

Amplitude Control for Parallel Lattices of Docked Modboats

Gedaliah Knizhnik and Mark Yim

Abstract—The Modboat is a low-cost, underactuated, modular robot capable of surface swimming. It is able to swim individually, dock to other Modboats, and undock from them using only a single motor and two passive flippers. Undocking without additional actuation is achieved by causing intentional self-collision between the tails of neighboring modules; this becomes a challenge when group swimming as one connected component is desirable. In this work, we develop a control strategy to allow parallel lattices of Modboats to swim as a single unit, which conventionally requires holonomic modules. We show that the control strategy is guaranteed to avoid unintentional undocking and minimizes internal forces within the lattice. Experimental verification shows that the controller performs well and is consistent for lattices of various sizes. Controllability is maintained while swimming, but pure yaw control causes lateral movement that cannot be counteracted by the presented framework.

I. INTRODUCTION

Aquatic systems that can dock, undock and reconfigure are of interest to researchers; they have significant potential to facilitate ocean research and infrastructure by providing mobile platforms to land helicopters or drones, building bridges for larger vehicles [1], or forming ocean-going manipulators. They can adapt to changing flow conditions or take precise measurements at small spatial scales. Nevertheless, work on such reconfigurable systems is not widespread. A small number of projects have used reconfiguration to multiplex swimming modes, such as AMOUR [2], and ANGELS [3] [4]. Both of these systems use robots capable of almost holonomic motion in 3D space, but only demonstrate minimal reconfiguration in 1D.

Extensive reconfiguration, however, has only been demonstrated — to the best of the authors’ knowledge — in two cases. The TEMP project built holonomic rectangular modules that could dock in a brickwork pattern and form a bridge [1] or assemble into arbitrary structures [5] [6]. The structures could actively resist deformation due to waves [7] and the undocked modules could move in formation, but movement of the entire structure was not considered. The Roboats project has designed similar rectangular modules capable of docking and reconfiguration, which are meant to form floating platforms in the canals of Amsterdam [8] [9]. Roboats can swim as a group, and can move using local information as long as the structure is symmetric [10] or through a learning approach for more arbitrary structures [11]. But both TEMP and Roboats require expensive modules capable of holonomic motion and rely extensively on this ability for effective motion control.

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

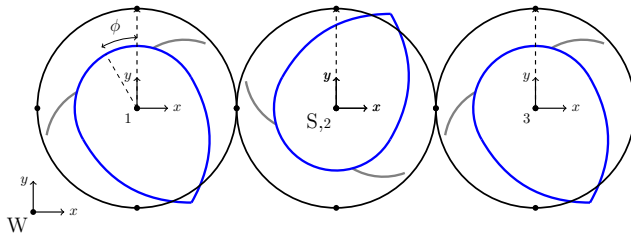


Fig. 1. Parallel lattice of three docked Modboats; for each boat, the top body is shown in black, while the bottom body/tail is in blue and flippers are in gray. Magnetic docking points are shown at the cardinal points of each boat, and the motor angle ϕ is the angle of the bottom body relative to the top body. Frame W is the fixed world frame, and frame S is the body-fixed frame at the COM, which coincides with the body-fixed frame of boat 2. The individual boat frames are aligned with the top bodies.

The Modboat, introduced in prior work [12] [13], is an inexpensive, underactuated planar modular robot. Passive flippers convert the oscillation of its two concentric bodies (see Fig. 1 and [13]) into propulsion and steering, with symmetric oscillations generating propulsion and asymmetric ones generating torques. We have also shown that the Modboat is capable of docking to and undocking from other modules to form a 4-connected structured arrangement [12], which we call a *lattice*. Control of this lattice, however, has not been attempted before this work, and is a significant challenge for several reasons:

- 1) Aquatic lattice control to date (e.g Roboats and TEMP) has required holonomic modules, but Modboat modules are underactuated. Force can be created in any direction on average over a long time-scale, but not sufficiently quickly to use holonomic approaches for control.
- 2) Modboat modules are connected via permanent magnets to enable latch-less docking/undocking [12]. The magnetic bond is relatively weak, however, and the magnets provide effectively no resistance to rotation. Angular acceleration between modules can cause them to undock if unchecked, as can undue lateral forces.
- 3) Modboat modules use intentional self-collision between their tails (see Fig. 1 and [12]) to undock without additional actuation. Swimming as a unit, however, requires avoiding unintentional undocking, which creates a complex constraint on allowable control inputs.

These challenges require significant attention to internal forces, since we cannot create arbitrary forces to balance the lattice and must account for possible changes in thrust direction as modules shift on their docks. Our long-term goal is to control arbitrary configurations of docked Modboats,

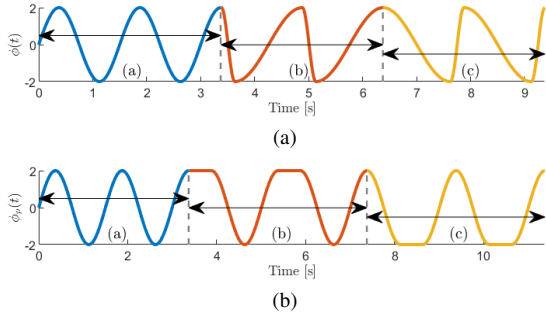


Fig. 2. Examples of (a) differential thrust and (b) inertial control waveforms used for single boat control as defined in prior work [14], with colors and arrows indicating regions in which the control parameters change. Neither is suitable for group control because phase differences between neighbors will occur.

but in this work we focus on *parallel lattices* (i.e. where all modules are placed on a line perpendicular to the desired “front” direction, such as shown in Fig. 1). The contribution of this work is thus a centralized control strategy for parallel lattices of Modboats that (1) functions for underactuated modules capable of thrust along a single axis, (2) minimizes internal forces within the lattice, and (3) guarantees no unintentional undocking of modules.

The rest of this work is organized as follows. In Sec. II we discuss a strategy for avoiding unintentional undocking, and develop a control formulation that follows this strategy in Sec. III. Sec. IV presents experimental verification of controller performance, which is discussed in detail in Sec. V.

II. AVOIDING UNINTENTIONAL UNDOCKING

As shown in Fig. 1, the bottom body of the Modboat (blue) and flippers (gray) are fully contained within the footprint of the top body (black) except for the tip of the tail. This ensures that the flippers of neighboring modules cannot mechanically interact, but the tails can be used to undock from neighboring modules by bringing them into contact (see Fig. 1) [12], which is essentially self-collision within the lattice. This is advantageous because docking and undocking can be performed without additional actuation, but it introduces a complex constraint during group control while swimming as a single unit.

We cannot use either of the single-module control modes defined in our prior work [14] and shown in Fig. 2 when the self-collision constraint is considered. Differential thrust, shown in Fig. 2a, uses frequency modulation of the motor angle to steer; inertial control, shown in Fig. 2b, uses pauses in the motor angle waveform to steer. Both control methods quickly lead to phase differences between the tail angles ϕ of neighboring boats (see Fig. 1) such that no guarantees can be made on avoiding self-collision. We also wish to avoid simply limiting the amplitude, which would enforce an overly restrictive thrust limit.

Consider two neighboring Modboats, each executing the waveform shown in (1) with amplitude A , centerline ϕ_0 , and angular frequency ω , where the subscripts i and k indicate the boat and the cycle respectively. The parameters of (1) are chosen at the beginning of each cycle and executed for

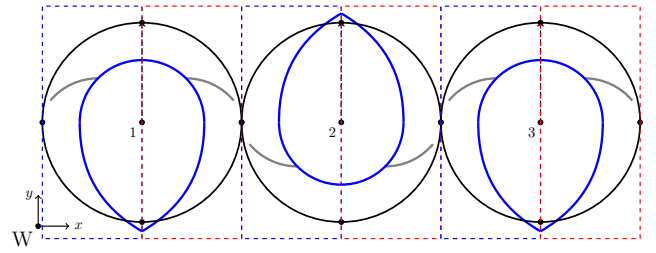


Fig. 3. Parallel lattice of three docked Modboats, with red-blue tiling used in proof for Theorem 1. Modboat 2 is shown in the reverse paddling configuration ($\phi = \phi_0 = \pi$ in (1)), while the rest are in the forward configuration ($\phi = \phi_0 = 0$), shown at $t = 0$.

a complete period. Considering the complex geometry of the tail, characterizing the set of parameters (ϕ_0, A, ω) for neighboring boats i and j that would lead to self-collision anywhere in the lattice is a complex task. We can instead consider a conservative approach that guarantees no unintentional undocking while still maintaining controllability for the lattice as a whole.

$$\phi_i(t) = (\phi_0)_{i,k} + A_{i,k} \cos(\omega t) \quad (1)$$

Assumption 1 (Phase Lock). *All boats are in phase with one another for all time, i.e. there is no phase offset in (1) and ω is the same for all boats¹. Control decisions are made simultaneously for all boats in the configuration at the end of each cycle.*

Assumption 2 (Forward/Reverse). *All boats can choose the centerline of rotation $(\phi_0)_i$ to be either 0 (forward) or π (reverse) in each cycle. No other angles are allowed.*

Under assumption 2 we can redefine (1) into (2). This minimizes the discontinuity in the wave when transitioning between choices of $(\phi_0)_i$, since $\cos((\phi_0)_i) = \pm 1$.

$$\phi_i(t) = (\phi_0)_{i,k} + A_{i,k} \cos(\omega t) \cos((\phi_0)_{i,k}) \quad (2)$$

Assumptions 1 and 2 ensure that each boat maintains its ability to create both forward and reverse thrust, but guarantee that collisions can not occur, as in Theorem 1.

Theorem 1 (No Unintentional Undocking). *When swimming under assumptions 1 and 2, and with waveform given by (2), we can guarantee that no unintentional undocking events will occur within a parallel lattice.*

Proof. Construct a lattice of neighboring red and blue regions, as shown in Fig. 3. For any lattice length, it is trivial to periodically tile. Consider some time t_0 when $\cos(\omega t_0) = 1$. Then $\phi_i = (\phi_0)_i + A \cos((\phi_0)_i) \forall i$, and

$$\phi_i = \begin{cases} (\phi_0)_i & t = t_0 + T/4 \\ (\phi_0)_i - A \cos((\phi_0)_i) & t = t_0 + T/2 \\ (\phi_0)_i & t = t_0 + 3T/4 \\ (\phi_0)_i + A \cos((\phi_0)_i) & t = t_0 + T \end{cases} \quad \forall i$$

¹The angular frequency ω can change between cycles k , although without loss of generality we assume it does not.

Thus the tail segments are **all** in red regions² when $t \in (t_0, t_0 + T/4) \cap (t_0 + 3T/4, t_0 + T)$, and blue regions when $t \in (t_0 + T/4, t_0 + 3T/4)$. Thus, at all times all tails occupy identically colored regions. By construction, no neighboring regions share a color, so no collisions are possible. \square

Note that waveform (2) and Theorem 1 rely exclusively on internal motor position. Thus their guarantees remain in place despite external disturbances, unless water conditions cause large enough forces to detach the magnets and undock modules. Dealing with disturbances of such magnitude is outside the scope of this work.

III. CONTROLLING THE LATTICE

Consider a Modboat lattice as shown in Fig. 1. A surge force F_y and torque τ applied to the lattice are sufficient to control it in the plane, and for $N \geq 2$ boats these can be arbitrarily set by the modules (within actuation limits). For $N > 2$, however, the lattice is overdetermined. Thus the challenge is to determine what forces each Modboat module must apply to steer the lattice.

Let frame W denote the fixed world frame, and let frame S represent the body-fixed structural frame centered at the center of mass (COM) of the lattice. For each boat $i \in [1, \dots, N]$, frame B_i is then attached to its own COM and aligned with the top body orientation. Without loss of generality, we can define $B_i \forall i$ to be aligned with S so that only a translation is needed to transform between them. We define x_i to be the distance along \hat{x}_S from S to B_i .

In the following we assume a planar model (i.e. the water is flat and no waves are present), and that there are no hydrodynamic interactions between boats in the lattice. The extent to which this latter assumption is valid is left to future work, but the controller performance presented in Sec. IV shows that it is not unreasonable.

A. Dynamic Model

When executing the waveform given in (2), each Modboat can be considered to be producing thrust $f_i \in [-f_{max}, f_{max}]$ on average along the direction given by $(\phi_0)_i$ over a full cycle of length T (where T is the period corresponding to the angular frequency ω). Under assumption 2 each force f_i acts along the \hat{y}_S (surge) axis.

We define v_y and a_y as the velocity and acceleration along the surge axis, respectively. We can then write the equations of motion for the lattice as in (3) and (4), where $m = \sum m_i$ is the total mass of the lattice, and I is the total moment of inertia about the z axis. The angular acceleration and velocity of the lattice are given by α and Ω , respectively, and C_L and C_R are drag coefficients in the forward and yaw directions, respectively. Note that the equations are written continuously

²The tail tip will eventually enter the neighboring boat's blue region. For the purposes of the proof, we consider a slightly interior point that remains within the red region.

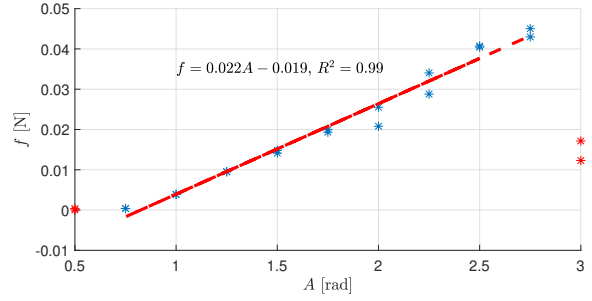


Fig. 4. The experimentally determined force vs. amplitude curve for a period of oscillation $T = 1.5$ s. Data points are shown in blue, and red stars indicate poor force generation due to incomplete flipper activation (at low amplitudes) or significant reverse thrust (at high amplitudes).

in time since the average thrust is used.

$$ma_y = \sum_i f_i - C_L |v_y| v_y \quad (3)$$

$$I\alpha = \sum_i f_i x_i - C_R |\Omega| \Omega \quad (4)$$

Our desired controller will track a commanded surge velocity v_c . Since this velocity will remain generally consistent, we can consider forward motion as occurring in steady state, in which thrust balances drag on average. Thus we can take $a_y = 0$. The yaw angle is allowed to change more frequently, so we do not assume steady state in yaw.

Let $\vec{f} \in \mathbb{R}^N$ represent the vector of boat forces $\vec{f} = [f_1 \dots f_N]^T$. We can then define a structural matrix P as in (5), and (3) and (4) can be rewritten in vector form (6).

$$P = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 \\ x_1 & x_2 & \dots & x_{N-1} & x_N \end{bmatrix} \quad (5)$$

$$P\vec{f} = \begin{bmatrix} C_L |v_c| v_c \\ I\alpha + C_R |\Omega| \Omega \end{bmatrix} \quad (6)$$

In (6), $P\vec{f} \in \mathbb{R}^2$ represents the surge force and yaw torque that steers the lattice as a whole. Applying suitable forward force will create surge velocity, while torquing the lattice allows it to maintain a desired yaw and heading. We can then use the Moore-Penrose inverse $P^+ = P^T (PP^T)^{-1}$ to find the forces desired from each boat, as in (7).

$$\vec{f} = P^+ \begin{bmatrix} C_L |v_c| v_c \\ I\alpha + C_R |\Omega| \Omega \end{bmatrix} \quad (7)$$

Using (7) to distribute forces among the modules results in a linear distribution along the x -axis of the lattice; a similar approach was used by Gabrich to distribute forces in a lattice of docked quadrotors [15]. Other distributions are possible, but a linear distribution most closely matches the internal dynamics that would be observed if the lattice were a single rigid body. As discussed in Sec. I, Modboat modules are docked using passive magnets that allow the units to rotate somewhat relative to one another, and significant intra-lattice forces can cause enough rotation to undock. Maintaining a rigid-body force distribution minimizes such forces and oscillation between neighboring modules.

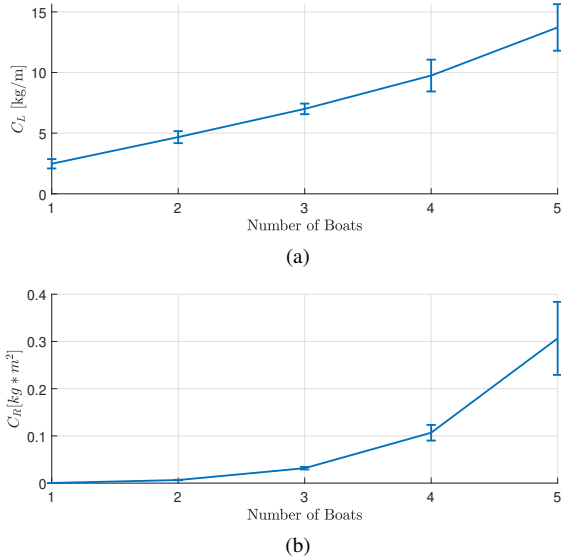


Fig. 5. Experimentally determined drag coefficients (a) C_L and (b) C_R vs. number of boats in a parallel configuration, as defined in (3) and (4). The linear drag coefficient C_L is roughly linear in the number of boats, while the angular drag coefficient C_R is roughly quadratic. Error bars indicate one standard deviation.

B. Control Input

The control structure given in (7) can track a desired surge velocity v_d and yaw angle θ_d . Since forward motion occurs in steady-state, it should be sufficient to use the desired velocity v_d as the commanded velocity v_c in (7). In practice it is observed that this is not sufficient, however, and using v_d directly results in very slow observed velocity v_{obs} . Instead, an artificial linear acceleration is calculated in (8) and (9) and used to integrate the commanded velocity as in (10). This discrepancy between v_c and v_d may be a result of hydrodynamic interactions between neighbors, but a full consideration is left to future work.

$$e_v = v_d - v_{obs} \quad (8)$$

$$\tilde{a}_y = K_{pv}e_v + K_{dv}\frac{de_v}{dt} \quad (9)$$

$$v_c = v_d + \int_0^t \tilde{a}_y d\tau \quad (10)$$

The yaw angle can be commanded by a standard PD control loop on the angular acceleration, as given in (11) and (12), where Θ and Θ_{des} are the observed and desired yaw angles for the lattice, respectively.

$$e_\Theta = \Theta_{des} - \Theta \quad (11)$$

$$\alpha = K_{p\Theta}e_\Theta + K_{d\Theta}\frac{de_\Theta}{dt} \quad (12)$$

C. Input Values

Eq. (7) gives the force distribution for each module and is valid for any single-thruster robot, but the required input for the Modboats is the choice of centerline $(\phi_0)_i$ and oscillation amplitude A_i . Theorem 1 guarantees there will be no unintentional undocking, so we can safely map the desired forces to those values. Since there is a practical

TABLE I

MASS AND INERTIA FOR EACH PARALLEL BOAT CONFIGURATION.

N	Boats	1	2	3	4	5
M	[kg]	0.66	1.32	1.98	2.64	3.30
I	[g * m ²]	2.05	11.8	36.8	84.8	164
C_L	[kg/m]	2.48	4.67	7.00	9.75	13.7
C_R	[g * m ²]	0.40	6.50	32.0	107	307

TABLE II

CONTROLLER PERFORMANCE ACROSS ALL CONFIGURATIONS AS IQR. RMS ERROR IS AFTER THE RISE TIME, FROM 0.3 – 0.9 OF THE STEP.

	Swim/Yaw only	Swim and Yaw
v_y RMS [cm/s]	[0.13, 0.38]	[0.23, 0.42]
v_y Rise [s]	[4.6, 6.4]	[4.5, 8.0]
θ RMS [rad]	[0.10, 0.24]	[0.065, 0.15] [†]
$\pi/2$ step θ Rise [s]	[3.7, 5.0]	N/A
π step θ Rise [s]	[3.9, 6.9] [†]	[5.7, 13.3]

[†] Statically significant improvement over the other case in each row.

maximum to the force the Modboats can generate, f_i is first capped by f_{max} . Leveraging the symmetry of the Modboat the centerline for each boat $(\phi_0)_i$ is determined by $\text{sgn}(f_i)$, and the amplitude is determined by $|f_i|$.

The mapping $f_i = f(A_i)$ can be determined experimentally as follows. We created a parallel lattice of three Modboats and commanded only the center module to execute the waveform in (2) with $T = 1.5$ s and $A \in [0.5, 3.0]$ rad. The period-wise surge velocity v_y was calculated for the lattice as in (13), and the steady-state velocity was converted to force via the quadratic drag relationship $f_i = C_L v_y^2$ and the drag coefficient as calculated below.

$$\vec{v}_T(t) = \frac{\vec{x}(t) - \vec{x}(t - T)}{T} \quad (13)$$

Fig. 4 shows the resulting mapping $f = f(A)$, which is linear within the range $A \in [0.75, 2.75]$ rad. Below 0.75 rad (with $T = 1.5$ s) the flippers do not fully open, so negligible thrust is produced. Above 2.75 rad, the tail rotates enough to produce significant reverse thrust during a portion of the cycle. Thus we intentionally limit the maximum allowable amplitude to 2.75 rad³.

The mass of an individual boat m_i can be measured, and the moment of inertia for an individual boat I_i can be calculated from its Solidworks model files. The total moment of inertia I can then be calculated via the parallel axis theorem for any lattice. The drag coefficients C_L and C_R can be experimentally calculated; a linear/angular impulse is delivered to the lattice, and the resulting linear/angular velocity is tracked. A nonlinear least-squares fit to a quadratic drag model then gives C_L and C_R . The resulting drag coefficients are shown in Fig. 5, where C_L is shown to be roughly linear and C_R is roughly quadratic.

IV. EXPERIMENTS

Experiments were performed in a 4.5 m × 3.0 m × 1.2 m tank of still water. An OptiTrack motion capture system

³In practice the maximum amplitude used experimentally is 2.5 rad, since higher amplitudes are found to cause the lattice to shake internally.

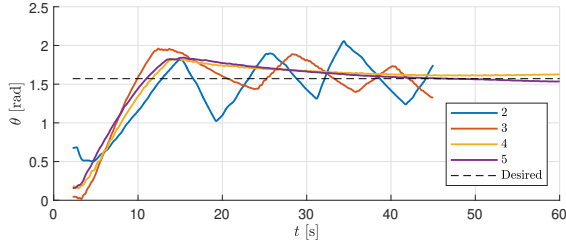


Fig. 6. Yaw response to a step input of $\pi/2$ rad for configurations of 2–5 boats. The controller provides comparable rise time performance regardless of the number of boats in the lattice, but oscillation amplitudes decrease as the lattice grows.

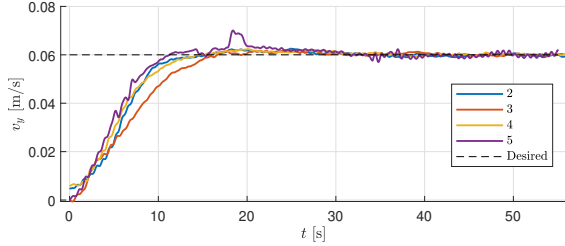


Fig. 7. Velocity response to a step input of 6 cm/s for configurations of 2–5 boats. The controller provides comparable performance regardless of the number of boats in the lattice.

provided planar position, orientation, and velocity data at 120 Hz, and a MATLAB script calculated control inputs and transmitted waveform parameters to the various boats over WiFi. All experiments were performed for configurations of 2–5 boats aligned in parallel (as in Fig. 1), and their measured parameters are presented in Table I.

The drag coefficients for parallel configurations of 1–5 boats⁴ were evaluated by applying impulses and performing a least-squares fit of a quadratic drag model to the resulting velocity data. The resulting coefficients are shown in Fig. 5, where the linear drag is roughly linear in the number of boats, and the angular drag is roughly quadratic.

Given the drag coefficients and the thrust-amplitude map in Fig. 4, we evaluated the yaw and velocity tracking abilities of the controller. Yaw control was evaluated by commanding $v_y = 0$ and a step input for θ_{des} ; results are shown in Fig. 6, and a sample trajectory is shown in Fig. 9. Velocity tracking was similarly evaluated by commanding a step input forward velocity of either 4 cm/s or 6 cm/s while not commanding a desired yaw; representative results are shown in Fig. 7. Finally, the combined controller was evaluated by commanding a desired yaw θ_{des} and velocity for 45 s and then changing the yaw to $\theta_{des} + \pi$ (i.e. a 180° turn) with the same velocity for another 45 s. A sample such trajectory is shown in Fig. 8.

To evaluate the stability of the controller for various configurations, we also considered how the system would behave under mismatched drag coefficients. To consider the most extreme example of this, the five-boat parallel configuration

⁴Although our controller is degenerate for a single boat, we include its parameters in Table I for completeness.

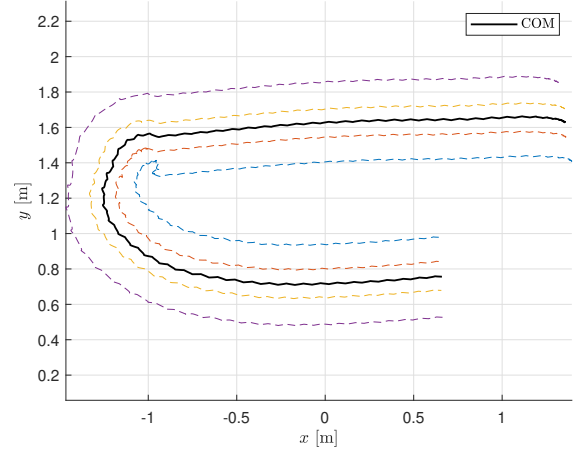


Fig. 8. Five boat configuration tracking a yaw of π rad (left) for 45 s, then tracking a yaw of 0 rad (right) for 45 s, all while maintaining a velocity of 6 cm/s. Each dashed color is an individual module, and the solid black line is the center of mass.

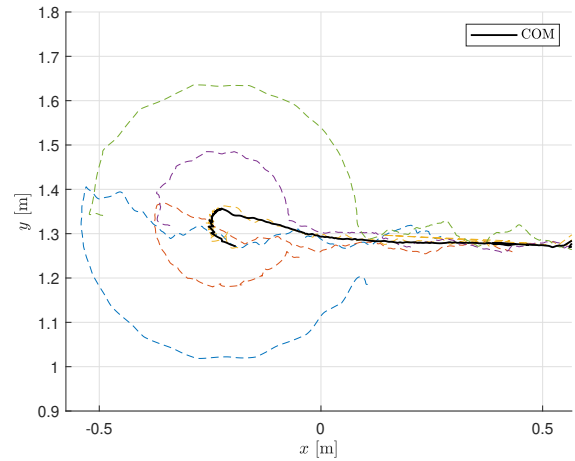


Fig. 9. Trajectory of five boat lattice responding to a π rad step input without a desired velocity. Dotted lines show individual boat trajectories, while the solid line follows the center of mass. The lattice turns in place until the desired yaw is achieved, and then drifts uncontrollably sideways due to the discontinuity in (2).

evaluations were repeated with the assigned drag coefficients as measured for a two-boat configuration, and the results are shown in Fig. 10.

V. DISCUSSION

The controller presented in Sec. III is intended to allow parallel lattices of Modboats to be controllable as a single unit. The experimental results presented in Sec. IV show that this is the case: parallel lattices of Modboats can be driven at a desired surge velocity and to a given yaw angle. With a suitable control law, this performance can easily be extended to waypoint tracking and more complex behaviors.

Velocity tracking, presented in Fig. 7 and summarized in Table II, is highly effective regardless of the lattice size. All tested lattices tracked the desired velocity to within 0.3 cm/s and achieve it within 5.7 s, which shows that the controller is not sensitive to lattice size. The controller can achieve velocities from 3.0–8.0 cm/s, bounded by the minimum

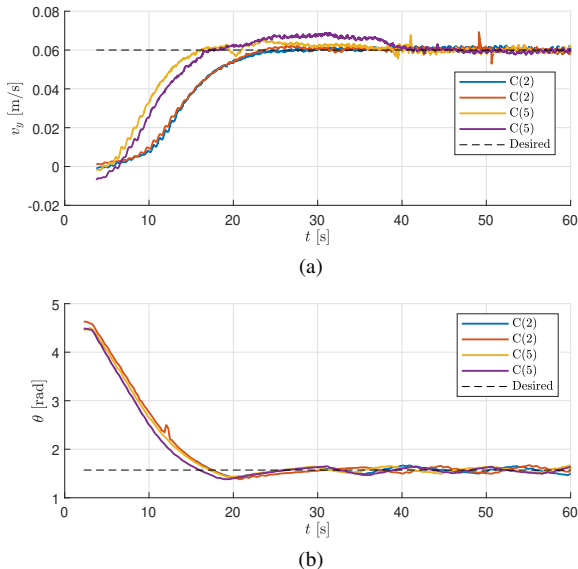


Fig. 10. (a) Velocity and (b) yaw step response for the 5 boat configuration where the drag coefficients are chosen as $C_L(n)$ and $C_R(n)$ with $n = 5$ and $n = 2$. Two tests are shown for each.

amplitude needed to activate the flippers and the maximum allowable amplitude. While a maximum speed of 0.5 body-lengths per second (the Modboat diameter is 15.2 cm) is not fast, it is sufficient. Velocity tracking performance is comparable during combined yaw/velocity testing for both rise time and RMS error. There is a noticeable drop in velocity during the turning maneuver (see Fig. 8), but it recovers similarly once the turn is complete.

Yaw can be tracked very accurately when the lattice is moving. Small adjustments to the amplitude of each boat’s oscillation yaws the lattice and tracks the desired heading to within 0.11 rad (6.5°), as shown in Table II. Yaw tracking suffers, however, when no velocity is desired; although the yaw can be driven to the desired value overall, the tracking error increases significantly, as shown in Table II. This occurs because the lattice attempts to remain in place, but finer adjustments require reversing thrust. If there is enough initial overshoot, significant oscillations around the desired value begin and are not damped by the derivative term, such as for the two and three boat lattices in Fig. 6 with $\pi/2$ rad step input and for the four and five boat lattices in Fig. 10b with π rad step input. Otherwise, a significant settling time is seen as the boats overcome the large inertia of the lattice, such as for the four and five boat lattices in Fig. 6. Pure yaw control, however, displays reasonable rise times that are not significantly affected by the step input, as shown in Table II, whereas rise time when swimming is significantly slower.

A secondary problem with controlling yaw alone is shown in Fig. 9; after achieving the desired yaw with minimal COM motion, the lattice drifts sideways. Since our controller can produce forces only along its surge axis, we cannot counteract this. Both issues stem from the same cause: the Modboats’ unique propulsive mechanism cannot smoothly transition from forward to reverse. Eq. (2) is discontinuous when $(\phi_0)_i$ changes, especially if A_i is small. The transition from forward to reverse thrust thus creates sideways forces

and yaw torques that disrupt the controller.

We also consider the sensitivity of the controller to the drag coefficients. Fig. 10 shows the controller performance when given incorrect drag coefficients. Behavior for velocity (Fig. 10a) matches what might be intuited: it takes longer to achieve the desired velocity, but the feedback controller is sufficient since the desired velocity is achievable. The final velocity tracking, which is based on a thrust/drag balance, is comparable to that achieved with the correct drag coefficients. The behavior in yaw (Fig. 10b) shows similar behavior, although the transient region is closer to the original behavior. This is likely because the $I\alpha$ term dominates the transient region, minimizing the effect of the mismeasured C_L . This behavior is encouraging, as it is likely that non-parallel configurations will be controllable even if their drag coefficients are not explicitly characterized.

VI. CONCLUSION

In this work we have presented a centralized approach that allows parallel lattices of underactuated modules to swim as a single unit. When applied to lattices of Modboats, it minimizes internal forces and guarantees no unintentional undocking. The controller is capable of tracking a desired surge velocity and yaw angle for configurations of various sizes, and has been experimentally verified for lattices consisting of two to five boats. Velocity tracking is shown to be highly effective, and yaw tracking is shown to be effective when the lattice is moving forward.

The controller struggles to track yaw angle while stationary, however, generating oscillations and sideways drift that cannot be counteracted by the given control law. This is poorly suited for docking or station keeping, both of which require precise orientation control while stationary. Future work will consider ways to add control of the lattice’s sway axis and reduce the observed oscillations.

A number of approaches for controlling smaller lattices — i.e. single Modboats — already exist [14], [16]. For larger lattices we theorize that our controller will continue to perform well, although problems may arise as the yaw authority of individual boats scales linearly with their distance from the center of mass, but the angular drag and inertia scale quadratically. Velocity tracking is likely to remain effective, but our testing tank is too small for larger configurations.

Finally, although we have considered only parallel lattices in this work, the control structure does not explicitly contain this restriction. Minor adjustments should allow the controller to function effectively for arbitrary lattices. Future work will consider the behavior of this controller for non-parallel lattices while still guaranteeing no unintentional undocking. We will also explore additional ways to avoid undocking that allow more flexible control inputs, which should increase the capabilities of the final lattice. Finally, we hope to consider the robustness of this controller to disturbances such as are commonly found in the ocean.

ACKNOWLEDGMENT

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

REFERENCES

- [1] 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.
- [2] 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.
- [3] S. Mintchev, E. Donati, S. Marrazza, and C. Stefanini, "Mechatronic design of a miniature underwater robot for swarm operations," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014, pp. 2938–2943.
- [4] S. Mintchev, R. Ranzani, F. Fabiani, and C. Stefanini, "Towards docking for small scale underwater robots," *Autonomous Robots*, vol. 38, no. 3, pp. 283–299, 8 2014.
- [5] J. Seo, M. Yim, and V. Kumar, "Assembly planning for planar structures of a brick wall pattern with rectangular modular robots," in *IEEE International Conference on Automation Science and Engineering*, 2013, pp. 1016–1021.
- [6] J. Seo, M. Yim, and V. Kumar, "Assembly sequence planning for constructing planar structures with rectangular modules," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 6 2016, pp. 5477–5482.
- [7] 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.
- [8] W. Wang, L. A. Mateos, S. Park, P. Leoni, B. Gheneti, F. Duarte, C. Ratti, and D. Rus, "Design, Modeling, and Nonlinear Model Predictive Tracking Control of a Novel Autonomous Surface Vehicle," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018, pp. 6189–6196.
- [9] W. Wang, T. Shan, P. Leoni, D. Fernández-Gutiérrez, D. Meyers, C. Ratti, and D. Rus, "Roboat II: A Novel Autonomous Surface Vessel for Urban Environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV (Virtual), 2020, pp. 1740–1747.
- [10] W. Wang, Z. Wang, L. Mateos, K. W. Huang, M. Schwager, C. Ratti, and D. Rus, "Distributed motion control for multiple connected surface vessels," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV (Virtual): Institute of Electrical and Electronics Engineers Inc., 10 2020, pp. 11 658–11 665.
- [11] S. Park, E. Kayacan, C. Ratti, and D. Rus, "Coordinated control of a reconfigurable multi-vessel platform: Robust control approach," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 5 2019, pp. 4633–4639.
- [12] G. Knizhnik and M. Yim, "Docking and Undocking a Modular Underactuated Oscillating Swimming Robot," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021, pp. 6754–6760.
- [13] G. Knizhnik, "Modboat: A single-motor modular self-reconfigurable robot," Feb 2022, <https://www.modlabupenn.org/modboats/>.
- [14] G. Knizhnik, P. Dezonias, and M. Yim, "Pauses Provide Effective Control for an Underactuated Oscillating Swimming Robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5075–5080, 10 2020.
- [15] B. Gabrich, G. Li, and M. Yim, "ModQuad-DoF: A Novel Yaw Actuation for Modular Quadrotors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 5 2020, pp. 8267–8273.
- [16] G. Knizhnik and M. Yim, "Thrust Direction Control of an Underactuated Oscillating Swimming Robot," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic (Virtual), 2021, pp. 8665–8670.