Multi-Robot Visual Control of Autonomous Soft Robotic Fish

Juan Salazar[1*], Levi Cai[2], Braden Cook[1], and Daniela Rus[1]

***Abstract*—The coordination and control of autonomous underwater vehicle (AUV) fleets in ocean exploration is a widely researched topic with much groundwork for traditional AUVs. Depending on the mission, AUV fleets can relax mission constraints on individual vehicles and improve a number of performance objectives (e.g. duration, sampling rate, area coverage). As missions begin to require navigation within more confined areas such as caves and coral reefs, however, safe interaction with such environments becomes more difficult for typical rigid AUVs and more feasible for soft, compliant underwater robots that can adaptively deform to their surroundings. Moreover, soft underwater robots show great promise as biomimetic vehicles that can take inspiration from nature's swimmers and help answer questions about their behavior, for instance about the schooling capabilities observed in many fish species. Unfortunately, few fully autonomous, self-contained underwater soft robots have been developed, let alone fleets of such robots. To address this, we present a milestone towards formation control of a fully autonomous, multi- soft robotic fleet inspired by fish schooling. We present a vision-based, leader-follower formation strategy using an untethered soft robotic fish (SoFi) platform that enables one SoFi robot to pursue another via a visual servoing behavior. Our system demonstrates basic formation control of a pair of fully autonomous, self-contained soft robotic fish without external input.**

***Index Terms*—soft robotics, robot fish, biomimetic, swarm, formation control, vision-based, fish schooling**

I. Introduction

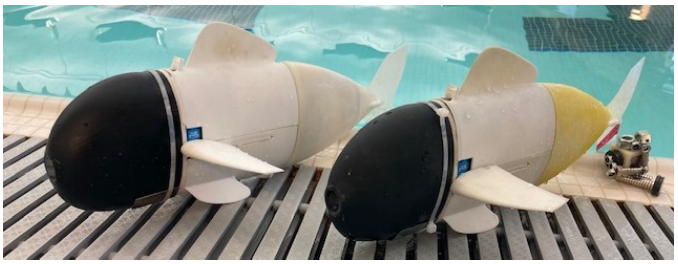


Fig. 1: Two SoFi robots. The follower is on the left and the leader is on the right.

In this work, we present the design and implementation of a *soft, bio-mimetic* multi-robot system for underwater monitoring and exploration that is *self-contained* and *fully autonomous*.

[1]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology

[2]Massachusetts Institute of Technology and Woods Hole Oceanographic Institution Joint Program

*Corresponding author email: salazarj@mit.edu.

The field of underwater robotics is seeing a growing interest in the coordination and control of multi-robot systems with promise to benefit several applications, including ocean sampling, mapping, tracking of marine life, and inspecting critical undersea infrastructure. Much work has been accomplished towards formation control of fleets of traditional autonomous unmanned vehicles (AUVs), including propeller-based and gliding vehicles [2], [5], [10], and yet a relatively small amount towards formation control of fleets of underwater soft robots. In this work, we aim to develop a multi- soft robotic platform for use in testing both hardware and software concepts that could expand the benefit of autonomous soft underwater vehicle fleets. In particular, we focus on implementing formation control for a school of soft robotic fish.

Self-contained, soft robots for underwater tasks Underwater soft robots have potential to compensate for certain drawbacks of traditional AUVs. While a typical AUV is rigid and offers limited flexibility and adaptivity to environments such as caves and reefs, a soft robot can excel in performing complex tasks in these areas due to its ability to adaptively deform [4], [9]. The sea star-inspired robot PATRICK demonstrated a notable contribution in closed-loop locomotion planning of mobile, untethered soft robots that can interact safely with their environments and which could one day reduce the invasiveness of our ocean exploration efforts [8]. Soft robots have also shown significant promise as biomimetic explorers, such as the aforementioned sea star and also undulatory swimmers. Some examples of these self-contained and soft, biomimetic underwater swimmers include fish, rays, and jellyfish [3], [11], [12], but are controlled in open-loop, which is insufficient to study multi-agent behaviors. Soft swimmers that use closed-loop control, using dielectric elastomer actuators, were developed to swiftly transition between different formations, which were based on typical natural swarming behaviors, via a global vision-based positioning system [13].

A soft snailfish-inspired robot developed at Zhejiang University in China demonstrated its free swimming capability using onboard power, control, and soft actuation at 10,900 below the ocean's surface around the Mariana Trench [7]. MIT's soft robotic fish (SoFi), which was developed at the Computer Science and Artificial Intelligence Laboratory, is an untethered biomimetic undulatory swimming robot that actuates a soft silicone elastomer tail to produce thrust and receives commands via acoustic signals sent by a nearby remote controller. SoFi performed multiple 40-minute swimming expeditions in the Pacific coral reefs and demonstrated its capability to navigate around marine life at various depths [6].

Nevertheless, these examples of underwater soft robots do not feature fully self-contained autonomous capability (e.g. PATRICK uses an external camera to track and pursue its targets) [8]. This motivated us to develop a vision-based autonomous capability for the SoFi platform with the goal of achieving formation control for a school of SoFi robots.

Fish-like, multi-robot systems While there have been numerous studies of fish-like robotic systems in hardware and, independently, algorithms development of multi-robot, schooling behaviors, there are few instances of full systems being developed to study both the hardware and software simultaneously. One particular and recent example is the BlueSwarm from Berlinger et al. [1], which consists of fish-like underwater robots that exhibited decentralized 3D formation control via the signaling and sensing of blue light among the robots. However, these systems are still based on relatively rigid robotic actuators.

While these works offer unique contributions to formation control of underwater robots, these robots either rely on an external system for autonomy or do not capitalize on the benefit of soft robots. Thus, we aim to address this by transforming the SoFi platform into a fully self-contained, autonomous multi- soft robotic platform that can help mimic and study fish schooling as well as test other concepts for autonomous soft underwater robot fleets.

II. SoFi Platform Overview

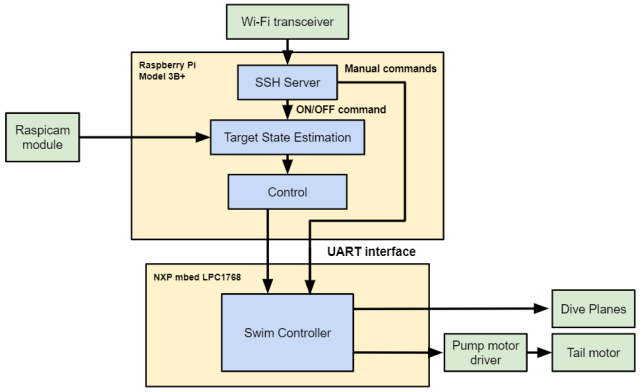


Fig. 2: Overview of SoFi software and hardware architecture for tracking.

In this section we summarize the vehicle concept, electronics and software architecture of our SoFi platform. We present a photo of two SoFi robots in Figure 1.

A. Vehicle Description

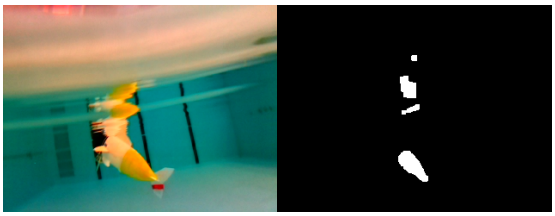
SoFi is an untethered, self-contained underwater system that propels itself via an undulating hydraulic silicone elastomer tail, which is designed to flex when one of its two internal chambers is pumped with liquid. The pump mechanism is an external gear pump, which in turn is driven by a brushed DC motor that can periodically alternate directions to achieve undulatory motion. By maintaining a cyclic alternating flow and varying the relative amount of liquid pumped into each of its two internal chambers, SoFi can perform turning motions with varying rates. In addition, SoFi is equipped with two symmetric control surfaces for pitch control and diving [6].

B. Software Architecture

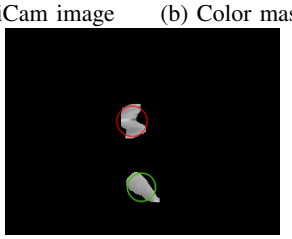
A software architecture overview diagram is illustrated in Figure 2. For sensory input, SoFi is equipped with a monocular Raspberry Pi camera module that is attached to a fisheye lens. The high-level computation and processing of sensor input and control is handled by an onboard Raspberry Pi 3B+, which interfaces via UART with an NXP LPC1768 microcontroller that receives the control commands and performs the electrical actuation of the pump and control surfaces. The high-level architecture on the Raspberry Pi was developed using the Robot Operating System (ROS) framework and the low-level architecture on the NXP microcontroller (developed in C) was carried over from the previous version of SoFi's software.

Originally, SoFi was designed to receive control commands via underwater acoustics from an acoustic remote operated by a nearby diver. For this work, we opted to remove the acoustic communications system to focus our efforts on developing a completely autonomous SoFi platform and instead we execute commands to start and stop SoFi's autonomy stack via SSH under a local Wi-Fi network. Once the autonomy stack is activated, SoFi independently computes its control commands based on sensory feedback from the camera.

III. Technical Approach



(a) Raw PiCam image (b) Color mask (yellow)



(c) Blob processing

Fig. 3: SoFi's visual tracking pipeline illustrated via sample frames. The raw image taken by the Raspberry Pi camera, which is shown in (a), is compressed and fed into a color mask targeting a pre-specified HSV range for yellow, resulting in the frame in (b). Finally, the mask frame is pruned of noise and small holes and then fed through a blob detector to find the centroid and radius of the largest shape by area, and false positives caused by surface reflections are rejected, as highlighted in (c).

As a milestone towards autonomous formation of soft robotic fish swarms, we devised a leader-follower coordination strategy in which one 'follower' SoFi robot pursues a second 'leader' SoFi robot via target state estimation and closed-loop control. To achieve this coordination over our robotic fish platform, we developed a new software architecture that enables a visual servoing behavior. Our efforts for this architecture were mainly divided between developing a target state estimation pipeline and a workflow for high-level control behavior.

A. Target State Estimation

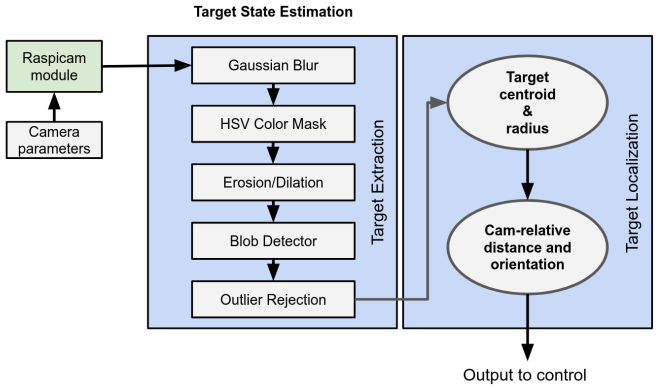


Fig. 4: Target visual tracking pipeline, an expansion of the module shown in Figure 2.

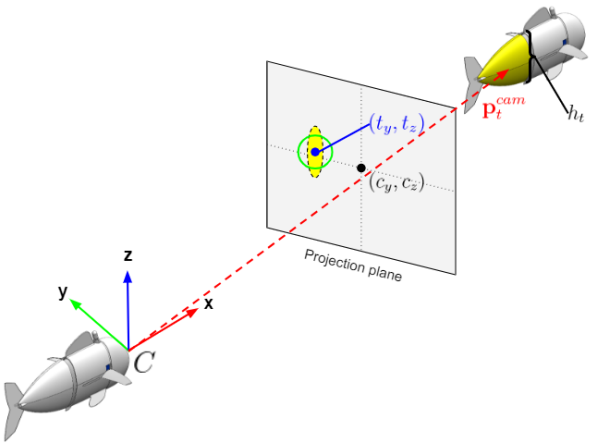


Fig. 5: Visualization of target location in SoFi's camera frame.

Our approach enables a follower SoFi robot to estimate the relative 3D position of a leader SoFi robot via monocular visual tracking. Our pipeline achieves this by performing color segmentation and blob detection on the image stream captured by the onboard Raspberry Pi camera. In this section, we summarize the two primary portions of the pipeline: 1) target detection and 2) state measurement and filtering. We present an overview of the pipeline in Figure 4.

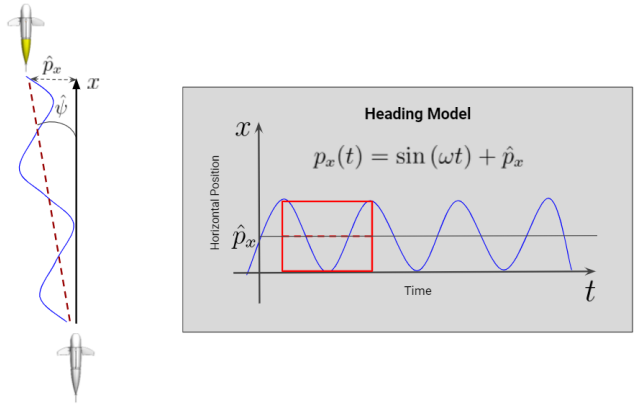


Fig. 6: Left: Illustration of filtered measurements $\hat{p}_y$ and $\hat{\psi}$. Right: Illustration of heading model with parameters ω and $\hat{p}_y$. The blue sinusoidal curve represents the raw target y-offset p_y in both illustrations.

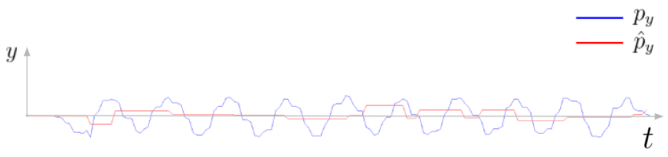


Fig. 7: Raw heading estimate $p_y(t)$ (blue) overlayed on filtered heading estimate $\hat{p}_y(t)$ (red).

1) Target Detection: At the start, a raw frame captured by the onboard camera is compressed and applied a Gaussian blur to reduce noise. We then transform the RGB image into the HSV scale and apply a color mask to capture a range of yellow hues, which is tuned for the deployment setting, and produce a color segmented image. Next, we eliminate small holes and further eliminate noise using morphological transformations on the color segmented image. We proceed by applying a simple blob detector to determine the centroid and radius of the largest color segmented region in the image by area. Figure 3 illustrates the evolution of an image frame at different stages of the pipeline. The bulk of the image processing operations described above were implemented using the OpenCV library. When operating in shallow waters, it is common to see reflections of target fish near the surface, as shown in Figure 3, to deal with this, in the case we detect multiple blobs of the correct shape, we reject all blobs except the bottom-most.

2) State Measurement and Filtering: Given the pixel coordinates (t_y, t_z) of the target centroid, radius of the detected blob in pixels r, the image center in pixel coordinates (c_y, c_z), the camera focal length f, and the height of the target object h_t, which are known a-priori via standard camera calibration and measurement of the tail sizes, we estimate the target's 3D position vector $\mathbf{p}_t^{cam} = (p_x, p_y, p_z)$ relative to the camera frame C (illustrated in Figure 5.)

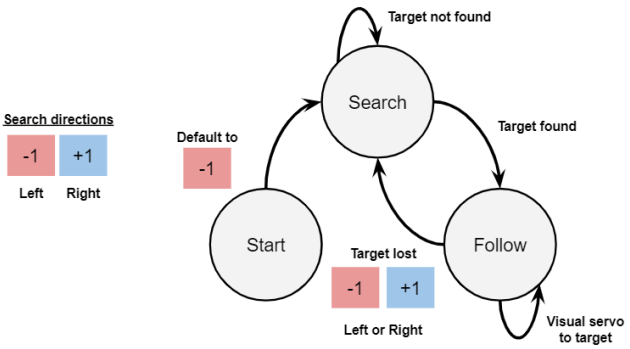


Fig. 8: High-level control finite state machine.

$$p_x = \frac{f h_t}{2r} \tag{1}$$

$$p_y = \frac{-p_x(t_y - c_y)}{f} \tag{2}$$

$$p_z = \frac{-p_x(t_z - c_z)}{f} \tag{3}$$

We also compute the heading angle ψ to the target around the z-axis as illustrated in Figure 5.

$$\psi = \arctan \frac{p_y}{p_x} \tag{4}$$

For a stream of detections over time, we smooth the profile of heading measurements using a second-order Butterworth low-pass filter due to the oscillation of the camera, which is induced by the tail undulation. First, we characterize the target's oscillation in the camera frame using a simple sinusoidal function parametrized by SoFi's undulation frequency ω and the target offset estimate $\hat{p}_y$ in the camera y-axis, as illustrated in Figure 6. Then, the filter outputs $\hat{p}_y$ as well as the filtered heading estimate $\hat{\psi}$.

B. Control Behavior

In order to ensure general robust performance and full autonomy of the SoFi, we implement a finite state machine similar to [1]. This enables a SoFi robot to transition between three states: START, SEARCH, and FOLLOW. In the START state, the robot runs all initialization tasks, such as loading calibrations and interfacing with embedded processors. SoFi then transitions into a SEARCH state after a pre-defined time. The SEARCH state is used for initial target acquisition as well as re-acquisition in case the target is lost. For initial acquisition, SoFi swims in a pre-defined circular pattern at a fixed depth. Once a target is acquired and filtered according to the sections above, SoFi transitions in the FOLLOW state. In this state, the filtered heading estimate $\hat{\psi}$ is fed into a hand-tuned PID controller, which produces a proportional steering control command using the error $E = \hat{\psi} - \psi_d$ where $\psi_d = 0$. Finally, in cases where a target is not detected by the visual tracking pipeline for 10 frames, SoFi transitions back into the SEARCH state. However, instead of swimming in a pre-determined direction, SoFi attempts to re-acquire the target by swimming in the same direction as it was previous to its tracking loss.

IV. Leader-Follower Formation Experiment

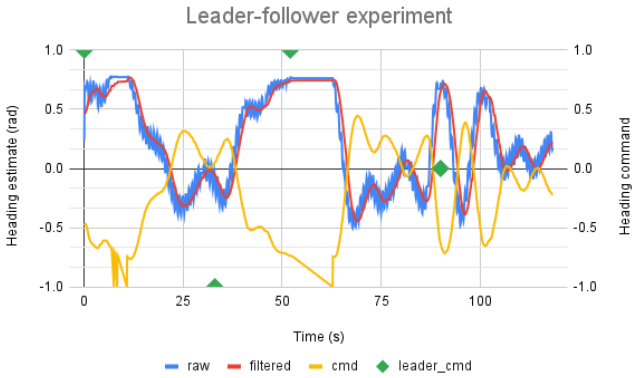


Fig. 9: A plot of the averaged target heading estimate and the resulting steering control command over time from a follower SoFi tracking a leader SoFi. Positive commands turn the robot towards the right in yaw, and negative goes left. The spike in command near 52sec is when the follower briefly loses sight of the leader. Corresponding qualitative visualizations are shown in Figure 10.

In this section, we describe our leader-follower system.

A. Experiment overview

To demonstrate our leader-follower coordination strategy, we tested SoFi's ability to fully autonomously follow another SoFi. In this demonstration, we used two SoFis, one which we will refer to as the *leader* and another as the *follower*. The goal is to estimate the follower's ability to robustly track and pursue the leader in a swimming pool. First, the leader is placed in the water ahead of the follower SoFi. The follower SoFi is then commanded to search and then follow the leader SoFi. The leader is manually driven via wireless commands sent from a shore-station computer through the SoFi ROS-interface. For this test, we commanded the leader SoFi to first swim right, left, right, then straight. Qualitative results are presented in Figure 10 and quantitative results based on the follower's estimates of the target relative state and corresponding control signals are reported in Figure 9.

B. Results and Discussion

From the qualitative results shown in Figure 10 and quantitative results in Figure 9, we can see that the follower SoFi is, in general, able to successfully track the manually driven leader SoFi. The follower "bounces" between a 0-error estimate and about 0.5 radians, as shown around the 25 second, 75 second, and 100 second regions, due to the lack of prediction about the overall future trajectory of the

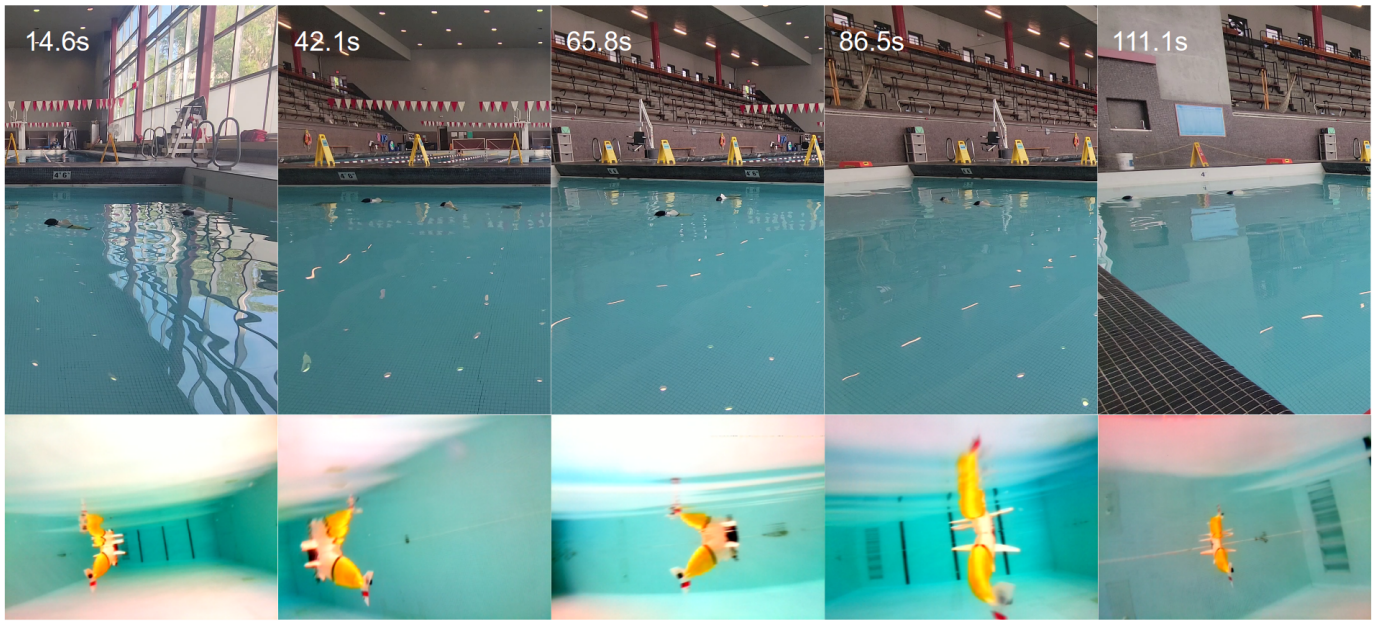


Fig. 10: Visual servoing behavior in action, the leader SoFi is manually driven and the follower is fully autonomous. Images above are from an external camera, below are the corresponding frames from the follower's camera. The leader initially starts to the front left of the follower and is instructed to swim to the right. The leader is commanded to turn left at 29 sec, right at 52 sec, and straight at 90 sec. The follower SoFi tracks it successfully throughout each maneuver. Times are manually synchronized between cameras and with the plot in Figure 9.

leader and the follower controller thus lags after reaching the correct setpoint. We also note that the recovery time worsens because as the linear distance between the follower and leader decreases, the rate of perceived rotation around yaw of the follower is much faster, which is currently not accounted for in the naive yaw-tracking controller. At 52 seconds, the follower loses sight of the leader. In these cases, the controller still swims in the correct direction at max rotational velocity, as indicated by the spike in control command near 60 seconds, due to our high-level state controller shown in Figure 8, and is able to re-acquire the target and continue tracking at the 62 second mark.

It is important to note that the leader fish also tends to drift to the left in open-loop control due to minor manufacturing-related issues and the presence of small perturbations from distant jets in the test pool. Manufacturing inconsistencies in the tails also contribute to differences in linear and rotational velocities of the two robots, which also affects general tracking performance and turning radii of each robot.

V. Conclusion and Future Work

In conclusion, to the best of our knowledge, we have demonstrated a first of its kind implementation of a fully-autonomous and self-contained, soft multi-robot system. We believe that these form the necessary initial steps towards enabling larger-scale efforts in soft-robotic swarms and their utility in studying animal behaviors as well as performing exploration and monitoring tasks that require additional environmental safety. Future work includes further hardware and software development to enable additional complexity and scalability for a wider range of tasks, such as larger schooling behaviors using 3 or more robots, and to test the platform in real-world conditions such as monitoring coral reefs. One example is to account for distance in the rotational controllers (in pitch and yaw), and to develop more consistent manufacturing procedures to ensure tail performance.

VI. Acknowledgment

This work was supported in part by the National Science Foundation under grant NSF EFRI 1830901 and DARPA under grant FA8750-20-C-0075. We are grateful for this support. Levi Cai was supported by the Department of Defense (DoD) through the National Defense Science Engineering Graduate (NDSEG) Fellowship Program. We thank former MIT students Jadorian Paul and Robert Williamson for their contributions to SoFi's software and hardware development. We also thank Professor Robert Katzschmann and Antares Zhang of ETH Zürich for their advice and support on hardware design and fabrication.

References

[1] Florian Berlinger, Melvin Gauci, and Radhika Nagpal. Implicit coordination for 3d underwater collective behaviors in a fish-inspired robot swarm. *Science Robotics*, 6(50):eabd8668, 2021.

[2] Edward Fiorelli, Naomi Ehrich Leonard, Pradeep Bhatta, Derek A. Paley, Ralf Bachmayer, and David M. Fratantoni. Multi-auv control and adaptive sampling in monterey bay. *IEEE Journal of Oceanic Engineering*, 31(4):935–948, 2006.

[3] Jennifer Frame, Nick Lopez, Oscar Curet, and Erik D Engeberg. Thrust force characterization of free-swimming soft robotic jellyfish. *Bioinspiration & biomimetics*, 13(6):064001, 2018.

[4] Kevin C Galloway, Kaitlyn P Becker, Brennan Phillips, Jordan Kirby, Stephen Licht, Dan Tchernov, Robert J Wood, and David F Gruber. Soft robotic grippers for biological sampling on deep reefs. *Soft robotics*, 3(1):23–33, 2016.

[5] Rui Gomes and Fernando Lobo Pereira. Attainable-set model predictive control for auv formation control. In *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, pages 1–6, 2018.

[6] Robert Katzschmann, Joseph Delpreto, Robert MacCurdy, and Daniela Rus. Exploration of underwater life with an acoustically-controlled soft robotic fish. 2018.

[7] Guorui Li, Xiangping Chen, Fanghao Zhou, Yiming Liang, Youhua Xiao, Cao Xunuo, Zhen Zhang, Mingqi Zhang, Baosheng Wu, Shunyu Yin, Yi Xu, Hongbo Fan, Zheng Chen, Wei Song, Wenjing Yang, Binbin Pan, Jiaoyi Hou, Weifeng Zou, Shunping He, and Wei Yang. Self-powered soft robot in the mariana trench. *Nature*, 591:66–71, 03 2021.

[8] Zach J. Patterson, Andrew P. Sabelhaus, Keene Chin, and Carmel Majidi. An untethered brittle star robot for closed-loop underwater locomotion. *CoRR*, abs/2003.13529, 2020.

[9] Steven I Rich, Robert J Wood, and Carmel Majidi. Untethered soft robotics. *Nature Electronics*, 1(2):102–112, 2018.

[10] Nicholas Rypkema and Henrik Schmidt. *Formation Control of a Drifting Group of Marine Robotic Vehicles*, pages 633–647. 03 2018.

[11] Z Wang, Y Wang, J Li, and G Hang. A micro biomimetic manta ray robot fish actuated by sma in 2009 ieee int. In *Conf. on Robotics and Biomimetics (ROBIO), Guilin, China*, pages 19–23, 2009.

[12] Zhenlong Wang, Guanrong Hang, Jian Li, Yangwei Wang, and Kai Xiao. A micro-robot fish with embedded sma wire actuated flexible biomimetic fin. *Sensors and Actuators A: Physical*, 144(2):354–360, 2008.

[13] Zhen Zhang, Tao Yang, Tianhao Zhang, Fanghao Zhou, Nuo Cen, Tiefeng Li, and Guangming Xie. Global vision-based formation control of soft robotic fish swarm. *Soft Robotics*, 8(3):310–318, 2021. PMID: 32654595.