles, fuzzy control, marine systems, model based control, neural control, neural networks, nonlinear control

1. INTRODUCTION

Due to the ever-increasing interest in underwater exploration, both for scientific and commercial benefit, considerable research effort in the area of underwater vehicle control can be noticed. With a shift in the focus of UUV research and development towards autonomous underwater vehicles (AUV), there has been a significant increase in the application of neural and fuzzy networks for control purposes in the last decade (Lorentz and Yuh, 1996). Due to the highly non-linear and dynamic characteristics of the underwater world control of UUVs is a far from trivial problem. Because of their flexibility and aptitude for dealing with non-linear problems, neural networks and fuzzy logic are expected to prove beneficial when used on autonomous craft. In this article, research endeavours focussed on the use of neural and fuzzy networks for the (semi-) autonomous control of underwater craft are examined. As there are several ways to incorporate neural and fuzzy networks into the control of an autonomous craft, a framework for categorising control strategies is defined. Using the presented theory to categorise the control approaches, recent research efforts in this field are reviewed. Several projects will be discussed and their control strategies will be categorized according to the presented framework. From the several examples characteristics of the three categories will be distilled and the advantages and disadvantages discussed.

2. CONTROL STRATEGIES

The use of neural networks (NNs) in control is versatile. As NNs are a good means of approximating non-linear functions, they can e.g. replace the conventional system identifier in model based control schemes. They can also control the dynamics of the craft directly by learning the inverse dynamics of the model. Another possibility is a NN learning the forward dynamics of the craft in order to consecutively train another NN that will act as a direct controller. As the craft is a highly non-linear system (Akkizidis and Roberts, 1998; Li *et al.*, 2002), a NN with at least one hidden layer is normally applied. It is thus capable of solving the non-linear classification problem. (Haykin, 1999, Ch. 4) Because of their ability to implement human expert knowledge, fuzzy controllers are often used in combination with NNs. Fuzzy logic is known to be highly useful in translating human expertise into a set of rules (L. H. Tsoukalas, 1997, Ch. 5) and thus may provide a controller with, although limited, intuitive knowledge of the proper control action. The lack of flexibility in terms of adaptability in fuzzy logic systems is obviated by the introduction of NNs in the controller that give the total controller the necessary adaptability (Akkizidis and Roberts, 1998). To create some kind of order in the possible applications of NNs in the control of underwater vehicles, the authors propose a classification into three major control strategies: (i) *combined control and learning, CCL* (also known in the literature as direct control), (ii) *separate control and learning, SCL* (known as indirect control) and (iii) *augmented control, AC*.

2.1 Combined Control and Learning

In the literature a variety of articles can be found on both practical and theoretical studies applying NNs that combine the tasks of control and on-line learning. Reasons for the popularity of this approach are, inter alia, that it is often possible simply to replace the old controller by the NN controller, that the complexity of the NN controller is limited and that learning is performed with the latest data. It should be noted that, due to the fact that online learning is performed in between control actions as explained below, this strategy is relatively demanding in terms of computational capacity.

In combined control and learning (CCL) (see figure 1), the controller system is used for control of the craft and training at consecutive time moments (Akkizidis and Roberts, 1998; Farrell *et al.*, 1990; Guo *et al.*, 1995; Kim and Yuh, 2001; Labonte, 2002; Seube, 1991; Venugopal *et al.*, 1992; Wang *et al.*, 1999*a*; Wang *et al.*, 1999*b*; Wang *et al.*, 2000; Wang and Lee, 2002; Yuh, 1990; Yuh and Lakshmi, 1993; Yuh, 1994).

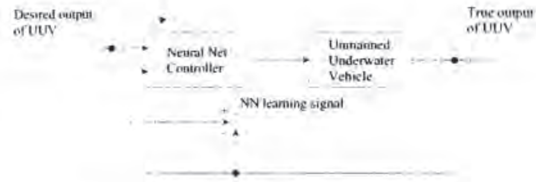

Fig. 1. Block level representation of combined control and learning

Operation of CCL is as follows. The desired system output is sent to the controller. The controller determines the necessary input to the system and the output of the system is monitored. Based on the actual output and the desired output of the system the NN is trained. The training cycle is performed in between (a series of) control actions. As training is performed online and continuously, the NN adapts to changing dynamics. Initial performance of the (untrained) controller can be improved by providing a simple conventional controller that is overruled once the NN is properly trained. Apart from performing initial control this controller can also act as a teacher(Guo *et al.*, 1995; Wang and Lee, 2002). Other architectures, mainly neuro-fuzzy controllers, use expert knowledge programmed into the fuzzy controller to perform initial control. Again learning is performed using the output of the system as an error signal(Akkizidis and Roberts, 1998; Kim and Yuh, 2001).

2.1.1. CCL: Example A In (Guo *et al.*, 1995) an architecture based on CCL is described. A two layer NN (2 inputs, 5 hidden nodes, 1 output node) controller is used to control motion in the horizontal plain while feedback of the control parameters is obtained from a compass. Back propagation is used to update the NN in two phases. In the first phase, the initialization phase, the NN is trained off-line with a linear controller as its automated teacher. After off-line training the obtained weights are used as initial weights in the controller. During actual deployment of the craft the weights are further updated using an objective function incorporating tracking error and control rate requirement.

The described controller is used to perform simulations and pool tests on a craft developed in the Department of Naval Architecture and Ocean Engineering of the National Taiwan University. The authors claim a good accordance between simulations and pool tests. Interesting to note is that the authors claim that the direct control strategy or Combined Control and Learning (CCL), is to be preferred over the indirect control strategy or Separate Control and Learning (SCL). As a reason the authors report that: "*Along the* *course of our study, we found that the network architecture of forward model for plant dynamics is very problem specific. It usually requires large network with recurrent connections.*" Due to the fact that, in CCL, learning is performed at a higher repetition frequency than in SCL, a feedforward network without recurrent connections can adapt itself to the changing dynamics. However, in SCL, the same forward model of the (possibly time-variant) dynamics is used for a relatively long period of time. As a result the network will have to represent the dynamic nature of the craft using e.g. recurrent connections leading to more sophisticated and more time consuming training algorithms.

2.1.2. CCL: Example B In (Wang and Lee, 2002) work on a NN control system for the ODIN (Omni-Directional Intelligent Navigator) (Yuh, 1994; Wang *et al.*, 1999*a*; Wang *et al.*, 1999*b*; Wang *et al.*, 2000; Kim and Yuh, 2001) is presented. The controller is based on the application of fuzzy basis functions (Wang and Mendel, 1992) as the antecedent of a fuzzy controller. Networks based on fuzzy basis functions can be seen as a special version of radial-basis functions making use of multivariate Gaussian functions(Haykin, 1999, Ch. 5). In the fuzzy basis functions however, the normal Gaussian function is replaced by a Gaussian membership function. As this fuzzy system can be described in a mathematical way as well as in a linguistic way, it is possible to determine or change the fuzzy basis functions either from linguistic rules or from in- and output data from the NN. Predecessors of this R-SANFIS (Recurrent Self-Adaptive Neuro-Fuzzy Inference System) are described in (Wang *et al.*, 1999*a*; Wang *et al.*, 1999*b*; Wang *et al.*, 2000; Kim and Yuh, 2001). Control of the craft is performed with the R-SANFIS controller in a feedforward loop while a PD (Proportional Derivative) controller is used in a feedback loop. Apart from minimizing the effect of disturbances, the PD controller output is also used as an error signal for updating the R-SANFIS parameters. Instead of one controller, six controllers are used for the forces and moments in six degrees of freedom to simplify the networks and thus speed up learning.

Several simulations were performed. The effect of learning on the performance is demonstrated in a simulation, comparing the performance of the R-SANFIS with and without online updating. The reported improvement in root mean square error (RMSE) of the desired and actual trajectory is about a factor of 2 to 3.5. In another simulation the R-SANFIS is compared to an adaptive controller as described in (Choi, 1995). Again the R-SANFIS with online learning shows a considerable improvement in RMSE of a factor of ≈ 8.8.

2.2 Separate Control and Learning

In contrast to Combined Control and Learning, Separate Control and Learning is a seldom-used strategy, probably due to the fact that this strategy requires a relatively complex architecture (see figure 2), which is illustrated by (Fujii and Ura, 1990; Ishii *et al.*, 1994; Ishii *et al.*, 1995; Ishii and Ura, 2000; Ura *et al.*, 1990). There are however several reasons to advocate this approach. Due to the separate loops for learning and control, learning does not have to be performed in the short time spans between control actions and is thus less computationally demanding. An additional effect is that more extensive, and possibly more appropriate, network architectures can be used. Recurrent networks for example, offer a means of representing time variant processes and are appropriate for filtering of noisy input data (Polycarpou and Ioannou, 1993). Also, the advantages of batch training can be exploited, as several input-output data points can be collected and used in one training cycle. When SCL is used, initially the craft is controlled by a conventional or fuzzy controller. Although this controller does only provide marginal control it does make sure that the craft shows a behaviour that is at least not harmful. During this period, which is sometimes referred to as the infancy of the craft, the NNs are trained. First, a NN learns to represent either the forward or inverse dynamics of the craft. If the forward dynamics are identified another NN will have to be trained with this forward model to obtain the desired control behaviour. In case the inverse dynamics are identified (a copy of) the NN can directly be used for control. Once the NN controller shows proper behaviour it takes over control fully. Learning is continued outside the control loop with a copy of this NN. At regular intervals the weights of the controlling NN are updated in order to account for changing parameters. One learning cycle can thus take up more than one control cycle. As a result there is no need for very fast processors. Alternatively more computationally demanding algorithms or network topologies (e.g. recurrent networks) can be used.

2.2.1. SCL: Example A A good example of this strategy is the controller called SONCS for Self-Organising Neural-net Control System (Fujii and Ura, 1990; Ishii *et al.*, 1994; Ishii *et al.*, 1995; Ishii and Ura, 2000; Ura *et al.*, 1990) that uses a so-called real-world and imaginary-world system. In the imaginary-world system, being a copy of the real-world system (which includes the craft dynamics), a controller is constantly trained using

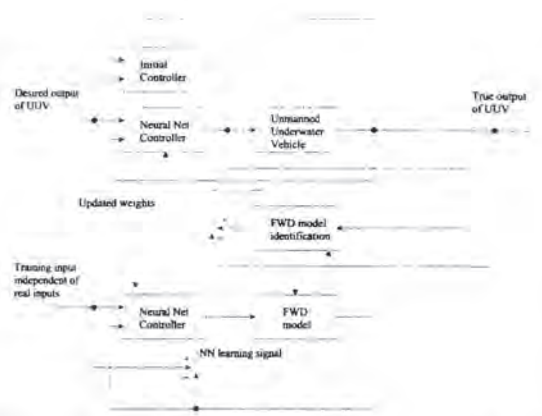

Fig. 2. Block level representation of separate control and learning

an online updated recurrent forward model of the craft dynamics. This model is regularly updated using inputs and outputs from the actual craft. At certain intervals the real-world controller is updated with the learned imaginary-world controller weights. Learning is thus performed in a loop totally independent of the control loop. To control the craft during its infancy a fuzzy controller is used that implements some rudimentary behaviours and is overruled once the NN controller is trained.

Figure 2 shows the architecture with the real-world and imaginary-world controllers. Back propagation is used to train the forward model and the imaginary world controller respectively. Tests are performed using the Twin-Burger test bed (Fujii *et al.*, 1993). According to the authors: "*The experiment shows that the robot is properly controlled to follow the target path, in spite of the existence of the current, and the control system has good performance... It can be concluded that the proposed control system shows good adaptability against the changes in the dynamic properties of the controlled object and its surroundings.*"

2.3 Augmented Control

Augmented control is based on enhancing conventional controllers that do not have the ability to adapt for changing parameters. A NN is added to the controller, often placed in parallel, and adds to the total control action to counteract the influence of unmodelled or poorly modelled dynamics, disturbances and other uncertainties. The most apparent advantage of Augmented Control is that it is normally possible to use the old architecture and place the new neural controller in parallel with the old controller (Campa *et al.*, 2000) or use it as, e.g., a feed forward controller. Again the NN can be trained outside the control loop and from this perspective there is thus no need for very fast processors. Although most examples of augmented control use feed forward networks combined with back propagation algorithms (Li *et al.*, 2002; Yamamoto, 1995; Pollini *et al.*, 1997), there are examples of architectures that employ other topologies such as recurrent networks (Kodogiannis *et al.*, 1996).

2.3.1. AC: Example A In (Li *et al.*, 2002) a two layer NN is used to augment a linear feedback controller. While a PD controller controls the linear part of the dynamics, the NN is designed to control the nonlinear uncertainties of the craft. This controller is implemented as shown in figure 3.

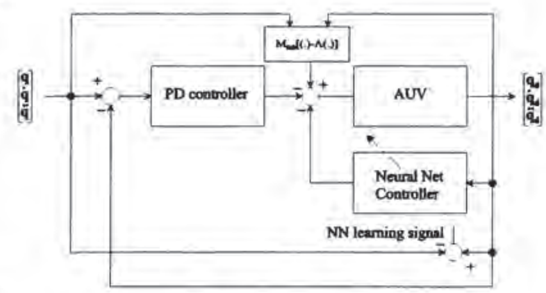

Fig. 3. Principal of operation of the augmented controller

Simulations with this controller and a model of the SAUV (Hong, 2000) (Semi-Autonomous Underwater Vehicle) developed in the Korea Research Institute of Ships & Ocean Engineering (KRISO) were undertaken. In the simulations the performance of the augmented controller was compared to the performance of a controller consisting of linear feedback control combined with a sliding mode controller. The simulation results show that the augmented controller approximates the nonlinear uncertainties in about ten seconds. Comparison with the linear feedback / sliding mode controller shows that after the initial ten seconds, the augmented controller shows far better tracking than the conventional controller. No numerical results are reported.

2.3.2. AC: Example B In (Kodogiannis *et al.*, 1996) a model predictive control system is augmented by using a neural network as the model. In this article control of the depth is performed on a craft with one degree of freedom. Future work will focus on control of a craft in six degrees of freedom. The control architecture is depicted in figure 4. In model predictive control (MPC) with NNs the latter are used as a forward model, predicting the future output of the system. The control action is then chosen such that it minimizes the difference between the predicted output and the desired output. Although in the control of the craft only one future estimate of the system output is used to determine the proper control action, the model was designed to predict five consecutive

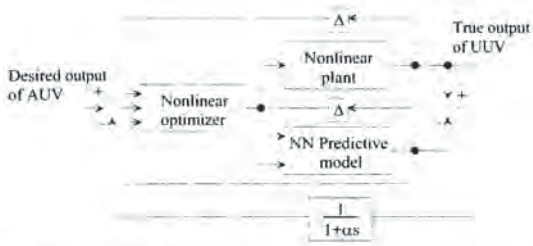

Fig. 4. Block schematic of the model predictive control scheme

future outputs. Two different NNs are tested for use as the forward model: a modified Elman network (Haykin, 1999, Ch. 15) and a newly proposed architecture called Autoregressive Recurrent NN (ARNN). Both networks use recurrent connections to capture time-variant information of the system.

Pool tests, in which the MPC scheme controlled a craft called the 'Aquacube', show that both NNs result in a good tracking of the desired path. For the ARNN the mean square error in prediction of the system output is 0.000516 while the mean squared tracking error is 0.0523. The Elman network performs slightly less with a mean squared error in system output prediction of 0.000748 and a mean squared tracking error of 0.0555. The authors claim that: "*The improvements obtained with recurrent networks are due to the fact that a minimum control effort was used to achieve the specific performance.*"

3. CONCLUSIONS

The presented examples of the various ways to use NN controllers for AUVs show that it is not trivial to point out the most promising architecture. Apart from the used strategy (CCL, SCL or AC) the chosen NN architecture, learning algorithms and test bed highly influence the obtained results. However some general characteristics have been found for the defined classes. Table 1 gives an overview of advantages and disadvantages encountered. As can be seen from table 1, SCL would generally be a more elaborate, but also (in terms of hardware) a more demanding approach. As it allows for elaborate NN architectures, SCL offers a good test bed for research focussing on the application of exotic structures and research endeavours to model the craft using NNs. CCL on the other hand, does need fast processors, but the NN itself is restricted to relatively simple networks. As a result this strategy would be more interesting for commercial applications in which one is not per se interested in good forward models of the craft. The same holds for AC. As it uses an existing controller it is probably the cheapest and quickest way to build a NN controller. One should however question how appropriate the original

Table 1. Advantages and disadvantages of the three approaches

CCL

\+ Simple architecture
Learning with newest data/circumstances
No FWD model necessary
Direct control is less demanding (Guo *et al.*, 1995)
Old controller can often be replaced

\- Computationally demanding

SCL

\+ Computationally less demanding
More appropriate NN architectures possible
More appropriate training algorithms possible
Batch training possible

\- Complex architecture
Controller is trained with model of the past

AC

\+ Simple implementation
Old architecture can be used
Computationally less demanding

\- Conventional controller might be totally inadequate if dynamics change drastically

conventional controller is for the underwater world and whether a high enough degree of robustness can be obtained. It is expected that CCL and SCL will be the prevalent strategies for future AUV control. CCL being an obvious choice for commercial AUVs, SCL mainly being interesting for (NN control) research studies.

4. ACKNOWLEDGEMENT

This work was supported by the Irish Research Council for Science, Engineering and Technology: funded by the National Development Plan.

REFERENCES

Akkizidis, I.S. and G.N. Roberts (1998). Fuzzy modelling and fuzzy-neuro motion control of an autonomous underwater robot. *5th International Workshop on Advanced Motion Control* pp. 641–46.

Campa, G., M. Sharma, A.J. Calise and M. Innocenti (2000). Neural network augmentation of linear controllers with application to underwater vehicles. *Proc. of the 2000 American Control Conference* **1**, 75–79.

Choi, S.K. (1995). An adaptive-learning control system for underwater robotic vehicles. *Ph.D. thesis. Department of Mechanical Engineering, Univ. of Hawaii.*

Farrell, J., B. Goldenthal and K. Govindarajan (1990). Connectionist learning control systems: submarine depth control. *Proc. of the 29th IEEE Conference on Decision and Control* **4**, 2362–7.

Fujii, T. and T. Ura (1990). Development of motion control system for auv using neural nets. *Proceedings of the (1990) Symposium on Autonomous Underwater Vehicle Technology* pp. 81–6.

Fujii, T., T. Ura, Y. Kuroda, H. Chiba, Y. Nose and K. Aramaki (1993). Development of a versatile test-bed "twin-burger" toward realization of intelligent behaviors of autonomous underwater vehicles. *Proc. of IEEE OCEANS '93* pp. 186–91.

Guo, J., F.C. Chiu and Chieh-Chih Wang (1995). Adaptive control of an autonomous underwater vehicle testbed using neural networks. *Proc. OCEANS '95* **2**, 1033–9.

Haykin, S. (1999). *Neural Networks. A comprehensive foundation.* 2 ed.. Prentice Hall. Upper Saddle River, New Jersey 07458.

Hong, S.W. et al. (2000). Development of technologies for navigation and manipulator system of a semi-autonomous underwater vehicle.

Ishii, K. and T. Ura (2000). An adaptive neural-net controller system for an underwater vehicle. *Control Engineering Practice* **8**, 177–84.

Ishii, K., T. Fujii and T. Ura (1995). An on-line adaptation method in a neural network based control system for auvs. *IEEE Journal of Oceanic Engineering* **2(3)**, 221–8.

Ishii, K., T. Ura and T. Fujii (1994). A feedforward neural network for identification and adaptive control of autonomous underwater vehicles. *1994 IEEE International Conference on Neural Networks* **5**, 3216–21.

Kim, T.W. and J. Yuh (2001). A novel neuro-fuzzy controller for autonomous underwater vehicles. *Proc. of the 2001 International conference on Robotics & Automation* pp. 2350–5.

Kodogiannis, P.J., G. Lisboa and J. Lucas (1996). Neural network modelling and control for underwater vehicles. *Artificial Intelligence in Engineering 1* pp. 203–12.

L. H. Tsoukalas, R.E. Uhrig (1997). *Fuzzy and Neural Approaches in Engineering.* John Wiley & Sons. New York.

Labonte, G. (2002). Fast adaptive control of a non-linear system by an adaline: motion in a fluid. *Proc. of the 2002 International Joint Conference on Neural Networks* **2**, 1837–41.

Li, Ji-Hong, Pan-Mook Lee and Sang-Jeong Lee (2002). Neural net based nonlinear adaptive control for autonomous underwater vehicles. *Proc. of the 2002 IEEE International Conference on Robotics and Automation* **2**, 1075–80.

Lorentz, J. and J. Yuh (1996). A survey and experimental study of neural network auv control. *Proc. of the 1996 Symposium on Autonomous Underwater Vehicle Technology* pp. 109–16.

Pollini, L., M. Innocenti and F. Nasuti (1997). Robust feedback linearization with neural network for underwater vehicle control. *Proc. of Oceans '97. MTS/IEEE* **11**, 12–16.

Polycarpou, M. M. and P.A. Ioannou (1993). Stable nonlinear system identification using neural network models. *Neural Networks in Robotics* pp. 165–77.

Seube, N. (1991). Neural network learning rules for control: application to auv tracking control. *Proc. of the 1991 IEEE Conference on Neural Networks for Ocean Engineering* pp. 185–96.

Ura, T., T. Fujii, Y. Nose and Y. Kuroda (1990). Self-organizing control system for underwater vehicles. *Proc. of OCEANS '90* pp. 76–81.

Venugopal, K.P., R. Sudhakar and A.S. Pandya (1992). On-line learning control of autonomous underwater vehicles using feedforward neural networks. *IEEE Journal of Oceanic Engineering* **17**, 308–319.

Wang, J.S. and C.S.G. Lee (2002). Self-adaptive recurrent neuro-fuzzy control for an autonomous underwater vehicles. *Proc. of the 2002 IEEE International conference on Robotics & Automation* pp. 1095–1100.

Wang, J.S., C.S.G. Lee and J. Yuh (1999*a*). An on-line self-organizing neuro-fuzzy control for autonomous underwater vehicles. *Proc. of the 1999 International Conference on Robotics & Automation* pp. 2416–21.

Wang, J.S., C.S.G. Lee and J. Yuh (1999*b*). Self-adaptive neuro-fuzzy control with fuzzy basis function network for autonomous underwater vehicles. *Proc. of the 1999 IEEE/RSJ Internationational Conference on Intelligent Robots and Systems* pp. 130–5.

Wang, J.S., C.S.G. Lee and J. Yuh (2000). Self-adaptive neuro-fuzzy systems with fast parameter learning for autonomous underwater vehicles. *Proc. of the 2000 IEEE International conference on Robotics & Automation* pp. 3861–6.

Wang, L. and J.M. Mendel (1992). Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks* **3, no. 5**, 807–14.

Yamamoto, I. (1995). Application of neural network to marine vehicle. *Proc. of the 1995 IEEE International Conference on Neural Networks* **1**, 220–4.

Yuh, J. (1990). A neural net controller for underwater robotic vehicles. *IEEE Journal of Oceanic Engineering* **15(3)**, 161–6.

Yuh, J. (1994). Learning control for underwater robotic vehicles. *IEEE Control Systems Magazine* **14(2)**, 39–46.

Yuh, J. and R. Lakshmi (1993). An intelligent control system for remotely operated vehicles. *IEEE Journal of Oceanic Engineering* **18(1)**, 55–62.