Wide-Speed Autopilot System for a Swimming Hexapod Robot

Philippe Giguere
Département d'informatique et Génie Logiciel
Laval University
Quebec City, Canada
philippe.giguere@ift.ulaval.ca

Yogesh Girdhar, Gregory Dudek
School of Computer Science
McGill University
Montreal, Canada
{yogesh,dudek}@cim.mcgill.ca

***Abstract*—For underwater swimming robots, which use the unconventional method of oscillating flippers for propulsion and control, being able to move stably at various velocities is challenging. This stable motion facilitates navigation, avoids blurring the images taken by a camera motion, and enables long-term observations of specific locations. Previous experiments with our swimming robot Aqua have shown that its autopilot system must adapt the control parameters as a function of speed. The reason is that the dynamics of both the robot and the thrusting system vary widely as a function of the overall velocity of the robot. In this paper, we present the results of manually tuning a stable autopilot system for this Aqua swimming robot. We employed a well-known technique called *gain scheduling* to allow for stable operation for velocities ranging from 0 to 40 cm/s, in real open sea conditions. Thus, our platform is now suitable for vision-based navigation in low light conditions as well as for extended observation through station-keeping. The results presented here are also a proof-of-concept that agile and reactive autonomous hovering is possible for flipper-based propulsion system.**

***Index Terms*—Underwater robotics; Oscillating Foils; Control; Gain Scheduling; Station-keeping;**

I. Introduction

Robots which can explore and survey underwater environments are useful for many applications, such as coral reef biodiversity monitoring, ship hull inspections, and surveillance. Underwater environments are inherently dangerous for humans, and hence the use of robots for such tasks is especially desirable. In particular, we are interested in using robots for inspecting coral reefs, where the goal is to adaptively sample the reef, collecting more data at visually surprising regions, and less data at previously seen regions lacking novelty. Apart from the challenge of information theoretic modeling of surprise and novelty of an observation, which we describe in [1], the other major challenge in instantiating our desired goal is to enable the underwater robot to swim at variable speeds. Although vehicles like Dorado [2] and Hugin [3] are extremely capable for surveying large underwater regions at high speed, their lack of maneuverability at low speeds makes them unsuitable for coral reef inspection task.

This work focuses on building an autopilot system for the Aqua robot [4][5], which uses six flippers for its motion rather than propellers and rudders. Aqua has 5 degrees of freedom (DoF) - yaw, pitch, roll, heave, and surge, and the proposed autopilot system allows the robot to maintain its depth and three rotational degrees of freedom, even at zero speed. Low light conditions, such as those found in deep water, requires that the camera be set to a slow shutter speed, which results in image blur in the presence of fast camera movement. Hence, a stable platform with limited oscillations is important for any vision based system in low light conditions. The autopilot presented in this work is an improvement over our previous work [1][6].

The rest of the paper goes as follows. In Section II, we present a number of underwater swimming robots as well as applications for underwater data collection. In Section III, we discuss Aqua's propulsion and proposed autopilot system, including the need for different autopilot parameters as a function of speed. In Section IV, we present the trial results in open sea, for numerous test cases. Finally, Section V concludes this paper.

II. Previous Work

A number of Autonomous Underwater Vehicles (AUV), have been designed recently with hovering capabilities in mind. Many of them employ traditional propeller-based thrusters; for example, MIT has developed the *Odyssey IV* class AUV with 4 DoF [7]. The *Hovering AUV*, a combined project between MIT and Bluefin Robotics, uses 5 thrusters to operate in 6 DoF [8]. The *SeaBED* AUV offers hovering in 3 DoF via its 4 thrusters [9]. Since system dynamics are highly dependent on the thrusting characteristic, direct comparison between our flipper-based platform and these propeller-based AUVs is not applicable.

Many swimming robots, employing propulsion strategies comparable to ours, have been developed. Menozzi et al. [10] have proposed an open loop control for a cylindrical hull shaped robot with with penguin-wing inspired fins placed in the front and the back. Each of the fins have two DoF, compared to a single DoF needed for Aqua. Crespi [11] has proposed designs of snake, boxfish and salamander as swimming robots, which use a central pattern generator model, inspired from those found in these vertebrates. Tonello [12] has studied and designed an AUV mimicking the swimming method of the Atlantic Salmon. Licht et al. [13] developed an AUV, nicknamed *Finnegan the RoboTurtle*, to study asymmetric flapping

foil propulsion efficiency. Finally, iRobot has a commercial version of a flipper-based vehicle called *Transphibian*, based on an earlier design by Nekton's research [14]. However, none of these papers has reported the design or performance of an autopilot for their system.

Autonomous underwater vehicles (AUV) have been used extensively for underwater data collections. Das et al. [15] have presented techniques to autonomously observe oceanographic features in the open ocean. Hollinger et al. [16] have studied the problem of autonomously studying underwater ship hulls by maximizing the accuracy of sonar data stream. Smith et al. [17] have looked at computing robot trajectories which maximize information gained, while minimizing the deviation from the planned path. Girdhar et al. [1] have explored coral reef inspection by fixing the robot path and only varying its speed based on a surprise score.

III. Motion Control and Autopilot

A unique characteristic of Aqua (depicted in Fig. 1) is that its six flippers serve both as propulsion mechanism as well as control surfaces. Thrust is generated by oscillating each of these flippers around an offset angle θ_{leg} with an amplitude B_{leg} and a fixed period of 0.24 *seconds*. At the same time, this offset angle θ_{leg} means that a lift force can be generated by the dynamic pressure impinging on them, if this angle is not parallel to the water flow. This is much like control surfaces on an airplane (e.g. ailerons) that are employed to rotate an airplane around its center of mass. The moments generated by the robot's flippers are highly dependent on the dynamic pressure, which can be approximated by:

$$Pressure \propto V_{surge}^2, \tag{1}$$

where V_{surge} is the forward velocity of the robot in water. Because of this relationship, the overall control and dynamic of the robot strongly depends on this forward velocity V_{surge} [6][18][19].

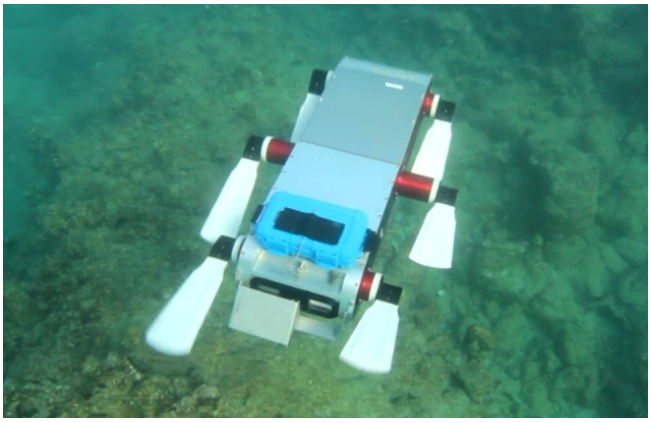

Fig. 1. Picture of the untethered Aqua robot used throughout the experiments in this paper. Photo credit: Y. Rekleitis.

Over the years, several swimming gaits that make use of θ_{leg} and B_{leg} have been developed for the Aqua robot. In particular, we developed one gait for moving the robot forward, while another (called *hovering* gait) was developed strictly for station-keeping purposes [20]. The latter gait compensates for the fact that, at $V_{surge} = 0$, the flippers can no longer generate lift forces due to the dynamic pressure of forward motion. A video showing an earlier version of the Aqua robot executing this gait can be seen in [21].

Because of the variability of dynamics of the robot with V_{surge}, an autopilot needs to adapt its parameters as a function of this velocity. In our initial autopilot experiments in 2008, an autopilot system was developed and tuned at a single velocity near 50 cm/s [22]. However, our use of the robot over a more varied speed envelope showed that the autopilot parameters must be adjusted properly at each velocity. Subsequent uses in 2012 showed that large rolling oscillations were present, when the autopilot was used in forward velocities $V_{surge} \approx 15$ *cm/s*, as shown in Fig. 2.

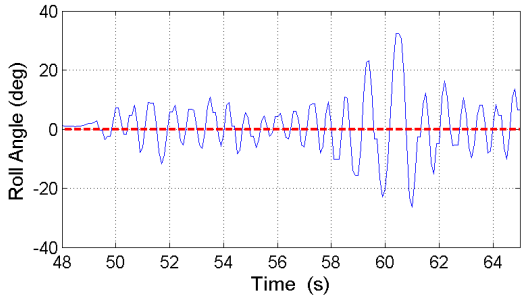

Fig. 2. Roll instability displayed by the autopilot system at low velocities V_{surge} during January 2012 trials. The dashed red line indicate the target roll angle of 0^o.

A. Autopilot Description

1) Attitude Control (Pitch, Roll and Yaw): By design, the coupling between the pitching, rolling and yawing axis on the robot is limited [20]. This decoupling comes partly from the left-right, front-back placement symmetry of the motor shafts rotating the flippers. It is also due to the swimming gaits themselves, which have been designed to provide independent moments in roll, pitch and yaw. For instance, a pitching moment is generated by displacing the front leg offsets θ_{leg} in opposite direction from the back legs (see Fig. 4 a). This produces a net pitching motion with little yawing or rolling moments. Similarly, rolling motion involves orienting the 3 left flippers in opposite direction to the 3 right flippers. Finally, yawing commands are executed by increasing the net thrust via amplitude B_{leg} on one side, while decreasing it by a similar amount on the opposite side.

This decoupling allows the decomposition of the attitude control problem into three independent problems, namely pitch (P), roll (R) and yaw (Y) control. Three servo loops are therefore used to maintain attitude. They rely on the classic Proportional-Derivative controller, illustrated in Fig. 3. The generic PD controller equation is:

$$C^A = K_P^A \varepsilon_A + K_D^A \frac{d}{dt} \varepsilon_A \tag{2}$$

with A representing one of the following three symbols: P for Pitch, R for Roll or Y for Yaw. ε_A is the measured error between the desired angle x_A and the measured angle y_A, for a given direction A:

$$\varepsilon_A = x_A - y_A. \tag{3}$$

All angles are expressed in radians. The command C^A sent to the robot is unitless, and generates approximately a linear rotation rate on that particular direction A [19].

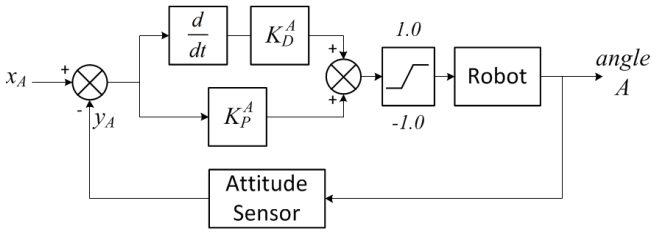

Fig. 3. Generic Proportional-Derivative (PD) servo loop used to control the attitude of the robot. The letter A represents the axis (P for Pitch, R for Roll and Y for Yaw). x_A is the desired angle. All commands (unitless) to the robot are limited to ± 1. The attitude sensor is the 3DM-GX1 Inertial Measurement Unit from Microstrain^{TM}.

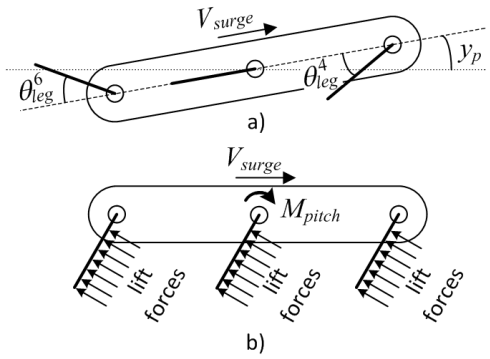

Fig. 4. Diagrams of the AQUA robot showing a) a pitching maneuver done by the flippers b) a heaving maneuver, both when a significant forward velocity V_{surge} is present. In the b) case, the lift forces on the flippers generate an undesired and significant pitching down moment M_{pitch}, making the robot tumble out of control.

2) Depth Control: Depth control of the robot can be accomplished via two different strategies [1]:

- by changing the target pitch angle x_P;
- by using a heaving command C^H with the $hovering$ gait.

For $V_{surge} > 15\ cm/s$, a change in pitch angles is able to produce a net upward or downward motion:

$$\dot{h} = V_{surge} sin(x_P), \tag{4}$$

where $\dot{h}$ is the time derivative of the depth h. For small angles x_P, this relationship is approximately linear via the familiar simplification $x_P \approx \sin x_P$. For this velocity regime, a simple proportional controller is used to maintain the depth h:

$$x_P = K_P^{Ph}(h_{target} - h). \tag{5}$$

This forms a cascaded loop, in conjunction with the $Pitch$ loop described in Eq. 2 as the inner loop.

At low-to-null V_{surge}, this approach cannot be used because the climbing rate $\dot{h}$ is not sufficient. Instead, vertical velocity $\dot{h}$ has to be generated by the thrusting ability of the flippers themselves, via a heaving command C^H. In our system, we used a modified Proportional-Integral (PI) controller (depicted in Fig. 5) to implement a separate control loop in order to maintain depth via this heaving command C^H. The presence of an integrator in the loop ensures that the system does not have a steady-state error, even when there is a bias in the system. In our case, this bias is the imperfect weight trim, i.e. the robot can display a slightly negative or positive buoyancy. Previous experiments showed that this trim bias can induce a depth error ε_h of 20-40 *cm* when no integrator is present with heaving.

The standard PI loop is expressed as:

$$C = K_P \varepsilon_h + K_I \int \varepsilon_h dt \tag{6}$$

with ε_h being the depth error. However, the use of an integrator such as in Eq. 6 can be problematic in a real system, for many reasons. First, in order to take into account mostly the long-term bias, the value of K_I must be very small. With short test times (2 minutes), the integrator does not have time to properly converge to eliminate the bias with a small value of K_I^H. On the other hand, a larger value means that the system incorporates too much large disturbances. To mitigate this problem, we clamped the error going into the integrator:

$$\varepsilon_h^{clamp} = clamp(\varepsilon_h, -L_\varepsilon, L_\varepsilon). \tag{7}$$

This has the desirable effect of limiting the rate of integration when very large errors are present (such as when we disturb the robot manually by a meter or so), while still fully taking into account the small errors ε_h due to trim biais, in steady-state. We selected a value of $L_\varepsilon = 0.30\ m$, as this was close to the expected steady-state error of our robot.

Second, we wanted to avoid dangerous integrator windups [23] and limit the maximum authority of the integrator's path (compared to the proportional path). For this, we used a *limited integrator*, which stops integrating its input when a positive or negative limit $\pm L_{Int}$ is reached. The equation of the depth controller, when using heaving, is thus:

$$C^H = K_P^{Hh}\varepsilon_h + K_I^{Hh} \int_{limited} \varepsilon_h^{clamp} dt. \tag{8}$$

This controller is depicted in Fig. 5.

B. Gain Scheduling

As mentioned above, the autopilot gains need to be changed as a function of the robot's velocity V_{surge}. A well known technique, called *gain scheduling*, solves this problem by finding the proper control parameters of a system at certain operating points of a *scheduling variable*. Between those points, the controller's parameter are interpolated. In our system, the velocity command V was used as *scheduling variable*, and we

TABLE I

AUTOPILOT GAINS FOR ATTITUDE CONTROLLERS P, R AND Y AND DEPTH CONTROLLER, AS A FUNCTION OF NORMALIZED FORWARD VELOCITY COMMAND V. THE CORRESPONDING VALUE V_{surge} WAS ESTIMATED FROM POOL TRIALS.

V	V_{surge} (cm/s)	K_P^P	K_D^P	K_P^R	K_D^R	K_P^Y	K_D^Y	K_P^{Ph}	K_I^{Hh}	K_P^{Hh}
1	62	3.5	0.0	1.25	0	3.5	1.5	-0.3	0	0
0.5	32	3.5	0.0	0.75	0	2.1	1.5	-0.3	0	0
0.3	21	3.5	0.0	0.75	0	3.5	1.5	-0.6	0	0
0.2	13	1.75	0.0	0. 50	0	1.75	1.0	0	-0.03	-0.5
0.0	0	0.875	0.40	0.25	0.10	1.75	1.0	0	-0.03	-0.5

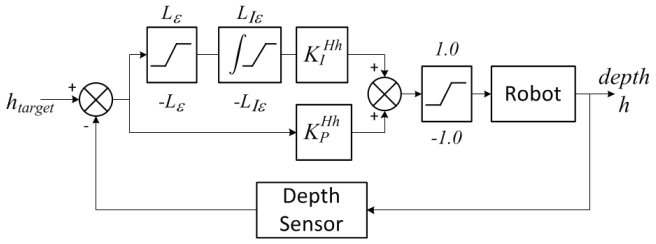

Fig. 5. PI servo loop for the control of depth at low or null V_{surge} velocities, using a heave command C^H. The integrator box is a *limited integrator* which stops integrating when reaching specific positive or negative thresholds.

employed linear interpolation. The operating points at which new parameters needed to be re-tuned were chosen based on the observed dynamic behavior of the robot during pool trials, one month before the sea trials. In particular, we noticed that the system dynamics changed more at low velocities, so more points were included in the table at low values of V. More trial-and-error tuning happened during sea trials in open ocean, and the final values are displayed in Table I.

From the values in Table I, we can see some general trends. At high velocities V, there was no need for differential gain in pitch (K_D^P) or roll (K_D^R). We strongly suspect that this is due to a intrinsic derivative component in forces, when the flippers are executing commands [19]. Indeed, pitching and rolling command for the gait active at these velocities V are executed by quickly changing the flipper's offset angle θ_{leg}. This induces short but rapid rotations of the flippers, resulting in a kicking forces proportional to the derivative of the command C^P or C^R. Since yaw commands C^Y are executed with a change in amplitude B_{leg} for any of our gaits, it lacks these kind of kicking forces [19]. Therefore, in the yaw (Y) direction, the autopilot required a derivative gain K_D^Y to improve its response at all velocities V.

To remove the unwanted oscillations at low velocities experienced during see trials a year earlier (and shown in Fig. 2), the roll proportional gain K_P^R had to be gradually reduced with velocity. At low velocities ($V < 0.2$), the intrinsic derivative forces are limited in pitch and roll, as the robot transition to the *hovering* gait. Consequently, derivative terms K_D^R and K_D^P are gradually introduced by the gain scheduler to compensate for the lack of these kicking forces.

Gain scheduling was also used to transition smoothly between the two control strategies for maintaining a constant depth $h_{desired}$. For $V > 0.2$, pitching is used to maintain depth, via the K_P^{Ph} variable (Eq. 5.) Just above $V > 0.2$, this gain is increased, to compensate for the reduced forward velocity term V_{surge} in Eq. 4, which directly affects the overall gain of the loop. Below $V < 0.2$, depth control via pitch is disabled, and depth control by heaving is enabled via the K_I^{Hh} integral gain and K_P^{Hh} proportional gain of Eq. 8.

IV. EXPERIMENTS : OPEN-SEA TRIALS IN BARBADOS

Experiments were conducted in an open-sea, uncontrolled environment in Barbados in January 2013. Waves of amplitude 10-30 *cm* were present, with some underwater currents around 5-10 cm/s. Tests were performed in shallow areas, with bottom depth between 4 to 13 *m*. Overall, the autopilot was used for over 200 minutes during the two-weeks sea trial, with success. A video showing the robot exploring the environment at low velocities with the autopilot enabled can be seen in [24]. Another video shows the robot performing station-keeping ($V = 0$) with the autopilot enabled in [25]. Note that this last video was taken during the tuning process, and therefore does not represent our best result.

In this Section, we present the results of experiments for two velocity conditions: low velocity and no velocity. Manual disturbances were applied on the robot for the no velocity experiments, to better stress the autopilot performances in this challenging regime. No depth change commands were issued in any of the trials. For the experiments with ($V > 0$), the heading of the robot was controlled by a curiosity-driven algorithm or a random walk algorithm. Therefore, we do not present any results for the yaw axis, as these commands were changing too rapidly for the robot to stabilize around that axis.

A. Trials at fixed, low velocity $V = 0.3$

Figs. 6 and 7 show two continuous trials of 6 minutes of the autopilot maintaining roll, pitch and depth at a constant commanded velocity of $V = 0.3$ (approximately 20 cm/s, estimated from pool trials). As we can see, there is no noticeable large oscillations present in the data due to autopilot instabilities. The standard deviation σ and mean μ values for the errors in these two trials are presented in Table II, for pitch, roll and depth. Overall, these results indicate that our autopilot system performs well in real open sea conditions, with roll errors generally less than 2.5^o. Pitch values tended to vary more, as it was used to maintain the depth of the system at that velocity. For example in Fig. 6 b), we can see

the pitch angle of the robot going to -9.5^o at $t = 205$ s. This was due to a needed depth correction, as depth had decreased at t=195 s in Fig. 6 c), and the robot had to pitch up in order to climb.

TABLE II
DISTRIBUTION OF ERRORS IN ROLL (R), PITCH (P) AND DEPTH h.

	Trial 1	Trial 2
Roll σ_R	2.16^o	1.69^o
Roll μ_R	0.97^o	0.04^o
Pitch σ_P	2.92^o	3.43^o
Pitch μ_P	-0.40^o	1.12^o
Depth σ_h	0.080 m	0.094 m
Depth μ_h	0.0085 m	0.037 m

B. *Trials at* $V = 0$

Because it is significantly more challenging to maintain attitude while hovering, great care was taken to tune the autopilot system in this regime. In particular, we will show how the various parameters helped improve the dynamic response of our system, when subjected to large disturbances. One such disturbance can been seen in Fig. 8, where the robot was tilted by 45^o by a diver. Moreover, we believe that achieving good station-keeping at $V = 0$ is the most significant contribution of our work.

Fig. 8. Aqua robot during one of the open-sea autopilot trials at $V = 0$, using the *hovering* gait. The robot was manually disturbed by the diver with a pitch angle of 45^o and then released.

1) Pitch Improvement: Fig. 9 a) shows the pitch robot response to an external disturbance before a derivative gain K_D^P, while Fig. 9 b) shows the response after tuning and adding a derivative gain K_D^P. The response of the robot is greatly improved, with the settling time going from 9 s in a) down to about $t = 6.0$ s in b). The number of oscillations has been greatly reduced as well, with a single overshoot present in the tuned system b). This demonstrates the agility of the *hovering* gait with this autopilot, as the pitch disturbance was significant (nearly 50^o.)

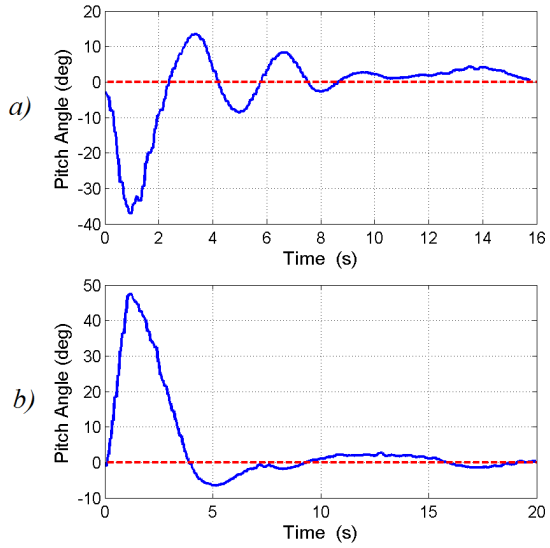

Fig. 9. a) Pitch response with *hovering* gait at $V = 0$ and the autopilot enabled, for a manual pitch disturbance at time $t = 1.0$ s, without derivative gain in place ($K_D^P = 0$) and an initial proportional gain $K_P^P = 1.75$. The presence of numerous secondary oscillations indicates that the system was under-damped and not properly tuned. b) Manual pitch disturbance at time $t = 1.2$ s, with derivative gain in place. On this particular test, the autopilot finished the correction at $t = 7.2$ s.

2) Roll Improvement: Fig. 10 a) shows the roll response instability of the autopilot, even without external disturbances. Fig. 10 b) shows the response after tuning and adding a derivative gain K_D^R. Again, the response of the robot is improved. In particular, the derivative term was able to remove the instabilities that were present before tuning the system. This tuned roll response has one more overshoot than pitch, however. This is due to the geometry of the robot, making it naturally less damped in the rolling direction than in the pitching direction.

3) Depth: Fig. 12 a) shows the depth h response of the autopilot when a large external disturbance is applied. Depth was maintained with heaving commands C^H, but without the use of an integrator ($K_I^{Hh} = 0$.) A steady-state error of approximately 30 *cm* can be seen from $t > 15$ s. On the other hand, Fig. 12 b) shows the system response when an integrator is present in the loop. In particular, the static error becomes very small as the system converges. The depth error distribution for a longer test segment of 80 seconds (without manual disturbance), is shown in Fig. 11. The average of this distribution is 1.2 *cm*, meaning that the robot was able to maintain depth, *on average*, with very good precision in an uncontrolled environment. This clearly demonstrates that maintaining depth with heaving commands is possible, and highlight the importance of having an integrator in this situation.

V. CONCLUSION AND FUTURE WORK

We have presented an autopilot designed for the flipper-based Aqua robot, which enables this robot to swim at a wide range of speed values, including zero speed. We have shown

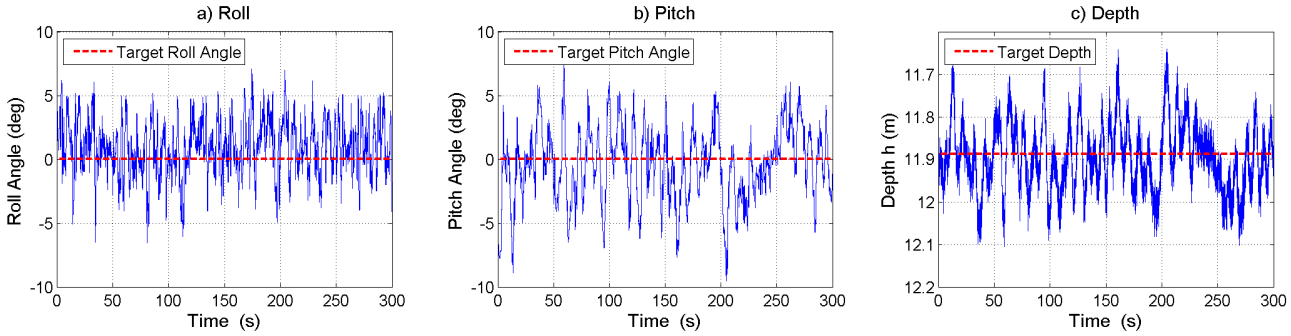

Fig. 6. Autopilot behavior during the sea trials. The speed was constant at $V = 0.3$.

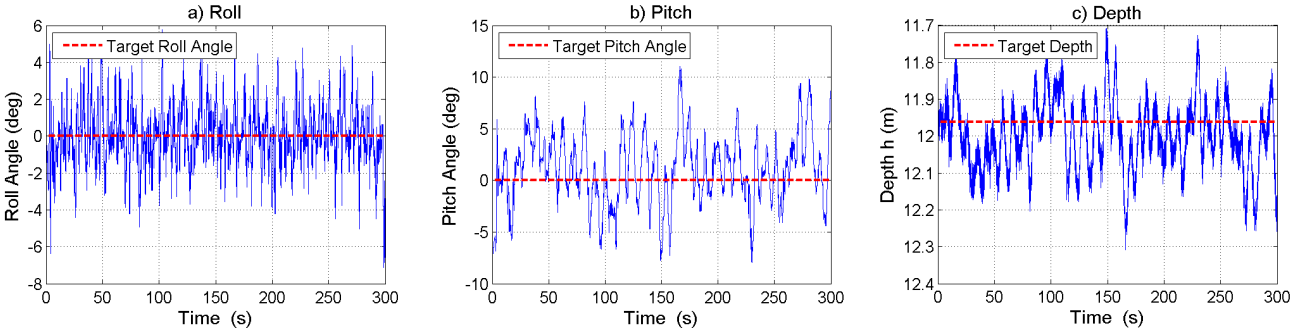

Fig. 7. Autopilot behavior during the sea trials. The speed was constant at $V = 0.3$.

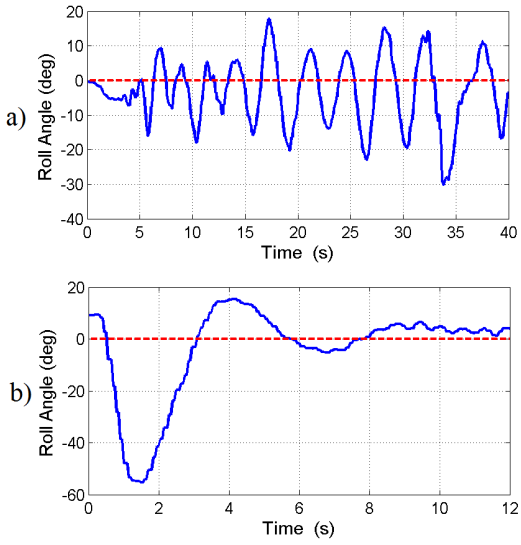

Fig. 10. a) Roll instability without any external disturbance, with *hovering* gait at $V = 0$ and the autopilot enabled. No differential gain was present ($K_D^R = 0$) in the system in this trial, and the proportional gain was $K_P^R = 0.5$. b) Manual roll disturbance at time $t = 1.8$ s, with derivative gain K_D^R in place and reduced proportional term $K_P^R = 0.25$. The small oscillations perceptible at $t > 8$ s are from the parasitic oscillation generated by the flippers' periodic thrust.

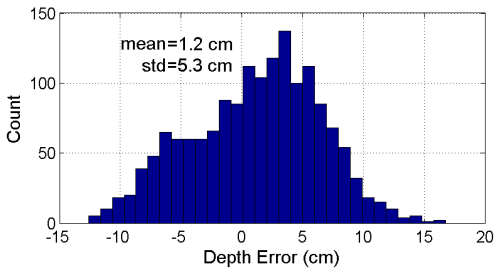

Fig. 11. Depth error distribution when the autopilot uses heaving commands $\tilde{C^H}$ to maintain depth at $V = 0$. The mean is 1.2 cm and the standard deviation is 5.3 cm.

that using the proposed autopilot, Aqua can achieve a very stable motion, with minimal oscillations. In particular, we were able to reduce considerably the instabilities and overshoots for the pitch and roll axis when the robot was subjected to large disturbances. We also demonstrated that the robot can maintain its depth with heaving motion with a precision in the cm range. Moreover, all these experiments were conducted in uncontrolled environments, in the presence of waves and currents. We believe that this was the first demonstration of a rigid-body, oscillating-foil swimming robot being able to maintain depth and attitude automatically.

As future work, we would like to investigate the application of machine learning techniques, such as *Reinforcement Learning*, to this control problem. We believe that there are further gains to be made, in particular since the *hovering* gait employed at zero speed was designed from simple heuristics. Finally, we believe that these autopilot improvements will facilitate other research efforts employing this robot, by now offering a stable and agile swimming platform.

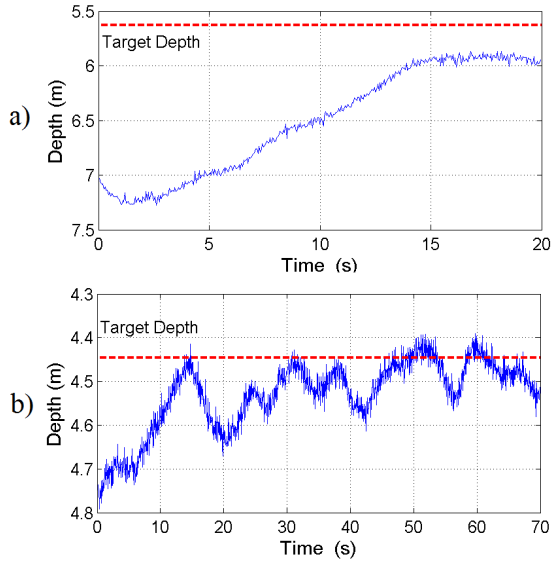

Fig. 12. a) Depth control via heaving commands C^H at $V = 0$ for a control loop *without* an integrator ($K_I^{Hh} = 0$.) A manual disturbance of 1.60 m was applied and released at $t = 2$ s, with the autopilot correcting the depth error afterwards. A steady state error is visible for $t > 15$ s. b) Test result for the autopilot system with an integral gain in the loop. The robot was left untouched at the beginning of the experiment. The depth error is seen as slowly decreasing over time, as the integrator compensates for the trim bias of the robot. At $t > 45$ s, the steady-state error is within ± 10 cm. Similar results were obtained when the robot was subjected to disturbances. The oscillations present with a period of ≈ 10 s correspond to the variation in water pressure induced by the surface waves.

REFERENCES

[1] Y. Girdhar, P. Giguère, and G. Dudek, "Autonomous Adaptive Underwater Exploration using Online Topic Modelling," in *International Symposium on Experimental Robotics (ISER)*, 2012. [Online]. Available: http://cim.mcgill.ca/ yogesh/publications/iser2012.pdf

[2] M. Sibenac, W. J. Kirkwood, R. Mcewen, F. Shane, and R. Henthom, "Modutar AUV for Routine Deep Water Science Operations," in *OCEANS '02 MTS/IEEE*, 2001, pp. 167–172.

[3] R. Marthiniussen, K. Vestgird, and R. A. Klepaker, "HUGIN-AUV concept and operational experiences to date," in *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, 2004, pp. 846–850. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1405571

[4] 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 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Edmonton, Alberta, Canada, August 2005.

[5] G. Dudek, P. Giguere, and J. Sattar, "Sensor-based behavior control for an autonomous underwater vehicle," in *Proceedings of the International Symposium on Experimental Robotics, ISER*, Rio de Janeiro, Brasil, July 2006.

[6] N. Plamondon, "Modeling and control of a biomimetic underwater vehicle," Ph.D. dissertation, McGill University, 2010.

[7] M. S. J. M. F. H. C. C. J. M. F. H. Justin Eskesen, Dylan Owens, "Design and performance of odyssey iv: A deep ocean hover-capable auv," Tech. Rep.

[8] E. M. J. D. S. K. Vaganay J, Gurfinkel L, "Hovering autonomous underwater vehicle - system design improvements and performance evaluation results," in *UUST*, 2009.

[9] M. Woolsey, V. Asper, A. Diercks, and K. McLetchie, "Enhancing niust's seabed class auv, mola mola," in *2010 IEEE/OES Autonomous Underwater Vehicles (AUV)*, 2010, pp. 1–5.

[10] A. Menozzi, H. Leinhos, D. Beal, and P. Bandyopadhyay, "Open-loop control of a multifin biorobotic rigid underwater vehicle," *Oceanic Engineering, IEEE Journal of*, vol. 33, no. 2, pp. 59 –68, april 2008.

[11] A. Crespi, "Design and control of amphibious robots with multiple degrees of freedom," Ph.D. dissertation, Ecole Polytechnique Fédérale De Lausanne, 2007.

[12] A. Tonello, "Control and guidance systems for the navigation of a biomimetic autonomous underwater vehicle," 2011.

[13] H. F. T. M. Licht S, Wibawa M, "Towards amphibious robots: Asymmetric flapping foil motion underwater produces large thrust efficiently," in *UUST*, 2009.

[14] J. H. L. Jr, J. Schumacher, N. Livingston, and M. Kemp, "Four flippers or two? tetrapodal swimming with an aquatic robot," *Bioinspiration & Biomimetics*, vol. 1, no. 1, p. 20, 2006.

[15] J. Das, F. Py, T. Maughan, T. O'Reilly, M. Messie, J. Ryan, G. S. Sukhatme, and K. Rajan, "Coordinated sampling of dynamic oceanographic features with underwater vehicles and drifters," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 626–646, Apr. 2012. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364912440736

[16] G. a. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," *The International Journal of Robotics Research*, vol. 32, no. 1, pp. 3–18, Nov. 2012. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364912467485

[17] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent Ocean Monitoring with Underwater Gliders: Adapting Sampling Resolution," *Journal of Field Robotics*, vol. 28, no. 5, pp. 714–741, 2011.

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

[19] P. Giguere, G. Dudek, and C. Prahacs, "Characterization and modeling of rotational responses for an oscillating foil underwater robot," in *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006.

[20] 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.

[21] MobileRoboticsMcGill, "Swimming gait (hovering) for station-keeping," http://www.youtube.com/watch?v=xh1DIUQtSvg.

[22] N. Plamondon and M. Nahon, "Trajectory tracking controller for an underwater hexapod vehicle," in *Oceans 08 MTS/IEEE*, Quebec City, Canada, September 2008.

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

[24] "Testing of an autopilot system at low velocities, in barbados 2013." http://youtu.be/V2V8HDSCXYs.

[25] "Testing of an autopilot system in hovering mode, in barbados 2013." http://www.youtube.com/watch?v=bYKZtfbx7X8.