IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Pauses Provide Effective Control for an Underactuated Oscillating Swimming Robot

Gedaliah Knizhnik, Philip deZonia, and Mark Yim

***Abstract*— We describe motion primitives and closed-loop control for a unique low-cost single-motor oscillating aquatic system: the Modboat. The Modboat is driven by the conservation of angular momentum, which is used to actuate two passive flippers in a sequential paddling motion for propulsion and steering. We propose a discrete description of the motion of the system, which oscillates around desired trajectories, and propose two motion primitives — one frequency based and one pause-based — with associated closed-loop controllers. Testing is performed to evaluate each motion primitive, the merits of each are presented, and the pause-based primitive is shown to be significantly superior. Finally, waypoint following is implemented using both primitives and shown to be significantly more successful using the pause-based motion primitive.**

I. Introduction

Affordable and capable propulsion has been a limiting factor in the development of modular autonomous surface and underwater vehicles (ASVs and AUVs). Significant work has gone into developing kayak, catamaran, or trimaran robots for ASVs [1], which are capable of powerful locomotion, but self-assembling modular systems have not been explored. The TEMP project [2] [3] developed modular ASVs capable of docking together and forming larger floating structures, but their modules were holonomic and required six actuators (two docking motors and four drive motors) to achieve this behavior. Similarly, work on modular AUVs has generally considered holonomic motion as a requirement, such as AMOUR V [4] or μAUV2 [5]. Actuators dominate the cost of a system, so a low-cost propulsion mechanism would enable development of larger systems and swarms of modular reconfigurable floating systems.

Reducing the number of actuators to one reduces cost but introduces control complexity. For self-assembling modular boats, docking together is what makes them unique. In order to enable docking, the precision placement of boats relative to each other is very important; thus precise trajectory following and position control must be solved to prove viability. Control is trivial for many aquatic systems but is complicated by the reduced actuation of low-cost systems.

A unique mechanism for providing motion with reduced actuation is internal rotor propulsion. Initially used for terrestrial locomotion, it was applied to the Chaplygin sleigh problem by Osborne and Zenkov, who considered using the movement of a fully actuated mass to steer the nonholonomic sleigh [6]. Kelly et al. introduced the Chaplygin Beanie, in which the mass is reduced to rotation only, and showed that it could be driven using proportional heading control [7], while Tallapragada and Fedonyuk showed that periodic impulses could also be used [8]. The same principle has also been used for climbing between two walls by Degani [9] [10].

The authors are with the GRASP Laboratory, University of Pensylvannia, Philadelphia, PA 19104. knizhnik@seas.upenn.edu

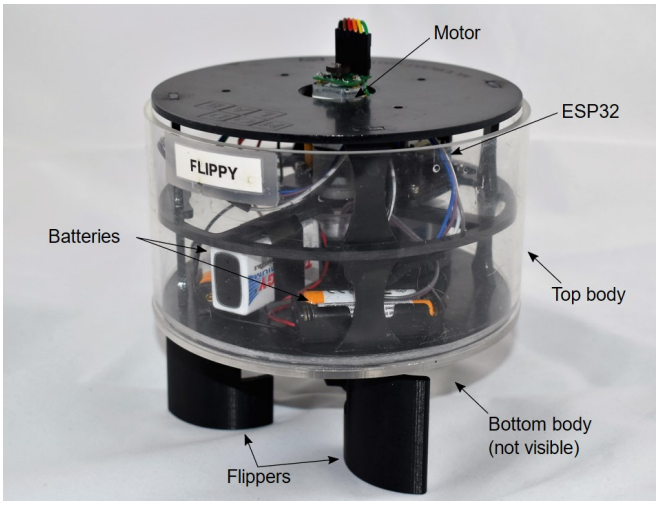


Fig. 1. The Modboat prototype, nicknamed Flippy. The bottom body is obscured beneath the top body [14].

Internal rotor propulsion has shown promise in aquatic locomotion as well. Kelly et al. applied the principles of the Chaplygin Beanie to an aquatic robot [7], while Tallapragada showed that the core propulsion mechanism is vortex shedding from the tail, driven by the internal rotor [11]. Refael and Degani introduced a unique design following this approach in which a rotating driving mass actuates a pair of passive flippers that generates propulsion and steering [12] [13]. We adapted this design to modular self-assembling behavior in previous work [14]. Our design — *the Modboat* — consists of two passive flippers mounted to a cylindrical bottom body, with hard stops defining fully open positions. A motor (under closed-loop position control) spins an upper body that functions as the driving mass, which forces the bottom body to spin in the opposite direction; a combination of drag and centrifugal action forces the leading flipper open against its hard stop, which then provides thrust. Reversing the motor pulls the open flap closed while forcing the opposite one open to provide thrust. An oscillation of the driving mass thus leads to a paddling motion that pushes the robot forward. This propulsion mechanism is inexpensive and capable of modular self-assembly.

In both previous systems [13] [14], the propulsion is driven by a sinusoidal oscillation in which steering is achieved by varying duty cycle and frequencies. The ability to turn is

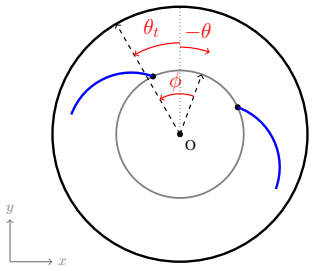


Fig. 2. A simplified top view of the Modboat. The bottom body is shown in gray with the flippers in blue, while the top body is black. The angles θ and θ_t are measured from the world frame y axis, while ϕ is the relative angle between the two bodies.

demonstrated, but neither work shows this principle following a trajectory. It turns out to be difficult to precisely control the orientation using frequency control. In this work we develop a **control system** and a **novel method for steering** the Modboat based on **pauses** and show it to be superior for trajectory following.

This paper is organized as follows. In Section II we introduce the dynamic model underlying the Modboat and in Section III we define the heading as the control variable. In Sections IV and V we propose two motion primitives for the system and associated controllers for each. Then in Section VI we evaluate the motion primitives and controllers, the results of which are discussed in Section VII.

II. Modeling

We define a **stroke** as a period during which the motor rotates in a single direction, i.e. either of the two cases in (6). A **cycle** is then defined as a full period, i.e. two strokes.

A. Continuous Model

A planar model is developed by Refael and Degani [13]. The assumptions made in developing the model are reproduced in our previous work [14], where a detailed description of the operating principles of the system is presented. The most relevant assumption for this work is that the translational velocity of the robot is negligible compared to its angular velocity. This allows the forces/torques acting on the flippers to be a function of the orientation of the bottom body θ and its derivatives only, which simplifies the model.

The orientation equation from the final equations of motion [14] is shown in (1), and a simplified diagram is presented in Fig. 2. I_b and I_t are the inertias of the bottom and top bodies, respectively, while $K[\mathrm{kg\,m^2}]$ and $C_r[\mathrm{kg\,m^2\,s^{-1}}]$ are constants. ϕ represents the relative angle between the bottom and top bodies. It is the only actuated variable and is taken as the input to the system (generally as a sinusoid) [13] [14].

$$(I_b + I_t)\ddot{\theta} = -K\,\mathrm{sgn}\,(\dot{\theta})\dot{\theta}^2 - C_r\dot{\theta} - I_t\ddot{\phi} \tag{1}$$

The three terms on the right hand side refer to — in order — thrust from the flippers, rotational drag opposing the rotation of the bodies in the water, and the angular momentum imparted by the actuated rotation of the top body.

B. Discrete Model

To remove the non-linearity in (1) and simplify the control problem, we propose a discrete model. Considering the orientation only, we observe that each stroke imparts angular momentum L as in (2), where we assume that the thrust provided by the flippers is the primary force, the duration of the integral is such that $\mathrm{sgn}\,(\dot{\theta})$ is constant throughout, and the subscript $i \in \{1, 2\}$ refers to the stroke.

$$L_i = -K\,\mathrm{sgn}\,(\dot{\theta}) \int \dot{\theta}^2 dt \tag{2}$$

Let the **overall orientation** Θ refer to some single orientation representative of a cycle and **overall angular velocity** $\Omega = \Delta\Theta/\Delta T$ to the change in overall orientation per cycle. The overall change in angular momentum ($\Delta L = L_2 - L_1$) is proportional to the **overall angular acceleration** $\Delta\Omega/\Delta T$; or $\Delta L = I_t \Delta\Omega/\Delta T$.

We assume that some map exists from the inputs that define $\phi(t)$ to ΔL, and from there to angular acceleration α imparted by a cycle, (3). Then for the new input α we can numerically integrate as in (4) for the overall orientation-only state $\begin{bmatrix}\Theta & \Omega\end{bmatrix}^T$ at cycle k, where the discretization time ΔT is a full cycle.

$$\alpha = f\left(\phi(t)\right) \tag{3}$$

$$\begin{bmatrix}\Theta \\ \Omega\end{bmatrix}_{k+1} = \begin{bmatrix}1 & \Delta T \\ 0 & 1\end{bmatrix}\begin{bmatrix}\Theta \\ \Omega\end{bmatrix}_k + \begin{bmatrix}0 \\ \Delta T\end{bmatrix}\alpha_k \tag{4}$$

This discrete linear model is far simpler than (1). By selecting inputs at the beginning of a cycle, letting the system run, and then evaluating at the end of the cycle, the problem of controlling the Modboat becomes more tractable.

III. Heading

Motivated by the orientation-only model in (4), we propose to focus control on the gross *direction of motion* of the Modboat, rather than the instantaneous orientation (which oscillates as it moves), and define the **heading** $h \in (-\pi, \pi]$ as the overall orientation representative of this direction of motion. The discrete control problem is then to define the input $\phi(t)$ for each cycle that will drive the heading to a desired value. For position control, this desired heading would be the angle from the (x, y) coordinates of the center of Modboat to the target.

The Modboat can only produce positive thrust; for a symmetric sinusoidal input $\phi(t)$ the net thrust will be generally towards the tail at the center-line of the rotation $\phi(t) = \phi_0$. We can therefore approximate the heading as the average orientation $\bar{\theta}_t$ over a cycle, where $\theta_t = \theta + \phi$ is the orientation of the top body (see Fig. 2). Thus if $\phi(t)$ is at a peak at t_i and then again at t_{i+1}, then we calculate the heading h as in (5).

$$h = \frac{1}{t_{i+1} - t_i}\int_{t_i}^{t_{i+1}} \theta_t(t)dt \tag{5}$$

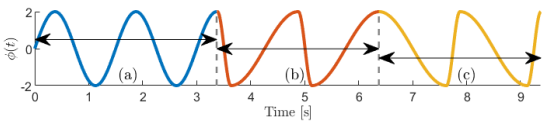


Fig. 3. An example frequency modulation waveform $\phi(t)$ as defined in (6) with three sections. The parameters $\begin{bmatrix} T_1 & T_2 \end{bmatrix}$ are given by (a): $\begin{bmatrix} 1.5 & 1.5 \end{bmatrix}$, (b): $\begin{bmatrix} 0.5 & 2.5 \end{bmatrix}$, and (c): $\begin{bmatrix} 2.5 & 0.5 \end{bmatrix}$, with $\begin{bmatrix} A & \phi_0 \end{bmatrix} = \begin{bmatrix} 2 & 0 \end{bmatrix}$ for all. Note that an extra time offset is added at the beginning because in practice the initial value of ϕ is 0.

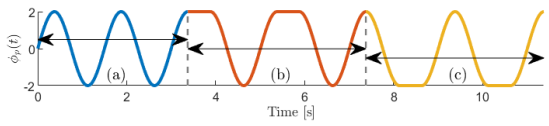


Fig. 4. An example pause modulation waveform $\phi_p(t)$ as defined in (9) with three sections. The base parameters are given by $\begin{bmatrix} T_1 & T_2 & A & \phi_0 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.5 & 2 & 0 \end{bmatrix}$, and the pause parameters $\begin{bmatrix} t_{ps} & t_{pl} \end{bmatrix}$ are given by (a): $\begin{bmatrix} 0 & 0 \end{bmatrix}$ (b): $\begin{bmatrix} 0 & 0.5 \end{bmatrix}$ and (c): $\begin{bmatrix} 0.75 & 0.5 \end{bmatrix}$. Note that an extra time offset is added at the beginning because in practice the initial value of ϕ is 0.

While the direction of motion should be accurately determined by the heading (i.e. the vehicle moves generally towards the direction it is facing), this method may be less accurate when external flows are present; translation not caused by the flippers may not be detected by (5).

IV. Frequency Modulation

A. Steering

Following Refael and Degani [13], we can propel the Modboat using inputs of the form given in (6) and illustrated in Fig. 3, which defines a piecewise continuous sinusoid with varying frequencies ω_1 and ω_2. T_1 and T_2 are the periods associated with *complete* rotations at frequencies ω_1 and ω_2, respectively, A is the amplitude, and ϕ_0 is the zero-orientation of the driving mass. The 4-tuple $\begin{bmatrix} T_1 & T_2 & A & \phi_0 \end{bmatrix}$ defines the input function.

$$\phi(t) = \begin{cases} \phi_0 + A\cos(\omega_1 t) & t \in \left[0, \frac{T_1}{2}\right) \\ \phi_0 - A\cos\left(\omega_2\left(t - \frac{T_1}{2}\right)\right) & t \in \left[\frac{T_1}{2}, \frac{T_1+T_2}{2}\right) \end{cases} \tag{6}$$

The robot can be "steered" by varying the periods of oscillation from $\phi(t)$, which forces an oscillation of $\theta(t)$. When $T_1 = T_2$, the oscillation is symmetric and results in oscillations about a straight line. When $T_1 < T_2$, however, the stroke that activates the left flipper — $t \in [0, T_1/2)$ — is faster than the stroke that activates the right. The angular momentum imparted by a single flipper from (2) shows that the faster stroke provides more thrust resulting in a clockwise rotation. We can therefore turn by lowering one of the periods, but in practice this method is often limited by the maximum velocity of the motor.

B. Control

The discrete system presented in (4) is controllable, so if the mapping f in (3) were known and invertible, it would be easy to design a controller using α and then invert it to obtain the control parameters $\begin{bmatrix} T_1 & T_2 & A & \phi_0 \end{bmatrix}$. In theory, f could be approximated numerically by solving the equation of motion (1) numerically on a reasonable submanifold of the input space $\mathbb{R}^4$ (where reasonable includes considerations such as $T_i > 0$, etc.). This depends on the accuracy of the model, which is known to lose accuracy as T_1 and T_2 diverge [13]. Moreover, it is likely that f shows dependence on exterior factors, such as flow velocity, and there is no guarantee that it is invertible, making its calculation difficult.

Instead we use PID control. For simplicity, we assume that A (which can be assumed to be related to translational velocity) and ϕ_0 are held constant, and only T_1 and T_2 will be varied. Because f^{-1} is not known, we assume — by an abuse of notation — that for an initial symmetric input $T_1 = T_2 = T$ the following inputs will be of the form $T_1 = T \pm \alpha_k$ and $T_2 = T \mp \alpha_k$, leaving the total period unchanged. Then a PID controller can be defined as (7) and (8) with the terms in (8) representing the proportional, derivative and integral terms respectively; because PID is a very general approach, it is possible that it will succeed despite all the simplifications made to derive it.

$$e_k = \Theta_{d,k} - \Theta_k \tag{7}$$

$$\alpha_k = K_p e_k + K_d \frac{d}{dt}(e_k) + K_i \int e_k dt \tag{8}$$

V. Pause Modulation

A. Steering

When driven using (6), each stroke imparts some angular momentum. When $T_1 = T_2$, the angular momenta from the two strokes cancel each other, resulting in no net change in overall angular velocity, a straight heading and a forward motion.

Consider introducing a pause into (6), beginning at time t_{ps} and lasting for t_{pl}; this waveform is given in (9), where $\phi(t)$ is defined by (6). For steering, we restrict t_{ps} to either 0 or $T_1/2$, i.e. either the peak or the trough of $\phi(t)$. An example waveform is shown in Fig. 4.

When the pause is reached $\dot{\phi} = 0$, but $\dot{\theta}$ lags $\dot{\phi}$ and is non-zero. Thus, for the duration of the pause the Modboat will drift, affected only by drag. When the cycle finishes, no net change in angular momentum has been imparted because $T_1 = T_2$. But the drift over $t \in [t_{ps}, t_{ps} + t_{pl}]$ causes a net change in heading.

$$\phi_p(t) = \begin{cases} \phi(t) & t \in \left[0, t_{ps}\right) \\ \phi(t_{ps}) & t \in \left[t_{ps}, t_{ps} + t_{pl}\right) \\ \phi(t - t_{pl}) & t \in \left[t_{ps} + t_{pl}, \frac{T_1+T_2}{2} + t_{pl}\right) \end{cases} \tag{9}$$

This motion primitive is capable of large magnitude turns since rotational drag is low; Fig. 5 shows that for a base cycle of $T_1 = T_2 = 1.5$s a pause $t_{pl} \lessapprox 4$s is sufficient to turn any amount in $(-\pi, \pi]$. Moreover, it is free in terms of actuator effort. However, this method does require more time; it may sometimes be impractical to wait too long to turn, especially in safety-critical scenarios.

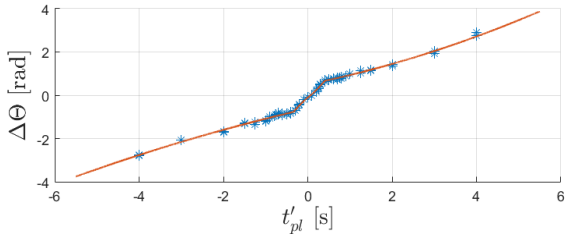


Fig. 5. Change in overall orientation per cycle for open-loop pause modulation over a base waveform defined by $\begin{bmatrix} T_1 & T_2 & A & \phi_0 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.5 & 2 & 0 \end{bmatrix}$. The signed pause-length t'_{pl} is defined as in (10). The data and best-fit function are shown.

B. Control

Although it is possible to modulate both frequency and pauses within the input waveform (9), for simplicity we choose to focus on modulating only the pause parameters.

To avoid discontinuity in angular velocity the pause start is restricted to either the peak or trough of $\phi(t)$: $t_{ps} \in \{0, T_1/2\}$. For simplicity, we define the **signed pause length** as a single input variable $t'_{pl} \in \mathbb{R}$, that maps to $\begin{bmatrix} t_{ps} & t_{pl} \end{bmatrix}$ as in (10). This allows us to consider a 1-dimensional control space consisting of t'_{pl} only.

$$\begin{bmatrix} t_{ps} & t_{pl} \end{bmatrix} = \begin{cases} \begin{bmatrix} 0 & \left|t'_{pl}\right| \end{bmatrix} & t'_{pl} < 0 \\ \begin{bmatrix} \frac{T_1}{2} & t'_{pl} \end{bmatrix} & t'_{pl} > 0 \end{cases} \tag{10}$$

The pause-waveform was tested on the Modboat by evaluating the trajectories produced for $t'_{pl} \in [-5, 5]$s for $\begin{bmatrix} T_1 & T_2 & A & \phi_0 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.5 & 2 & 0 \end{bmatrix}$. The orientation of the boat is plotted in each case, and the orientation at successive peaks is taken as the overall orientation. The resulting trajectories are found to be linear in overall orientation as a function of time with $\Omega(t_{pl})$ as the overall angular velocity from the slope of the line of best fit.

The resulting function is reasonably one-to-one, so we use an optimizer [15] to solve for a piecewise-linear continuous function that best fits the data. The results are multiplied by the cycle-time $(T_1 + T_2)/2 + t_{pl}$ to get the change in overall orientation per cycle, shown in Fig. 5.

This function — implicitly inverted — defines a one-to-one map from $\Delta\Theta \to t_{pl}$, which can be used directly as a controller. The model in (4) can thus be simplified further to a **single-integrator**. For any desired heading, we use (11) to directly compute $\Delta\Theta$ and obtain the needed input t'_{pl}.

$$\Theta_{k+1} = \Theta_k + \Delta\Theta_k(t'_{pl}) \tag{11}$$

In the single-integrator model from (11) the control input fully corrects the overall orientation by the end of the cycle. We must therefore modify the heading calculation in (5). We replace t_i in (5) with $t_i + t_{ps} + t_{pl}$, i.e. we use the end of the pause as the start of integration.

VI. Experiments

The Modboat was tested in a $4.5\text{m} \times 3.0\text{m} \times 1.2\text{m}$ tank of water (shown in Fig. 6) equipped with an OptiTrack motion capture system that provided planar position, orientation, and velocities at 120Hz. A MATLAB interface was used to record the data, calculate the heading and control parameters, and send commands to the boat for course correction. A set of base parameters $\begin{bmatrix} T_1 & T_2 & A & \phi_0 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.5 & 2 & 0 \end{bmatrix}$ was heuristically determined to provide reasonable open-loop performance[1] and was used for **all** experiments. For the frequency controller, this defines the T_1 and T_2 that are modified by α_k in (8), while for the pause controller this is the set of constant parameters over which t'_{pl} is varied.

TABLE I

Summary of experimental results for step input tests. RMS error is measured only after the rise time.

Metric	Pause		Frequency	
	Median	Interquartile Range	Median	Interquartile Range
Rise time [cycles]	0.82	0.75–0.88	4.2	3.8–5.7
Dist. to rise [m]	0.12	0.11–0.16	0.39	0.36 – 0.62
RMS error [rad]	0.17	0.083 – 0.20	0.25	0.21 – 0.32

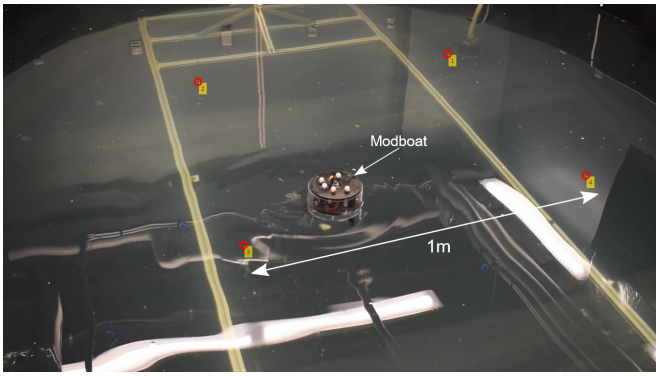


Fig. 6. A photo of the experimental setup, demonstrating the Modboat during a waypoint tracking experiment. The approximate locations of the waypoints are superimposed on the image.

Each controller was tested for its response to step inputs. The Modboats were placed in the tank and — for all tests — commanded a desired heading of π. The boats were released facing either approximately in the correct direction or $\pm\pi/2$ offset from it, which simulated step inputs of $\{-\pi/2, 0, \pi/2\}$; the boat would begin swimming in the direction it faced and then correct towards the fixed desired orientation. Due to floating drift the evaluations performed on the results aim to be agnostic to initial conditions.

For each step response test, we evaluated the rise-time, the distance traveled over the rise period, and the steady-state error over multiple iterations. The results for all tests and starting positions are shown in Table I. Both controllers perform well, but the pause controller shows superior performance in all three metrics. A sample response from the dataset for each controller is shown in Fig. 7; the pause controller responds far more quickly than the frequency

[1]This set of parameters provides an oscillation fast and large enough to fully open the flippers, while being slow enough to be symmetric to within the motor and micro-controller specifications.

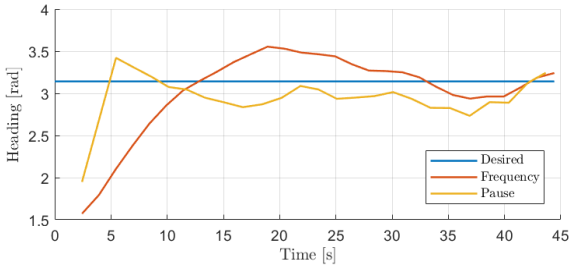
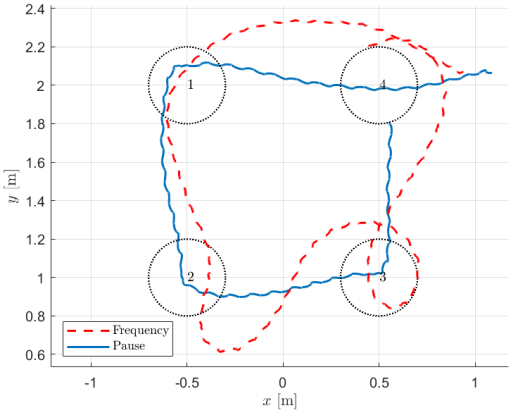


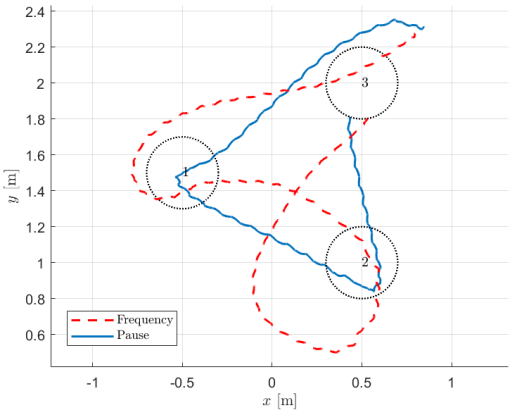
Fig. 7. Sample response of the two controllers to a step input of $\approx \pi/2$; control begins when the first heading measurement is obtained, at 1 cycle. The frequency controller lags significantly behind the pause controller.

controller. We note that the frequency controller in Fig. 7 begins at a disadvantage, but its response is sufficiently worse as to not be attributable to this factor, and this effect persists across starting conditions. This is clear in Table I, where the step magnitude is not considered.

Since both controller designs assume the inability to plan exact trajectories, we approximated trajectory planning by a series of discrete waypoints. The Modboat was directed to a waypoint until it was within 20cm, at which point the next waypoint in sequence would be swapped into the controller. The waypoints are far enough apart that approximately 10 – 20 cycles are required before reaching the next one. We tested the ability of the controllers to swim in different patterns; each test was run for 2min, although it was terminated early if the final waypoint was reached. The resulting trajectories are shown in Fig. 8, where the pause controller successfully completes each task while the frequency controller struggles. A heading vs. time plot for the square trajectory in Fig. 8a is shown in Fig. 9.



(a)



(b)

Fig. 8. Waypoint following executed by both controllers; the frequency controller is marked by red dashes, and the pause controller by a blue line. The waypoints are indicated by dotted circles and are labeled by the order in which they were commanded.

VII. Discussion

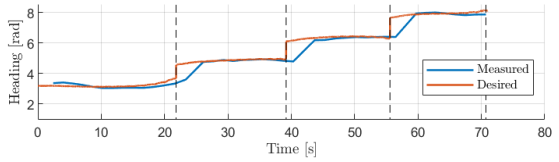
Table I shows the results of step input tests comparing the two controllers over all inputs. The large interquartile range shown by the frequency controller is reasonable since we consider step inputs of 0 and $\pi/2\text{rad}$, but it stands in stark contrast to the consistency shown by the pause controller across inputs. The pause controller shows superiority in rise-time and distance traveled during the rise-period; it achieves the correct orientation in one cycle and only $\approx 12\text{cm}$, regardless of the step input. The frequency controller, meanwhile, takes much longer to rise and travels $3-5$ times as far. An example of this response is shown in Fig. 7.

The advantage in steady-state error, shown as RMS error after the rise period, is less clear but still evident. The pause controller shows a lower median and a smaller range across the various step-inputs, demonstrating more consistent and better performance. The error range ($0.083-0.20\text{rad}$ or $4.6-11°$) is not ideal, but is still lower than the error for the frequency controller, despite the integral term in the PID loop. The addition of an integral term in the pause controller can reduce steady-state error further.

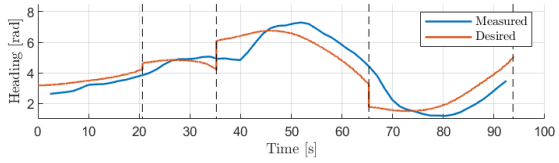
The advantages provided by the pause controller are demonstrated further in the waypoint following tests shown in Fig. 8. The pause controller is capable of swimming in square (Fig. 8a), triangular (Fig. 8b), and linear (not shown) patterns while following reasonably linear trajectories between the waypoints, even when they are only $\approx 1\text{m}$ apart. The frequency controller struggles with following linear trajectories between the waypoints and shows increasingly worse performance when sharper turns are required.

These differences are largely due to the significant advantages displayed by the pause control in rise-time and distance traveled during rise (see Table I). The pause controller is able to turn quickly and sharply (see Fig. 9a), allowing it to follow tight configurations of waypoints. The frequency controller requires a longer time to turn and much more distance to do so, forcing it to struggle with tight configurations. This can be seen in Fig. 9b; the heading lags the desired value and does not converge, and the associated translation causes the desired heading to vary significantly.

The close placement of the waypoints in Fig. 8 was largely due to the constraints of our testing environment. In many ocean applications the waypoints may be much further apart, and this would allow the frequency controller more space and time to correct, resulting in better waypoint following. But



(a)



(b)

Fig. 9. Heading vs. time plots for the waypoint experiments in Fig. 8a. The pause controller is shown in (a), the frequency controller in (b). Vertical dotted lines show waypoints reached.

these results still demonstrate the fundamental limitations of the frequency-driven motion primitive and the advantages of the pause-driven motion primitive. Moreover there are still aquatic applications that require relatively precise placement, and of the two primitives considered only the pause controller is capable of such tasks.

VIII. Conclusions

In this work we have described two motion primitives for the Modboat, whose design is presented in detail in our previous work in [14]. The first primitive — **frequency modulated steering** — was first proposed by Refael and Degani in [13] but was developed into closed-loop control in this work, while the second — **pause modulated steering** — is a novel contribution described in this work. For each motion primitive, we have also presented a closed-loop controller to drive the measured heading to a desired value.

Through experimental evaluation of step-response and waypoint tracking, we have shown that our novel pause modulation primitive significantly outperforms the frequency modulation primitive due to its ability to sharply correct step inputs of effectively arbitrary magnitude. This makes the pause modulation controller well suited for waypoint tracking on the order of 1m and sensor placement within $\approx 0.4\text{m}$ in calm waters. We note that these distances are derived heuristically from the simple orientation-only approach to trajectory planning used in this work, and that more intelligent trajectory planning that takes translation into account could be used to increase precision.

This work considered only one set of base parameters over which both frequency and pause modulation were applied. This choice was made to create a tractable problem and to evaluate the feasibility of pause control as a unique motion primitive, but neglects the possibilities of the full parameter space. We intend to explore the parameter space including varying the period of oscillation and its amplitude, as well as relaxing the requirement that $T_1 = T_2$ (i.e. combining frequency and pause modulation). We will also evaluate our control strategy in the presence of external flows, which will be crucial in allowing our system to perform in unpredictable ocean environments. The presence of waves may require a full 3D consideration rather than a planar model, while currents may require quantifying translation as well as orientation.

We will develop docking procedures for multiple Modboats as we explore group dynamics and group propulsion. Additional sensing approaches applicable to ocean environments will also be explored, as readily-available GPS data cannot provide the orientation measurements we need for control.

Acknowledgment

We thank Dr. M. Ani Hsieh for the use of her instrumented water basin in obtaining all of the testing data.

References

[1] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, "Unmanned surface vehicles: An overview of developments and challenges," *Annual Reviews in Control*, vol. 41, pp. 71–93, 2016.

[2] I. O'Hara, J. Paulos, J. Davey, N. Eckenstein, N. Doshi, T. Tosun, J. Greco, J. Seo, M. Turpin, V. Kumar, and M. Yim, "Self-assembly of a swarm of autonomous boats into floating structures," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014, pp. 1234–1240.

[3] J. Paulos, N. Eckenstein, T. Tosun, J. Seo, J. Davey, J. Greco, V. Kumar, and M. Yim, "Automated Self-Assembly of Large Maritime Structures by a Team of Robotic Boats," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 958–968, 2015.

[4] I. Vasilescu, C. Detweiler, M. Doniec, D. Gurdan, S. Sosnowski, J. Stumpf, and D. Rus, "AMOUR V: A Hovering Energy Efficient Underwater Robot Capable of Dynamic Payloads," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 547–570, 2010.

[5] H. Hanff, K. Schmid, P. Kloss, and S. Kroffke, "μaUV2 - Development of a minuscule autonomous underwater vehicle," in *13th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Lisbon, Portugal, 2016, pp. 185–196.

[6] J. M. Osborne and D. V. Zenkov, "Steering the chaplygin sleigh by a moving mass," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, 2005, pp. 1114–1118.

[7] S. D. Kelly, M. J. Fairchild, P. M. Hassing, and P. Tallapragada, "Proportional heading control for planar navigation: The Chaplygin beanie and fishlike robotic swimming," in *2012 American Control Conference (ACC)*, Montreal, QC, 2012, pp. 4885–4890.

[8] P. Tallapragada and V. Fedonyuk, "Steering a Chaplygin Sleigh Using Periodic Impulses," *Journal of Computational and Nonlinear Dynamics*, vol. 12, no. 5, 2017.

[9] A. Degani, H. Choset, and M. T. Mason, "DSAC - Dynamic, Single Actuated Climber: Local stability and bifurcations," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, 2010, pp. 2803–2809.

[10] A. Degani, "Dynamic single actuator robot climbing a chute: Period-doubling bifurcations: analysis and experiments," *Meccanica*, vol. 51, no. 5, pp. 1227–1243, 2016.

[11] P. Tallapragada, "A swimming robot with an internal rotor as a nonholonomic system," in *2015 American Control Conference (ACC)*, Chicago, IL, 2015, pp. 657–662.

[12] G. Refael and A. Degani, "Momentum-driven single-actuated swimming robot," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015, pp. 2285–2290.

[13] ——, "A Single-Actuated Swimming Robot: Design, Modelling, and Experiments," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 94, pp. 471–489, 2018.

[14] G. Knizhnik and M. Yim, "Design and Experiments with a Low-Cost Single-Motor Modular Aquatic Robot," in *2020 17th International Conference on Ubiquitous Robots (UR)*, Kyoto, Japan, 2020, pp. 233–240.

[15] C. Jekel, "Fitting a piecewise linear function to data," https://jekel.me/2017/Fit-a-piecewise-linear-function-to-data/, 2017.