Autonomous Guidance and Control for an Underwater Robotic Vehicle

David Wettergreen, Chris Gaskett, and Alex Zelinsky
Robotic Systems Laboratory
Department of Systems Engineering, RSISE
Australian National University
Canberra, ACT 0200 Australia
[dsw | cg | alex]@syseng.anu.edu.au

Abstract

Underwater robots require adequate guidance and control to perform useful tasks. Visual information is important to these tasks and visual servo control is one method by which guidance can be obtained. To coordinate and control thrusters, complex models and control schemes can be replaced by a connectionist learning approach. Reinforcement learning uses a reward signal and much interaction with the environment to form a policy of correct behavior. By combining vision-based guidance with a neurocontroller trained by reinforcement learning our aim is to enable an underwater robot to hold station on a reef or swim along a pipe.

1 Introduction

At the Australian National University we are developing technologies for underwater exploration and observation. Our objectives are to enable underwater robots to autonomously search in regular patterns, follow along fixed natural and artificial features, and swim after dynamic targets. These capabilities are essential to tasks like exploring geologic features, cataloging reefs, and studying marine creatures, as well as inspecting pipes and cables, and assisting divers. For underwater tasks, robots offer advantages in safety, accuracy, and robustness.

We have designed a guidance and control architecture to enable an underwater robot to perform useful tasks. The architecture links sensing, particularly visual, to action for fast, smooth control. It also allows operators or high-level planners to guide the robot's behavior. The architecture is designed to allow autonomy of at various levels: at the signal level for thruster control, at the tactical level for competent performance of primitive behaviors and at the strategic level for complete mission autonomy.

We use visual information, not to build maps to navigate, but to guide the robot's motion using visual servo control. We have implemented techniques for area-based correlation to track features from frame to frame and to 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.

Approaches to motion control for underwater vehicles, range 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 and ignore coupling between motions. They use dynamic models of the vehicle and make simplifying assumptions that can limit the 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.

By combining vision-based guidance with a neurocontroller trained by reinforcement learning our aim is to enable an underwater robot to hold station on a reef, swim along a pipe, and eventually follow a moving object.

1.1 Kambara Underwater Vehicle

We are developing a underwater robot named Kambara, an Australian Aboriginal word for crocodile. Kambara's mechanical structure was designed and fabricated by the University of Sydney. At the Australian National University we are equipping Kambara with power, electronics, computing and sensing.

Kambara's mechanical structure, shown in Figure 1, has length, width, and height of 1.2m, 1.5m, and 0.9m, respectively and displaced volume of approximately 110 liters. The open-frame design rigidly supports five thrusters and two watertight enclosures. Kambara's thrusters are commercially available electric trolling motors that have been modified with ducts to improve thrust and have custom power amplifiers designed to provide high current to the brushed DC motors. The five thrusters enable roll, pitch, yaw, heave, and surge maneuvers. Hence, Kambara is underactuated and not able to perform direct sway (lateral) motion; it is non-holonomic.

A real-time computing system including main and secondary processors, video digitizers, analog signal digitizers, and communication component is mounted in the upper enclosures. A pan-tilt-zoom camera looks out through the front endcap. Also in the upper enclosure are proprioceptive sensors including a tri-

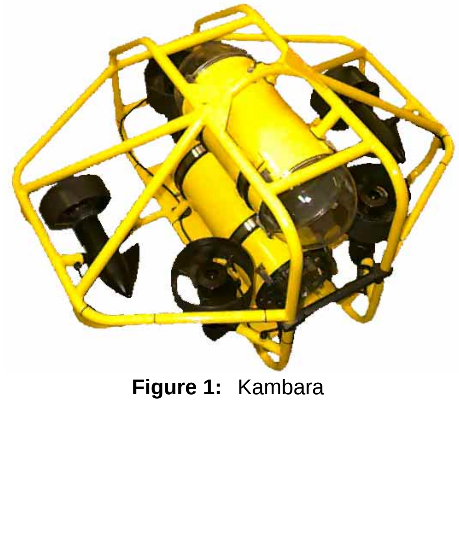

Figure 1: Kambara

axial accelerometer, triaxial gyro, magnetic heading compass, and inclinometers. All of these sensors are wired via analog-to-digital converter to the main processor.

The lower enclosure, connected to the upper by a flexible coupling, contains batteries as well as power distribution and charging circuitry. The batteries are sealed lead-acid with a total capacity of 1200W. Also mounted below are depth 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 for processing by the vision-based guidance processes.

2 Architecture for Vehicle Guidance

Kambara's software architecture is designed to allow autonomy at various levels: at the signal level for adaptive thruster control, at the tactical level for competent performance of primitive behaviors, and at the strategic level for complete mission autonomy.

The software modules are designed as independent computational processes that communicate over an anonymous broadcast protocol, organized as shown in Figure 2. The Vehicle Manager is the sole downstream communication module, directing commands to modules running on-board. The Feature Tracker is comprised of a feature motion tracker and a feature range estimator as described in section 3. It uses visual sensing to follow targets in the environment and uses their relative motion to guide the Vehicle Neurocontroller. The Vehicle Neurocontroller, described in 4, learns an appropriate valuation of states and possible actions so that it can produce control signals for the thrusters to move the vehicle to its goal. The Thruster Controller runs closed-loop servo control over the commanded thruster forces. The Peripheral Controller drives all other devices on the vehicle, for example cameras or scientific instruments. The Sensor Sampler collects sensor information and updates the controllers and the State Estimator. The State Estimator filters sensor information to generate estimates of vehicle position, orientation and velocities. The Telemetry Router moves vehicle state and acquired image and science data off-board.

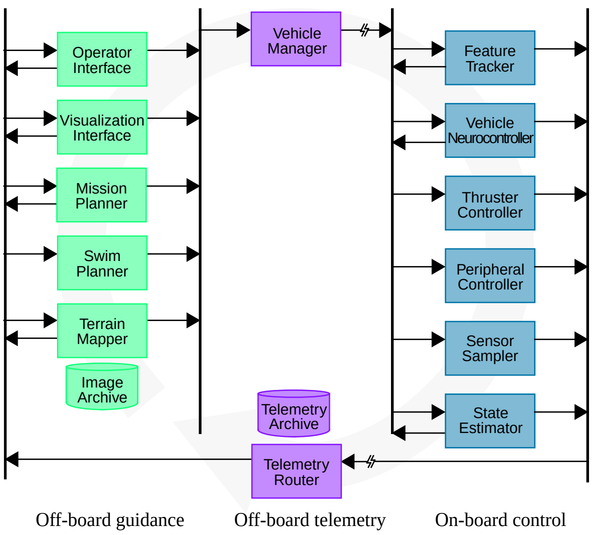

Figure 2: Architecture for vehicle guidance and control

The Visualization Interface will transform telemetry into a description of vehicle state that can be rendered as a three-dimensional view. The Operator Interface interprets telemetry and presents a numerical expression of vehicle state. It provides method for generating commands to the Vehicle Interface for direct teleoperation of vehicle motion and for supervisory control of the on-board modules.

The Swim Planner interprets vehicle telemetry to analyze performance and adjust behavior accordingly, for example adjusting velocity profiles to better track a pattern. A Terrain Mapper would transform data (like visual and range images) into maps that can be rendered by the Visualization Interface or used by the Swim Planner to modify behavior. The Mission Planner sequences course changes to produce complex trajectories to autonomously navigate the vehicle to goal locations and carry out complete missions.

2.1 Operational Modes

The software architecture is designed to accommodate a spectrum of operational modes. Teleoperation of the vehicle with commands fed from the operator directly to the controllers provides the most explicit control of vehicle action. While invaluable during development and some operations, this mode is not practical for long-duration operations. Supervised autonomy, in which complex commands are sequenced off-board and then interpreted over time by the modules on-board, will be our nominal operating mode. Under supervised autonomy, the operator's commands are infrequent and provide guidance rather than direct action commands. The operator gives the equivalent of "swim to that feature" and "remain on station". In fully autonomous operation, the operator is removed from the primary control cycle and planners use state information to generate infrequent commands for the vehicle. The planners may guide the vehicle over a long traverse, moving from one target to another, or thoroughly exploring a site with no human intervention

3 Vision-based Guidance of an Underwater Vehicle

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 making a strong argument that visual imagery could be used to guide an underwater vehicle.

Detailed models of the environment are often not required. 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 robot.

3.1 Area-based Correlation for Feature Tracking

The feature tracking technique that we use as the basis for visual servoing applies area-based correlation to an image transformed by a sign of the difference of Gaussians (SDOG) operation. A similar feature tracking technique was used in the visual-servo control of an autonomous land vehicle to track natural features.[5]

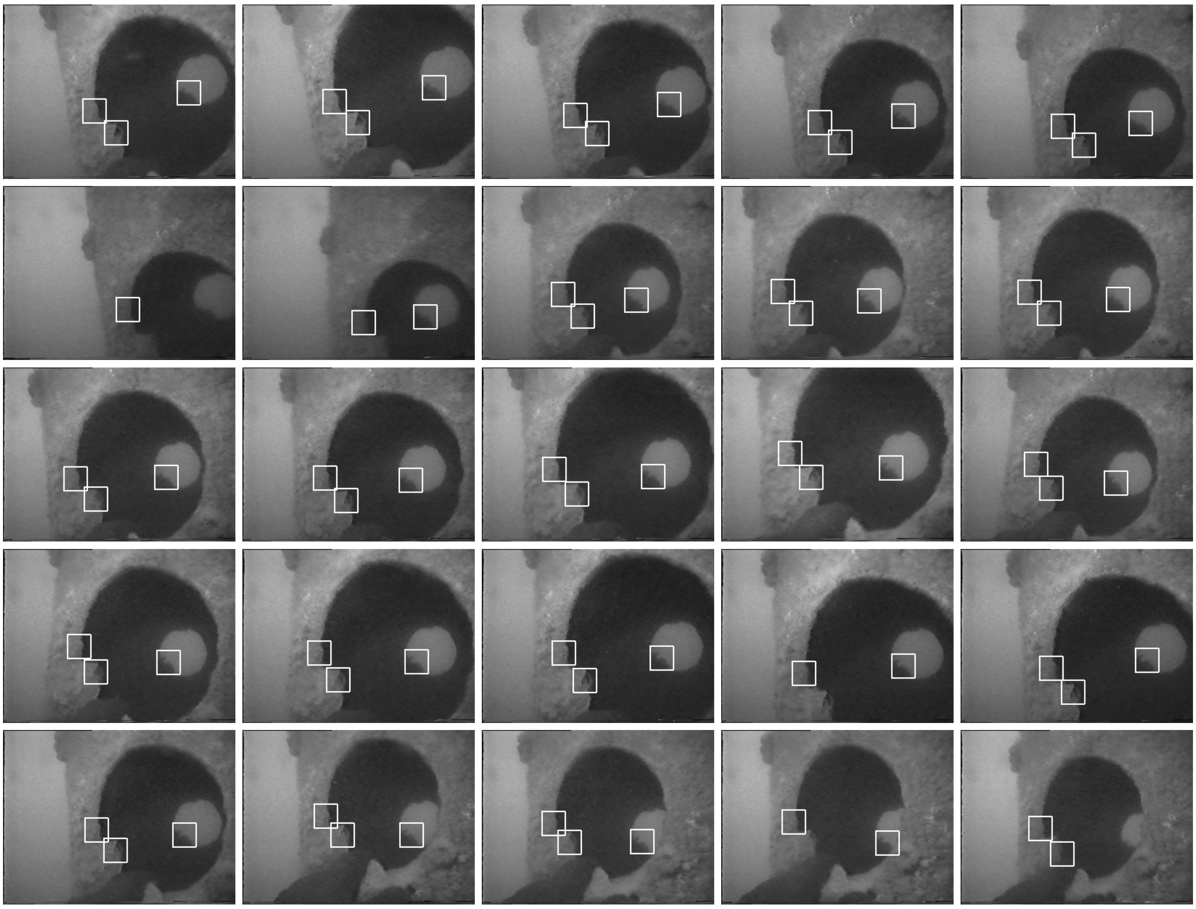

Figure 3: Every tenth frame (top left across to bottom right) in a sequence of 250 images of an underwater support pile recorded at 15Hz. Boxes indicate three features tracked from the first frame through the sequence.

Input images are subsampled and 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.[6] 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 (the number of pixels which differ), the minimum of which indicates the best match.

3.2 Tracking Underwater Features

We are verifying our feature tracking method with actual underwater imagery. Figure 3 shows tracking three features through 250 images of a support pile.

The orientation and distance to the pile changes through this 17 second sequence. Some features are 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 Vehicle Neurocontroller of Kambara's position relative to the target.

3.3 Vehicle Guidance from Tracked Features

Guidance of an AUV using our feature tracking method requires two correlation operations within the Feature Tracker, as seen in Figure 4. The first, the feature motion tracker, follows each feature between previous and current images from one camera while the other, the feature range estimator, correlates between left and right camera images. The feature motion tracker correlates stored feature templates to determine the image location and thus direction to each feature. Range to a feature is determined by correlating features in both left and right stereo images to find their pixel disparity. This disparity is then related to an absolute range using camera intrinsic and extrinsic parameters which are determined by calibration.

The appearance of the features can change drastically as the vehicle moves so managing and updating feature templates is crucial part in reliably tracking features. We found empirically that updating the feature template at the rate at which the vehicle moves a distance equal to the size of the feature is sufficient to handle appearance change without suffering from excessive accumulated correlation error.[5]

The direction and distance to each feature are fed the Vehicle Neurocontroller, The neurocontroller requires vehicle state, from the State Estimator, along with feature positions to determine a set of thruster commands. To guide the AUV, thruster commands become a function of the position of visual features.

4 Learned Control of an Underwater Vehicle

Many approaches to motion control for underwater vehicles have been proposed, and although 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

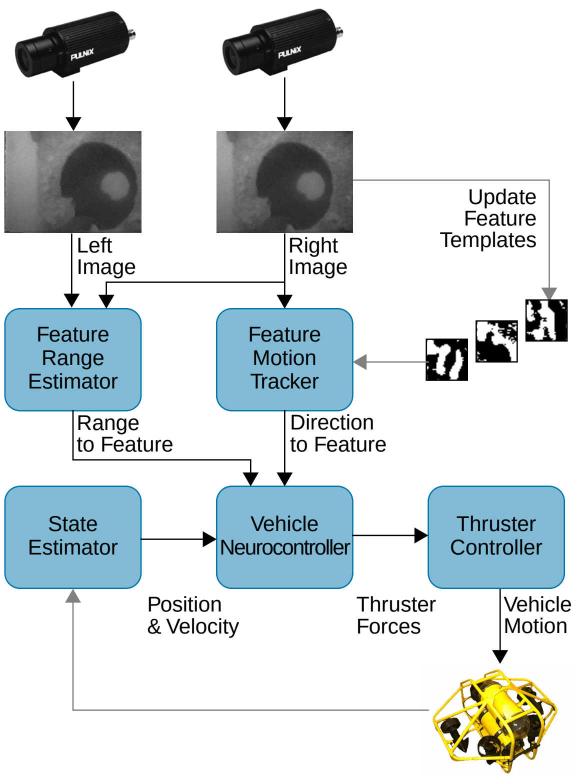

Figure 4: Diagram of the AUV visual servoing system

degrees-of-freedom, for example yaw and surge, and assume motion along some dimensions can be controlled independently. These controllers usually require a dynamic model and simplifying assumptions that may limit operating regime and robustness.

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

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, nonlinear control techniques are needed to obtain high performance.[7]

Nonlinear models of underwater vehicles have coefficients that must be identified and some remain unknown 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.[8]

4.1 Neurocontrol of Underwater Vehicles

Control using artificial neural networks, neurocontrol, [9] 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.

A model of system dynamics is not required. An appropriate controller is developed slowly through learning. Control of low-level actuators as well as high-level navigation can potentially be incorporated in one neurocontroller.

Several different neural network based controllers for AUVs have been proposed. [10] Sanner and Akin [11] 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. Ishii, Fujii and Ura [12] 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. Yuh [10] proposed several neural network based AUV controllers. Error at the output of the controller is also based on a critic.

4.2 Reinforcement Learning for 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. The lack of an explicit a priori model reduces reliance on knowledge of the system to be controlled.

Reinforcement learning addresses the problem of forming a *policy* of correct behavior through observed interaction with the environment. [13] The strategy is to continuously refine an estimate of the utility of performing specific actions while in specific states. 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 the future. 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.

Q-learning [14] is an implementation method for reinforcement learning in which a mapping is learned from a state-action pair to its value (Q). The mapping eventually represents the utility of performing an particular action from that state. The neurocontroller executes the action which has the highest Q value in the current state. The Q value is updated according to:

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

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

Q-learning is normally considered in a discrete sense. High-performance control 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. When states and actions are continuous, the learning system must generalize between similar states and actions. To generalize between states, one approach is to use a neural network.[15] An interpolator can provide generalization between actions.[16] Figure 5 shows the general structure of such a system.

A problem with applying *Q*-learning to AUV control is that a single suboptimal thruster action in a long sequence does not have noticeable effect. Advantage learning [17] is a variation of *Q*-learning

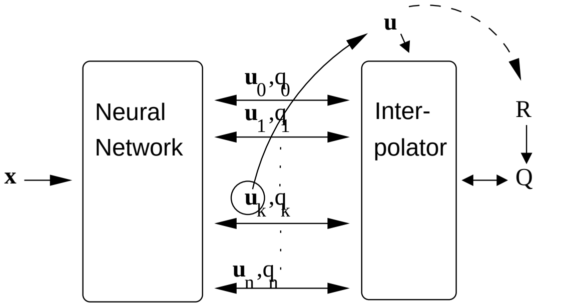

Figure 5: A Q-learning system with continuous states and actions as implemented in the neurocontroller.

which addresses this by emphasizing the difference in value between actions and assigning more reward to correct actions whose individual effect is small.

Kambara's neurocontroller [18] is based on advantage learning coupled with an interpolation method [16] for producing continuous output signals.

4.3 Evolving a Neurocontroller

We have created a simulated non-holonomic, two degree-of-freedom AUV with thrusters on its left and right sides, shown in Figure 6. 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.

The simulated AUV is given a goal at 1 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.

Every 200 time steps, a new goal is randomly generated until the controller has experienced 40 goals. A graph showing the performance of 140 neurocontrollers, trained with advantage learning is shown in the box-and-whisker plot of Figure 7. All controllers (100%) 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. This learning method converges to good performance quickly and with few and small-magnitude spurious actions.

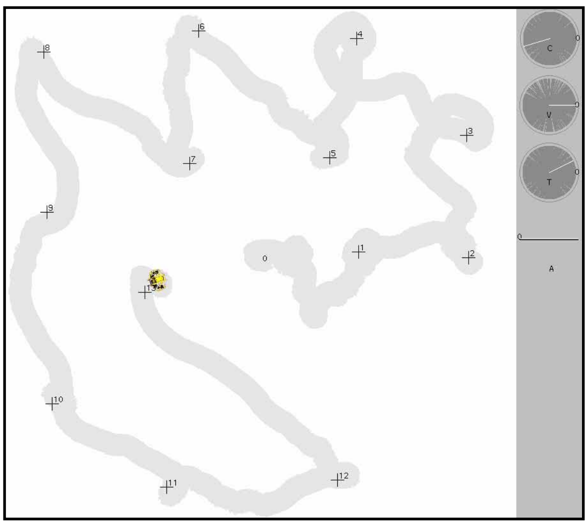

Figure 6: Kambara simulator while learning to control motion and navigate from position to position. The path between goals becomes increasingly direct.

The next experiments 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. Experiments in linking the vision system to the controller can then commence.

A significant challenge lies in the nature and effect of live sensor information. We anticipate bias, drift, and non-white noise 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.

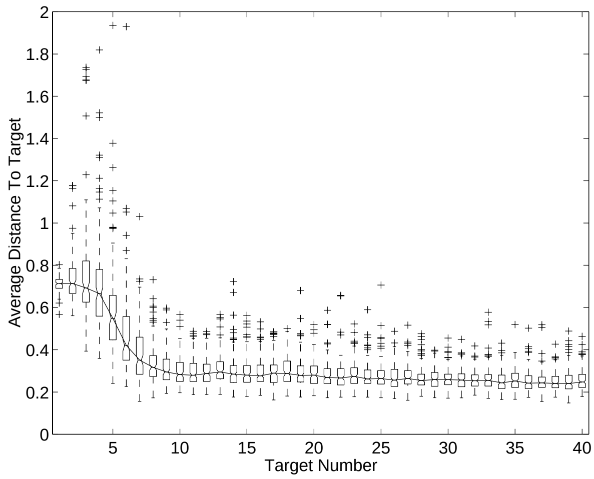

Figure 7: Performance of 140 neurocontrollers trained using advantage learning. Box and whisker plots with median line when attempting reach and maintain 40 target positions each for 200 time steps.

5 Commanding Thruster Action

The task of Vehicle Neurocontroller is simplified if its commanded output is the desired thrust force rather than motor voltage and current values. The neurocontroller need not learn to compensate for the non-linearities of the thruster, its motor and amplifier. Individual thruster controllers use force as a desired reference to control average motor voltage and current internally.

Considerable effort has been applied in recent years to developing models of underwater thrusters.[19][20][21] This is because thrusters are a dominant source of nonlinearity in underwater vehicle motion.[19] 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.

We have measured motor parameters including friction coefficients and motor inertia and begun in-tank tests measure propeller efficiency and relationships between average input voltage and current, motor torque, and output thrust force. Using a thruster model [21] and these parameters, the neurocontrollers force commands can be accurately produced by the thrusters.

6 Estimating Vehicle State

In order to guide and control Kambara we need to know where it was. where it is, and how it is moving.

This is necessary for long-term guidance of the vehicle as it navigates between goals and for short-term control of thruster actions. Continuous state information is essential to the reinforcement learning method that Kambara uses to learn to control its actions.

Kambara carries a rate gyro to measure its three angular velocities and a triaxial accelerometer to measure its three linear accelerations. A pressure depth sensor provides absolute vertical position, an inclinometer pair provide roll and pitch angles and a magnetic heading compass measures yaw angle in a fixed inertial frame. Motor voltages and currents are also relevant state information. The Feature Tracker could also provide relative position, orientation and velocity of observable features.

These sensor signals, as well as input control signals, are processed by a Kalman filter in the State Estimator to estimate Kambara's current state. From ten sensed values (linear accelerations, angular velocities, roll, pitch, yaw and depth) the filter estimates and innovates twelve values: position, orientation and linear and angular velocities.

The Kalman filter requires models of both the sensors and the vehicle dynamics to produce its estimate. Absolute sensors are straightforward, producing a precise measure plus white Gaussian noise. The gyro models are more complex to account for bias and drift. A vehicle dynamic model, as described previously, is complex, non-linear, and inaccurate. All of our models are linear approximations.

There is an apparent contradiction in applying model-free learning to develop a vehicle neurocontroller and then estimating state with a dynamic model. Similarly, individual thruster controllers might be redundant with the vehicle neurocontroller. We have not fully reconciled this but believe that as practical matter partitioning sensor filtering and integration, and thruster control from vehicle control will facilitate learning. Both filtering and motor servo-control can be achieved with simple linear approximations leaving all the non-linearities to be resolved by the neurocontroller.

If the neurocontroller is successful in doing this, we can increase the complexity (and flexibility) by reducing reliance on modelling. The first step is to remove the vehicle model from the state estimator, using it only to integrate and filter data using sensor models. Direct motor commands (average voltages) could also be produced by the neurocontroller, removing the need for the individual thruster controllers and the thruster model. Without the assistance of a model-based state estimator and individual thruster controllers the neurocontroller will have to learn from less accurate data and form more complex mappings.

7 Conclusion

Many important underwater tasks are based on visual information. We are developing robust feature tracking methods and a vehicle guidance scheme that are based on visual servo control. 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.

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 neurocontrollers.

Acknowledgements

We thank Wind River Systems and BEI Systron Donner for their support and Pacific Marine Group for providing underwater imagery. We also thank the RSL Underwater Robotics team 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] 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.

[6] K. Nishihara, "Practical Real-Time Imaging Stereo Matcher", Optical Engineering, vol. 23, pp. 536-545, 1984.

[7] T. Fossen, "Underwater Vehicle Dynamics," *Underwater Robotic Vehicles: Design and Control*, J. Yuh (Editor), TSI Press, pp.15-40, 1995.

[8] K. Goheen, "Techniques for URV Modeling," *Underwater Robotic Vehicles: Design and Control*, J. Yuh (Ed), TSI Press, pp.99-126, 1995.

[9] P. Werbos, "Control," *Handbook of Neural Computation*, F1.9:1-10, Oxford University Press, 1997.

[10] J. Yuh, "A Neural Net Controller for Underwater Robotic Vehicles," IEEE Journal of Oceanic Engineering, vol. 15, no. 3, pp. 161-166, 1990.

[11] R. M. Sanner and D. L. Akin, "Neuromorphic Pitch Attitude Regulation of an Underwater Telerobot," IEEE Control Systems Magazine, April 1990.

[12] K. Ishii, T. Fujii, T. Ura, "An On-line Adaptation Method in a Neural Network-based Control System for AUV's," IEEE Journal of Oceanic Engineering, vol. 20, no. 3, July 1995.

[13] L. Kaebling, M. Littman, A. Moore, "Reinforcement Learning: A Survey," Journal of Artificial Intelligence Research, vol. 4, pp. 237-285, 1996.

[14] C. Watkins, *Learning from Delayed Rewards*, Ph.D. Thesis, University of Cambridge, England,1989.

[15] L.-J. Lin. "Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching" Machine Learning Journal, 8(3/4), 1992.

[16] L. Baird, A. Klopf, "Reinforcement Learning with High-dimensional, Continuous Actions," Technical Report WL-TR-93-1147, Wright Laboratory, 1993.

[17] M. Harmon, L. Baird, "Residual Advantage Learning Applied to a Differential Game," International Conference on Neural Networks, Washington D.C, USA, June 1995.

[18] 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.

[19] 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.

[20] 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.

[21] R. Bachmayer, L. Whitcomb, M. Grosenbaugh, "A Four-Quadrant Finite Dimensional Thruster Model," IEEE OCEANS'98 Conference, Nice, France, pp. 263-266, September 1998.