

An Application of Neural Networks to the Guidance of Free-Swimming Submersibles

D.P. Porcino

J.S. Collins

Engineering Dept., Royal Roads Military College, FMO Victoria B.C. Canada VOS 1B0

Published in Proceedings of the International Joint Conference on Neural Networks,
January, 1990, Washington, D.C. pp. II-417 – II-420

ABSTRACT

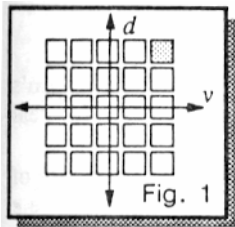
Most neural network models use *learning with a teacher* to train a great many simple processing elements. However the Adaptive Search Element/Adaptive Critical Element (ASE/ACE) model developed by Barto, Sutton, and Anderson, and elaborated on in this paper, is comprised of just two relatively complex units using *learning with a critic*. The ASE controls the physical system, and the ACE criticizes the performance of the ASE in an effort to accelerate learning. In this paper, the basic model is extended to the realm of complex numbers, and the output is time averaged, yielding a system that has a continuously valued control output. The system learns quite quickly.

In the simplest organisms possessing nervous nets, mechanisms exist to control reflex behaviours. In more complex organisms, these reflexes become hidden by the sophistication afforded by a more complex nervous system, but, they still exist. (Blackburn & Nguyen, 1988). In light of these facts, we will examine a most basic reflex behaviour: a reflex that keeps a submersible robot a certain distance above the ocean floor.

INTRODUCTION

A.H. Klopff suggested in 1982 that neurons could learn using operant conditioning: they would learn to attain certain states while avoiding others. In the field of neuronal modelling therefore, the principles of behavioural psychology provide a way to train unsupervised systems.

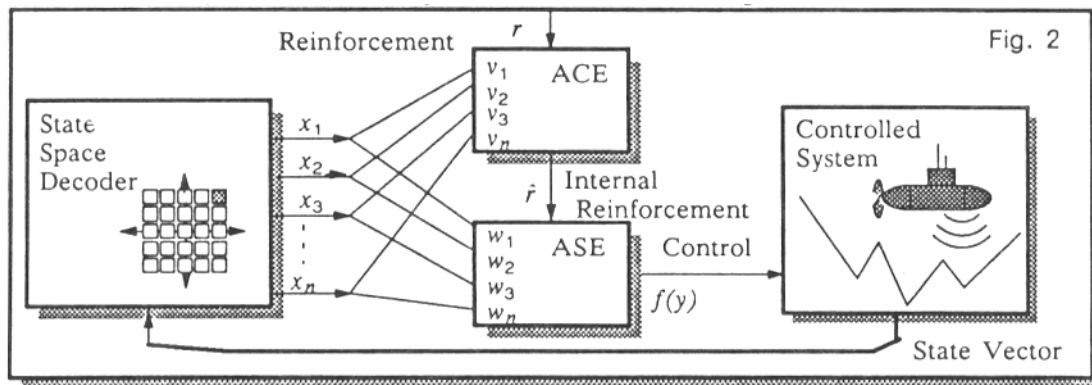
The basis- for the ASE and the ACE is the "boxes" system of D. Michie and R.A. Chambers (1968). State space is partitioned into "boxes," each with its own resident "demon" which non-deterministically chooses a control action appropriate to that state. All the demons together form a metaphor for the input synapses of a single neuron.



The state space diagram in Figure I shows displacement versus vertical velocity partitioned into boxes. The shaded box shows velocity and displacement both large and positive. The demon within this box must choose an action that will be most likely to prevent a failure from happening.

Learning - adaptation - is based on an estimate of how long the system would continue to operate after particular actions. When the system fails, the individual estimates in each box must be updated to make another similar failure less likely.

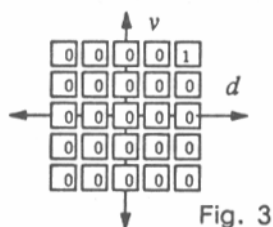
Figure 2 shows the simulation's ASE/ACE controlling a submersible robot equipped with a thruster for forward motion, and up and down facing thrusters for maneuvering:



The only information available to the robot is the return from its sonar, which gives the distance to the ocean floor (actually the displacement from the desired distance above the bottom), and a failure signal (reinforcement), supplied whenever the robot exceeds the allowable vertical error. The robot doesn't know what criterion generates the failure signal, and must work through trial and error to avoid the punishing failure signal.

DESCRIPTION OF THEORY AND APPLICATION - THE ADAPTIVE SEARCH ELEMENT

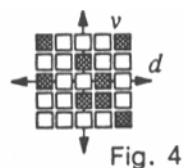
The state space decoder divides the state vector up - and generates therefore a unique and finite set of combinations. The element whose quantization parameters best match the state vector gives an output of one, and every other element outputs zero.



As in Figure 1, Figure 3 shows a state with a large positive velocity and displacement.

Upon failure, every element of the decoder will be set to zero. This is an important condition for proper operation of the equations.

The output of the decoder, x , is passed to the ASE, each element being connected to a particular synapse, having a weight of w .



This diagram shows how the synaptic weights, w , might appear before much learning has taken place. Dark squares represent negative weights - or a high likelihood that system will decide to thrust up, and light squares represent positive weights - the probability of thrusting down.

The original model of Barto, Sutton, and Anderson used real synaptic weights to generate a binary output. In this version, complex weights consisting of a real and an imaginary part were used to allow a four-valued output. A

binary output allows up and down thrust - the complex four-valued output allows the thrusters to be shut off as well.

The following equation describes the output of the ASE, $y(t)$, at time t :

$$y(t) = f \left[\sum_{i=1}^n w_i(t)x_i(t) + noise(t) \right] \quad (1)$$

The addition of a *noise* signal having a zero-centered Gaussian distribution causes the ASE's output to be governed by chance. The noise is important during early training to ensure that the system fully explores the state space. The noise should be *gradually* shut off to fine tune the weights, and *completely* shut off after training is complete to ensure consistent and repeatable behaviour.

$$f(x) = \begin{cases} +1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases} \quad (2a)$$

$$f(x) = \begin{cases} +1, & \text{if } x_r > 0 \text{ and } x_i > 0 \\ -1, & \text{if } x_r < 0 \text{ and } x_i < 0 \\ 0, & \text{otherwise} \end{cases} \quad (2b)$$

The output of the ASE is passed through a thresholding function f . This quantizer provides the system's nonlinearity - utterly necessary for any computation. Equation (2a) describes the quantizer for real numbers, and (2b) shows the quantizer extended to use complex numbers.

The controller's most recent choices were most likely responsible for the current success or failure of the system, and therefore the most *eligible* to have their weights changed. A complex eligibility trace e decays at a rate governed by δ , ($0 \leq \delta \leq 1$). The inclusion of the output of the ASE, term $y(t)$, means that eligibility will be positive decaying to zero if the ASE's output was positive, and negative decaying to zero otherwise.

$$e_i(t+1) = \delta e_i(t) + (1 - \delta)y(t)x_i(t) \quad (3)$$

$$w_i(t+1) = w_i(t) + ar(t)e_i(t) \quad (4)$$

- (3) The eligibility remembers how long ago a choice was made at a particular synapse, and what that choice was.
 (4) This equation shows how weights change through time.

As shown in (4), the system learns through operant conditioning - setting the reinforcement signal r to -1 is an aversive stimulus - punishment. A positive value for reinforcement would be reward, or positive reinforcement. (The reinforcement signal as supplied to the ASE in our complex-valued model is real, but internally it is treated as $r+ri$.)

As long as the system is functioning properly, the reinforcement signal is held at zero, and *the weights do not change*. On the other hand, if a failure occurs, the negative value of the reinforcement will penalize the most recent decisions (highest eligibility synapses) that undoubtedly led to the failure, making those decisions less likely to occur in the future.

DESCRIPTION OF THEORY AND APPLICATION - THE ADAPTIVE CRITICAL ELEMENT

When failure finally occurs, one particular sequence of decisions would probably have been better than the sequence that occurred. Based on that, the system learns. Unfortunately, this learning only occurs upon failure - which is to say, infrequently. The ACE is added to provide a continuous feedback signal to the ASE. Reinforcement now arrives all the time - for good choices as well as for bad - so learning progresses at a much quicker rate.

The ACE receives the same decoder signal as the ASE, and also has a memory trace for every box, like the ASE. Rather than storing a best decision probability in each box, the ACE stores a prediction, $v(t)$, of the reinforcement

the environment will provide for a particular state. Each synapse therefore contains a rating of that state's dangerousness - synapses corresponding to dangerous states will contain a strong prediction of failure.

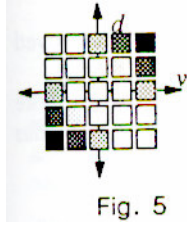


Figure 5 shows what the ACE eventually learns about the submersible control problem - if displacement and velocity are both large and of the same sign, failure will soon occur. The white squares show where no reinforcement arrives from the environment, and the darker squares show that a failure signal is likely to occur very soon.

This equation shows how the ACE makes its prediction:

$$p(t) = \sum_{i=1}^n v_i(t)x_i(t) \quad (5)$$

Where: v holds the ACE's internal weights, and x is the singleton output of the decoder. At failure the ACE's prediction is zero, as every decoder element will be zero.

The ACE sends the ASE the internal reinforcement signal:

$$\hat{r}(t) = r(t) + \gamma p(t) - p(t-1) \quad (6)$$

Upon failure, the output of the decoder, and consequently the ACE, $p(t)$, will be zero, but the external reinforcement, r , will be -1. (Usually r is zero.) The internal reinforcement will be the difference between the previous reinforcement signal, $p(t-1)$, and r . A fully predicted failure causes no reinforcement. Incorrect actions leading up to the failure will be punished in accordance with how much the ACE's eligibility traces have decayed.

The predictions of failure will be reinforced, because they were accurate predictions.

In the absence of external signals, the discount term, γ , causes the internal signals to gradually die away. The internal reinforcement is the difference between the discounted predicted reinforcement $\gamma p(t)$, and the previously predicted reinforcement, $p(t-1)$. Understanding (6)'s difference calculation is the key to understanding the whole operation:

If the system moves from a low danger state to a high danger state, the internal reinforcement will be negative, punishing the system. If, on the other hand, the system moves from a high danger state to a lower danger state, internal reinforcement will be positive: reward. From this information, the system learns to avoid dangerous states.

The equations governing learning in the ACE and the ASE are very similar:

$$v_i(t+1) = v_i(t) + \beta[r(t) + \gamma p(t) - p(t-1)]\bar{X}_i(t) \quad (7)$$

Notice that the term inside the square brackets is the same as the internal reinforcement signal r in (6). In form and substance, (7) is almost the same as the ASE's weight change (4), except for the reinforcement term, and the form of the eligibility, in this case, $X_i(t)$. The ACE's eligibility trace equation is almost the same as the ASE's (3).

$$\bar{X}_i(t+1) = \lambda \bar{X}_i(t) + (1-\lambda)x_i(t) \quad (8)$$

THE SUBMERSIBLE MODEL AND SIMULATION RESULTS:

Simple equations describe the simulated physical system's kinetics. Friction was ignored, and the submersible was given a unit mass and neutral buoyancy. The horizontal component of the robot's velocity is held constant. State space is quantized into 64 boxes, as finer partitioning offers no particular advantage. Partitioning is exponential, giving finest control near the equilibrium point. If the system state moves into one of the extreme displacement partitions, a crash or surfacing (failure) has occurred, and the trial is terminated.

The ASE/ACE parameters were set as in Barto et al 1982, as these were quite reasonable. Real parameters were changed to complex with the imaginary and real components equal to each other. It is the interdependencies of the two components of an imaginary number as it undergoes manipulation that makes the solution robust and useful.

The vertical thrusters (governed by the ASE's output) yield thrusts of -1, 0, or +1. The solution achieved by the system is quite interesting: The robot drifts back and forth within the safe corridor, thrusting only when necessary to stay within it.

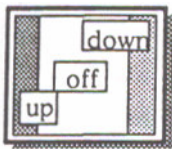


Fig. 6

The solution was refined by adding continuous control of the output through time averaging over four or five steps. The weights generated are more difficult to understand than those shown in Figure 6, as the solution is characterized by short, graduated thrusts which quickly return the system to an almost zero vertical velocity. Unlike the previous solution that had clear-cut boundaries surrounding the decision regions, the time average solution has a scattering of different choices associated with each synapse. When averaged with neighbouring synapses, a graduated thrust results.

For use in practical situations, a successful solution should be *frozen* beforehand and the ACE removed so that the behaviour of the system is completely predictable in the field.

The ASE/ACE system learns quickly and generalizes readily to varying terrain. It was able to take advantage of the time averaging of the output. Figure 7 shows a number of trial runs. Truncated lines show where failures occurred.

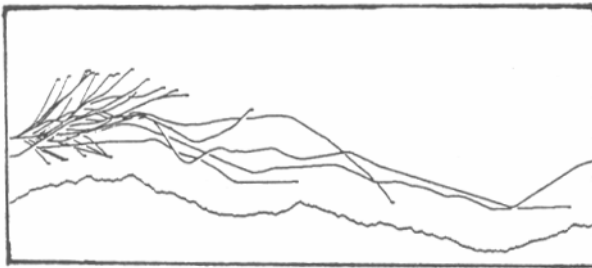


Fig. 7a. The earliest stages of training. The system makes many mistakes at first, but soon learns to negotiate the terrain.

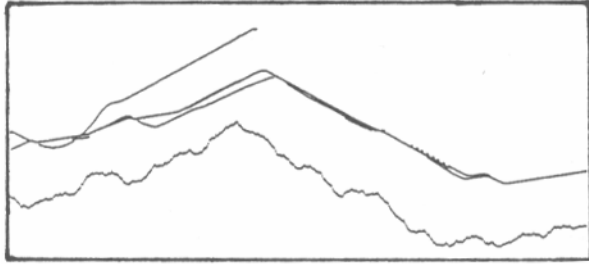


Fig. 7b. The next time interval, with new terrain supplied. The system fails once, and then successfully negotiates the terrain.

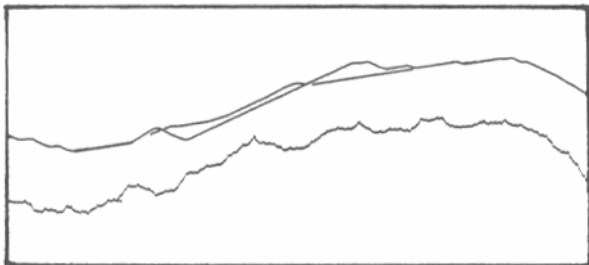


Fig. 7c. The terrain is changed again, and it is apparent that the system has generalized its solution, and can function well over any terrain.

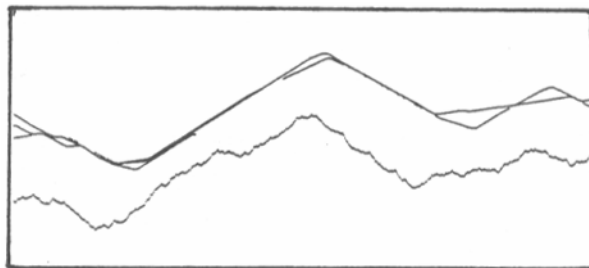


Fig. 7d. Much later in the training. The system has no problems with arbitrary terrain. Performance is qualitatively similar over all terrains, the solution is robust.

REFERENCES

Barto A.G., R.S.Sutton, & C.W. Anderson. **Neuronlike adaptive elements that can solve difficult learning control problems.** IEEE Transactions SMC, 1982. SMC-13: 834-846

Blackburn M.R., & H.G. Nguyen. **An Artificial Neural Network for Autonomous Undersea Vehicles.** Naval Ocean Systems Center, San Diego California. Technical document 1318, July 1988.

Klopf, A.H. **The Hedonistic Neuron: A Theory of Memory, Learning, and Intelligence.** Hemisphere Press, Washington DC, 1982.

Michie D., & R.A. Chambers. **BOXES: An experiment in adaptive control,** in *Machine Intelligence 2*, E. Dale and D. Michie, Eds. Oliver and Boyd, Edinburgh. pp. 137-152 (1968)