Enabling Autonomous Capabilities in Underwater Robotics

Junaed Sattar, Gregory Dudek, Olivia Chiu, Ioannis Rekleitis, Philippe Giguère, Alec Mills, Nicolas Plamondon,
Chris Prahacs, Yogesh Girdhar, Meyer Nahon, and John-Paul Lobos

Abstract—Underwater operations present unique challenges and opportunities for robotic applications. These can be attributed in part to limited sensing capabilities, and to locomotion behaviours requiring control schemes adapted to specific tasks or changes in the environment. From enhancing teleoperation procedures, to providing high-level instruction, all the way to fully autonomous operations, enabling autonomous capabilities is fundamental for the successful deployment of underwater robots. This paper presents an overview of the approaches used during underwater sea trials in the coral reefs of Barbados, for two amphibious mobile robots and a set of underwater sensor nodes. We present control mechanisms used for maintaining a preset trajectory during enhanced teleoperations and discuss their experimental results. This is followed by a discussion on amphibious data gathering experiments conducted on the beach. We then present a tetherless underwater communication approach based on pure vision for high-level control of an underwater vehicle. Finally the construction details together with preliminary results from a set of distributed underwater sensor nodes are outlined.

I. INTRODUCTION

Many different underwater applications can be automated with the use of robots. Environmental monitoring and damage assessment, security applications, oil and gas installation monitoring and repair, and pipeline inspection are among the most well known. In general humans are limited to short visits underwater and only to limited depths. As such, the underwater environment holds many opportunities for deployment of robotic agents. At the same time, limited visibility, constraints in communications, as well as the application of external forces to the robot from water currents make underwater operations very challenging.

We have developed a robotic vehicle named Aqua, shown in Fig. 1, that has the potential to operate autonomously and in conjunction with stationary sensors. Our vehicle has passed through several design and re-design cycles [1], [2], [3], [4] and it has been tested in a variety of environments. Moreover, the mechanical design, the implementation of basic sensing capabilities, and the development of basic behaviours have undergone significant change. Currently, two prototypes are being deployed; see Fig. 1 and 2. A detailed description of the mechanical and electronic aspects of Aqua can be found in [2], [1] and is outside the scope of this paper.

The operation of any robot can be characterized according to the amounts of input required from a human operator. The most common mode in underwater vehicles is complete teleoperation; where the operator is required to constantly monitor and control the robot, reacting to continuous sensor feedback. Such operations allow no autonomy to the robot, are quite challenging, and require a well trained operator.

Centre for Intelligent Machines, McGill University, junaed@cim.mcgill.ca
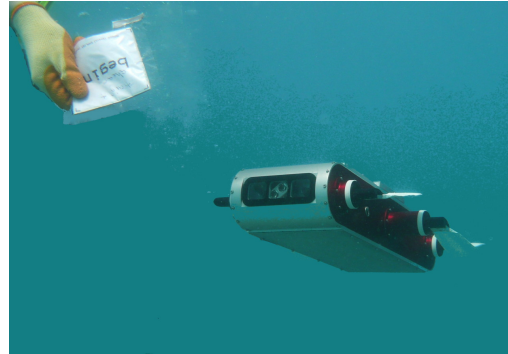
Fig. 1. Tetherless Aqua robot is instructed using a cue-card.

By adding a controller to handle small disturbances, the user interface can be augmented and a significant cognitive load can be removed from the operator. Different such control algorithms have been developed in order to maintain attitude or follow an intended trajectory despite disturbances. The next level of autonomy is to equip the robot with a set of basic behaviours such as: move forward, turn, hover, go to depth, surface, etc. At this stage the operator can communicate the commands with or without a tether, yet still maintain direct supervision.

The control capabilities developed for the augmented user interface are crucial for the robot's abilities to accurately execute the given instructions. The communication abilities of our tetherless underwater vehicles are restricted to processing simple visual cues, as they are not equipped with other communication means. As the autonomous capabilities increase the instructions become more high-level. At the same time, the robot needs to acquire a better understanding of its pose and of the environment. The Aqua robot has unique amphibious capabilities. In full autonomous mode, we envision the robot to be walking off the beach, entering the water, collecting data and returning back to shore. Some of the components that are essential to the realization of the above mentioned scenario are discussed in this paper.

The next section presents a brief overview of related work. Section III discusses several different control schemes for stable trajectory following. Next we present classification results for identifying the environment the robot operates in. The visual communication capabilities underwater are discussed in Section V. The development and successful deployment of a set of underwater sensor nodes is presented in Section VI. Finally, we close with conclusions and a discussion of future work.

II. RELATED WORK

The Aqua robot is an embodiment of the quest for an autonomous, stable amphibious legged vehicle, and as such our

work is exploring different aspects of control theory, machine vision, supervised learning and of course underwater vehicle construction. We present some of the work done in the past in these areas.

Our robots are a direct descendant of the RHex family of robots [5], which were designed as hexapod walking vehicles. Several generations of RHex robots have been produced, and the Aqua robots incorporate the mobility of the original RHex robot, and the ruggedness built into the later generations [1], coupled with the ability to work underwater.

A number of controllers for underwater robots have been explored by other researchers. For example, Yuh designed nonlinear and adaptive controllers for an underwater vehicle [6]. Smallwood et al. designed and tested six different types of controllers for the John Hopkins University Remotely Operated Underwater Vehicle (JHUROV) [7], and Encarnação et al. developed a controller for following a 3D path [8]. However, a controller has not been previously designed for a hexapod paddled robot. Using a model and simulation tests developed for Aqua in [9], a few different controllers have been developed and implemented on the robot.

Visual servoing is a well-studied area of robotics [10], one which combines the theories of active vision into practises in real-world applications. Robotic assembly lines, autonomous aircraft control (*i.e.* landing, hovering etc), robot-assisted surgery [11], navigation and guidance of underwater robots are applications where visual servoing has been highly successful. While tracking underwater swimmers visually has not been explored in the past, some prior work has been done in the field of underwater visual tracking and visual servoing for autonomous underwater vehicles (AUVs). Naturally, this is closely related to generic servo-control. The family of algorithms developed are both of the offline and on-line variety. The on-line tracking systems, in conjunction with a robust control scheme, provide robots the ability to visually follow targets underwater [12].

Gesture-based robot control has been explored extensively. This includes explicit as well as implicit communication between human operators and robotics systems. Several authors have considered specialized gestural behaviours [13] or strokes on a touch screen to control basic robot navigation. Skubic *et al.* have examined the combination of several types of human interface components, with special emphasis on speech, to express spatial relationships and spatial navigation tasks [14].

The underwater domain poses certain unique challenges that render a lot of the assumptions of terrestrial robotics problematic. An underwater robot has six degrees of freedom, and maneuvering with six degrees of freedom creates serious complications. Substantial progress has been made in designing the hardware and algorithms for underwater robots, and much of the research is directed in creating an AUV for operator-independent exploration of underwater environments. The traditional approach to propel undersea vehicles is by using propellers or thrusters. Although sim-

Fig. 2. A tethered Aqua robot performs a 45° turn guided by a PD controller.

ple by design, these vehicles lack the maneuverability and agility seen in fish and other marine species. For an AUV, efficient energy consumption is critical, and thrusters are not an energy efficient approach to station keeping underwater [15]. Among other efforts to propel underwater vehicles, the RoboTuna project at Massachusetts Institute of Technology (MIT) is well known. The RoboTuna project [16] attempted to create a fish-like underwater vehicle, with a propulsion system mimicking those found in fish, hence creating an example of Biomemetic Robotics applied in underwater environments. The MIT Sea-Grant Program has an extensive program to create new underwater platforms for deep ocean explorations, including AUVs. The flapping-foil fish robot [17] is an example of an experimental, high-maneuverability robot created by the Tow Tank Lab under the Sea Grant Project. In the next section we present different algorithms for the control of Aqua, specifically designed for underwater environments.

III. TRAJECTORY TRACKING EXPERIMENT

There are various types of controllers appropriate for use with this type of vehicle. A controller can be used to stabilize the vehicle and thus allows the pilot to steer the vehicle with more ease. Alternatively, a controller can replace the pilot altogether and enable a vehicle to follow a predetermined path on its own. Both types of controllers can be used together or separately and they can take on different forms.

This section presents the results obtained during the stability and trajectory tracking experiments performed in open water trials. The objective of the stability experiment is to determine the robot's ability to react to moment impulses acting on it. The autopilot experiments test trajectory tracking algorithms in a dynamic environment and compares PD and model-based control.

A. Stability Augmented System

A stability augmented system (SAS) differs from an autopilot in that it does not ensure that the robot follows a trajectory. Rather, it aims to return all state perturbations to zero, and thus reduces the impact of external disturbances on the system. This is done by closing the feedback loop in the system and returning the measured states of the robot to
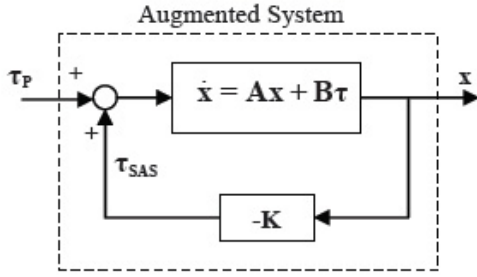
Fig. 3. Block diagram of a SAS

the controller as seen in Fig. 3. In this figure, the equation of motion of the system is represented by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\tau, \tag{1}$$

where $\mathbf{x} = [u \ v \ w \ p \ q \ r \ x \ y \ z \ \phi \ \theta \ \psi]^T$ is the state vector with u, v, w as the body-frame components of the robot's velocity; p, q, r are the body-frame components of the angular velocity; x, y, z are the vehicles center of mass position in the inertial frame; and ϕ, θ, ψ are the vehicle's Euler angles. The vector $\tau = [f_{x1} \dots f_{x6} \ f_{z1} \dots f_{z6}]^T$ represents the control forces due to the paddles where f_{xi} is the force provided by the ith paddle in the x-direction and f_{zi} in the z-direction. The paddles achieve these control forces by varying the oscillation period, amplitude and offset of each paddle. Finally, $\mathbf{A}$ and $\mathbf{B}$ are state matrices derived using numerical finite differencing for the steady state forward velocity $0.5 m/s$, which was verified in [18]. The control input τ_{SAS} is provided by the SAS, while the input τ_P is provided by the high-level controller such as a pilot or an autopilot. The general notion is that the high-level controller "sees" a new augmented system that is more stable then the original non-augmented system (with $\tau_{SAS} = \mathbf{0}$).

The SAS can be a proportional controller and take the form

$$\tau = -\mathbf{K}\mathbf{x} \tag{2}$$

where $\mathbf{K}$ is the gain matrix. If the above equation is substituted into Eq. (1) the system then becomes

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} \tag{3}$$

and $(\mathbf{A} - \mathbf{B}\mathbf{K})$ can be replaced by $\mathbf{A}'$ such that

$$\dot{\mathbf{x}} = \mathbf{A}'\mathbf{x} + \mathbf{B}\tau_{\mathbf{P}} \tag{4}$$

This augmented system represented by $\mathbf{A}'$ is more stable and it is what the high-level controller acts on. Since the augmented system is more stable, the design of the autopilot can focus on following a trajectory without being concerned about the stability of the vehicle.

The stability of the augmented system can be evaluated by looking at the eigenvalues of the $(\mathbf{A} - \mathbf{B}\mathbf{K})$ matrix. This can be done by solving the following equation

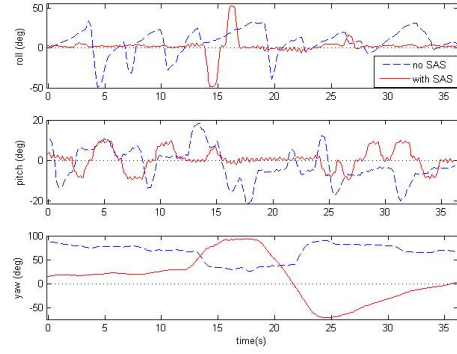$$\det |s\mathbf{I} - (\mathbf{A} - \mathbf{B}\mathbf{K})| = 0 \tag{5}$$

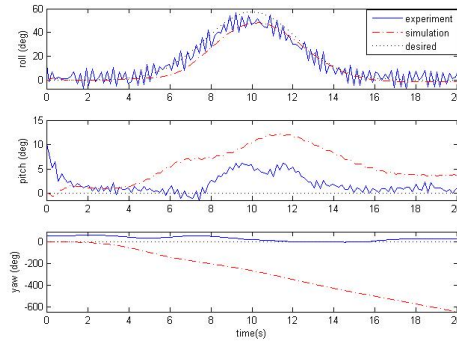Fig. 4. Experimental data using the SAS.

Fig. 5. Experimental data with the autopilot.

Since the eigenvalues are also the poles of the system, any eigenvalues with a positive real part would indicate instability of the system. Therefore the $\mathbf{K}$ matrix is found by solving Eq. 5 such that the eigenvalues of $(\mathbf{A} - \mathbf{B}\mathbf{K})$ all have negative real parts.

A $\mathbf{K}$ matrix is designed and implemented in simulation for Aqua when it is swimming with a forward velocity of approximately 0.5m/s. This is then tested in the ocean with Aqua swimming forward and impulse roll and pitch commands were periodically given to the robot. Fig. 4 presents results from two trials. The dashed (blue) line represents a trial without the use of the SAS and the red line represents a trial that used the SAS. As can be seen, with the SAS, the roll and pitch angles returns more quickly to zero after each impulse. Furthermore, the roll and pitch angles remains closer to zero with the use of the SAS. For more information please refer to [18].

The SAS has also been tested during a coral transect. The pilot expressed a vast improvement in the controllability of the robot and the ease in carrying out the coral transect.

B. Autopilot

Next a PD autopilot is designed to be used with the SAS. This is done mostly through trial and error and implemented in simulation. The predetermined path required the roll angle to follow a Gaussian curve, which is represented by the dotted (black) line in the Fig. 5. The other angles were required to remain at zero. The autopilot is then implemented on the robot and it is found that the simulation data did not correspond to the experimental results. This can be clearly seen in Fig. 5.
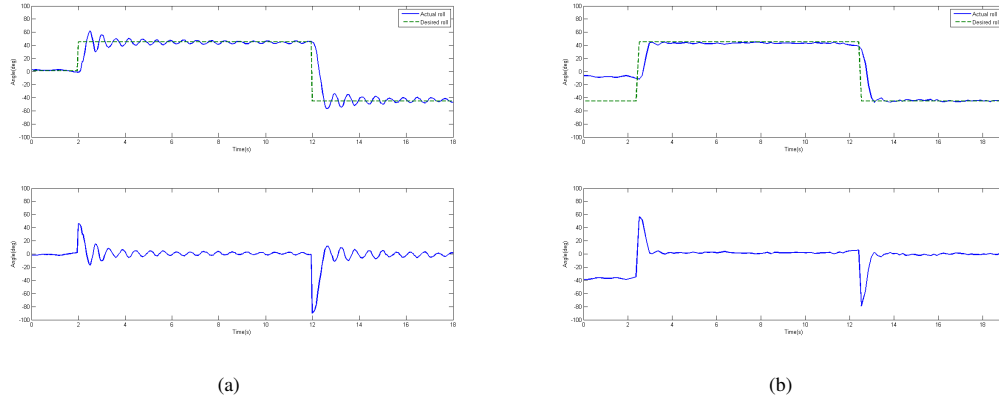
Fig. 6. (a) Tracking of a roll command using a proportional controller. (b) Tracking of a roll command using a model-based controller. In both, the top graph shows the actual and desired angle and the bottom one the tracking error.

Both the simulation and experimental data show the robot following the roll trajectory. However, the experimental data show a large amount of oscillations, which are not accounted for in the simulation. The discrepancies arise from the difference in mapping of the autopilot output to the paddle movements.

C. Controllers

Two other controllers have been developed, that do not use the SAS, and have been tested on the robot. The first controller is a simple PD controller, in which, the force is given by

$$\mathbf{L} = \mathbf{K}_p(\phi_d - \phi) + \mathbf{K}_d(p_d - p) \tag{6}$$

where $\mathbf{L}$ is the moment about the x-axis, $\mathbf{K}_p$ is the proportional gain, $\mathbf{K}_d$ the derivative gain, p the roll rate and ϕ the roll angle. The pitch and yaw motion is controlled using an expression similar to 6. The difference is that the desired pitch and yaw angle was zero.

The second controller is a model-based linearizing (MBL) controller. Its objective is to cancel the nonlinear terms in the equation of motion of the robot. The vehicle model can be found in [19], [20]. The controller took the following form:

$$\begin{aligned} \mathbf{L} = &\ L_{\dot{p}}\dot{p}_d + (\mathbf{I}_{zz} + \mathbf{M}_{\dot{q}} - \mathbf{I}_{yy} - \mathbf{N}_{\dot{r}})qr + \mathbf{K}_p(\phi_d - \phi) \\ &+ \mathbf{K}_d(p_d - p) - \mathbf{L}_{p^2}|p|p \end{aligned} \tag{7}$$

where $\mathbf{I}_{zz}$ and $\mathbf{I}_{yy}$ are moments of inertia, $\mathbf{M}_{\dot{q}}, \mathbf{N}_{\dot{r}}$, and $\mathbf{L}_{p^2}$ are hydrodynamic derivatives, q is the pitch rate and r is the yaw rate. The two nonlinear terms of Eq. 7 are the Coriolis force (second term) and the hydrodynamic damping force (third term). They cancel the Coriolis and hydrodynamic damping force present in the equation of motion of the robot, giving a linear system.

Only a roll trajectory is used for two reasons. A magnetic compass is used for yaw measurement in the robot, and the magnets in the motors, along with other electromagnetic disturbances, introduce significant noise in this measurement. For measuring the roll, we use the IMU, and hence the sensing in roll is the most accurate while sensing the yaw is

much noisier. Second, as the robot was operating in shallow water a pitch trajectory would have brought the robot close to the sea floor.

Fig. 6(a,b) show how the robot was able to follow the prescribed trajectory. The settling time was found to be around 3.2 seconds for the PD controller and 0.7 second for the MBL controller. We can also observe an overshoot with the PD controller. Based on these results, we can conclude that the MBL outperforms the PD.

Selecting appropriate gains for the implemented controllers depends on the type of the environment the robot operates in. Different gains are used at depth and different ones when the robot is at the edge of the water, or on sand. Next, we present an automated approach for environment identification.

IV. DATA GATHERING FOR ENVIRONMENT IDENTIFICATION

One ongoing goal is to simplify and improve environment detection for mobile robots, as demonstrated by Giguère *et al.* [21]. Terrain identification is a crucial task for autonomous legged robots. It enables the possibility to modify or switch gaits when terrain conditions require it. Mapping capabilities are also improved by providing extra information.

During beach trials, we collect robot sensor data as it walked on different types of beach environments (dry sand, wet sand, water). Data is also collected as the robot freely bobbed in the water's surf zone. Transitions between terrain types are also collected. For example, Fig. 7 shows on-board sensor signals as the robot walks away from wet sand near the water, towards the drier section of the beach. Changes in signal patterns are perceptible as the transition is crossed.

Increasing the autonomy of terrain detection and discovery requires improvement in clustering methods. Some of our efforts concentrate on the development of clustering algorithms exploiting time-dependencies between samples. This time-dependency arises from continuity of terrains in real world. The clustering problem is also simplified by reducing the dimensionality of the sensor data. This is accomplished

by concatenating the sensor data collected over full gait periods, and then applying Principal Component Analysis (*PCA*). Figure 8 shows the two principal components for the combined rolling angular velocity and six motor currents, for the same sequence as shown in Fig. 7.
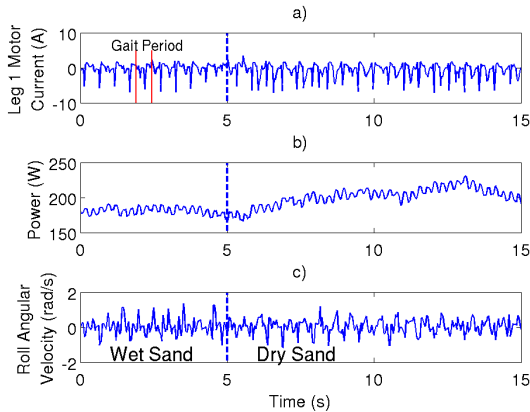

Fig. 7.   a) leg 1 motor current, b) average power consumption and c) rolling angular velocity as the robot moves from hard wet sand to a drier and softer section of the beach. The transition occurs around time = 5 *s*. Significant changes in signal patterns can be seen. Time duration of a single gait period is shown in a) with a pair of vertical red lines.
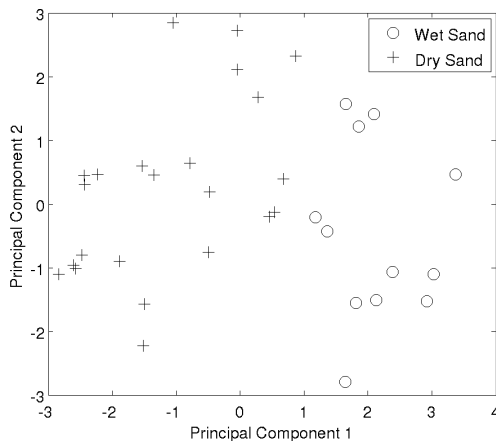

Fig. 8.   Reduced dimensionality for the sequence displayed in Fig. 7. Each point represents the data after performing PCA, for complete gait periods. The initial data is the concatenation of roll gyro rate and six motor currents estimator sampled 23 times each, over a complete gait period. The clusters are non-overlapping, yet are not necessarily easy to distinguish without labels.

V. VISION-GUIDED TETHERLESS OPERATIONS

The Aqua robot is capable of being visually guided, by active visual servoing on colour targets, and also by visual interaction with a human operator using engineered visual patterns. The pure visual servoing system is based on a colour-feature tracker and a PID controller pair. This helps the robot to track and follow targets of interest. In this case, the diver can only modulate the activities of the robot by

making gestures that are interpreted by a human operator on the surface. In the following two subsections, we briefly discuss a visual human-robot communication paradigm, and the tetherless operation mode that results from using that scheme.

A. Visual Human-Robot Interaction

While the servoing system enables the robot to track human scuba divers, a human-operator still needs to remain in the control loop to modify the behaviour of the robot, in particular when instructed to do so by an accompanying diver. To address this issue, we created a symbolic visual control scheme for a human operator (i.e. a diver) to control the behaviour of the robot, without the need of any human operators in the loop. Our visual communication scheme, called *RoboChat* [22], decodes fiducial tags (bar-code like markers) as input signals to the robot. RoboChat has been designed as a *visual language*, with a strict grammar that specifies the input tokens to the language, as well as the form of legal sentences. These sentences can be atomic commands (*i.e.* "turn left"), complex sequence of commands, or even semantically higher-level constructs (*i.e.* "go to the reef 50 feet from the shore and perform a transect"). RoboChat does not depend on the actual fiducial scheme being used to signal the robot, although currently we are using the ARTag [23] marker scheme for our operations. The RoboChat vocabulary is quite expressive, and thus requires only a small number of tags to embed all the necessary commands required by a diver to operate the robot. Furthermore, by incorporating gestures into the RoboChat scheme, the diver only needs to carry 2 to 3 markers, but is still able to express a large number of commands. In such cases, the different gestures as well as the properties of the performed gestures themselves contribute to the large vocabulary size. While we do not have a real-time implementation of the gestures on-board the robot yet, it has a working implementation of the basic RoboChat scheme. This ability to directly communicate with the robot renders the fiber-optic tether redundant for robot control, and indeed, the robot is currently capable of operating tetherlessly, controlled solely by the tags and the RoboChat scheme.

B. Vision-guided autonomous control

With the aid of the RoboChat scheme, together with the visual tracking and servoing mechanism, the robot can demonstrate the ability to operate without a tethered remote controller. The vision computer runs a client program that implements a subset of the RoboChat language, and also the visual tracking and servoing algorithms. The robot controller code is a different executable that runs on the control computer, and at power-up, both these programs come on-line. The vision client immediately goes into tag detection mode and waits for the human operator to show it tags. Once it detects a valid tag (or a set of tags, correctness of which is enforced by the RoboChat language grammar), the vision client communicates with the control client over the network using the UDP protocol and sends robot behaviour
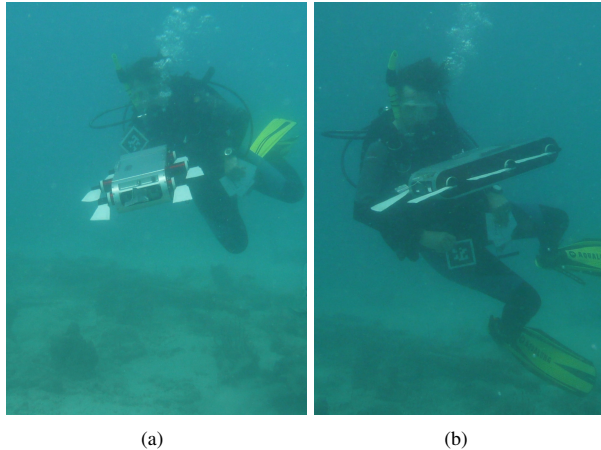
(a)   (b)

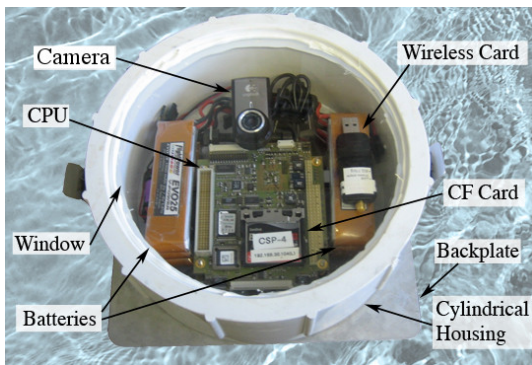Fig. 9.   (a) Robot is instructed to turn right. (b) The robot swims to the right

Fig. 10.   An annotated photo of an underwater sensor node.

commands, and reads back robot responses. Once put into swimming mode, the vision client has affordance over the five controllable degrees of motion of the robot, and can also engage the visual servoing system to track and follow an object underwater in full autonomous mode. An example of open-water tetherless operations can be seen in Fig. 9, where the robot makes a 90-degrees right-turn after being instructed by the diver.

VI. CAMERA SENSOR NODES

The Aqua robots were augmented with a set of underwater sensor nodes with capabilities of recording visual data. The data collected can be used independently for monitoring sections of the coral reef, or in collaboration with Aqua for estimating the pose of Aqua with respect to the sensor nodes.

A. Technical Details

At the heart of each node is an industrial computer in the PC/104 form-factor, of Pentium-M class. This is powered using a bank of Lithium-Ion rechargeable batteries, which give more than five hours of operational life in the current configuration. The board connects to a USB2-class high-bandwidth camera, which records video data at 2.0MP at 3Hz, and stores the data on an on-board CompactFlash memory card. A reed switch allows for the system to be turned on and off while underwater. There is a wireless card for 802.11g connectivity, when the nodes are above water.
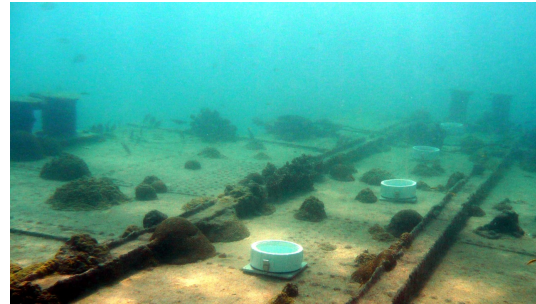

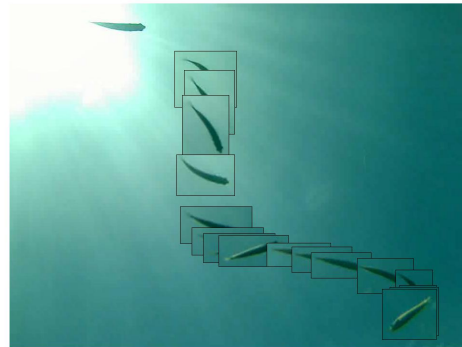Fig. 11.   A set of nodes placed on top of the "Joly Rogers" wrecked barge.

Fig. 12.   A temporal mosaic of a fish (of the Wrasse family) swimming in the field of view of a sensor node.

The entire system is enclosed in a custom waterproof case as shown in Fig. 10. The nodes were operated routinely at a depth of up to 8 meters, while the waterproof case was tested up to a depth of 18 meters.

B. Experimental work

During the sea trials, the nodes were deployed in various configurations and in various locations. They remained stationary during the data capture phase. Many hours of video data was recorded of underwater activity such as scuba divers, schools of fish and, of course, the robot swimming. The data is being analyzed and using advanced tracking algorithms [24], the objects in the video are tracked offline, see Fig. 12, and their paths will be used to mosaic the node images and self-localize. It is also hoped that it will be possible to localize the robot using the data from the nodes.

C. Future work

Once the tracking algorithm is complete, it will be installed in the nodes themselves. This way the processing can happen in real time, while making use of much less data. One scenario where this would be useful is if we wish to have to robot communicate with the nodes underwater. Since underwater communication is slow, it is important that the nodes only transmit the most useful information. If they are counting fish for a marine biology experiment, it is much more efficient for the nodes to process the data and just transmit a number than to transmit the entire video file.

Future hardware upgrades, similar to [25] include: An accelerometer, to monitor possible motion due to currents; a depth gauge; an exterior thermometer; an inductive sensor to
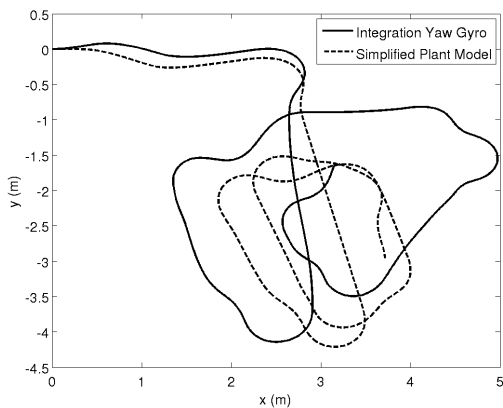
Fig. 13. Estimated 2-D trajectory of the robot using the internal gyro and a simplified plant model for a sequence of 35 *seconds*. The simplified model was a first-order low-pass filter. More sophisticated models for this robot are available and will be employed.

detect when a robot is in proximity; and an LED panel for communication with a robot or a human.

VII. Conclusions

This paper summarizes our efforts in creating an autonomous underwater legged vehicle. The project represents ongoing research over the past few years, and as we have mentioned, a significant amount of work needs to be done to create a truly autonomous underwater mobile robot. In the domain of sensor networks, we have presented preliminary work on image mosaicing from multiple sensor nodes. One future goal of our research is to achieve mutual localization of the robot and the nodes, using image data acquired from both entities. Also, an important step towards autonomous reef monitoring requires the robot to perform simultaneous localization and mapping (SLAM) in real-time. We have acquired monocular footage of coral reef from the robot's camera as it performed a transect over the reefs. We aim to use this data to develop algorithms which will give the robot the ability to localize in underwater environments, as well as map reefs to some degree of accuracy. Preliminary results comparing the IMU data with a simple plant-model of the robot can be seen in Fig. 13. For untethered operation, a gesture-based visual language scheme is currently being implemented for on-board operations. As further future work, we are looking at environmental sensing and mapping to automatically switch between different gaits as the robot transits from one type of surface to another.

References

[1] C. Prahacs, A. Saunders, M. K. Smith, D. McMordie, and M. Buehler, "Towards legged amphibious mobile robotics," in *The Inaugural Canadian Design Engineering Network (CDEN) Design Conf.*, Montreal, Canada, July 2004.

[2] G. Dudek, M. Jenkin, C. Prahacs, A. Hogue, J. Sattar, P. Giguère, A. German, H. Liu, S. Saunderson, A. Ripsman, S. Simhon, L. A. Torres-Mendez, E. Milios, P. Zhang, and I. Rekleitis, "A visually guided swimming robot," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Edmonton, Alberta, Canada, August 2005.

[3] M. Theberge and G. Dudek, "Gone swimmin'," *IEEE Spectrum*, pp. 38–43, June 2006.

[4] G. Dudek, P. Giguere, C. Prahacs, S. Saunderson, J. Sattar, L.-A. Torres-Mendez, M. Jenkin, A. German, A. Hogue, A. Ripsman, , J. Zacher, E. Milios, H. Liu, , P. Zhang, M. Buehler, and C. Georgiades, "Aqua: An amphibious autonomous robot," *IEEE Computer Magazine*, vol. 40, no. 1, January 2007.

[5] R. Altendorfer, N. Moore, H. Komsuoglu, M. Buehler, H. B. Jr., D. McMordie, U. Saranli, R. Full, and D. E. Koditschek, "RHex: A biologically inspired hexapod runner," *Autonomous Robots*, vol. 11, pp. 207–213, 2001.

[6] J. Yuh, "Control of underwater robotic vehicles," in *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS)*, July 1993.

[7] D. Smallwood and L. Whitcom, "Model-based dynamic positioning of underwater robotic vehicles: Theory and experiment," *IEEE Journal of Ocean Engineering*, vol. 29, no. 1, pp. 169–186, January 2004.

[8] P. E. ao and A. Pascoal, "3d path following for autonomous underwater vehicl," in *Proc. of the IEEE Conf. on Decision and Control*, vol. 3, 2000, pp. 2977–2982.

[9] C. Georgiades, M. Buehler, and M. Nahon, "Simulation and control of an underwater hexapod robot," in *Proc. of 14th Int. Symp. Unmanned Untethered Submersible Technology*, 2005.

[10] S. A. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, October 1996.

[11] S. Lavallee, L. Brunie, B. Mazier, and P. Cinquin, "Matching of medical images for computed and robot assisted surgery," in *Proc. of the Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society, Vol.13*, 1991, pp. 39–40.

[12] J. Sattar, P. Giguere, G. Dudek, and C. Prahacs, "A visual servoing system for an aquatic swimming robot," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, August 2005, pp. 1483–1488.

[13] D. Kortenkamp, E. Huber, and P. Bonasso, "Recognizing and interpreting gestures on a mobile robot," in *13th National Conf. on Artifical Intelligence*, 1996.

[14] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, "Spatial language for human-robot dialogs," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 34, no. 2, pp. 154–167, May 2004.

[15] C. Georgiades, A. German, A. Hogue, H. Liu, C. Prahacs, A. Ripsman, R. Sim, L. A. Torres-Mendez, P. Zhang, M. Buehler, G. Dudek, M. Jenkin, and E. Milios, "AQUA: An aquatic walking robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 4, Sendai, Japan, September 2004, pp. 3525–3531.

[16] A. H. Techet, F. S. Hover, and M. S. Triantafyllou, "Separation and turbulence control in biomimetic flows," *Flow, Turbulence and Combustion*, vol. 71, pp. 105–118, October 2003.

[17] S. Licht, V. Polidoro, M. Flores, F. Hover, and M. S. Triantafyllou, "Design and projected performance of a flapping foil AUV," *IEEE Journal Of Oceanic Engineering*, vol. 29, no. 3, pp. 786–794, 2004.

[18] O. Chiu, M. Nahon, and N. Plamondon, "Stability augmentation for a hexapod underwater vehicle," in *Int. Symposium on Unmanned Untethered Submersible Technology (UUST)*, Durham, NH USA, August 19th -22nd 2007, pp. 119–127.

[19] N. Plamondon and M. Nahon, "Control of an underwater robot using flexible oscillating paddles," in *Proc. of the 15th Int. Symposium on Unmanned Untethered Submersible Technology*, NH, 2007.

[20] C. Georgiades, "Simulation and control of an underwater hexapod robot," Master's thesis, McGill University, Montréal, Canada, 2005.

[21] P. Giguere, G. Dudek, C. Prahacs, and S. Saunderson, "Environment identification for a running robot using inertial and actuator cues," in *Proc. of Robotics Science and System (RSS 2006)*, August 2006.

[22] G. Dudek, J. Sattar, and A. Xu, "A visual language for robot control and programming: A human-interface study," in *Proc. of the Int. Conf. on Robotics and Automation ICRA*, Italy, 2007.

[23] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *CVPR '05: Proc. of the 2005 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 590–596.

[24] G. Stein, "Tracking from multiple view points: Self-calibration of space and time," *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 521–527, 1999.

[25] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," *Proc. of the 3rd Int. Conf. on Embedded networked sensor systems*, pp. 154–165, 2005.