Flow-Based Rendezvous and Docking for Marine Modular Robots in Gyre-Like Environments

Gedaliah Knizhnik, Peihan Li, Mark Yim, and M. Ani Hsieh

Abstract— **Modular self-assembling systems typically assume that modules are present to assemble. But in sparsely observed ocean environments, modules of an aquatic modular robotic system may be separated by distances they do not have the energy to cross, and the information needed for optimal path planning is often unavailable. In this work we present a flow-based rendezvous and docking controller that allows aquatic robots in gyre-like environments to rendezvous with and dock to a target by leveraging environmental forces. This approach does not require complete knowledge of the flow, but suffices with imperfect knowledge of the flow's center and shape. We validate the performance of this control approach in both simulations and experiments relative to naive rendezvous and docking strategies, and show that energy efficiency improves as the scale of the gyre increases.**

I. Introduction

Modular, self-reconfigurable robotic systems (MSRRs) that can function in aquatic environments have the potential to facilitate detailed oceanographic studies, provide repair and maintenance to ocean infrastructure, and create landing platforms or bridges for drones or larger vehicles. But operating in an ocean environment comes with challenges for distributed robotic system: the environments under study may be several orders of magnitude larger than the individual robotic modules and just the act of spreading out or converging to a single location may be difficult.

Prior work has already considered the significant challenges for modular systems of determining the proper shape for a task [1]–[5], finding a feasible assembly order that does not violate any motion or docking constraints [6]–[10], and actually executing this assembly order in an efficient manner [10]–[13]. These works, however, assume that modules are *present* and available to assemble. In an oceanographic setting robots may be sparsely distributed over tens or hundreds of kilometers such that it is not trivial to command them to return or converge. For resource constrained robots with limited actuation and power-budgets, motion control and path planning strategies that leverage ocean currents can significantly prolonged the operational lifespan of these platforms.

One approach taken in the literature is to generate *time and energy optimal paths* for autonomous marine vehicles (AMVs) that leverage environmental flows [14]–[23]. But these approaches require full knowledge of the flow field, which is particularly difficult to obtain. And while existing strategies can account for ocean current forecast uncertainties in planning [24]–[28], they still need complete knowledge of the flow. Alternatively, the path planning problem for converging aquatic modules can be considered as *rendezvous* in a flow, but existing work [29]–[31] assumes complete knowledge of the flow field and results have mostly been theoretical.

The contribution of this work is the development of **flow-based rendezvous and docking strategies** for resource-constrained robotic modules that (unlike in prior work) do not have full knowledge of the flow dynamics in **gyre-like flows**. These strategies can be used to fuse persistent monitoring of dynamic environments like the ocean by sparse robot teams [32] with the flexibility of MSRRs to tackle unpredictable tasks, and allow **self-assembly over large distances**. These strategies are developed using the Modboat platform [33] (shown in the inset of Fig. 1), an underactuated aquatic surface MSRR developed by the authors. Modboats can dock in a rectangular lattice through passive magnetic docks [34], can move as a single unit in arbitrary lattice configurations [35], [36] despite each having only one motor, and are an ideal resource-constrained test platform.

The rest of the paper is organized as follows: gyre-like environments are defined and the problem statement is formalized in Sec. II, Our flow-based rendezvous and docking (FBRD) controller is presented in Sec. III, and our experimental setup and results are detailed in Sec. IV. Experimental and simulation-based validation is discussed in Sec. V. Conclusions and directions for future work are presented in Sec. VI.

II. Problem Formulation

A. Gyre-Like Flow Fields

A 2D gyre-like flow-field field $\mathbb{W} \subset \mathbb{R}^2$ has the form given by (1), where $\vec{x} \in \mathbb{W}$. The dissipative dynamics of the flow are given by $V_d : \mathbb{W} \mapsto \mathbb{R}^2$, and the non-dissipative dynamics by $V_f : \mathbb{W} \mapsto \mathbb{R}^2$. The solutions or trajectories of (1) form roughly concentric orbits.

$$\dot{\vec{x}} = V_f(\vec{x}) + V_d(\vec{x}), \tag{1}$$

A more complete and formal list of properties of gyre-like flow fields is provided in our prior work [32], but we summarize the most critical properties of $V_f(\vec{x})$ here:

1) **Star-shaped:** any line outward from the center of the gyre intersects a trajectory of $V_f(\vec{x})$ only once.
2) **Monotonically changing angular velocities:** There exist regions in the flow where the angular velocity changes monotonically with distance from the center.

This work was supported by the National Science Foundation Award IIS-1812319.

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

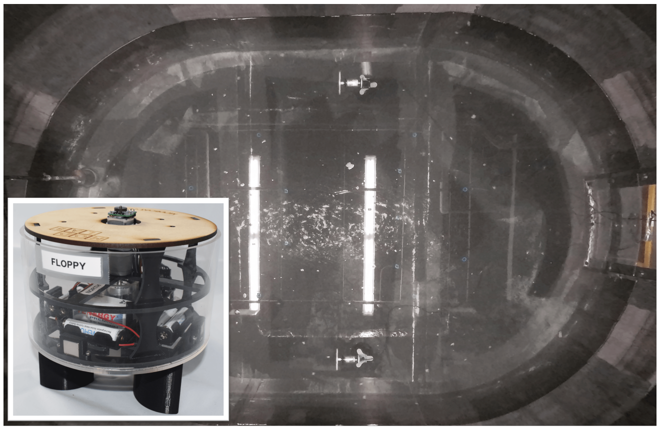


Fig. 1. An overhead view of the racetrack shaped experimental tank. The two propellers are visible at the top and bottom of the image, and a glass observation window is present on the right-hand side. Inset: a photo of one of the Modboat modules used in this work. The Modboat is described in [33], and the controller used in this work is developed in [32], [37].

These properties will enable us to synthesize motion control strategies for robots in gyre-like flows that rely on minimal and imperfect information about the gyre shape and center.

B. Problem Statement

Consider a system of aquatic modules comprising a modular self-reconfigurable robotic system (MSRR) in a gyre-like flow. Some of the modules form a lattice, while others are scattered throughout the environment performing various tasks. Individual modules may be tasked to rendezvous with the lattice and dock to a particular location within it, to faciliate reconfiguration or to recharge and offload data. In a sparsely populated ocean environment, the modules may be kilometers apart, and may not have the resources to make a journey to the lattice for docking.

Our prior work [32] showed that imperfect knowledge could be used without a full flow model to generate energy-efficient orbits. In this work, we argue that similar techniques can be used for energy efficient rendezvous. As in our prior work [32], assume that we have a robot that can act as a pointable thruster on a time-scale much less than the period of the flow[1], and **imperfect knowledge of**:

- The shape of the gyre orbits.
- The gyre center.
- The robot's location relative to the gyre center.

Problem: Given a single mobile module and a less mobile lattice of docked modules, design a controller capable of

1) energy-efficient rendezvous with minimal information about the flow, and
2) precise docking over short distances to specific locations in the lattice.

III. Methodology

The task of docking in a gyre-like flow as considered in this work and described in Sec. II is split into three stages. Let the characteristic length of a Modboat lattice be given by d_{lat}, and the distance between the lattice and an incoming module be given by d_{sep}. Then **flow-based rendezvous and docking (FBRD)** is split into the following stages:

1) **Rendezvous:** driving d_{sep} from $d_{sep} \geq d_{lat}$ to $d_{sep} \sim d_{lat}$ — i.e. bringing the incoming module to the lattice from some distance away.
2) **Following:** maintaining $d_{sep} \sim d_{lat}$ while waiting for an available dock — i.e. waiting close enough to the lattice to activate docking when ready, while not being accidentally captured in the wrong position.
3) **Docking:** bringing $d_{sep} \to 0$ — i.e. actually docking to the desired opening in the lattice.

Stage 1 is particularly important for energy efficiency since ocean gyres can range from tens of meters to hundreds of kilometers in size, and the task of rendezvous over such large distances is likely to be very energy intensive. Reducing the cost by leveraging the surrounding currents at this stage would provide significant benefits to modular ocean systems.

A. Rendezvous

The flow-based control strategy in our prior work [32] showed over 70% energy reduction when orbiting a gyre by using the flow for orbital motion and thrusting only radially to adjust the orbital period. We propose to enable energy-efficient **rendezvous** by using this strategy to reduce the difference in phase between the mobile individual module and the less mobile lattice it is docking to.

As shown in prior work [32], the flow field can be mapped to a circular orbital model as in (2), where $r \in \mathbb{R}_+$, $\Theta \in [0, 2\pi)$, using a linear map $g : (x_1, x_2)^T \in \mathbb{W} \mapsto (r, \Theta)^T \in \mathbb{C}$, such that $\Theta = \arctan\left(\frac{x_2 - \vec{x}_z(2)}{x_1 - \vec{x}_z(1)}\right)$ (the angle from the gyre center to the robot).

$$\mathbb{C} : \begin{bmatrix} \dot{r} \\ \dot{\Theta} \end{bmatrix} = \begin{bmatrix} -\nu(r) \\ \Omega(r, \Theta) \end{bmatrix}, \tag{2}$$

Letting the subscript lat indicate the lattice and mod indicate the incoming module, we can define the angular error and rendezvous controller as in (3) and (4), where $K_{ccw} = 1 \quad (-1)$ if the flow is counterclockwise (clockwise) and $K_{out} = 1(-1)$ if faster flow is further out from (closer in to) the center. We choose some small phase difference ϵ_Θ as the desired separation to await the docking maneuver, and we choose $\pm\epsilon_\Theta$ such as to minimize $|e_\Theta|$.

$$e_\Theta = K_{ccw} \left(\Theta_{lat} - \Theta_{mod} \pm \epsilon_\Theta\right) \tag{3}$$

$$r_{des} = r_{lat} + K_p K_{out} e_\Theta \tag{4}$$

Eq. (4) has the effect of driving the desired radius to r_{lat} at the same time as the phase difference $e_\Theta = 0$, thus bringing the module to some small distance from the lattice. In (3), K_{ccw} adjusts the sign of the error such that positive (negative) error indicates phase lag (lead) regardless of the direction of flow, while in (4) K_{out} ensures that phase lag is compensated for by moving to faster water, while lead causes movement to slower water.

[1] On realistic ocean timescales, effectively all robots satisfy this property, and the Modboat specifically is shown to act this way in [37].

The Modboat desaturated thrust direction (DTD) controller [37] is then given a heading as per (5) to track the desired radius $r_{des} \pm \epsilon_r$ for some small value of ϵ_r. Eq. (5) thrusts either outward or inward (at maximum thrust u_{max}) from the approximate gyre center to adjust the orbit of the incoming module.

$$(u, \theta_r) = \begin{cases} (u_{max}, \quad \Theta_{mod}) & r_{mod} \leq r_{des} - \epsilon_r \\ (0, \Theta_{mod}) & \|r_{mod} - r_{des}\| < \epsilon_r \\ (u_{max}, -\Theta_{mod}) & r_{mod} \geq r_{des} + \epsilon_r \end{cases} \tag{5}$$

B. Following

Once the **rendezvous** controller in Sec. III-A has brought $d_{sep} \sim d_{lat}$, we can now consider the problem of docking. Since rendezvous uses only the *location* of the lattice, it is possible that the module arrives when the lattice is oriented unfavorably for the desired docking position. If the lattice is itself large, the module is low on energy, or the flow is particularly strong, naively swimming to the target dock may be unfeasible.

Thus we consider the energy efficient methodology of **following** the lattice — i.e. maintaining some desired separation distance $d_{des} \sim d_{lat}$ — while waiting for a favorable docking position. If we define Θ_{sep} as the heading from the lattice to the module, then we can define the following controller as in (6).

$$(u, \theta_r) = \begin{cases} (u_{max}, \quad \Theta_{sep}) & d_{sep} \leq d_{des} - \epsilon_{sep} \\ (0, \Theta_{sep}) & \|d_{sep} - d_{des}\| < \epsilon_{sep} \\ (u_{max}, \pi - \Theta_{sep}) & d_{sep} \geq d_{des} + \epsilon_{sep} \end{cases} \tag{6}$$

Eq. (6) has the effect of thrusting away from the lattice when the module comes too close, and towards it when the module is too far, with ϵ_{sep} defining a safe ring within which the module can drift freely. As long as the flow is not divergent in the region occupied by the lattice and the module, we can expect that both will maintain a relatively constant separation in both radius and phase under this control law with minimal energy input.

Once the desired docking position is in view (via angle and distance criteria), the individual module can thrust towards it using the *homing* strategy[2] described in our prior work [34].

C. Lattice Control

Transition from **following** stage to the **docking** stage in Sec. III-B requires that the desired dock location be in view, but the **following** controller does not actively pursue moving towards the desired dock location. Instead, the lattice is tasked with rotating to bring the desired location into view.

Prior work [35], [36] has demonstrated that a lattice of docked Modboats, such as the one shown in Fig. 2, can be controlled as a single unit without unintentional undocking maneuvers. This is done by using (7) to map a surge force f_{lat} and yaw torque τ_{lat} applied to the lattice to individual boat forces $\vec{f}$. The structural matrix P in (8) encodes the lattice structure (x_i represents the lateral distance of module i from the center of mass), and P^+ represents the Moore-Penrose pseudo-inverse of P [35], [36]. Each module then achieves this force by modulating its oscillation amplitude to set the magnitude and choosing a centerline of either 0 or π rad to control force direction.

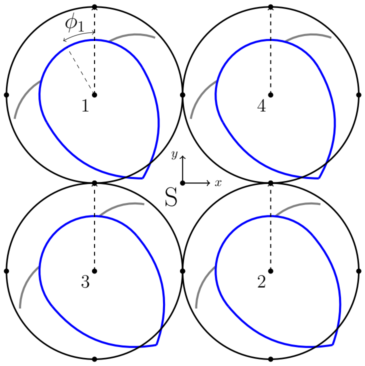


Fig. 2. Configuration of four docked Modboats in a square lattice; for each boat, the top body is shown in black, while the bottom body/tail is in blue and flippers are in gray. ϕ represents the angle of the tail relative to the front of the module, and intentionally colliding the protruding tails causes modules to undock [34].

$$\vec{f} = P^+ \begin{bmatrix} f_{lat} & \tau_{lat} \end{bmatrix}^T \tag{7}$$

$$P = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 \\ x_1 & x_2 & \dots & x_{N-1} & x_N \end{bmatrix} \tag{8}$$

Notably, however, when used only for yaw control this strategy causes uncontrollable sideways drift. This occurs because precise yaw control requires frequent shifts from positive to negative thrust, and the resulting large centerline transitions impart sideways forces to the structure [35], [36]. Since the proposed docking method requires precise alignment, such sideways drift would be highly disruptive, and pure yaw or yaw-rate control would be desirable.

To alleviate this, we modify the strategy for group yaw control presented in our prior work [35], [36], by assigning each module to either a CW or CCW group. Each module can then maintain its centerline for as long as control is active, and changing between positive and negative torque is done by changing which group is thrusting, instead of by changing centerlines. As long as each group has at least two modules, pure torques can be generated, and no sequence of adding additional modules can leave either group with fewer than two members.

To use this control approach, we start with the structural matrix P as defined in [35], [36], and selectively zero columns to create P_{ccw} and P_{cw} (with (9) as an example for the lattice in Fig. 2). The appropriate matrix is then chosen based on the desired torque τ_{lat}, as in (10).

$$P_{ccw} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ x_1 & x_2 & 0 & 0 \end{bmatrix}, \quad P_{cw} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & x_3 & x_4 \end{bmatrix} \tag{9}$$

$$P = \begin{cases} P_{ccw} & \tau_{lat} > 0 \\ P_{cw} & \tau_{lat} < 0 \end{cases} \tag{10}$$

[2] Note that the strategy in [34] begins with *distancing* — i.e. achieving distance along the perpendicular approach vector. This phase is skipped here because this distance is already enforced by the **following** stage.

TABLE I

Coefficient values used in (3), (4), and (11) for the simulation results presented in Tab. II and Fig. 3.

Γ [m^2/s]	a [m]	K_{ccw}	K_{out}	K_p [m/rad]	v_{max} [m/s]
0.0565	0.05	1	-1	0.75	0.08

TABLE II

Simulated rendezvous success rate by injected Gaussian noise standard deviation σ applied to the flow-field.

σ [m/s]	0.000	0.001	0.010	0.100	0.200	0.500
# Successes	49	49	49	49	48	29
# Tests	50	50	50	50	50	50

Using this strategy of assigning forces to individual modules, a simple PID loop (see our prior work [34]), can control either the yaw or yaw-rate of the lattice without causing sideways drift, as will be shown in Sec. IV.

IV. Results

Experiments were conducted in a $4.5\ \text{m} \times 3.0\ \text{m} \times 1.2\ \text{m}$ racetrack shaped water tank, shown in Fig. 1, equipped with an OptiTrack motion capture system providing planar position, orientation, and velocity data at up to 120 Hz. A single gyre was created by placing two horizontally mounted propellers, spinning at $200\ \text{rpm}$, along the straight edges of the tank as seen in Fig. 1. In this configuration, the flow is shaped by the walls of the tank and forms a clockwise gyre with faster flow towards the outside. The maximum observed fluid velocity at the outer boundary was $0.18\ \text{m/s}$.

A. Simulation

Simulation was used to evaluate the robustness and repeatability of the flow-based **rendezvous** controller when not constrained by the boundaries of our experimental facility, as well as to assess its performance at more realistic gyre-flow scales. In simulation the rendezvous controller (4) is implemented to calculate the desired orbital radius, and then the individual module and the lattice are treated as point particles with velocity inputs and values shown in Table I. Convergence occurs when the module and lattice are $7.5\ \text{cm}$ apart. We considered a Rankine vortex model as given in (11), and the maximum velocity at which the individual module can move $v_{max} = 8.0\ \text{cm/s}$ is set to approximately match the Modboat max velocity under DTD control of $9.3\ \text{cm/s}$ [32].

$$v_r = 0, \quad v_\theta(r) = \frac{\Gamma}{2\pi} \begin{cases} r/a^2 & r \leq a \\ 1/r & r > a \end{cases} \tag{11}$$

To evaluate robustness and repeatability, simulations were run with randomly distributed lattices and modules with varying levels of zero-mean Gaussian noise to simulate uncertainty in the flow model. Γ, as given in Tab. I, was set to give a maximum velocity of $0.18\ \text{m/s}$ (at the interior boundary), which matches the maximum observed experimental velocity (at the outer boundary). Success rates for robustness to noise are shown in Tab. II, showing high robustness until the noise level significantly exceeds thrust capacity.

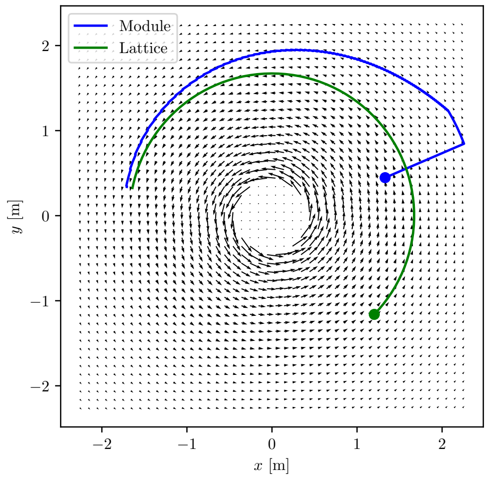


Fig. 3. Sample results of a rendezvous simulation in a Rankine vortex under FBRD. The module is marked in blue, and the lattice in green, with both start locations indicated by a circle marker. The module starts ahead of the lattice, and so moves quickly to slower water to await the rendezvous.

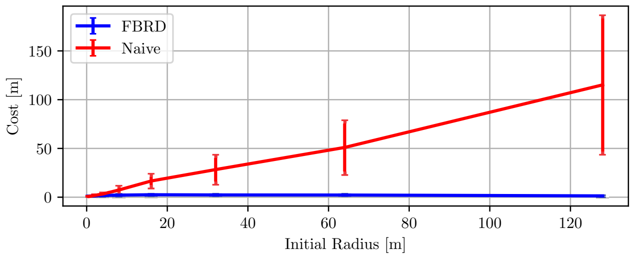


Fig. 4. Cost of FBRD and naive approach as the dimension of the Rankine vortex grows. FBRD maintains constant cost, while the naive approach cost grows linearly with gyre dimension. Cost is calculated as the distance traveled as a result of the controller input, without the gyre.

To evaluate the performance of FBRD at more realistic gyre scales and compare it to the *naive approach* of constantly swimming towards the lattice target, simulations were run at varying scales. For varying values of an initial radius R, the value of Γ was adjusted so that $v_\theta(R) = 0.18\ \text{m/s}$, and the module and lattice were spawned at random locations within the ring defined by $r \in [R-0.2, R+0.2]\ \text{m}$. Cost was measured as distance traveled due to the controller, and cost scaling is shown in Fig. 4, showing that FBRD becomes significantly more efficient than the naive approach as the scale become more realistic.

B. Experiments

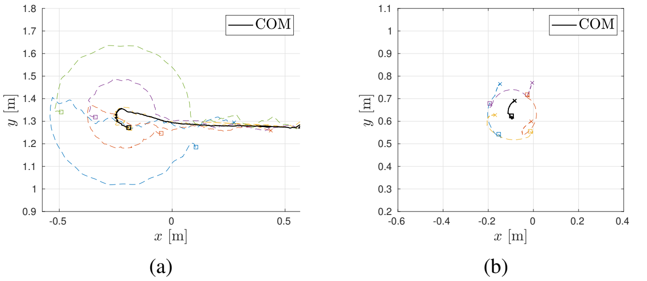
We experimentally evaluated the ability of the lattice controller described in Sec. III-C to control yaw alone without causing sideways drift by commanding step changes in orientation *without active flow*. As shown in Tab. III and Fig. 5, modified lattice control significantly reduced lateral drift observed in prior work [35], [36] while preserving controllability. The modified controller was then evaluated for its ability to control *yaw-rate* in the presence of a flow. The lattice was commanded to track varying rotational velocities

TABLE III

MODIFIED LATTICE YAW CONTROLLER PERFORMANCE, AS $\mu \pm \sigma$.

	Original [35], [36]	Modified[†]
Lateral drift w/o flow [cm/s]	2.1 ± 0.37	0.18 ± 0.060

[†] Statically significant difference over the other strategy.



(a)

(b)

Fig. 5. Samples of pure yaw correction for (a) a five boat line under prior collective control [35], [36] and (b) a four boat square under the modified control presented in Sec. III-C, with individual boats shown as dotted lines. Starts (ends) indicated with squares (Xs). Prior control in (a) results in large lateral drift, but modified control in (b) results in minimal lateral drift.

TABLE IV

MODIFIED LATTICE YAW-RATE CONTROLLER PERFORMANCE, AS $\mu \pm \sigma$.

ω_{des} [rad/s]	-0.1	-0.05	0.0
$100\omega_{obs}$ [rad/s]	-7.8 ± 1.3	-6.8 ± 1.7	-3.8 ± 1.1
RMS Err [rad/s]	0.022	0.018	0.038

TABLE V

COEFFICIENT VALUES USED IN (3), (4), AND (5) FOR THE EXPERIMENTAL RESULTS PRESENTED IN TAB. VI.

K_{ccw}	K_{out}	K_p [m/rad]	ϵ_Θ [rad]	ϵ_r [m]
-1	1	0.5	$(0.7\text{ m})/r^*_{lat}$	0.05

* The angle is adjusted to track 0.7 m from the lattice at any radius.

(about its own center of mass), and the performance was tracked. Results are given in Tab. IV.

Evaluation of the overall flow-based rendezvous and docking (FBRD) controller was performed by testing the end-to-end system, with coefficients given in Tab. V. We considered a square lattice composed of four Modboats (see Fig. 2), with a single Modboat serving as the mobile module. A dock location was manually selected on the square lattice, and both the module and the lattice were released from random positions in the flow and allowed to attempt to dock until a dock occurred or the test timed out at 4 minutes. The individual Modboat module was able to rendezvous with, follow, and dock to the lattice in all cases that did not encounter the tank boundaries, resulting in thirty successful tests. A sample rendezvous and docking trajectory is shown in Fig. 6a, which demonstrates the general behavior of the controller: swim out to take advantage of faster water, wait while gaining on the lattice, and then swim back to transition to following and docking.

We also evaluated docking in a gyre flow using the the naive docking approach presented in our prior work [34] integrated with DTD control [37] as a control case. Despite not being designed with flow in mind, the naive approach successfully rendezvoused and docked in fourteen out of fifteen cases[3], and a sample trajectory is shown in Fig. 6b, which demonstrates the general behavior of this approach: follow a curved trajectory defined by the movement of the mobile module towards a drifting target before transitioning to docking. Active lattice control was not used for these tests.

Since the goal of FBRD is energy-efficiency when rendezvousing over large distances, we compared both overall energy cost and the energy cost per distance traveled for our flow-based strategy and the naive approach strategy, and the results are presented in Tab. VI. To ensure a fair comparison, six iterations were run with an initial (enforced) phase separation of $\pi/2$ rad, and four iterations with an initial (enforced) phase separation of π rad. Since (5) is either fully on or off, we use *time the controller is on (time on)* as a proxy for energy consumed by the controller.

A primary issue encountered in our experiments, however, was the impact of testing facility dimensions. The experimental tank is 3.0m wide, but the Modboats are 0.14m in diameter, so there are ≈ 10 body lengths between the center of the gyre and its edge on the shortest side, or ≈ 5 body-lengths for the square lattice of four Modboats. This means that the Modboats frequently interact with the control limits, either in the form of the fastest flow at the outer wall, or the slowest flow on an interior virtual boundary that prevents interaction with the gyre center. These issues were not encountered in simulation, where the boundaries were not present as described in Sec. IV-A.

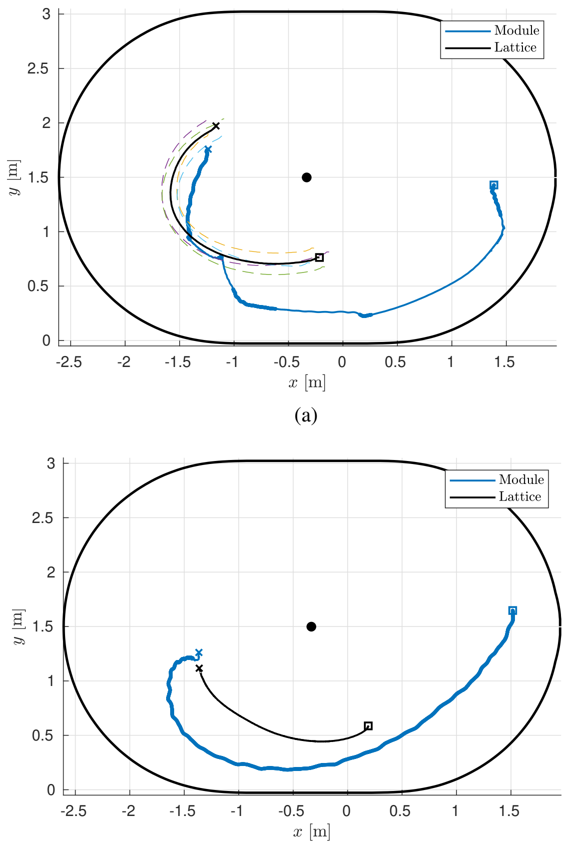
V. DISCUSSION

The **flow-based rendezvous and docking (FBRD) controller)** in this work was motivated by the need for improved energy efficiency when traveling over large distances in gyre-like flows. Tab. VI shows that our flow-based approach provides significant efficiency improvement when compared to naively swimming towards the target, despite limited knowledge of the flow. This allows resource constrained modules to travel far on a limited energy supply and rendezvous with other modules or lattices in an efficient way.

The top half of Tab. VI compares the total energy cost of the two approaches and highlights comparable cost for FBRD and the naive approach at experimental scales, but our flow-based approach is expected to be far more economical in a larger gyre. For a gyre with characteristic length L, the naive approach has linear cost $C_{naive} \sim L$, since — in the worst case — the module must cross the entire gyre. FBRD, on the other hand, has constant cost[4] $C_{flow} \sim 2K_p$. It is therefore reasonable to expect that our controller may be less economical at small scales, but far more economical

[3] We note that the last case also successfully rendezvoused and docked, but to the incorrect target location.

[4] In general, the motion of the flow-based rendezvous consists of changing its radius by $K * e_\Theta(0)$, which is constant in L, and then returning to the original radius, which is also constant in L.



(a)

(b)

Fig. 6. Sample trajectory for (a) FBRD and (b) naive approach, with the tank boundary and gyre center indicated. Start (end) positions are marked with squares (Xs). Individual lattice modules are shown as dotted lines in (a). Thicker sections of module plot indicate the regions in which control is on, with the continuous control on region on the left side corresponding to docking mode.

at realistic ocean scales[5]. Fig. 4 shows this growth in cost (measured as the distance traveled as a result of the controller) as the gyre grows to 128 m in radius, and the pattern can be expected to continue to more realistic scales.

Simulations also demonstrate that our rendezvous approach is robust to noise in the flow model, as shown in Tab. II. The module and lattice successfully converged in 98% of cases even in the presence of noise, with the failure attributable to timeout, which indicates that our controller is likely to succeed even in the presence of ocean current variations. When the noise significantly exceeds the max robot velocity, however, the success rate drops significantly. While this likely indicates that more work is needed to rendezvous in stormy or particularly violent water, part of the failures can be attributed to the final following and docking stages, which were not implemented in simulation. Thus in many cases the module made it to the lattice, but could not converge completely under the *rendezvous* law alone.

Part of the economy of FBRD comes from cooperation with the lattice, which adjusts its orientation to present an available docking position. The lattice has a natural rotational velocity when in a gyre, which depends on its shape and the particular flow. The lattice's ability to adjust its yaw-rate (shown in Tab. IV) is poor in terms of accuracy, but — critically — it can still shift its mean rotational velocity either down to allow successful docking or up to bring docking positions into view more quickly. Since the primary goal — adjusting yaw/yaw-rate without drifting sideways (see Fig. 5) — is achieved, rendezvous and docking can still occur despite poor tracking performance.

TABLE VI

ENERGY COST AND EFFICIENCY (ENERGY PER DISTANCE TRAVELED) FOR BOTH METHODS BY INITIAL PHASE SEPARATION, AS $\mu \pm \sigma$.

	$e_\Theta(0)$	$\pi/2$ [rad]	π [rad]	Overall
Energy Cost [s]	**Flow**	$19 \pm 3.4^\dagger$	53 ± 14	—
	Naive	27 ± 7.4	42 ± 16	—
Efficiency [s/m]	**Flow**	$3.3 \pm 0.47^\dagger$	$3.3 \pm 1.1^\dagger$	$3.3 \pm 0.67^\dagger$
	Naive	6.7 ± 1.6	9.6 ± 2.6	7.7 ± 2.3

† Statically significant difference over the other strategy.

VI. Conclusion

In this work, we have presented a **flow-based rendezvous and docking controller** that allows individual aquatic modules in a modular self-reconfigurable robotic system (MSRRs) to cover large distances to converge on a lattice they are assembling, despite resource contraints and limited knowledge of the gyre-like environments they inhabit. This is achieved by using imperfect knowledge of the gyre center and the module's location relative to it to thrust radially and adjust the angular velocity with which the gyre flow carries the module. Once the distance between the module and the lattice is decreased to a manageable magnitude, our method uses minimal thrust to maintain separation from the lattice, while the lattice rotates to present a valid docking position that the module can then dock to.

We have shown experimentally that this method is successful and can be repeatably executed as an end-to-end system. While equivalent to a naive docking approaches at small scales, our controller provides increased efficiency of travel and simulations show that it becomes more economical at realistic ocean gyre scales, where both time-scales and distances are large. We have also shown in simulation that the rendezvous controller is robust to noise in the flow-field and converges effectively and repeatably, especially when boundary effects that limit experimentation are removed.

In future work, we hope to experimentally evaluate the performance of our system in realistic gyre conditions at a much larger scale, which will avoid boundary conditions and validate the cost predictions of our simulations. More complex gyre shapes and arrangements will also be considered, such as the wind-driven double gyre model, as well as the challenge of learning the needed gyre information from local measurements. Future work will also consider the effect of decentralized coordination on the system, in which the lattice and module communicate with each other directly to achieve

[5] Our tank has $L = 3.0$m, while our experimental $K_p = 0.5$. Thus $C_{naive} \sim 3$ and worst case $C_{flow} \sim 2\pi K_p \approx 3.14$.

docking, rather than being guided by a centralized system.

References

[1] G. Jing, M. Yim, H. Kress-Gazit, and T. Tosun, "An End-To-End System for Accomplishing Tasks with Modular Robots," *Robotics: Science and Systems*, vol. 2, no. 7, 2016.

[2] G. Jing, T. Tosun, M. Yim, and H. Kress-Gazit, "Accomplishing high-level tasks with modular robots," *Autonomous Robots*, vol. 42, no. 7, pp. 1337–1354, 10 2018.

[3] A. M. Mehta, J. DelPreto, K. W. Wong, S. Hamill, H. Kress-Gazit, and D. Rus, "Robot Creation from Functional Specifications," in *Springer Proceedings in Advanced Robotics*. Springer Science and Business Media B.V., 2018, vol. 3, pp. 631–648.

[4] T. Tosun, G. Jing, H. Kress-Gazit, and M. Yim, "Computer-Aided Compositional Design and Verification for Modular Robots," in *Springer Proceedings in Advanced Robotics*. Springer Science and Business Media B.V., 2018, vol. 2, pp. 237–252.

[5] T. Tosun, "Addressing Tasks Through Robot Adaptation," Ph.D. dissertation, University of Pennsylvania, 2018.

[6] 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.

[7] 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.

[8] D. Saldana, B. Gabrich, M. Whitzer, A. Prorok, M. F. Campos, M. Yim, and V. Kumar, "A decentralized algorithm for assembling structures with modular robots," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, 12 2017, pp. 2736–2743.

[9] A. Costa, A. Abdel-Rahman, B. Jenett, N. Gershenfeld, I. Kostitsyna, and K. Cheung, "Algorithmic Approaches to Reconfigurable Assembly Systems," in *IEEE Aerospace Conference Proceedings*, vol. 2019-March. IEEE Computer Society, 3 2019.

[10] C. Liu, Q. Lin, H. Kim, and M. H. Yim, "SMORES-EP, a Modular Robot with Parallel Self-assembly," in *arXiv: 2104.00800*, 2021.

[11] D. Saldana, B. Gabrich, G. Li, M. Yim, and V. Kumar, "ModQuad: The flying modular structure that self-assembles in Midair," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 2018, pp. 691–698.

[12] 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.

[13] 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.

[14] B. Garau, A. Alvarez, and G. Oliver, "Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an a* approach," in *2005 IEEE International Conference on Robotics and Automation (ICRA)*, April 2005, pp. 194–198.

[15] D. Kruger, R. Stolkin, A. Blum, and J. Briganti, "Optimal auv path planning for extended missions in complex, fast-flowing estuarine environments," in *Robotics and Automation, 2007 IEEE International Conference on*, April 2007, pp. 4265–4270.

[16] J. Witt and M. Dunbabin, "Go with the flow: Optimal auv path planning in coastal environments," in *Australasian Conference on Robotics and Automation (ACRA)*, 2008.

[17] T.-B. Koay and M. Chitre, "Energy-efficient path planning for fully propelled auvs in congested coastal waters," in *OCEANS - Bergen, 2013 MTS/IEEE*, 2013, pp. 1–9.

[18] D. Rao and S. B. Williams, "Large-scale path planning for underwater gliders in ocean currents," in *Australasian Conference on Robotics and Automation (ACRA)*, Sydney, 2009.

[19] A. Chakrabarty and J. Langelaan, "Uav flight path planning in time varying complex wind-fields," in *2013 American Control Conference*, June 2013, pp. 2568–2574.

[20] M. Eichhorn, "Optimal routing strategies for autonomous underwater vehicles in time-varying environment," *Robotics and Autonomous Systems*, vol. 67, pp. 33 – 43, 2015.

[21] M. Otte, W. Silva, and E. Frew, "Any-time path-planning: Time-varying wind field + moving obstacles," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2575–2582.

[22] D. N. Subramani and P. F. Lermusiaux, "Energy-optimal path planning by stochastic dynamically orthogonal level-set optimization," *Ocean Modelling*, vol. 100, pp. 57 – 77, 2016.

[23] D. Kularatne, S. Bhattacharya, and M. A. Hsieh, "Optimal path planning in time-varying flows using adaptive discretization," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 458–465, 2018.

[24] M. T. Wolf, L. Blackmore, Y. Kuwata, N. Fathpour, A. Elfes, and C. Newman, "Probabilistic motion planning of balloons in strong, uncertain wind fields," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 1123–1129.

[25] A. A. Pereira, J. Binney, G. A. Hollinger, and G. S. Sukhatme, "Risk-aware path planning for autonomous underwater vehicles using predictive ocean models," *Journal of Field Robotics*, vol. 30, no. 5, pp. 741–762, 2013.

[26] W. H. Al-Sabban, L. F. Gonzalez, and R. N. Smith, "Wind-energy based path planning for unmanned aerial vehicles using markov decision processes," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 784–789.

[27] V. Huynh, M. Dunbabin, and R. Smith, "Predictive motion planning for auvs subject to strong time-varying currents and forecasting uncertainties," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1144–1151.

[28] D. Kularatne, H. Hajieghrary, and M. A. Hsieh, "Optimal Path Planning in Time-Varying Flows with Forecasting Uncertainties," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, 9 2018, pp. 4857–4864.

[29] C. Wei, H. G. Tanner, X. Yu, and M. A. Hsieh, "Low-range interaction periodic rendezvous along lagrangian coherent structures," *Proceedings of the American Control Conference*, vol. 2019-July, pp. 4012–4017, 7 2019.

[30] C. Wei, X. Yu, H. G. Tanner, and M. A. Hsieh, "Synchronous Rendezvous for Networks of Active Drifters in Gyre Flows," *Springer Proceedings in Advanced Robotics*, vol. 9, pp. 413–425, 2019.

[31] X. Yu, M. A. Hsieh, C. Wei, and H. G. Tanner, "Synchronous Rendezvous for Networks of Marine Robots in Large Scale Ocean Monitoring," *Frontiers in Robotics and AI*, vol. 6, p. 76, 9 2019.

[32] G. Knizhnik, P. Li, X. Yu, and M. A. Hsieh, "Flow-Based Control of Marine Robots in Gyre-Like Environments," in *2022 International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, 5 2022, pp. 3047–3053.

[33] G. Knizhnik, "Modboat: A single-motor modular self-reconfigurable robot," Feb 2022, https://www.modlabupenn.org/modboats/.

[34] 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.

[35] G. Knizhnik and M. Yim, "Amplitude Control for Parallel Lattices of Docked Modboats," in *2022 International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, 5 2022, pp. 3027–3033.

[36] G. Knizhnik and M. Yim, "Collective Control for Arbitrary Configurations of Docked Modboats," *arXiv:2209.04000 [cs.RO]*, 9 2022.

[37] 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.