3D Trajectory Synthesis and Control for a Legged Swimming Robot

David Meger[1], Florian Shkurti[1], David Cortés Poza[1], Philippe Giguère[2] and Gregory Dudek[1]

Abstract— Inspection and exploration of complex underwater structures requires the development of agile and easy to program platforms. In this paper, we describe a system that enables the deployment of an autonomous underwater vehicle in 3D environments proximal to the ocean bottom. Unlike many previous approaches, our solution: uses oscillating hydrofoil propulsion; allows for stable control of the robot's motion and sensor directions; allows human operators to specify detailed trajectories in a natural fashion; and has been successfully demonstrated as a holistic system in the open ocean near both coral reefs and a sunken cargo ship. A key component of our system is the 3D control of a hexapod swimming robot, which can move the vehicle through agile sequences of orientations despite challenging marine conditions. We present two methods to easily generate robot trajectories appropriate for deployments in close proximity to challenging contours of the sea floor. Both offline recording of trajectories using augmented reality and online placement of fiducial tags in the marine environment are shown to have desirable properties, with complementary strengths and weaknesses. Finally, qualitative and quantitative results of the 3D control system are presented.

Fig. 1. The Aqua robot completes a vertical climb alongside a cargo vessel that rests on the ocean floor, prompted by a visual marker placed next to the ship by a human operator, seen near A. This is one of many new stable behaviors made possible by our technique. Image best viewed in color.

I. INTRODUCTION

This paper presents a number of tightly integrated control and human robot interface (HRI) components, developed for the hexapod autonomous underwater vehicle (AUV), Aqua [1], which enable agile, 3D, terrain-adapted trajectories. A 6-DOF motion controller extends the previous quasi-planar 2D operating modes of our vehicle while maintaining stability and responsiveness. Additionally, we describe two HRI paradigms that are convenient enough to be deployed by non-expert and preoccupied support divers but also rich enough to describe dynamic motions through space. The combined system has been evaluated during a recent sea trial where it performed numerous tasks that would have been impossible for previous iterations of our vehicle (e.g., autonomously tracing the 3D contour of a shipwreck, shown in Figure 1).

The core of our method is a trajectory controller that allows a swimming robot propelled by oscillating hydrofoils to track almost arbitrary 6-DOF motions rapidly, but stably. Examples of these motions are provided in Figure 2. Aqua's propulsion system is based on the periodic motion (i.e., *kicking*) of 6 independent flippers, so a careful accounting of forces and attitudes is required. Our controller combines robust computation of 6-DOF quantities, such as the body-frame commands required to meet world-frame constraints, spherical linear interpolation to find shortest paths in rotation-space, and model-free feedback control. To address

the practical challenge of tuning controller gains in new and variable environmental conditions, our system performs semi-autonomous gain selection, with the robot searching subsets of the parameter space interspersed with oversight from an operator.

The trajectory controller opens the door for countless behaviors and applications ranging from fish tracking to hull inspection. While a full sensor autonomy system would be well served by such a motion controller, in the current work we consider the common model of a collaborative team of human divers and their robot assistant. Underwater HRI is well-documented to be challenging due to the limited mobility and cognitive load facing human divers. We present two HRI schemes based on visual fiducial markers: design via augmented reality and terrain-based tag programming. Both methods are simple to understand and well-suited to describing the types of 3D motion that we seek. A key aspect of this work is that it allows a human mediator (i.e. a scuba diver) to specify the robot trajectories in real time and to adapt trajectories to the undersea environment.

The next section will describe existing systems and highlight the novelty of the work presented here. We will then describe our legged swimming platform and the novel 3D control regime that we have employed. Section IV describes the HRI programming tools that we have developed. Our experimental section provides quantitative analysis of the control performance and reports the results of deploying the HRI approaches in complex subsea locales. Finally, we will outline future work.

[1]Mobile Robotics Lab, Centre for Intelligent Machines, McGill University `{dmeger,florian,davidcp,dudek}@cim.mcgill.ca`
[2]Département d'informatique et génie logiciel, Faculté de science et génie, Université Laval `philippe.giguere@ift.ulaval.ca`
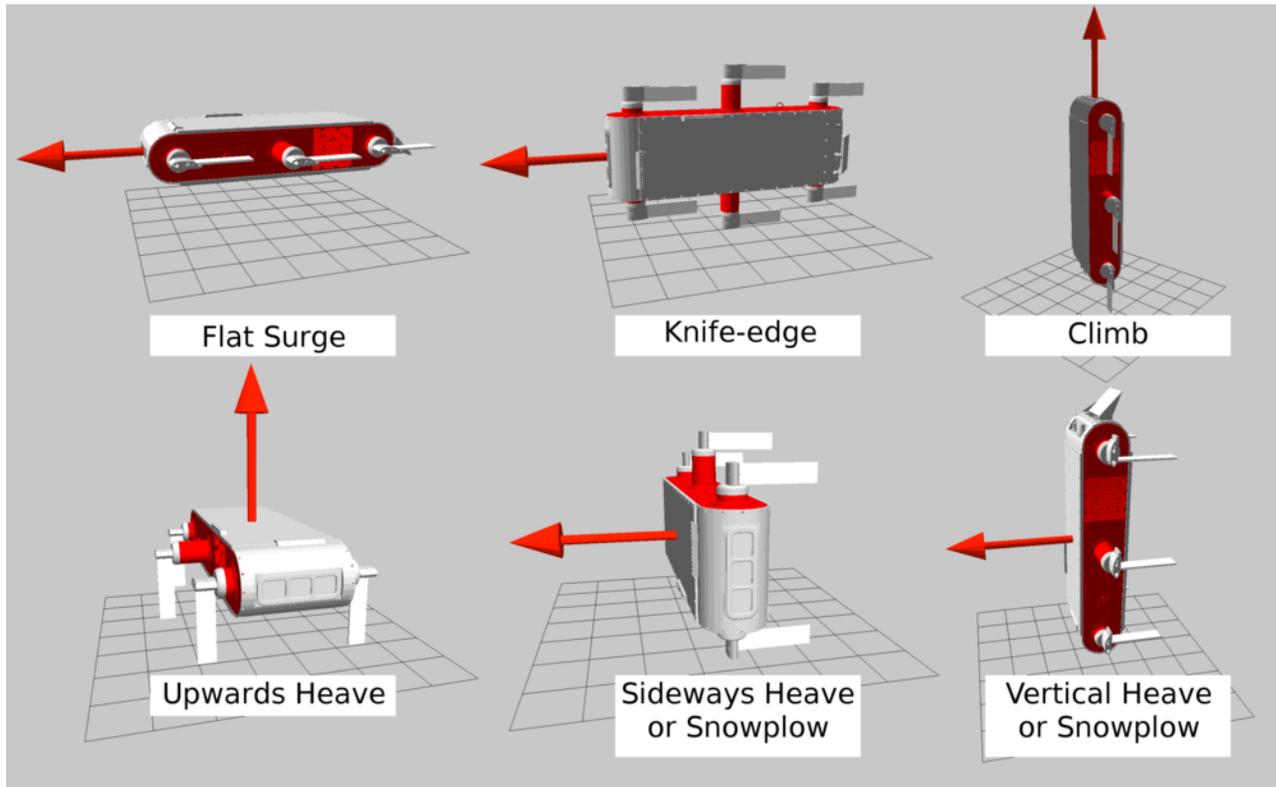
Fig. 2. Six example motions shown are from the larger set achievable by our 3D controller. Red arrows indicate directions of motion, explicitly named surge (robot's X direction) and heave (robot's Z direction), as indicated. No sway motions (robot's Y direction) are shown since our robot cannot produce thrust in this dimension. Previous work on the Aqua vehicle almost exclusively focused on motions in the left-most column, which also represent the least challenging control problems.

II. RELATED WORK

The problem of designing and controlling stable AUVs has been studied by many authors [2], [3], [4], [5] on a variety of platforms. While techniques range widely depending on vehicle characteristics and goals, feedback control on orientation sensors is a common theme. While controllers based on fluid dynamics properties can be accurate, they are also often complex. In our case, deployment on a physical robot with limited computing resources was essential, precluding intensive flow dynamic simulations. Rather, we employ a simple, but practical linear feedback controller [6].

Previous authors have undertaken hydrodynamic motion analysis for the Aqua vehicle itself [7], [8], [9]. This work has yielded several versions of a model-based controller for simple input commands. These controllers allowed for only one angular dimension at a time to vary from the standard downward looking configuration and considered a single fixed forward thrust.

Humans dynamically adapt the way we move based on conditions, and this same behaviour is crucial for robots. In the underwater domain, [10] notes that gain scheduling is needed for stable control of a swimming robot across multiple speeds due to nonlinear drag effects. Online parameter tuning for a swimming vehicle controller was suggested by [11], although their work considers only the theoretical aspects of the problem and only discusses the optimization of a single parameter value. We consider control of the vehicle at arbitrary body orientations and at a variety of speeds (consider Figure 2). This introduces a wide range of disturbances, and modes of propulsion, and has motivated our development of semi-autonomous gain selection.

Our work differs notably from prior studies that use cameras for semi-autonomous underwater operation [1], [12], [13], [14], [15], in particular due to the complex and vision-guided trajectories we consider. As we descend very near to underwater structures, 3D motions such as those we develop are essential to avoid the plentiful obstacles and the paths between obstacles become more complex to design, requiring HRI solutions.

III. 3D TRAJECTORY CONTROL FOR LEGGED SWIMMING ROBOTS

This section describes a feedback controller that allows our vehicle to swim precise trajectories in the space of 3D attitude and depth. This module closes the loop around the vehicle's sensors, producing flipper rotation (kicking) commands based on the error between the robot's current and target states. We build upon an existing low-level gait generator, which translates desired thrusts in the robot frame into flipper motions in an open loop fashion. The gait generator has previously been paired with a 2D attitude controller that achieved stable flat swimming [1]. This paper
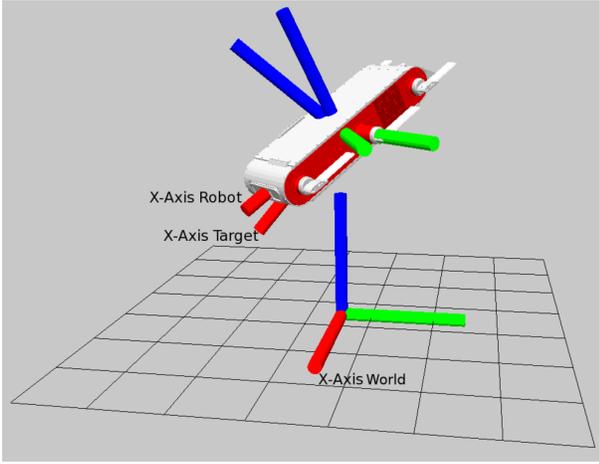
Fig. 3. The 3D coordinate frames involved in our control problem.

Fig. 4. An overview schematic of our vehicle controller.

represents the first demonstration of controlling the Aqua vehicle under arbitrary 3D attitudes.

Control in 3D requires mapping all necessary sensor and motion data into a common coordinate frame, as visualized in Figure 3. We correct positional errors in depth using computations robust to being upside-down or vertical. In order to stabilize motions and to reject disturbances, we implement PD control using an inertial measurement unit (IMU) and a depth sensor as feedback. As our robot is non-holonomic, jointly satisfying target depth and attitude are conflicting goals. We employ a cascade controller [6], composed of a depth loop surrounding an attitude loop, to mediate between factors. This prioritizes depth control, which is important for safe operations underwater. The final component of our system is a semi-autonomous gain adaptation behaviour that allows human operators to more easily optimize controllers for a family of motions and to adapt to changing environmental conditions.

A. Hexapod Swimming Gaits

The mapping between a desired motion of the robot's body and the required leg movements is highly complex, as it involves interactions in the force-domain between the various legs, the hydrodynamics created when vortices from one leg are "shed" to the next, and interactions with exogenous factors such as ocean currents. Simple swimming strategies, called gaits, have been developed in previous publications (e.g., [1]). In this work, we re-use the existing gait module and describe its properties and inputs here as background.

Gaits rely on hand-crafted behaviors that coordinate motions of the six flippers. These behaviors are based on an abstraction that approximates the flippers as steerable thrusters. Under this approximation, absolute thrust can be controlled via the oscillation amplitude and its direction by changing an offset angle around which the oscillation is performed. By combining the action of these six virtual thrusters, simple rate commands (either angular or translational) can be executed in open-loop. For example, a pure rolling motion can be
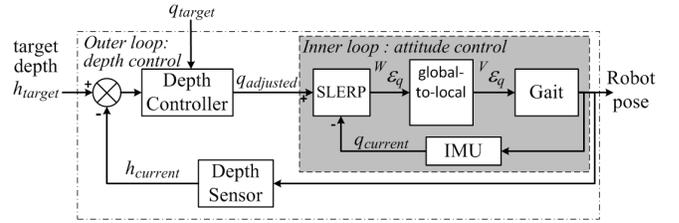
achieved by having all the left virtual thrusters pointing upwards and the right virtual thrusters pointing downward.

The gait controller accepts 5 of 6 possible instantaneous thrusts. Sway, along the robot's Y axis, is not possible due to the physical orientation of the flippers, making this vehicle non-holonomic. Therefore, the input to the gait module is 5 vehicle frame desired thrusts at each time: roll, pitch, yaw, heave and surge. Due to inherent approximations and the need to re-orient the flippers to achieve a new thrust, the actual response of the robot to any requested motion is difficult to model and may have complex time-domain properties. This motivates our pursuit of a closed-loop attitude controller to drive the gaits.

B. Feedback Control for 3D Swimming

1) Inputs: Our controller receives target trajectories from the users and data from the sensors as its inputs. A trajectory is an n length sequence, $p^n = \{p_1, ..., p_n\}$, where each element, $p_i = \{q_i, f_i, h_i\}$, specifies a target global frame orientation in quaternion form, q_{target}, vehicle frame heave and surge thrusts, f_{target} and depth below the water's surface, h_{target}. A 3DM-GX1 IMU from Microstrain Inc. measures the vehicle's orientation, acceleration, and angular velocity. In this work, we utilize only the orientation reading as a quaternion, $_V^W q$, which represents the vehicle's attitude in the IMU's reference (or *world*) frame. A pressure sensor reads the depth below the surface of the water, $h_{current}$.

2) Coordinate Frames: The first step in achieving 3D control is the correct handling of the multiple frames of reference related to various system quantities. The IMU's reference frame, defined by the gravity vector, acts as our world coordinates. This implies that the ideal behavior is for the robot to move such that the IMU readings exactly match the commanded orientations. The robot's gait controller accepts instantaneous vehicle frame velocity commands, so we must robustly transform geometric information between the two frames. Our depth sensor is located at the back of the vehicle, so its readings must be corrected based upon the vehicle's current orientation.

3) Depth Control Loop: The *depth control* loop, shown in Figure 4, is responsible for modifying the robot's orientation such that it will climb or descend to maintain a specified target depth, h_{target}. The robot can produce a motion along the global Z axis by modifying its orientation to direct thrust as needed. In symbols, the target orientation q_{target} is modified to achieve this Z motion via a depth-correcting

rotation q_{depth}. The final target angle to the inner controller takes the form

$$q_{adjusted} = q_{depth} \otimes q_{target} \qquad (1)$$

with $\otimes$ representing quaternion multiplication. We derive q_{depth} by forming a quaternion from the unit vector, k, and angle of rotation about that vector, θ, as

$$q_{depth} = \begin{bmatrix} k_x sin(\theta/2) \\ k_y sin(\theta/2) \\ k_z sin(\theta/2) \\ cos(\theta/2) \end{bmatrix}. \qquad (2)$$

The angle of rotation, specified by vector k, is selected to achieve climb or descent as needed, accounting for the current thrust direction. This desired vector is perpendicular to both the vehicle's world frame thrust and the world Z axis, so it can be computed using the cross product

$$k = ({}^W_V q \, {}^V f_{target}) \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \qquad (3)$$

The magnitude of rotation towards the desired depth, θ, is determined by a PD controller, based on the current depth error, $\varepsilon_h = h_{target} - h_{current}$, as

$$\theta = K_P^{depth} \varepsilon_h - K_D^{depth} \frac{d\varepsilon_h}{dt}. \qquad (4)$$

This correction angle is capped to $\pm 45°$ to limit the climb rate of the robot, which is desirable when operated in conjunction with human divers who cannot safely change depth rapidly. Our nested controller can become susceptible to oscillation, as there are two dependent levels of feedback occurring. For this reason, we have found that derivative control within the depth loop, K_D^{depth} is essential.

4) Attitude Control Loop: Our controller's inner loop attempts to generate gait commands that bring the orientation of the vehicle, $q_{current}$, to the orientation requested by the outer loop, $q_{adjusted}$. Each of these quantities is a rotation in 3D and so their difference, our control error, cannot simply be computed by subtraction. Instead, we consider the distance of the shortest path through the space $SO(3)$. Topographically, this shortest path is the *great circle* along the unit sphere which connects the unit vectors representing the two orientations. Our goal is to drive the robot along such a path through angle-space until it arrives at the target angle $q_{adjusted}$.

In order to send instantaneous commands, we extract the current local orientation change needed to progress along the angle-space path. This interpolation of the path into small segments along the great circle is referred to as spherical linear interpolation (SLERP) [16]. It produces a world frame rotation error

$$^W\varepsilon_q = SLERP(q_{current}, q_{adjusted}, u) \qquad (5)$$

with interpolation parameter u controlling the size of the angle step. The underlying gait controller accepts vehicle

frame thrust commands, so this world frame angle step must be transformed into the robot's coordinates

$$^V\varepsilon_q = {}^V_W q \, {}^W\varepsilon_q. \qquad (6)$$

Finally, we extract the components of $^V\varepsilon_q$ in the roll, pitch and yaw Euler angle representation accepted by the gait module: ε_r, ε_p and ε_y. PD feedback control is applied in order to output unitless rate commands compatible with the gait module, in the roll, pitch and yaw axes

$$r_{gait} = K_P^r \varepsilon_r - K_D^r \frac{d\varepsilon_r}{dt} \qquad (7)$$

$$p_{gait} = K_P^p \varepsilon_p - K_D^p \frac{d\varepsilon_p}{dt} \qquad (8)$$

$$y_{gait} = K_P^y \varepsilon_y - K_D^y \frac{d\varepsilon_y}{dt} \qquad (9)$$

In order to reduce the noise amplification effects of signal differencing, a first-order Infinite Impulse Response filter with a time-constant period on the order of one cycle of leg motion ($\sim 250 \, ms$) is applied to smooth each signal. This also alleviates the issue of unwanted oscillations due to the periodic nature of the gait.

C. Online Gain Adaptation

Our robot is non-spherical, which causes highly varying disturbances as the robot changes orientation. For example, swimming directly forward causes the narrow and rounded front shell to cut through the water, while so called *heave* motions push large amounts of water with the flat top of the robot. The measured drag ratio between these different motions is roughly a full order of magnitude (10x). To achieve stable motion in both cases, different sets of controller gains are required for each motion, slowing the development of an agile system. To address this, we have developed a semi-automated procedure to select our controller's gains in novel situations.

The robot adapts its controller gains while swimming by searching for parameters that minimize control error over short sequences. While numerous parameter search methods could be viable, we have implemented an efficient coordinate descent, which selects one parameter dimension at a time, line searching over values of each dimension, fixing the minimum value obtained, and then moving to the next dimension. We will describe the efficacy of this search procedure in our experimental results, but for now we make several observations about its desirable properties. We note that our approach can operate without a model of the robot's control system or its disturbances – it is model-free. Coordinate descent also has the important practical advantage of maintaining the majority of parameter dimensions unchanged in any given step, so that the robot's effective behavior changes slowly and safety can be maintained. In underwater robotics, the safety of nearby humans and the marine ecosystem are paramount. A final advantage of modifying a single parameter at a time is to build the confidence of human operators and to facilitate offline interpretation of the test results by a human, an important consideration in mixed-initiative robot operations.
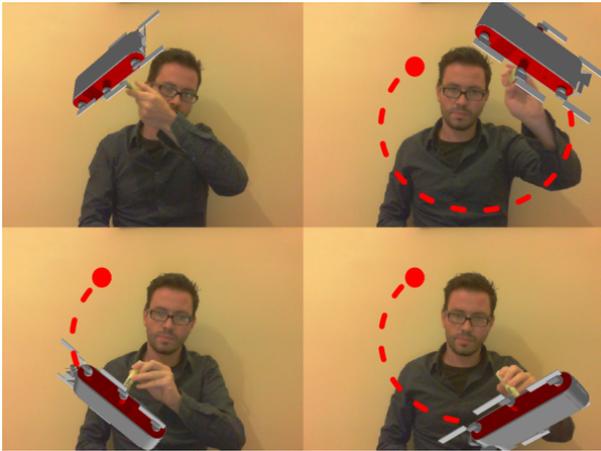
Fig. 5. A human operator specifies a looping trajectory using our augmented reality interface. Note that the swimming robot that appears in each image is rendered by computer graphics and is not physically present. The rendered pose of the robot reflects a point along the trajectory, which is stored in a format suitable for input to our trajectory controller. Later, the robot replays the specified path with high fidelity, in this case swimming in a vertical loop.

IV. SYNTHESIZING 3D UNDERWATER TRAJECTORIES

In challenging underwater environments, semi-autonomous operation is often preferred. Full sensor-guided autonomy is not within the reach of most platforms due to cost, reliability, or other limitations. For example, the close-to-object operations that we study here must occur in relatively deep water, which precludes the use of a GPS antenna and acoustic underwater localizers add significant financial overhead. However, human scientists regularly participate in the day-to-day routine of underwater science and are a reliable source of guidance. We describe several techniques that allow an operator to synthesize detailed 3D underwater trajectories suitable for our vehicle, which we have developed and tested in the ocean.

A. Parameterized-Shape Trajectories

Our first approach allows a human operator to specify target robot trajectories by composing segments of parameterized shapes, such as squares and circles, with different orientation, speed and depth settings for each individual shape. In principle, by careful choice of the parameters and potentially the use of a very large number of segments, the operator is able to produce any arbitrary trajectory. However, choosing the shape family and parameters is difficult to visualize even in a lab scenario, and is essentially impossible for a diver in deep water. So, we have continued to consider several more convenient synthesis paradigms.

B. Trajectory Design via Augmented Reality

Our second synthesis approach is inspired by the notion of a human "conductor" forming paths with natural gestures that the robot mimics afterwards. To assist the operator in visualizing the scenario our system provides augmented reality feedback in the form of a virtual robot rendered on top of the live image feed. The response of the augmented reality robot allows easy comprehension of what will be the real robot's target behaviour during trajectory playback. Figure 5 provides an example of the human operator recording a trajectory. A sample of the corresponding robot motion can be seen in the video submission accompanying this paper, which is also available on the Internet[1].

Mapping human hand motion into robot swimming motion requires a translation between two different physical spaces. Several mappings have been considered during our system development and the following was reported by the operator to be well suited for the task due to its simple intuitive nature:

- The position of the tag along the camera's viewing direction is mapped to the robot depth's below the water surface.
- Changes in the position of the tag in the camera's image plane between video frames are projected onto the forward direction of the robot to determine its surge.
- Rotations of the tag are mapped to rotations of the robot.

Although we postpone full evaluation of this HRI paradigm for future work, we received significant positive feedback from test operators. The robot was able to perform a number of trajectories in the ocean, both near reefs and a sunken ship, collecting valuable sensory data while operating safely without environmental impact.

C. Terrain-Based Programming

The final trajectory synthesis method that we studied focuses on achieving very close proximity between the vehicle and the terrain. While each of the previous methods allows for the construction of trajectories that conform loosely to the shape of the sea floor, the vehicle operates without feedback during trajectory execution, which means that a reasonable safety margin must be maintained. The approach that we refer to as *Terrain-Based Programming* utilizes visual fiducial markers placed by human operators in the environment itself in order to achieve precise near-terrain trajectories. The robot detects the visual markers using its cameras, which is reliable due to the highly distinctive nature of the visual patterns and the robust image processing used for detection [17]. Each time that the robot detects a tag, it recovers an unique numeric ID and solves for the relative pose between its camera and the tag.

A vocabulary, known to the human operators, maps between available tag IDs and the matching motions. The operators construct a desired robot trajectory by choosing the position and orientation of several tags, so that each will be seen by the robot after it completes execution of the previous motion. Tags are often placed at key landmarks in the environment, for example to enable the robot to avoid a sensitive obstacle, and at points of desired robot behaviour change, for example where a local visual survey should be conducted. The relative pose between the robot

[1]Address: http://www.cim.mcgill.ca/~dmeger/IROS2014_3DTrajectories/
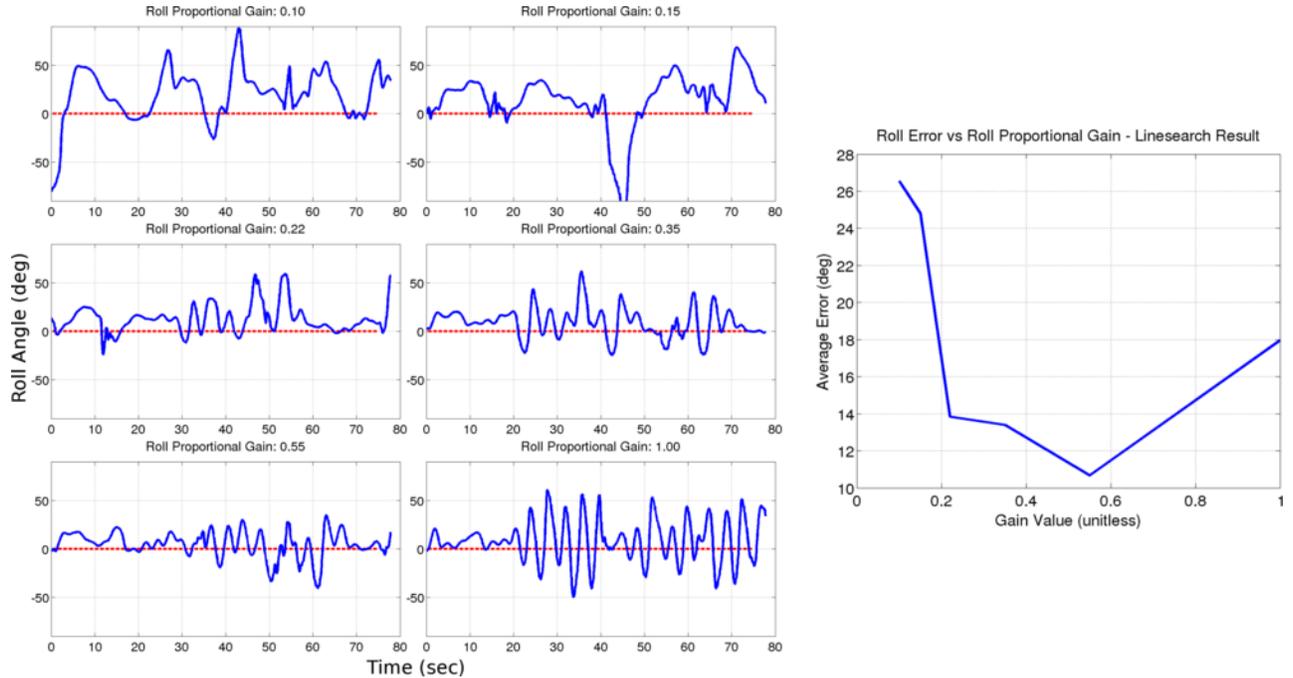
Fig. 6. Roll tracking with various gains attempted automatically by the robot during a line-search run conducted in the ocean. The error summary on the right shows that the gain value 0.55 produced minimal error and was selected. Note that the snowplow motion shown here pushes significant amounts of water in front of the robot, causing large natural disturbances, and large errors compared to other motions. This is one of the most challenging elementary motions to control on the Aqua vehicle.

and the tag acts as a global frame position correction and can influence the robot behaviour, such as to remove drift. Figure 8 provides an example of tag-based shipwreck inspection. We will discuss the specific vocabulary used and resulting robot trajectory in our experiments section.

V. EXPERIMENTS

We have evaluated our system in the open waters of the Caribbean Sea, off the coast of Barbados. In all cases, a team of divers operated the Aqua vehicle without tether, pre-loading software and parameters to be executed when the robot had been brought to sufficient depths and conditions. In Section V-A, we quantitatively study the stability achieved by our depth and attitude controller over a range of desired robot trajectories. Here we do not focus on trajectories that are particularly meaningful for the local terrain, but rather choose those that allow us to understand the system's behavior. Section V-B describes evaluation of our holistic system, including the trajectory synthesis, its flexibility and efficacy, as well as the control system's ability to achieve that trajectory.

A. 3D Control Evaluation

Our initial set of ocean trials were conducted in tracts of water with few obstacles where the stability of our controller could be verified and tuning could be done safely. The controller was driven with several fixed swimming behaviors (i.e., each trajectory was composed simply of one orientation and speed, repeated for a duration). These trials were used to observe the stability of our controller and to execute

TABLE I

TRACKING ERROR FOR VARIOUS MOTION CLASSES. BOTH HELICAL MOTIONS ROTATED THE ROBOT AT A CONSTANT $36°$ EACH SECOND ABOUT ITS ROLL AXIS.

Motion	Roll Error (deg)	Pitch Error (deg)	Yaw Error (deg)	Depth Error (m)
Flat	0.4	1.6	0.3	0.05
Knife	1.7	3.0	1.1	0.06
Snowplow	7.5	3.7	2.0	0.20
Circular Helix	7.2	6.3	6.0	0.11
Square Helix	13.0	6.6	7.3	0.17

the online parameter optimization procedure. We have summarized the average errors for a variety of both simple and composite robot motions in Table I. The increasing errors are an indication of the difficulty level associated with each motion pattern. For example, the vertical heave maneuver has a much larger drag force than the flat or knife-edge maneuvers. As mentioned earlier, this increase in drag renders motion less stable. We additionally provide results on our controllers stability in Figure 7. The motion command, a circular helix, involves simultaneous and continuous change in all angular dimensions, which is a stress-test for an attitude controller. Maintaining depth errors within $11cm$ for this behaviour indicates a relatively well-tuned system.

We report results from semi-autonomous gain selection in Figure 6. In this case, the robot attempts to line-search over the values of its roll proportional gain, $^r_P K$, to find the value which yields the minimal error swimming performance under

current conditions. In order to be robust to environmental disturbances and to allow convergence of the controller, each gain value is maintained for $T = 75s$, parameter that can be modified in the field. In this trial, the initially low gain values lead to poor tracking of the target angle. As the gain was increased, the robot moved ever faster towards the desired orientation, with the largest attempted gain leading to significant oscillatory behavior. Of the values attempted in this run, the gain 0.55 was selected, as it yielded the minimal average error over the trial. Note that this procedure was run live in the open ocean, so that all disturbances such as current, sub-optimal buoyancy, and physical variations in the robot had realistic values.

B. Synthesized Trajectory Execution

We have utilized all three of our trajectory generation methods for carrying out a selection of sample tasks in the ocean.

1) Parameterized-shape trajectories: As a proof of concept and to ensure the coverage of the full range of target robot attitudes, several trajectories were constructed as collages of parameterized shape trajectories: (a) a right-angle-based path where the robot traverses several edges of a cuboid, with a variety of orientations and camera directions; and (b) a circle-based path, where the robot orbits several central points of interest, observing them from a dense set of the possible viewing directions. The accompanying video is most appropriate for demonstrating the repertoire of motions performed in this dance-like trajectory. We were able to achieve all robot motions that were attempted, at the cost of a relatively large effort needed to produce the choreography via source code.

2) Trajectories designed via augmented reality: We recorded trajectories with our augmented reality system and played them back in the ocean at locations where collection of visual data was desirable. We asked our human operators to produce trajectories that took the robot through a wide range of visual angles, and that also varied the robot's depth significantly. Again, several example trajectories appear in our video submission, and our operators reported that they observed the robot carrying out all aspects of the task that they had intended.

3) Terrain-based programmed trajectories: We specifically focused on guiding the robot to pass into close proximity with a sunken cargo vessel, located near the coast of Barbados, in the Caribbean Sea. In order to capture the visual experience of this location with high detail, the robot swam less than 2 meters from the surface of the vessel. The robot followed the contour of the vessel due to the strategic placement of tags at the edges joining various elements of its construction:

- A tag encoding "ascend vertically to the deck" was placed on the sea floor near to each one of the ship's outer walls.
- A tag encoding "descend vertically to the sea floor" was placed at the very edge of the ship's deck, at the last
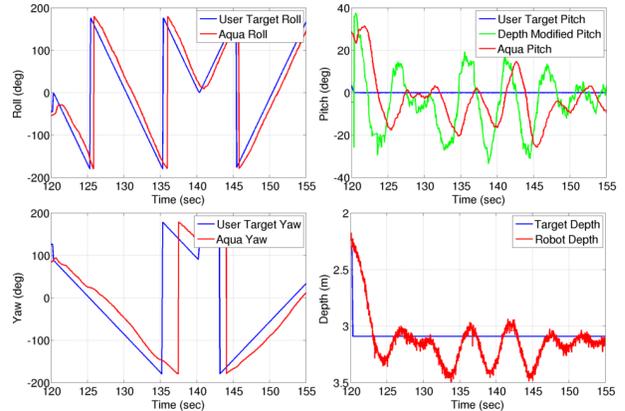

Fig. 7. Control of the acrobatic "circular helix" maneuver, one of our family of parameterized shapes. With constant forward speed, the robot rotates in roll (top-left) and yaw (bottom-left), to form a circle in space. Disturbances from a constant depth target (bottom-right) are corrected by modifying the pitch angle (top-right), through 3D-aware computation. Over the 35 second maneuver, our system achieves an average depth error of 11 cm.

location the robot passed over before heading out into open ocean.
- An "execute a U-turn" tag was placed on the ocean floor several meters away from the ship in both directions, to start the behavior again from the opposite side.

Figure 8 demonstrates a sampling of the motions made by our robot during the executing of this ship survey. The robot was able to make crisp turns to swim entirely forwards, upwards, or downwards, and this let the robot pass very near to both the flat deck of the ship, as well as to the outer walls of its hull. The data collected by the robot during this trial gives a unique perspective on the marine life present in the ship's ecosystem, and on the condition of various parts of the vessel. In the future, such trajectories may be utilized for inspecting operational vessels, to inspect for signs of damage and for security reasons. We note that the visual tags have easy-to-understand translations into robot motions and that no specialized robotics knowledge is necessary to program our system in this way. Therefore, this technology will allow teams of unspecialized divers to direct automatic survey platforms.

VI. CONCLUSIONS

We have described a system that allows our robot to survey and navigate in parts of the marine environment such as beside and between potentially complex benthic (sea floor) structures. We have presented the details of a control system for an under-actuated hydrofoil vehicle that features quaternion angle representation, cascading depth and attitude feedback loops, SLERP command smoothing and online gait selection. This system can achieve agile robot motions, but requires rich input trajectories in order to reach its full potential. Therefore, we have paired it with several human robot interaction techniques, including a conductor style augmented reality gesture interface and a spontaneously created environmental landmark map based on
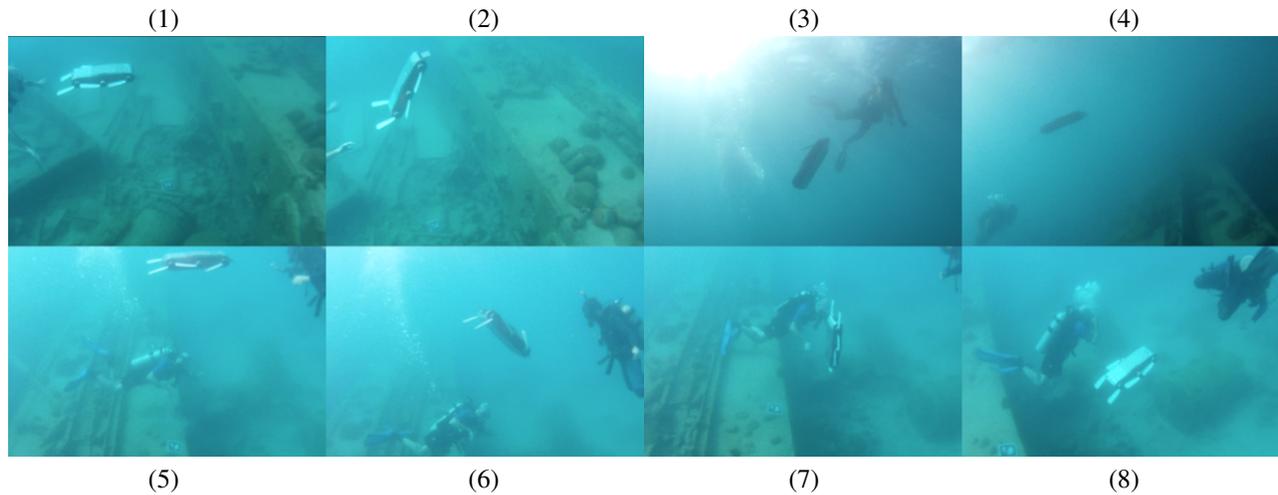
Fig. 8. A turn-by-turn account of one pass over the contour of a shipwreck. From left to right and top to bottom, the robot: (1) approaches from the sand; (2) views an "up" tag and ascends vertically along the ships hull; (3) crests at a depth above the deck; (4) swims flat above the deck; (5) views a "down" tag; (6) turns downwards; (7) descends near to the ship's opposite side; and (8) finally levels out, again swimming flat over the sandy ocean bottom. Best viewed in color.

a vocabulary of available actions. We have demonstrated the effectiveness of these approaches in the Caribbean Sea, near to several coral heads as well as a sunken cargo ship, with our robot able to achieve many desirable poses that would be challenging or impossible for existing systems.

The demands of the marine environment, especially in the context of human-robot interaction, are substantial and the interaction modalities we have outlined are merely samples what can be achieved. The HRI aspect of our system will be more formally verified and extended, potentially through use by interested marine biologists and oceanographers.

REFERENCES

[1] G. Dudek, M. Jenkin, C. Prahacs, A. Hogue, J. Sattar, P. Giguere, 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 *Proceedings of Intelligent Robots and Systems*, Edmonton, Alberta, Canada, August 2005.

[2] S. Mohan and A. Thondiyath, "A non-linear tracking control scheme for an under-actuated autonomous underwater robotic vehicle," *International Journal of Ocean System Engineering*, vol. 1, no. 3, pp. 120 − 135, 2011.

[3] J. Vaganay, L. Gurfinkel, M. Elkins, D. Jankins, and K. Shurn, "Hovering autonomous underwater vehicle - system design improvements and performance evaluation results," in *Proceedings of Unmanned Untethered Submersible Technology*, 2009.

[4] D. Boskovic and M. Krstic, "Global attitude/position regulation for underwater vehicles," *Inernational Journal of Systems Science*, vol. 30, no. 9, pp. 939 − 946.

[5] D. Font, M. Tresanchez, C. Siegentahler, T. Pallej, M. Teixid, C. Pradalier, and J. Palacin, "Design and implementation of a biomimetic turtle hydrofoil for an autonomous underwater vehicle," *Sensors*, vol. 11, no. 12, pp. 11 168–11 187, 2011.

[6] A. Visioli, *Practical PID Control*. Springer-Verlag, 2006.

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

[8] P. Giguere, G. Dudek, and C. Prahacs, "Characterization and modeling of rotational responses for an oscillating foil underwater robot," in *Proceedings of Intelligent Robots and Systems*, Beijing, China, October 2006.

[9] N. Plamondon and M. Nahon, "Trajectory tracking controller for an underwater hexapod vehicle," in *Proceedings of Oceans*, Quebec City, Canada, September 2008.

[10] P. Giguere, Y. Girdhar, and G. Dudek, "Wide-speed autopilot system for a swimming hexapod robot," in *Proceedings of Computer and Robot Vision*, 2013.

[11] G. Bartolini and A. Pisano, "Black-box position and attitude tracking for underwater vehicles by second-order sliding-mode technique," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 14, pp. 1594 − 1609, 2010.

[12] J. Sattar, P. Giguere, and G. Dudek, "Sensor-based behavior control for an autonomous underwater vehicle," *International Journal of Robotics Research (IJRR)*, vol. 28, no. 6, pp. 701–713, June 2009.

[13] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard, "Visually navigating the rms titanic with slam information filters," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[14] M. Dunbabin, K. Usher, and P. Corke, "Visual motion estimation for an autonomous underwater reef monitoring robot." in *Proceedings of Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, P. I. Corke and S. Sukkarieh, Eds., vol. 25. Springer, 2005, pp. 31–42.

[15] M. Bryson, M. Johnson-Roberson, O. Pizarro, and S. Williams, "Automated registration for multi-year robotic surveys of marine benthic habitats," in *Proceedings of Intelligent Robots and Systems*. IEEE, 2013, pp. 3344–3349.

[16] K. Shoemake, "Animating rotation with quaternion curves," in *Proceedings of ACM SIGGRAPH*, 1985.

[17] M. Fiala, "Artag, a fiducial marker system using digital techniques," in *Proceedings of Computer Vision and Pattern Recognition*, vol. 2, June 2005, pp. 590–596.