Collective Control for Arbitrary Configurations of Docked Modboats

Gedaliah Knizhnik and Mark Yim

Abstract—The Modboat is a low-cost, underactuated, modular robot capable of surface swimming, docking to other modules, and undocking from them using only a single motor and two passive flippers. Undocking is achieved by causing intentional self-collision between the tails of neighboring modules in certain configurations; this becomes a challenge, however, when collective swimming as one connected component is desirable. In this work, we develop a centralized control strategy to allow arbitrary configurations of Modboats to swim as a single steerable vehicle and guarantee no accidental undocking. We also present a simplified model for hydrodynamic interactions between boats in a configuration that is tractable for real-time control. We experimentally demonstrate that our controller performs well, is consistent for configurations of various sizes and shapes, and can control both surge velocity and yaw angle simultaneously. Controllability is maintained while swimming, but pure yaw control causes lateral movement that cannot be counteracted by the presented framework.

Index Terms—Cellular and modular robots, marine robotics, underactuated robots, cooperating robots.

I. INTRODUCTION

Aquatic systems that can dock, undock and reconfigure are of interest to researchers and industry; they have 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. Such applications can be accomplished by systems composed of large numbers of modules, but conventional wisdom has been that the individual modules must be capable of holonomic motion [1]–[4], which makes them expensive and limits their number. Allowing the individual modules to be under-actuated would reduce their cost and allow scaling such systems to be more efficient.

In a docked aquatic system, however, challenges exist even when the individual modules are capable of holonomic motion. These challenges include robust docking and undocking [2], [5], assembly and disassembly of floating structures from individual modules [6], [7] and from substructures, minimizing disturbances and forces within the configuration [1], [2], achieving consensus without a centralized controller [8], and finding the optimal distribution of effort across the structure [9], [10]. If we seek to reduce the cost and complexity of the modules by reducing their overall mobility, these problems become much harder.

Most non-docking aquatic robots are built in either a thruster-rudder or a differential thrust arrangement [11], and the ability of such systems to move competently on their own has been widely explored in the literature. In either arrangement the ability of an individual module to thrust is restricted to a linear direction in a rigid docked configuration, with the ability to yaw greatly reduced due to the relative scaling of thrust and inertia that grows unfavorably for modular systems [12]. We can therefore consider modules that are only capable of thrust along a single axis as a reasonable model of a general aquatic surface robot, and must find a non-trivial thrust distribution to allow such modules to collaborate when docked.

A further restriction is added when considering more unique individual modules. Modboats, introduced by the authors in prior work [13], utilize a unique combination of a single motor and passive flippers for propulsion and steering (originally introduced by Refael and Degani [14], [15]), and have been shown to be capable of complex motions when swimming individually [16], [17]. Modboats are capable of docking in a rectangular lattice through passive magnetic docks and use a tail rigidly connected to their actuating body to undock without additional actuators [18], reducing cost, mass and complexity. However, this tail — while enabling undocking — introduces a significant restriction on allowable thrusts and motions when collective motion is desirable. Naive application of control strategies, without consideration of this restriction, would immediately generate undesirable undocking behavior, and the docked structure would quickly disintegrate. This motivates us — as will be presented in Sec. V — to consider a configuration of modules that can each thrust along a single axis where all these axes are aligned.

Moreover, a significant issue to consider when controlling any aquatic robots swimming in close proximity, and especially when they are docked together, is the effect of hydrodynamic interaction between individual actuators. While this effect certainly exists for conventional propeller-craft [19], it is a significant and complex phenomenon for tail or flipper driven swimmers, such as the Modboat. Significant work in the literature has been dedicated to exploring the effects of fluid interactions, vortices, and formation on fish swimming [20]–[29]. At the very least, we must acknowledge that actuator dynamics must be adjusted when applied to configurations of coordinating swimmers.

The contribution of this work, therefore, is to develop a relatively simple centralized control strategy that can be applied to arbitrary rectangular configurations of docked aquatic swimmers, each capable of thrusting along a single axis, aligned with each other. We show how such a strategy can be implemented using the Modboat while guaranteeing that modules will not unintentionally undock during collective swimming, and demonstrate that a relatively simple model of hydrodynamic interaction provides strong performance when tracking yaw and surge velocity.

This work is an extension of our prior work [30], in which such a strategy was shown for parallel configurations only.

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

Consider a set of swimming robotic modules arranged on a rectangular lattice, as shown in Fig. 1. Each module is capable of producing both positive and negative thrust along its own body-fixed $y$ axis on average over a period of length $T$, and we assume that all the $y$ axes are aligned with a body-fixed frame $S$ at the center of mass (COM) of the configuration. For the sake of clarity in this work we will use both 2-dimensional and linear numbering for the modules in the configuration where appropriate; a tuple of subscripts indicates a 2-dimensional numbering, and a single subscript indicates linear numbering. Any suitable mapping between the two can be used for conversions.

As long as the number of modules $N \geq 2$ and the configuration is at least two modules wide, the structure is controllable under these assumptions. Appropriate choices of forces $f_i$, $i \in [1, N]$, for each individual module can generate desired surge (forward, or $y$ axis) forces and yaw torques on the configuration, which is sufficient to control it in the plane. No express control is given along the $x$ axis, so disturbances along the $x$ axis are assumed to be small.

Define $v_y$ and $a_y$ as the velocity and acceleration of the COM along the $y$ axis, respectively, and $\Omega$ and $\alpha$ as the angular velocity and acceleration of the structure, respectively. We can then write the dynamics of the configuration as in (1) and (2), where $x_i$ is the distance from the COM to module $i$ along the $\hat{x}_S$ axis, $CL$ and $CR$ are drag coefficients, and $m$ and $I$ are the mass and moment of inertia of the configuration, respectively.

\begin{align}
ma_y &= \sum_i f_i - CL \sgn(v_y) v_y^2 \\
I\dot{\alpha} &= \sum_i f_i x_i - CR \sgn(\Omega) \Omega^2
\end{align}

Equations (1) and (2) can be rewritten in matrix form by defining the structural matrix $P$ as in (3). This gives (4), where $\vec{f} = [f_1 \ f_2 \ \ldots \ \ f_N]^T$ is the vector of module forces. The right hand side of (4) then represents the surge force and yaw torque that need to be applied to the structure to generate a desired surge velocity $v_c$ and to track a desired yaw angle $\Theta$.

\begin{align}
P &= \begin{bmatrix}
1 & 1 & \ldots & 1 & 1 \\
x_1 & x_2 & \ldots & x_{N-1} & x_N
\end{bmatrix}
\end{align}

\begin{align}
PF &= \begin{bmatrix}
ma_y + CL|v_c|v_c \\
I\dot{\alpha} + CR|\Omega|\Omega
\end{bmatrix}
\end{align}

For the purposes of control, we consider a case in which the desired surge velocity is (relatively) constant over time, while the desired yaw angle is varied to steer. This allows us to assume steady-state in the velocity equation and set $a_y = 0$.

Eq. (4) can then be solved for the individual module forces $\vec{f}_i$, as in (5), by using the Moore-Penrose pseudo-inverse $P^+ = P^T(PP^T)^{-1}$.

\begin{align}
\vec{f}_i &= P^+ \begin{bmatrix}
CL|v_c|v_c \\
I\dot{\alpha} + CR|\Omega|\Omega
\end{bmatrix}
\end{align}

Using (5) to distribute forces among the modules results in a linear distribution along the $x$-axis of the configuration; a similar approach was used by Gabrich to distribute forces in a configuration of docked quadrotors [12]. Other distributions are possible, but a linear distribution most closely matches the internal dynamics that would be observed if the configuration were a single rigid body. As discussed in Sec. IV, Modboat modules are docked using passive magnets that allow rotation, and significant intra-configuration forces can cause modules to undock. Maintaining a rigid-body force distribution minimizes such forces and oscillation between neighboring modules.

A. Control Input

Eq. (5) allows the configuration of Modboats to track a desired surge velocity $v_d$ and yaw angle $\theta_d$. Surge motion is assumed to occur at steady-state on average, so it should be enough to use the desired velocity $v_d$ as the commanded velocity...
velocity \( v_c \) in \( \text{[5]} \). In practice, it is observed that this is not sufficient, however, so \( v_d \) is instead used as a feedforward term, and a PD controller acts as an artificial acceleration to adjust the commanded velocity \( v_c \). The controller is given in \( \text{(5)} \) and \( \text{(7)} \), where \( v_{\text{obs}} \) is the observed surge velocity of the configuration.

\[
e_v = v_d - v_{\text{obs}}
\]

\[
v_c = v_d + \int_0^t \left( K_{pe} e_v + K_{dv} \frac{dv}{dt} \right) \, dt
\]

The yaw angle can be commanded by a standard PD control loop on the angular acceleration, as given in \( \text{(8)} \) and \( \text{(9)} \), where \( \Theta \) and \( \Theta_{\text{des}} \) are the observed and desired yaw, respectively. The angular velocity \( \Omega \) used in \( \text{(5)} \) is then the observed angular velocity of the configuration.

\[
e_{\dot{\Theta}} = \Theta_{\text{des}} - \Theta
\]

\[
\alpha = K_{p\dot{\Theta}} e_{\dot{\Theta}} + K_{d\dot{\Theta}} \frac{d\dot{\Theta}}{dt}
\]

### B. Drag Coefficients

Using \( \text{[5]} \) to drive the Modboat configuration requires knowledge of the constants relevant to the system, namely the moment of inertia \( I \) and the linear and angular drag coefficients \( C_L \) and \( C_R \). The moment of inertia for an individual boat \( I_i \) can be calculated from its Solidworks model files, and the total moment of inertia \( I \) can then be calculated in a straightforward way via the parallel axis theorem for any configuration.

The drag coefficients \( C_L \) and \( C_R \) can be experimentally calculated; a linear/angular impulse is delivered to the configuration, and the resulting linear/angular velocity is tracked. A nonlinear least-squares fit to a quadratic drag model then gives \( C_L \) and \( C_R \) when the moment of inertia and mass are known. The resulting drag coefficients are shown in Fig. 2 and given in Table I for parallel configurations, where \( C_L \) is shown to be roughly linear and \( C_R \) is roughly quadratic.

However, it is clearly impractical to use this method for arbitrary configurations, since it requires experimental evaluation for every possible configuration. Two factors combine to allow us to avoid such labor intensive experimentation. First, experimental evaluation on a small selection of non-parallel configurations (see Fig. 3) shows that their drag coefficients correspond reasonably well to the value given based on their projection for parallel configurations, as shown in Fig. 2. Second, as we will show in Sec. VII, even using significantly mismatched drag coefficients results in only minor performance penalties after feedback is applied.

Thus, we project the configuration onto its axes and use the drag coefficients predicted for parallel configurations when applying \( \text{[5]} \) for control. For linear drag \( C_L \), this is the width along the \( x_S \) axis \( x_w \). For rotational drag \( C_R \) we use the maximum width along either the \( x_S \) or the \( y_S \) axes.

### III. Waveform

In Sec. II we posited a set of modules that can produce positive and negative thrust along their \( y \)-axes. It remains to be shown how this can be achieved with the Modbot, and how to guarantee that the resulting system is collision-free under the control implementation of Sec. I.

When Modbot modules swim alone, their top body section acts as an inertial rotor to allow the propulsive bottom body to rotate in the water \( \text{[13]} \). Nevertheless, some of the motion goes into the top body, and this must be accounted for in single boat control schemes \( \text{[16], [17]} \). However, when multiple Modboats are docked together the dock acts to significantly reduce the rotation of the top body. This allows us to equate the orientation of the bottom body \( \theta \) and the motor angle \( \phi \) (shown in Fig. 1), and consider the input waveform directly as a measure of propulsion.

Consider a set of Modboats executing the waveform given in \( \text{[10]} \), where the centerline is given by \( \phi_0 \) and the amplitude by \( \Lambda \), and the subscript \( i \) indicates the particular module. Wherever necessary, we assume that a waveform is executed with its set of parameters for a single cycle, and then the parameters are updated for the next cycle; a subscript indicating...
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).

\[ f = 0.0224 A - 0.019, R^2 = 0.99 \]

IV. HYDRODYNAMIC INTERACTIONS

As noted in Sec. [11] it is difficult to accurately model hydrodynamic interactions between swimming bodies located close to one another. Significant work has been done in the literature to determine the effects of swimmers in a configuration [20–29], but — to the best of the authors’ knowledge — none has achieved a model simple enough for effective control use. This is especially difficult for flapping swimmers, where the time scale of the wake behavior is comparable to that of the actuation.

In our prior work considering only parallel configurations [30] we assumed no hydrodynamic interactions between horizontal neighbors for ease of control, and the resulting performance was reasonable enough to justify this simplifying assumption. For non-parallel configurations, however, this assumption can no longer be justified, as performance decreases significantly when it is made\(^1\) Some hydrodynamic modeling is therefore necessary, but it is desirable that the model be readily integrated into the dynamics and control method presented in Sec. [11] which has already been validated [30].

The most obvious wake interaction to consider when extending to arbitrary configurations is the interaction between vertical neighbors. Since the Modboat’s thrust is based on the relative velocity between its flippers and the fluid, swimming in the rearward wake of another boat would be expected to decrease the thrust produced. For a first approximation we consider interactions only along the \( y \) axis of Fig. [1] (i.e. the wake does not spread laterally). This is reasonable since — in any case — we are restricting thrust to be along the \( y \) axis.

We consider the simplest model of hydrodynamic interactions: a linear thrust loss due to interactions with the wake of neighbors in front. Thus, if Sec. [III] introduced the thrust map \( f = f_0(A) \) for a Modboat swimming alone, then under the thrust loss model each boat \( i \) would produce \( f_i = \alpha_i f_0(A_i) \) for some constant \( \alpha_i \leq 1 \).

Since we are considering wakes that move only vertically, it is sufficient to consider a column of the configuration to determine the coefficients \( \alpha_i \). We can therefore experimentally model this thrust loss by placing a column of Modboats on a thrust stand and measuring the thrust produced at various oscillation amplitudes. All the Modboats were given the same amplitude \( A \); first one boat only was actuated, then the first

\(^1\) Although thrust is maintained up to 2.75 rad, in practice amplitudes higher than 2.5 rad cause the configuration to shake internally.

\(^2\) E.g. yaw tracking error while swimming for the \( L \) configuration increases by 89%, with mean 0.21 rad and IQR [0.15, 0.26] rad. Compare with mean 0.11 rad and IQR [0.051, 0.17] rad in Table [IV] when hydrodynamic effects are modeled.
two, and then three[4] Fig. 5 shows the experimental setup, and the results are given in Fig. 6.

The data shown in Fig. 6 is reasonably linear, so we are justified in pursuing a linear thrust model. To fit the coefficients $\alpha$, we found the line of best fit for the base case (with slope $m_0$) in Fig. 6 and then determined the best fit slopes for lines sharing an $x$ intercept for the remaining data sets[4]. Then the $\alpha$ coefficients can be found via (11), where $\alpha(k)$ indicates the coefficient $\alpha$ for a boat that has $k-1$ Modboats in front of it. The results are given in Table II wake effects reduce the rearward Modboats’ thrusts by around 30%.

$$\alpha(k) = \frac{m_k - m_{k-1}}{m_0} \quad k \in [1, 3]$$ \hspace{1cm} (11)

Since the thrust map $f_0(A)$ is linear, under this model we can compensate for thrust loss by increasing the commanded amplitude $A_i$ by a factor $1/\alpha_i$. It is therefore convenient to define a wake compensation factor using the coefficient $\alpha$. Recall that, under the wake model we have assumed, the thrust reduction for any boat $i$ can depend only on $k$, where there are $k-1$ boats in front of it.

**Definition 1** (Wake compensation Factor). Let the wake compensation factor $\gamma$ of a boat $i$ be given by:

$$\gamma(k) = \frac{\alpha(k-1)}{\alpha(k)} = \frac{A(k)}{A(k-1)} \quad k \in \mathbb{Z}^*$$ \hspace{1cm} (12)

Then it is always true that for a boat $(i, j)$ that has a rearward value $k$, $A(i, j) = \gamma(k)A(i, j+1)$. That is $\gamma$ defines the ratio between the amplitude of a boat and its forward neighbor. The wake compensation factors measured for the Modboats are given in Table II.

While the measurements and models presented in this section cannot capture the entirety of the hydrodynamic interactions and are a significant oversimplification, the results presented in Sec. VI demonstrate that even this first order approximation is sufficient — in conjunction with feedback control — to provide a reasonable model of system behavior.

V. AVOIDING UNINTENTIONAL UNDOCKING

In Sec. II we presented a controller for an arbitrary configuration of modules that can produce thrust along a single aligned axis, and in Sec. III we showed that Modboats can act like such a module when docked and averaged over a full cycle of length $T$.

The challenge for control of a configuration of Modboats, however, lies in the multiplexed function of the bottom body tail. 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. 10[18]), which is essentially self-collision within the configuration. This is advantageous because docking and undocking can be performed without additional actuation, but introduces a complex constraint when swimming as a unit.

It therefore remains to show that the waveform defined in Sec. III is sufficient to avoid unintentional self-collisions for all allowable inputs. More general solutions likely exist but are complex to find and define; they are deferred to future work.

Formally, we can define some consequences of the application of the waveform (10) under the restriction $\phi_0 \in \{0, \pi\}$ rad. Assumption II stems directly from the lack of a
As long as the maximum wake compensation factor occurs within an arbitrary configuration of docked Modboats, the phase offset term in (10). Similarly, assumption 2 formalizes the choice of \( \phi_0 \in \{0, \pi\} \) rad.

**Assumption 1 (Phase Lock).** All boats are in phase with one another for all time, i.e. there is no phase offset in (10) and \( \omega \) is the same for all boats. Control decisions are made concurrently 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 rad (forward) or \( \pi \) rad (reverse) in each cycle. No other angles are allowed.

We can then propose Theorem 1 in conjunction with the rearward wake compensation factor in Definition 1.

**Theorem 1 (No Unintentional Undocking).** When swimming with waveform given by (10) under assumptions 1 and 2, we can guarantee that no unintentional undocking events will occur within an arbitrary configuration of docked Modboats as long as the maximum wake compensation factor \( \gamma \) is sufficiently small.

**Proof.** Construct a configuration of Modboats as in Fig. 7. The tail protrusion is small enough to guarantee no collisions with diagonal neighbors, so we must consider only collisions with horizontal and vertical neighbors.

For **horizontal neighbors**, construct a lattice of neighboring red and blue regions, as shown in Fig. 7. For any configuration width (along the \( x \) axis), it is trivial to periodically tile. Consider any horizontal row of the configuration at some time \( t_0 \) when \( \cos(\omega t_0) = 1 \), and let \( \phi_0 = 0 \) for all boats. Then \( \phi_i = (\phi_0)_i + A \cos((\phi_0)_i) \forall i \), and

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

Thus the tail segments are all in red region \( (a) \) 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 horizontal collisions are possible.

**Vertical collisions** can be similarly considered. The linear distribution of forces created by the Moore-Penrose pseudo-inverse is guaranteed to generate identical forces \( f_i \) for boats with identical displacements \( x_i \) (see Appendix A). This guarantees that all boats in a vertical column will make the same choice of \( \phi_0 \). Collisions caused by boat \( (i, j) \) selecting \( \phi_0 = \pi \) rad and its forward neighbor \( (i, j + 1) \) selecting \( \phi_0 = 0 \) rad are thus impossible.

Additional consideration must be given to the wake compensation coefficient \( \gamma \). For any boat \( (i, j) \), we have \( \phi(i,j) = A \) and \( \phi(i, j - 1) = \gamma A \) at \( t = t_0 \). If \( \gamma \) is sufficiently large, we could have \( \phi(i,j) \approx \pi \) while \( \phi(i, j - 1) \approx 0 \), leading to a collision, and this occurs for \( \gamma \geq 1.9 \) (see Appendix B). Table III shows the maximum \( \gamma \) in the configurations to be 1.49, so collisions are impossible.

### VI. Experiments

Modboat configurations were experimentally evaluated in a 4.5 m \( \times \) 3.0 m \( \times \) 1.2 m tank of still water, equipped with an OptiTrack motion capture system providing planar position, orientation, and velocity data at 120 Hz. A MATLAB script calculated forces via (5) and used the mapping in Fig. 4 and wake-compensation parameters to determine the required amplitude and centerline for each boat. These parameters were

\[ (a) \quad (b) \quad (c) \quad (d) \quad (e) \]

Table III shows the controller coefficients used in (6), (9).

| \( K_{pv} \) [s\(^{-1}\)] | \( K_{dv} \) | \( K_{d\phi} \) [s\(^{-2}\)] | \( K_{d\Theta} \) [s\(^{-1}\)] |
|-------------------|-----------|----------------|----------------|
| 0.6               | 0.1       | 1.0            | 0.2            |

The 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.
TABLE IV
Controller RMS performance as IQR. Parallel configurations are summarized as Par. Non-parallel configurations are summarized as NP.

|           | V_y RMS [cm/s] | Yaw RMS [rad] |
|-----------|----------------|---------------|
|           | Vel. only      | Vel. & Yaw    | Yaw only       | Vel. & Yaw    |
| 2         | 0.12, 1.29    | 0.25, 0.89    | 0.28, 0.36    | 0.099, 0.21  |
| 3         | 0.14, 0.41    | 0.27, 0.48    | 0.14, 0.17    | 0.058, 0.17  |
| 4         | 0.15, 0.29    | 0.20, 0.32    | 0.10, 0.19    | 0.035, 0.077 |
| 5         | 0.19, 0.34    | 0.20, 0.42    | 0.073, 0.10   | 0.064, 0.13  |
| Par.      | 0.13, 0.38    | 0.23, 0.42    | 0.10, 0.24    | 0.065, 0.15  |
| NP        | 0.25, 0.39    | 0.24, 0.63    | 0.17, 0.23    | 0.062, 0.16  |

TABLE V
Controller yaw rise time (0.3–0.9 of the step) performance from yaw only tests as IQR. Parallel configurations are summarized as Par. Non-parallel configurations are summarized as NP.

|           | \(\pi/2\) rad step [s] | \(\pi\) rad step [s] | Overall [s] |
|-----------|-------------------------|----------------------|--------------|
| 2         | 2.9, 3.8*              | 3.0, 3.4*            | 2.9, 3.6    |
| 3         | 3.7, 3.8*              | 4.4, 4.5*            | 3.7, 4.5    |
| 4         | 4.9, 5.4*              | 6.1, 6.6*            | 5.1, 6.4    |
| 5         | 4.9, 5.1*              | 7.3, 7.7*            | 5.0, 7.5    |
| Par.      | 3.7, 5.0               | 3.9, 6.9             | 3.7, 5.8    |
| NP        | 2.7, 3.3†              | 4.1, 5.1             | 2.9, 4.4†   |

TABLE VI
Controller velocity rise time (0.3–0.9 of the step) performance from velocity only tests as IQR. Parallel configurations are summarized as Par. Non-parallel configurations are summarized as NP.

|           | 4 cm/s step [s] | 6 cm/s step [s] | Overall [s] |
|-----------|----------------|----------------|--------------|
| 2         | 3.8, 12.6*     | 4.1, 4.9*      | 4.0, 8.7    |
| 3         | 3.7, 5.1*      | 4.4, 8.7*      | 4.1, 6.9    |
| 4         | 5.8, 6.5*      | 4.8, 6.1*      | 5.3, 6.3    |
| 5         | 5.2, 7.8*      | 5.6, 6.2*      | 5.4, 7.0    |
| Par.      | 4.4, 7.2      | 4.6, 6.1      | 4.6, 6.4    |
| NP        | 5.8, 7.2      | 7.5, 9.2†     | 6.2, 8.5†   |

Testing was performed on four parallel configurations, from two to five boats in a parallel line (i.e. along the \(\hat{z}_S\) axis), and on five non-parallel configurations, which are shown in Fig. [10]. The results are summarized in Tables [IV][V][VI]. We also considered how the system would behave under mismatched drag coefficients, since it is impossible to experimentally evaluate the drag coefficients for each configuration. To consider the most extreme example of this mismatch, the five boat parallel configuration was also tested with artificially assigned drag coefficients for the two-boat parallel configuration. Representative results for these evaluations are shown in Fig. [13].

VII. DISCUSSION

The controller presented in this work is intended to provide a general formulation for control of surge velocity and yaw for arbitrary configurations of Modboats. Thus, we would expect that — for a successful controller — performance is consistent and independent of the particular configuration. The experimental results presented in Sec. [VI] show that this is the case: arbitrary configurations of Modboats can be driven at a desired surge velocity and to a given yaw angle. With a suitable outer control law, this performance can easily be extended to waypoint tracking and more complex behaviors.

Velocity tracking, presented in Fig. [11] for parallel configurations and summarized in Table [IV] is highly effective regardless of the configuration. All of the evaluated configurations were able to track the desired velocity to within 0.33 cm/s despite varying shapes and interfaces with the surrounding fluid, and non-parallel configurations have statistically identical performance to parallel configurations, indicating that our simple thrust-loss model is sufficient for good performance in velocity tracking.

Yaw tracking, presented in Fig. [12] for parallel configurations and summarized in Table [V] is similarly agnostic to the particular configuration. Yaw tracking is excellent when swimming, as small adjustments to the amplitude of each boat's oscillation yaw the configuration and track the desired heading to within 0.11 rad (6.5°). Yaw tracking suffers, however, when no velocity is desired; although the yaw can be driven to the desired value overall, the tracking error...
Fig. 9. Five boat parallel configuration tracking a yaw of $\pi$ 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 the solid black line is the center of mass.

Fig. 10. Five boat L configuration tracking a yaw of $\pi$ rad (left) for 45 s, then tracking a yaw of 0 rad (right) for 45 s, all while maintaining a velocity of 4 cm/s. Each dashed color is an individual module, and the the solid black line is the center of mass. Note the oscillations during the turn are the result of deformation of the configuration due to fluid forces.

increases significantly. This occurs because the configuration is attempting 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 out by the derivative gain, such as for the two and three boat configurations in Fig. 12 for a $\pi/2$ rad step input and for the four and five boat configurations in Fig. 13a for a $\pi$ rad step input. Otherwise, a significant settling time is observed as the boats attempt to overcome the large inertia of the configuration, such as for the four and five boat configurations in Fig. 12.

A secondary problem with controlling yaw alone is shown in Fig. 14 after achieving the desired yaw with minimal motion of the center of mass (COM), the configuration drifts uncontrollably along its $x$ axis. Since our controller can produce forces only along its surge axis, we have no way of counteracting this behavior. Both issues stem from the same cause: the Modboats’ unique propulsive mechanism cannot smoothly transition from forward to reverse. Eq. (10) is strongly discontinuous whenever $(\phi_0)_i$ 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.

Nevertheless, Table IV shows that the yaw tracking performance is remarkably consistent across configuration shapes, and non-parallel configurations have statistically identical performance to parallel configurations, indicating that our simple thrust-loss model is sufficient for good performance in yaw tracking.

More significant distinctions emerge when considering the rise-time achieved by the controller. Fig. 13a shows that while using the wrong drag coefficient results in comparable tracking performance, it does result in worse velocity rise time. This is reasonable, since drag is the only term in the velocity equation of (5). Similarly, Table V shows that non-parallel configurations have rise-times that are generally worse than those of parallel configurations. Drag coefficients for non-parallel configurations are determined through projection and underestimated (see Sec. II-B and Fig. 2), so it is reasonable that longer rise-times result.

Rise-times for yaw do not show this same pattern, and non-parallel configurations show a small improvement over parallel configuration behavior, as shown in Table V. This is consistent with the behavior observed when the drag is mismatched, as in Fig. 13b, where the performance is not significantly affected. Having no significant shift in performance is reasonable, since in the yaw equation of (5) we observe that $I\alpha \gg C_R|\Omega|\Omega$. The inertia term — which is calculated far more accurately — dominates over the effect of the less well modeled drag.

Fig. 11. Velocity response to a step input of 6 cm/s for configurations of 2-5 boats in parallel. The controller provides comparable performance regardless of the number of boats in the configuration.

Fig. 12. Velocity response to a step input of 6 cm/s for configurations of 2-5 boats in parallel. The controller provides comparable performance regardless of the number of boats in the configuration.

VIII. CONCLUSION

In this work we have presented a centralized control approach that allows arbitrary configurations of underactuated modules to swim as a single unit, as long as they can produce thrust along a single axis. When applied to configurations of Modboats, this strategy minimizes internal forces, guarantees no undocking, and allows the non-rigid

Note that (10) is discontinuous whenever $A_i$ changes even if $(\phi_0)_i$ is constant. We do not observe significant effects from this, however, and it is the normal mode of operation for the control strategy presented in this work.
configuration to behave similarly to a rigid body. Using a minimal model of fluid interactions between neighboring modules and a small set of drag coefficient measurements, our controller is capable of effectively tracking a desired surge velocity and yaw angle for a variety of differently shaped configurations. These results have been verified experimentally for configurations consisting of two to five boats, in nine different configurations.

Velocity tracking is shown to be highly effective, and yaw tracking while swimming forward is similarly accurate. The controller struggles to track yaw while stationary however, managing to get within the desired region but generating oscillations and sideways drift that the control law cannot counteract. Thus this controller is poorly suited for docking or station keeping, both of which require precise orientation control while stationary, and is well suited for transportation of objects or collective travel. Future work will consider ways to extend the control law to the configuration’s sway axis and reduce the observed oscillations in yaw.

A number of approaches for controlling smaller configurations — i.e. single Modboats — already exist [16], [17]. For larger configurations 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. An additional issue to consider is the wake compensation factor and the maximum allowable amplitude causing increased clipping as the configurations grow vertically. Velocity tracking is likely to remain effective, but our testing tank is too small for larger configurations or extensive maneuvering.

Finally, although the current control formulation is sufficient to guarantee no unintentional collisions, it is not necessary. In fact, for $N \geq 3$ boats it should be possible to simultaneously control both translation axes and rotation, rendering any arbitrary configuration into a holonomic vehicle despite the limitations of each module. Finding a control law to accomplish this, however, would require dynamic solutions that avoid the complex collision space within an arbitrary configuration. Future work will investigate such control laws.

ACKNOWLEDGMENT

We thank Dr. M. Ani Hsieh for the use of her instrumented water basin in obtaining all of the testing data, and Peihan M. Li and Julia Dase for their help running experiments for this work.

REFERENCES

[1] J. Paulos, N. Eckenstein, T. Tosun, J. See, 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. O’Hara, J. Paulos, J. Davey, N. Eckenstein, N. Doshi, T. Tosun, J. Greco, J. See, 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.
PROOF OF IDENTICAL FORCES FOR VERTICAL NEIGHBORS

Our proof for avoiding collisions between vertical neighbors in Sec. [V] assumes that vertical neighbors make the same choice of centerline $\phi_0$, since non-adherence to this restrictions results in collisions in many cases. In fact, this is a mathematical consequence of the use of the matrix $P$ and its pseudo-inverse, since boats with identical values of $x_i$ (which includes all vertical neighbors) will have the same force output.

Consider the structural matrix $P$ parameterized by its columns given in (13). Note that, as per (5), each column has the form $p_i = [1 \ x_i \ T^2]$. Hence,

$$ P = \begin{bmatrix} p_1 & \cdots & p_N \end{bmatrix} $$

(13)

In the pseudo-inverse, let $A = (PP^T)^{-1} \in \mathbb{R}^{(2\times2)}$. Then (14) shows the form of the pseudo-inverse $P^+$. We see from (14) that if we have two columns $i$ and $j$ of $P$ that are identical, such that $p_i = p_j$ (i.e. boats in the same vertical column) then the corresponding rows $i$ and $j$ of $P^+$ will also be identical.

$$ f = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} = P^+ \tilde{F} = \begin{bmatrix} p_i^T A \cdot \tilde{F} \\ \vdots \\ p_N^T A \cdot \tilde{F} \end{bmatrix} $$

(15)

Eq. (15) translates from the configuration frame forces $\tilde{F}$ to the individual boat forces $f_i$, and we can see that if $x_i = x_j$, then $p_i = p_j$ and $f_i = f_j$. Since the centerline $\phi_0$ is determined by $\text{sgn} (\phi_0)$, we can begin the proof for the dynamics by analyzing the model for $x_i = x_j$.
is unaffected. The potential for collision provided by the increased amplitude is addressed in Appendix B.

APPENDIX B
PROOF OF NON-COLLISION DESPITE AMPLITUDE INCREASE

The Modboat tail is parameterized in \((r, \theta)\) by (16), where \(r_t\) is the radius of the top body, \(r_p\) is the protrusion of the tail tip, and \(\phi\) is the tail angle (values given in Table VII for the Modboats used in this work). The subscript \((-\pi, \pi]\) indicates wrapping to that range, and \(\theta_w\) is a parameter setting the width of the protruding region. Eq. (16) causes the radius to increase linearly in proportion to angle. This allows two boats to smoothly increase the distance between them when undocking.

\[
r(\theta) = r_t + r_p \left( 1 - \frac{\left| (\phi - \theta)(-\pi, \pi] \right|}{\theta_w} \right)
\]

We can numerically model the entire collision space for two neighboring boats, which results from intersections between two tails modeled by (16) separated by \(2r_t\), and the resulting space is shown in Fig. 15 for a front-back neighbor pair (recall that \(\phi_i = 0\) has the tail tip pointing rearward, and \(\phi_i = \pi\) has the tail tip pointing forward).

We consider Modboats executing waveforms of the form given in (10). Assumption 1 guarantees that \(\omega\) is the same for both boats, and since we consider only vertical neighbors we know that both must make the same choice of centerline \(\phi_0\) as a consequence of the matrix inverse in (5). Consider also that — as dictated by our approach — boats 1 and 2 have their amplitude related as in (12).

| Parameter | \(r_t\) [m] | \(r_p\) [m] | \(\theta_w\) [rad/deg] |
|-----------|-------------|-------------|----------------------|
| Value     | 0.0762      | 0.015       | 0.62/35              |

Then we have:

\[
\frac{d\phi_2}{d\phi_1} = \frac{d\phi_2/dt}{d\phi_1/dt} = \frac{-A_2 \sin (\omega t) \cos \left((\phi_0)_2\right)}{-A_1 \sin (\omega t) \cos \left((\phi_0)_1\right)}
\]

\[
= \frac{A_2}{A_1} = \gamma(2)
\]

Eq. (19) shows that trajectories in the phase space shown in Fig. 15 are lines with slope dictated entirely by \(\gamma\). Moreover, because of Assumption 1 all trajectories must pass through the point \((0,0)\).

Thus we can consider the effect of \(\gamma\) by drawing a horizontal line passing through the origin and slowly increasing the slope until a collision occurs at \(\gamma_{max} \approx 1.9\). Since all \(\gamma\) values provided in Table I are below \(\gamma_{max}\), no collisions will occur.

---

Fig. 15. Numerical model of collision space between horizontal neighbors with boat 2 behind boat 1. The green region represents no collision, while the red diamond marks the collision space.