REINFORCEMENT LEARNING FOR A VISUALLY-GUIDED AUTONOMOUS UNDERWATER VEHICLE

David Wettergreen, Chris Gaskett, and Alex Zelinsky
Robotic Systems Laboratory
Department of Systems Engineering, RSISE
Australian National University
Canberra, ACT 0200 Australia
Phone: +61-2-6279-8686; Fax: +61-2-6279-8688
[dsw | cg | alex] @syseng.anu.edu.au

Abstract

Reinforcement learning uses a scalar reward signal and much interaction with the environment to form a policy of correct behavior. We have applied this technique to the problem of developing a controller for an autonomous underwater vehicle and have achieved reliable off-line development of stable controllers.

Many important underwater tasks rely upon on visual observation of underwater features. We have devised a feature tracking method and a vehicle guidance scheme that are also based on visual observation of features. We have obtained results in reliably tracking features in underwater imagery, not for map building but to guide an underwater vehicle. Using visual servo control techniques, feature position can be used directly to guide motion.

Introduction

At the Australian National University we are developing an autonomous underwater vehicle for tasks in exploration and inspection. Our objectives are to enable submersible robots to autonomously search in regular patterns, to follow along fixed natural and artificial features, and ultimately, to swim after dynamic targets. These capabilities are essential to tasks like cataloging reefs, exploring geologic features, and studying marine creatures, as well as inspecting pipes and cables, and assisting divers.

There are many approaches to the problem of motion control for underwater vehicles, ranging from traditional control to modern control [1][2] to a variety of neural network-based architectures [3]. Most existing systems control limited degrees-of-freedom, for example yaw and surge, and assume motion along other dimensions can be controlled independently. The implementation of these controllers usually requires a dynamic model of the vehicle and a number of simplifying assumptions which may limit its operating regime and/or robustness. The modeling process is expensive, sensitive, and unsatisfactory.

We have sought an alternative. We are developing a method by which an autonomous underwater vehicle (AUV) learns to control its behavior directly from experience of its actions in the world. We start with no explicit model of the vehicle or of the effect that any action may produce. Our approach is a connectionist (artificial neural network) implementation of model-free reinforcement learning. The AUV learns in response to a reward signal, attempting to maximize its total reward over time. Our approach differs from supervisory learning approaches which require that the “correct” control be known (from pre-existing models) so that it can be used to train the neural network. Another consideration that we address is that unlike most learning systems, which consider discrete states, control of an AUV requires evaluation of continuous input signals and, for smooth control, requires continuous output signals.

Controlling its own motion is part of the AUV's challenge, another is determining where to go. We are developing the use of visual information, not to build maps to navigate, but for visual servo control.[4] We have implemented techniques for area-based correlation to track feature from frame to frame and estimate range by matching between stereo pairs. A mobile robot can track features and use their motion to guide itself. Simple behaviors regulate position and velocity relative to tracked features in order to provide control inputs and immediate reinforcement to the learning system. In this manner we intend our AUV to hold station on a reef, swim along a pipe, or perform a repeatable search of the sea floor.

Underwater Vehicle Control

Various approaches to the problem of motion control for underwater vehicle have been proposed, and although many working systems exist, there is still a need to improve their performance and to adapt them to new vehicles,

tasks, and environments. Most existing systems control limited degrees-of-freedom, for example yaw and surge, and assume motion along dimensions can be controlled independently. The implementation of these controllers usually requires a dynamic model of the vehicle and a number of simplifying assumptions which may limit their operating regime and/or robustness. Movement between two points is typically considered a navigation problem, separate from the control problem.

Traditional methods of control for vehicle systems proceed from dynamic modelling to the design of a feedback control law that produces control inputs to compensate for deviation from the desired motion. This is predicated on the assumption that the system can be well-modelled and that specific desired motions can be determined.

Non-traditional, specifically connectionist (artificial neural network), approaches to motion control, can avoid much of the modelling difficulty. Instead, networks are constructed without any model of system dynamics. An appropriate controller is developed through training; appropriate actions to move the vehicle along the desired path slowly emerge. Control of low-level actuators as well as high-level navigation can potentially be incorporated in one neurocontroller.

Control of AUVs

Small, slow-moving underwater vehicles present a particularly challenging control problem. The dynamics of such vehicles are nonlinear because of inertial, buoyancy and hydrodynamic effects. Linear approximations are insufficient and nonlinear modelling and control techniques are needed to obtain high performance.[5]

Nonlinear models of underwater vehicles have many coefficients that must be identified. Some model parameters remain unknown either because they are unobservable or because they vary with un-modelled conditions. To date, most controllers are developed off-line and only with considerable effort and expense are applied to a specific vehicle with restrictions on its operating regime.[6]

Considerable effort has been made in recent years to developing accurate models of thrusters.[7][8][9] This is because thrusters are a dominant source of nonlinearity in vehicle motion.[7] Every thruster is different either in design or, among similar types, due to tolerances and wear, so parameter identification must be undertaken for each one. With thrusters precisely modeled, the task of coordinating all the thrusters is built upon the dynamic model of the whole system which takes to form: $M(\dot{\boldsymbol{v}}) + C(\boldsymbol{v})\boldsymbol{v} + D(\boldsymbol{v})\boldsymbol{v} + g(\boldsymbol{x}) = \tau$ where M is the inertia matrix including added mass, C is the matrix of Coriolis and centripetal terms, D is the hydrodynamic damping and lift matrix, g is the vector of restoring forces and moments, and $\boldsymbol{\tau}$ is the vector of control inputs; $\boldsymbol{x}$ is the vector of positions and orientations and $\boldsymbol{v}$ is the vector of linear and angular velocities.[5]

Yoerger and Slotine proposed a series of single-input/single-output continuous-time controllers by using sliding mode techniques and demonstrated the robustness of these systems in the presence of uncertainties.[1] Sliding mode techniques enable stable control of the system over a wide operating regime, as required for an AUV. Another advantage is that adaptation can be incorporated to modify the control law as it reaches the limits of its operating regime. Cristi proposed an adaptive sliding mode controller based on a primary linear model and bounds on nonlinear disturbances.[2] Refinements to sliding mode controllers continue to produce one or two-dimensional controllers.[10][11][12] General, full degree-of-freedom solutions to control of freely moving underwater vehicles remain elusive.

Neurocontrol of AUVs

Control using artificial neural networks, neurocontrol, [13] offers a promising method of designing a nonlinear controller with less reliance on developing accurate dynamic models. Controllers implemented as neural networks can be more flexible and are suitable for dealing with multi-variable problems.

Several different neural network based controllers for AUVs have been proposed. [14] Sanner and Akin [15] developed a pitch controller trained by back-propagation. Training of the controller was done off-line in with a fixed system model. Output error at the single output node was estimated by a critic equation based on the pitch error. Ishii, Fujii and Ura [16] developed a heading controller based on indirect inverse modelling. The model was implemented as a recursive neural network which was trained offline using data acquired by experimentation with the vehicle and then further training occurred on-line. Error at the output of the controller was estimated by propagating through the model to the single output node which drove the steering thrusters differentially. Yuh [14] has proposed several neural network based AUV controllers. Error at the output of the controller was based on a critic equation which uses an estimate of the upper bounds of the vehicle inertia matrix to assign error to individual outputs. The

controller learned in simulation. Venugopal [17] used a similar arrangement to Yuh except that a gain matrix was inserted between the controller and the system model. It reduced the reliance on known parameters of the vehicle but made assumptions about the interactions between various directions of motion.

The resulting performance of these controllers is promising. The ability to learn or at least refine the controller on-line in real time has been demonstrated [16], as has the ability to cope with changing system parameters [18]. These controllers all use a system model, whether it is fixed, learned or refined. It is possible to develop a neurocontroller in which the system model is not required at any stage. In situations where the required controller is less complex than the system model, this type of model-free approach may be the most appropriate.

Reinforcement Learning for Vehicle Control

In creating a control system for an AUV, our aim is for the vehicle to be able to achieve and maintain a goal state, for example station keeping or trajectory following, regardless of the complexities of its own dynamics or the disturbances it experiences. We are developing a method for model-free reinforcement learning with multiple continuous states and multiple continuous actions. The lack of an explicit a priori model makes the system adaptable and reduces reliance on knowledge of the system to be controlled. The system identifies both the structure and parameterization of an effective controller at once.

Reinforcement Learning

Reinforcement learning addresses the problem of forming a *policy* of correct behavior through observed interaction with the environment. [20] The general strategy is to use statistical techniques and dynamic programming methods to continuously refine an estimate of the utility of performing a specific action while in a specific state. The *value* of an action is the reward received for carrying out that action, plus a discounted sum of the rewards which are expected if optimal actions are carried out from all future states.

A distinguishing characteristic of reinforcement learning is that a single scalar reward is received for each action taken. The reward comes from a critic which evaluates ongoing progress (unlike supervised learning where the correct output is required at each step to train the network).

In the case of an AUV, an extended sequence of thruster commands is required to reach a goal. At any instant it is difficult to determine whether individual thrusters are behaving correctly. Only after a period of time can their collective performance be evaluated. The reward follows, often with some delay, an action or sequence of actions. Reward could be based on distance from a target, roll relative to vertical or any other measure of performance. The controller learns to choose actions which, over time, will give the greatest total reward.

The delay before reward leads to the *temporal* credit assignment problem, identifying which parts of a composite action caused the reward is the *structural* credit assignment problem. The scalar reward value on its own does not give enough information to determine what part of a composite action was beneficial, or what part of the state information was important for determining the choice of this action. Thus reinforcement learning systems require time for exploration of the state and action spaces.

Q-learning[21] is an implementation method for reinforcement learning in which a mapping is learned from a state-action pair to a value called *Q*. The mapping eventually represents the utility (in the long run) of performing an particular action from that state. The neurocontroller then measures the state, chooses the action which has the highest *Q* value and executes it. The *Q* function is updated according to the equation:

$$Q(\mathbf{x}, \mathbf{u}) = (1-\alpha)Q(\mathbf{x}, \mathbf{u}) + \alpha[R + \gamma \max_u Q(\mathbf{x}_{t+1}, \mathbf{u}_{t+1})]$$

where Q is the expected value of performing action **u** in state **x**; **x** is the state vector; **u** is the action vector; R is the reward; α is a learning rate and γ is the discount factor. Initially $Q(\mathbf{x},\mathbf{u})$ is strongly influenced of the immediate reward, R, of performing an action **u** given state **x** but, over time, it comes to reflect the long-term utility of the action.

Continuous States and Actions

Many real world control problems require actions of a continuous nature, in response to continuous state measurements. But most learning systems, indeed most classical AI techniques, are designed to operate in discrete (or symbolic) domains. High-performance control of mobile robots cannot be adequately carried out with coarsely coded inputs and outputs. Motor commands need to vary smoothly and accurately in response to continuous changes in state.

Systems in which state and action are discretized scale poorly, as the number of state and action variables increases the size of table required grows exponentially. Accurate control requires that variables be quantized finely, but as these systems may fail to generalize between similar states and actions they require large quantities of training data. Using a coarser representation of states leads to a state aliasing problem in which different world states appear to be the same. It is possible to avoid these discretization problems entirely by using learning methods which can deal with continuous states and actions.

Continuous State and Action *Q*-learning

Q-learning (and many other reinforcement learning algorithms) are normally considered in a discrete sense. This allows implementation in a simple lookup table. When states and actions are continuous, it is necessary to generalize between similar states and actions. To generalize between states, one approach is to use a neural network.[22] An interpolator can provide generalization between actions. [23] Partial derivatives of the interpolator can be used to back-propagate the error in the *Q*-value as an error signal to the neural network. Figure 1 shows the general structure of such a system.

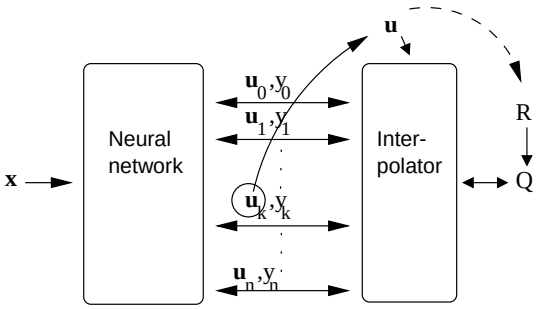

Figure 1: A Q-learning system with continuous states and actions.

A problem with using *Q*-learning when applied as in AUV control is that at any brief instant the action being undertaken does not have a large contribution to the movement of the vehicle. One suboptimal thruster action in a long sequence does not have noticeable effect. Advantage learning addresses this by emphasizing the *relative* advantage of each action.

Advantage Learning

Advantage learning [24] is an enhancement to *Q*-learning which emphasizes the differences in value between actions. In cases where an action does not lead to a catastrophic event, the value of an action can vary only slightly from other actions. This leads to a situation in which the *Q*-value varies widely between states, but only slightly between actions. The problem is compounded as the time intervals between control actions get smaller. As the *Q*-value is only approximated for continuous states and actions it is likely that most of the approximation power will be used representing the values of the states rather than actions in states. The relative values of actions will be poorly represented, resulting in a poor policy. In advantage learning the value of the optimal action is the same as for *Q*-learning, but the lesser value of non-optimal actions is emphasized by a scaling factor. This makes a more efficient use of the approximation resources available.

By applying the technique of advantage learning with continuous states and actions we can evolve an neurocontroller that can move the AUV from place to place, so now we need a method of generating guidance about where to go.

Visual-servoing for Underwater Vehicle Guidance

Many tasks for which an AUV would be useful or where autonomous capability would improve effectiveness, are currently teleoperated by human operators. These operators rely on visual information to perform tasks, so evidently visual imagery can form the basis for guiding underwater vehicles. [25] Other sensing modalities, such as acoustics, are certainly useful, but most of what needs to be done is done visually.

Detailed models of the environment often do not need to be derived from the visual imagery. There are some situations in which a three-dimensional environment model might be useful but, for many tasks, fast visual tracking of features or targets is necessary and sufficient.

Visual servoing is the use of visual imagery to control the pose of the robot relative to (a set of) features.[4] It applies fast feature tracking to provide closed-loop position control of the robot. We are applying visual servoing to the control of an underwater vehicle.

Area-based Correlation for Feature Tracking and Range Estimation

The feature tracking and range estimation technique we use as the basis of visual servoing applies area-based correlation to an image transformed by a sign of the difference of Gaussians (SDOG) operation. An overview of the

method appears in Figure 2. A similar feature tracking technique was used in the visual-servo control of an autonomous land vehicle to track natural features. [26]

This correlation method exploits the invariance of the sign of the zero crossing in the Laplacian of the Gaussian of an image. Even in the presence of noise and image intensity shifts, this sign information is stable. [27] Binary correlation offers more efficient implementation over other schemes, such as sum-of-squared differences and frequency domain matching, since it uses logical rather than arithmetic operations to match the binary sign information, and operates on several pixels at once.

Input images are subsampled, then processed using a difference of Gaussian (DOG) operator. This operator offers many of the same stability properties of the Laplacian operator, but is faster to compute. By selecting different Gaussian sizes, the filter can be adjusted to the input images. The blurred sub-images are then subtracted and binarized based on sign information. This binary image is then correlated with an SDOG feature template matching a small window of a template image either from a previous frame or from the paired stereo frame. A logical exclusive OR (XOR) operation is used to correlate the feature template with the transformed sub-image; matching pixels give a value of zero, while non-matching pixels will give a value of one. A lookup table is then used to compute the Hamming distance (i.e. the number of pixels which differ), the minimum of which indicates where the best match occurred. In the simplest case, the minimum of the correlation matrix indicates the best match. When the minimum is not dominant other statistical methods are needed to identify the location of the best match.

Each feature, and specifically its transformed template, must exhibit sufficient texture to be distinctive from its surroundings. We have found that even without artificially structured targets, natural environments contain sufficient features of an appropriate scale. Currently features are chosen manually but interest operators and selection techniques [28] will soon eliminate this requirement.

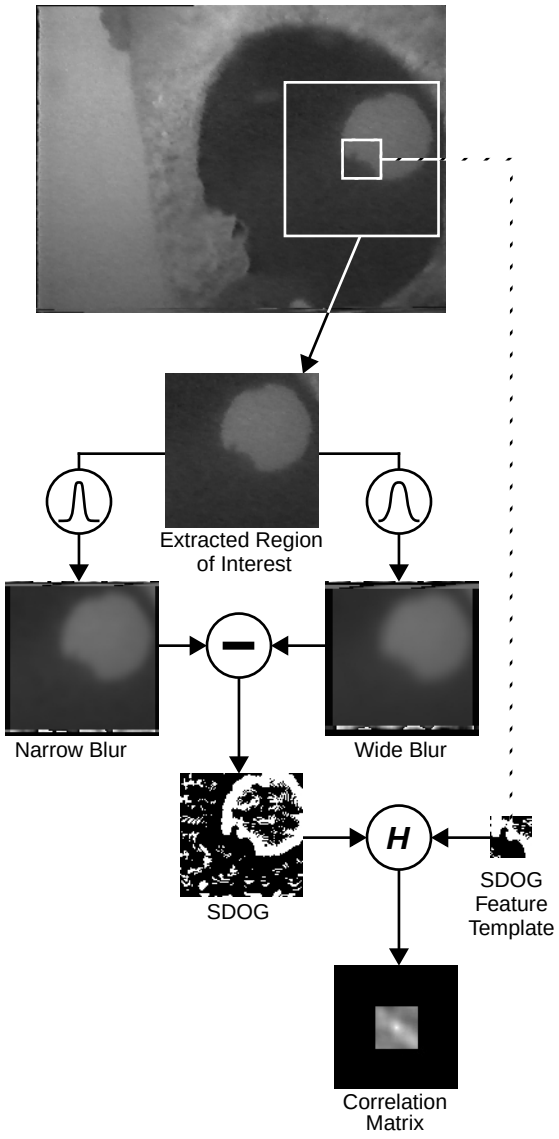

Figure 2: Diagram of the correlation method which performs a sign of the difference of Gaussians (SDOG) transform on a portion of the image and then computes the Hamming distance with stored feature templates to produce a correlation matrix.

The appearance of the features can change drastically as the vehicle moves and the greatest change in appearance typically occurs when the vehicle nears the target, within two or three meters for the size of feature we use. At longer distances, the image to image change in appearance, and pixel correlation, is slight. Simply updating the template every correlation cycle would seem to solve this aspect change problem, but small (single pixel) tracking errors integrate each time the template is updated. In the worst case, this causes the correlator to slide off the feature of interest. By using the same template for several correlation cycles, the effect of accumulated error can be reduced.We have found empirically that updating the template at the frequency at which the vehicle moves a distance equal to the size of the target is sufficient to handle appearance change without suffering from excessive accumulated correlation error.[26] When trying to maintain constant position with respect to a feature it is best, however, to maintain a fixed set of feature templates.

Vehicle Guidance from Tracked Features

We apply this correlation method to guide an AUV with two correlation processes. One, the Feature Motion Tracker follows each target between previous and current images from one camera while the other, the Feature Range Estimator correlates between left and right stereo images to find pixel disparity and from that estimate range.

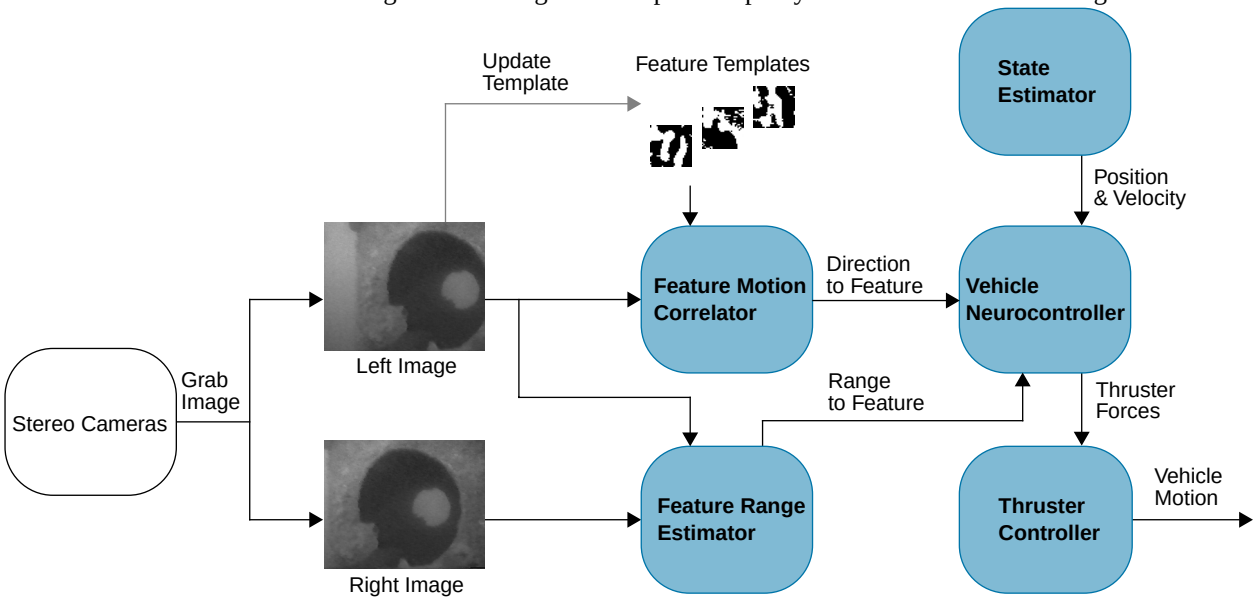

Figure 3: Diagram of the AUV visual servoing system

Figure 3 shows an overview of the visual servoing system. Input imagery comes from a stereo pair of cameras mounted on the AUV. The motion of features in the input images and their range from the camera can therefore be determined relative to the AUV. The Feature Motion Correlator uses stored feature templates to determine the direction to each feature. Feature optical flow can also be determined. Range to a feature is determined by the Feature Range Estimate by correlating features in both left and right stereo images to find their pixel disparity. This disparity is then related to an absolute range based on camera intrinsic and extrinsic parameters which have been determined by calibration. The direction and distance to each feature are then fed the Vehicle Neurocontroller, The neurocontroller requires vehicle state, from the State Estimator, along with feature positions to determine an action, a set of thruster commands.

To guide the AUV, thruster commands become a function of the position of visual features—differential changes in feature position are related to differential changes in the AUV motion.

Application to an Experimental System

Kambara Underwater Vehicle

Our underwater vehicle is named Kambara, an Australian Aboriginal word for crocodile. Kambara's mechanical structure was designed and fabricated by the University of Sydney. It is a simple, low-cost underwater vehicle suitable as a test-bed for research in underwater robot autonomy. At the Australian National University we have undertaken the task of equipping Kambara with power, electronics, computing and sensing.

Kambara's mechanical structure, shown in Figure 4, is an open frame which rigidly supports five thrusters and two watertight enclosures. The frame has length, width, and height of 1.2m, 1.5m, and 0.9m, respectively and displaced volume of approximately 110 liters.

Figure 4: Kambara

Kambara's five thrusters enable roll, pitch, yaw, heave, and surge maneuvers. Hence, is underactuated and not able to perform direct sway (lateral) motion; it is non-holonomic.

Mounted in the upper enclosure is a real-time computing system including main and secondary processors, video digitizers, analog signal digitizers, and communication components. A pan-tilt-zoom camera looks out through the front endcap. Also in the upper enclosure are proprioceptive sensors including a triaxial accelerometer, triaxial gyro, magnetic heading compass, and inclinometers. All of these sensors are wired via analog-to-digital converter to the main processor. These sensor signals, as well as control signals, are processed by an extended Kalman filter (in the State Estimator of Figure 3) to produce a continuous estimate of Kambara's state.

The lower enclosure, connected to the upper by a flexible coupling, contains batteries as well as power distribution and charging circuitry. The batteries are 12V, sealed lead-acid with a total capacity of 1200W. Also mounted down below are depth, temperature and leakage sensors.

In addition to the pan-tilt-zoom camera mounted in the upper enclosure, two cameras are mounted in independent sealed enclosures attached to the frame. Images from these cameras are digitized and mapped into the main processor's memory for processing by the feature motion and range estimation processes.

Evolving a Neurocontroller

Kambara's neurocontroller is based on the advantage learning algorithm [24] coupled an interpolation method [23] for producing continuous control signals. No explicit model of the vehicle is given a priori. The controller generates continuous outputs based on continuous state information. In [29] we describe the algorithms in detail.

We have created a simulated non-holonomic, two degree-of-freedom AUV with thrusters on its left and right sides. The simulation includes linear and angular momentum, and frictional effects. Virtual sensors give the location of targets in body coordinates as well as linear and angular velocity, all with absolute certainly. An image of the simulator during offline learning is shown in Figure 5.

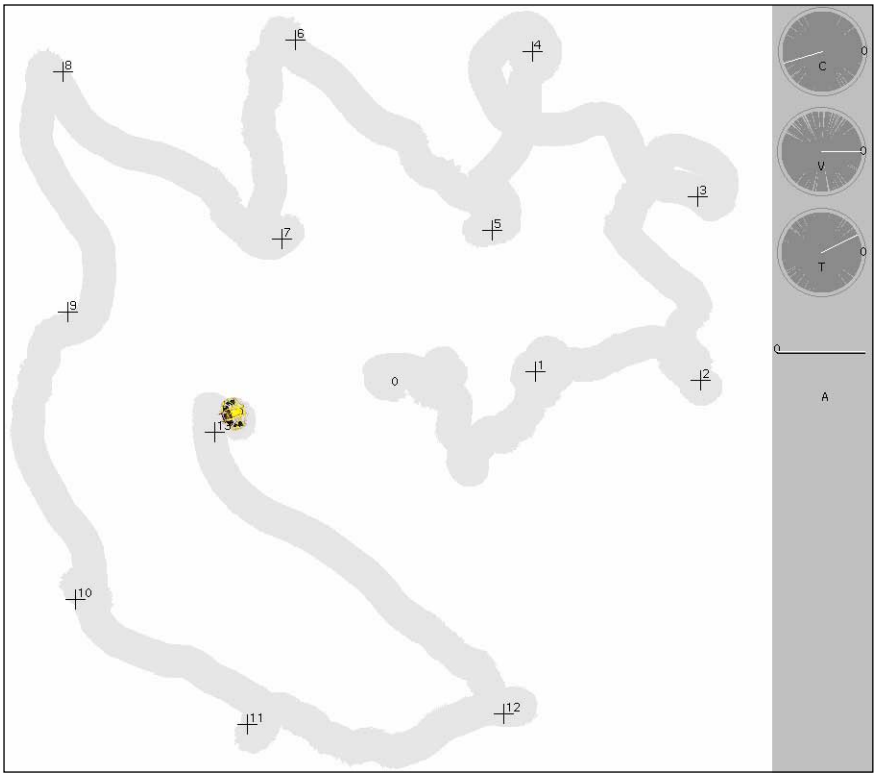

Figure 5: Appearance of the Kambara simulator while learning to control motion and navigate from position to position. The path between goals becomes increasingly direct.

In earlier results using *Q*-learning alone [29], simulated AUVs reach their first (randomly positioned) goal location (within small bounds) about 70% of the time. The AUVs initially take a circuitous route, moving off in the wrong direction and spiralling around. Somewhat less than half of the controllers could reach all in series of 10 targets. Part

of the difficulty is obtaining successful controllers is in designing the reward function. There is little established theory on this. We have empirically settled on the negative of the distance to the target (.i.e the reward for being at the goal is zero). There are some algorithmic advantages in having non-positive reward values.

We now report that 100% of neurocontrollers converge to acceptable performance. The experimental method is that the simulated AUV is given a goal at 2 units of distance away in a random direction. For 200 time steps the controller receives reward based upon its ability to move to and then maintain position at the goal. A purely random controller achieves an average distance of 1.0. A hand-coded controller, which produces apparently good behavior by moving to the target and stopping, achieves 0.25 in average distance to the goal over the training period. This testing method ensures that controllers which guide the AUV to the goal but do not maintain their position don't receive a high rating.

Every 200 time steps, a new goal is randomly generated until the controller has experienced 40 goals. Graphs comparing 140 controllers trained with *Q*-learning and 140 trained with advantage learning are shown in the box-and-whisker plots in Figure 6. Again, all controllers learn to reach each goal although some display occasionally erratic behavior, as seen by the outlying "+" marks. Half of the controllers perform within the box regions, and all except outliers lie within the whiskers. Advantage learning converges to good performance more quickly that *Q*-learning and with many fewer and smaller magnitude spurious actions. Gradual improvement is still taking place at the 40th target.

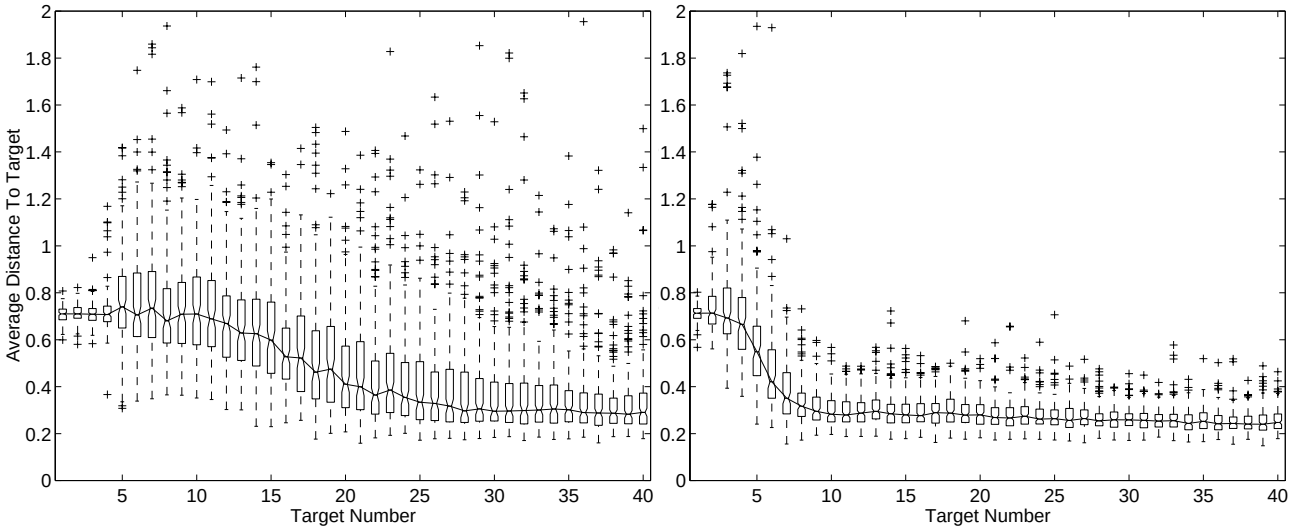

Figure 6: Performance of 140 neurocontrollers trained using Q-learning (left) and advantage learning (right). Box and whisker plots with lines following median performance when attempting to maintain zero distance from each target.

The next obvious experiments, which are currently underway, are to add additional degrees of freedom to the simulation so that the controller must learn to dive and maintain roll and pitch, and to repeat the procedure in the water, on-line, with the real Kambara. A significant challenge lies in the nature and effect of live sensor information. We anticipate drift, systematic error, in our vehicle state estimation. How this will effect learning we can guess by adding noise to our virtual sensors but real experiments will be most revealing.

Tracking a set of Underwater Targets

We have begun work verifying our feature tracking method with actual underwater imagery. In Figure 7, in images of an underwater pile, we provide an example of tracking three features through 250 frames. Note the changing orientation and distance to the pile through this 17 second sequence and that the imagery is low contrast and soft focus. Some features are occasionally lost and then reacquired while the scene undergoes noticeable change in appearance. The changing position of the features provides precisely the data needed to inform the neurocontroller of Kambara's position relative to the target. It is not difficult to imagine Kambara using this visual information to hold station on this pile or to dive and complete an inspection task.

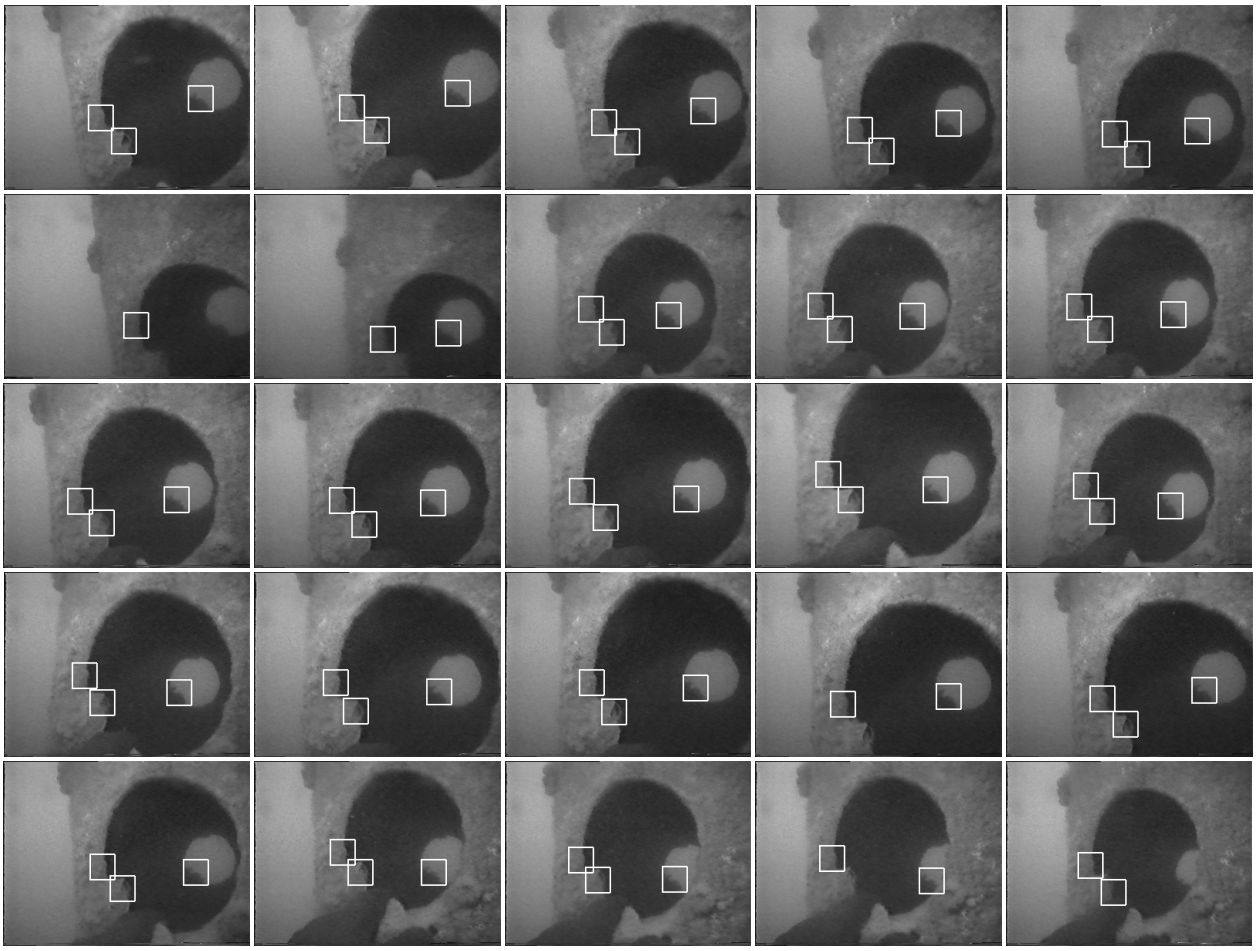

Figure 7: Every tenth frame (top left to bottom right) in a sequence of 250 images of an underwater support pile recorded at 15Hz. Boxes indicate three features tracked from the first frame through the sequence.

Conclusion

There are many approaches to the problem of underwater vehicle control, we have chosen to pursue reinforcement learning. Our reinforcement learning method seeks to overcome some of the limitations of existing AUV controllers and their development, as well as some of the limitations of existing reinforcement learning methods. In simulation we have shown reliable development of stable controllers.

Many important underwater tasks are based on visual information. We are developing robust feature tracking methods and a vehicle guidance scheme that are also based on visual information. This seems to be an appropriate way to make underwater vehicles autonomous. We have obtained initial results in reliably tracking features in underwater imagery and have adapted a proven architecture for visual servo control of a mobile robot.

Acknowledgements

We thank WindRiver Systems and BEI Systron Donner for their support and Pacific Marine Group for providing underwater imagery. We also thank the Underwater Robotics project team: Samer Abdallah, Terence Betlehem, Wayne Dunston, Ian Fitzgerald, Chris McPherson, Chanop Silpa-Anan and Harley Truong for their contributions.

References

[1] D. Yoerger, J-J. Slotine, “Robust Trajectory Control of Underwater Vehicles,” IEEE Journal of Oceanic Engineering, vol. OE-10, no. 4, pp.462-470, October1985.

[2] R. Cristi, F. Papoulias, A. Healey, “Adaptive Sliding Mode Control of Autonomous Underwater Vehicles in the Dive Plane,” IEEE Journal of Oceanic Engineering, vol. 15, no. 3, pp. 152-159, July 1990.

[3] J. Lorentz, J. Yuh, “A survey and experimental study of neural network AUV control,” IEEE Symposium on Autonomous Underwater Vehicle Technology, Monterey, USA, pp 109-116, June 1996.
[4] S. Hutchinson, G. Hager, P. Corke, “A Tutorial on Visual Servo Control,” IEEE International Conference on Robotics and Automation, Tutorial, Minneapolis, USA, May 1996.
[5] T. Fossen, “Underwater Vehicle Dynamics,” *Underwater Robotic Vehicles: Design and Control*, J. Yuh (Editor), TSI Press, pp.15-40, 1995.
[6] K. Goheen, “Techniques for URV Modeling,” *Underwater Robotic Vehicles: Design and Control*, J. Yuh (Ed), TSI Press, pp.99-126, 1995.
[7] D. Yoerger, J. Cooke, J-J Slotine, “The Influence of Thruster Dynamics on Underwater Vehicle Behavior and Their Incorporation Into Control System Design,” IEEE Journal of Oceanic Engineering, vol. 15, no. 3, pp. 167-178, July 1990.
[8] A. Healey, S. Rock, S. Cody, D. Miles, and J. Brown, “Toward an Improved Understanding of Thruster Dynamics for Underwater Vehicles,” IEEE Journal of Oceanic Engineering, vol. 20, no. 4., pp. 354-361, July 1995.
[9] R. Bachmayer, L. Whitcomb, M. Grosenbaugh, “A Four-Quadrant Finite Dimensional Thruster Model,” IEEE OCEANS’98 Conference, Nice, France, pp. 263-266, September 1998.
[10] A. Healey, D. Lienard, “Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles,” IEEE Journal of Oceanic Engineering, vol. 18, no. 3, pp. 327-338, July 1993.
[11] L. Rodriques, P. Tavares, and M. Prado, “Sliding Mode Control of an AUV in Diving and Steering Planes,” IEEE OCEANS’97 Conference, Halifax, Canada, pp. 576-583, 1997.
[12] G. Bartolini, E. Punta, E. Usai, “Tracking Control of Underwater Vehicles including Thruster Dynamics by Second-Order Sliding Modes,” IEEE OCEANS’98 Conference, Nice, France, September1998.
[13] P. Werbos, “Control,” *Handbook of Neural Computation*, F1.9:1-10, Oxford University Press, 1997.
[14] J. Yuh, “A Neural Net Controller for Underwater Robotic Vehicles,” IEEE Journal of Oceanic Engineering, vol. 15, no. 3, pp. 161-166, 1990.
[15] R. M. Sanner and D. L. Akin, “Neuromorphic Pitch Attitude Regulation of an Underwater Telerobot,” IEEE Control Systems Magazine, April 1990.
[16] K. Ishii, T. Fujii, T. Ura, “An On-line Adaption Method in a Neural Network-based Control System for AUV's,” IEEE Journal of Oceanic Engineering, vol. 20, no. 3, July 1995.
[17] K. P. Venugopal, R. Sudhakar, and A. S. Pandya, “On-line Learning Control of Autonomous Underwater Vehicles using Feedforward Neural Networks,” IEEE Journal of Oceanic Engineering, vol.17, no. 4, pp. 308-318, October 1992.
[18] K. Ishii, T. Fujii, T. Ura, “Neural Network System for On-line Controller Adaptation and Its Application to Underwater Robot,” IEEE Intl. Conf. on Robotics and Automation, Leuven, Netherlands, pp. 756-761, 1998.
[19] J. Guo, F. Chiu, C-C. Wang, “Adaptive Control of an Autonomous Underwater Vehicle Testbed Using Neural Networks,” OCEANS’95, San Diego, USA, pp. 1033-1392, 1995.
[20] L. Kaebling, M. Littman, A. Moore, “Reinforcement Learning: A Survey,” Journal of Artificial Intelligence Research, vol. 4, pp. 237-285, 1996.
[21] C. Watkins, *Learning from Delayed Rewards*, Ph.D. Thesis, University of Cambridge, England,1989.
[22] L.-J. Lin. “Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching” Machine Learning Journal, 8(3/4), 1992.
[23] L. Baird, A. Klopf, “Reinforcement Learning with High-dimensional, Continuous Actions,” Technical Report WL-TR-93-1147, Wright Laboratory, 1993.
[24] M. Harmon, L. Baird, “Residual Advantage Learning Applied to a Differential Game,” International Conference on Neural Networks, Washington D.C, USA, June 1995.
[25] J. Santos-Victor, J. Sentieiro, “The Role of Vision Underwater Vehicles,” IEEE International Symposium on Autonomous Underwater Vehicle Technology, AUVT’94, Boston, USA, July 1994.
[26] D. Wettergreen, H. Thomas, and M. Bualat, “Initial Results from Vision-based Control of the Ames Marsokhod Rover, “IEEE International Conference on Intelligent Robots and Systems, Grenoble, France,1997.
[27] K. Nishihara, “Practical Real-Time Imaging Stereo Matcher”, Optical Engineering, vol. 23, pp. 536-545, 1984.
[28] J. Shi, C. Tomasi, “Good Features to Track,” IEEE Conference on Computer Vision and Pattern Recognition, CVPR’94, Seattle, USA, June 1994.
[29] C. Gaskett, D. Wettergreen, A. Zelinsky, “Reinforcement Learning applied to the control of an Autonomous Underwater Vehicle,” Australian Conference on Robotics and Automation, Brisbane, Australia, pp. 125-131, March 1999.