A HYDRAULIC, JELLYFISH-BASED SOFT ROBOT

By

KE YANG

A thesis submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Mechanical and Aerospace Engineering

Written under the direction of

Aaron Mazzeo

And approved by

_____

_____

_____

New Brunswick, New Jersey

October, 2015

ABSTRACT OF THE THESIS

A HYDRAULIC, JELLYFISH-BASED SOFT ROBOT

By KE YANG

Thesis Director:

Professor Aaron Mazzeo

This thesis describes the design, fabrication, and characterization of a soft robotic jellyfish. Living jellyfish are some of the most efficient swimmers in the ocean and exhibit unique capabilities for manipulating fluid flows for efficient propulsion. The fabricated soft vehicles in this work emulate some of the characteristics of living jellyfish and their propulsive mechanisms have the potential to influence the future design of efficient underwater vehicles. With respect to fabrication, the author uses silicones of varied hardness to mimic the flexibility and shape of biological jellyfish. Characterization and testing of the biomimetic vehicles facilitate comparisons to living jellyfish and provide physical insights into the dependence of efficiency on timed gaits and formed vortices. Based on calculated costs of transport, a slow gait results in the highest locomotive efficiency. These results suggest that further optimization of the designed vehicles and gait will yield future propulsion systems with even higher efficiency.

ACKNOWLEDGMENT

It is my great pleasure to express my deep sense of gratitude to my advisor Dr. Aaron Mazzeo, it is him who guided me, encouraged me to keep working on the robotic jellyfish projects.

And it is also my privilege to thank Melanie A. Cotton, who has been a great researcher helping pushing the jellyfish project moving forward. It would not be possible to get so many experiments and analysis done, and so many iterations of the outlines of the publication done without her. She is one of the smartest, the most dedicated person I have ever worked with. It is my honor to work with her.

I am very thankful to Jingjin Xie, Suze Zhang, Yanjun Wang, Xiangyu Gong, Eugene Kim, and Ge Song, who have all spent nights helping me work on experiments and collecting data.

I am very thankful to Mr.John Petrowski, who had helped me make the electro-hydraulic system for the jellyfish robot.

DEDICATION

This dissertation is gratefully dedicated to my beloved parents.

Table of Contents

ABSTRACT OF THE THESIS ii

ACKNOWLEDGMENT iii

DEDICATION iv

Table of Contents v

1. Introduction 1

2. Experimental Design 5

2.1. Linear Actuation and Contraction of the Bell of the Jellyfish 5

2.2. Fabrication of the Bladder and Bell Segments of the Jellyfish 9

2.3. Hydraulic Power Delivered to Jellyfish 13

2.4. Characterization and Measurement 13

3. Results and Discussion 17

4. Conclusions 29

Appendix A: Supplementary Figures 30

Appendix B: Supplementary Table 36

Appendix C: List of supplementary Movies 37

Appendix D: Matlab Code 38

Appendix E: Arduino Code 47

Appendix F: Formula used in this work 52

References.....53

1. Introduction

Currently, underwater vehicles and aquatic robots do not achieve the efficiency attained by many biological species. Furthermore, most artificial vehicles are composed entirely or primarily of rigid materials, making them unsafe for use around people and wildlife. Metallic propellers are dangerous, may rust, and may need to be replaced if they are not properly treated. Copper and other toxic chemicals found in antifouling paint and treatments used on propellers contaminate water ecosystems. The use of propellers and rigid materials in most underwater vehicles poses a threat to people and wildlife. Propellers are energy inefficient, cause injuries to people or animals, and destroy marine plants. In order to explore possible solutions to these problems, we built a robotic system to mimic the PER feature present in moon jellyfish (*Aurelia aurita*), which is considered the primary reason some jellyfish species swim with remarkably high efficiency. The use of a soft, flapping underwater vehicle would offer a safer alternative to traditional underwater vehicles and could have potential applications in the search and rescue of people or animals and in the observation of wildlife. The proposed flapping propulsion of jellyfish robots could not become ensnared by plants (and damage them in the process), so it would be suitable for underwater vehicles which travel in shallow waters. Furthermore, the jellyfish robot platform was easy to modify. The researcher had the freedom to change the control settings, the curvature of the jellyfish bell, the localized stiffness of the jellyfish robot body (i.e. by using different silicone rubber materials, Mold Star® 30 and Ecoflex® 30, at various locations on the robot body), and other parameters.

This project was motivated by the now widely accepted theory of PER, developed by Gemmell, et al. [1] using the observation and measurement of live jellyfish. However,

while studying biological jellyfish helped us to explore the behavior and features of real jellyfish, it did not afford us the freedom to isolate and to examine each factor independently (e.g. the curvature of the jellyfish bell, the flexibility of the bell structure, the gait of the jellyfish propulsion). In order to obtain this freedom, we developed a jellyfish robot platform that mimicked the swimming pattern of live jellyfish and that displayed PER, a key feature of *Aurelia aurita*. We did not understand the impact of gait on PER, as it was impossible to test multiple gaits on a biological specimen. A robotic platform allowed for complete manipulation of structure and of gait, providing the opportunity to fully comprehend their effect on PER.

It was once widely held that jellyfish were inefficient swimmers[2][3][4]. However, the currently held belief is that jellyfish are the most efficient swimmers, due in part to PER, as described by Gemmell, et al. Researchers have already developed several jellyfish robots, but these robots offered little insight into the swimming mechanisms of jellyfish. The jellyfish developed by Nawroth, et al. [5] was completely soft, but due to its fragility, it was not suitable for a test bed. Furthermore, the living jellyfish species this robot mimicked does not exhibit PER (a second period of acceleration was not present in the velocity plots in their paper). The jellyfish robot built by Priya, et al. [6] was not completely soft, as it relied on metallic actuators embedded inside the structure of its bell; it used the heat generated by an electric current to drive the shape-memory-alloy (SMA) to bend during the contraction stage. This limited the robot's ability to swim under different conditions, e.g. in very cold water, which slowed the contraction of the robot or in very warm water, which made it impossible for the robot to revert to its original shape. Meanwhile, our jellyfish robot used hydraulic power, and most of its structure consisted of

silicone rubber, so it was highly flexible, while still durable enough for a test bed, and capable of swimming in cold or hot water.

Since its discovery, PER has piqued interest in the propulsion of jellyfish. The basic swimming mechanism of biological jellyfish capable of exhibiting PER consists of four phases, including a static phase. As seen in Figure 1, the jellyfish begins in a completely static state. Then, the bell contracts to produce the starting vortex [7][8][9][10][11][12], which rotates outwardly, from inside the bell of the jellyfish to outside. After this contraction stage, the relaxation stage begins with the recoil of the bell, producing the stopping vortex, which rotates inwardly, from outside the bell to inside, opposing the starting vortex. For several jellyfish species (including the moon jellyfish), the stopping vortex continues rotating to draw more fluid into the region underneath the bell structure during the second half of the relaxation stage. The amassed pressure within this region results in a second period of acceleration for the jellyfish [1], which is indicative of the PER stage.

However, we found that altering the speed of actuation resulted in formations of vortices that differ from this originally proposed mechanism seen in biological jellyfish.

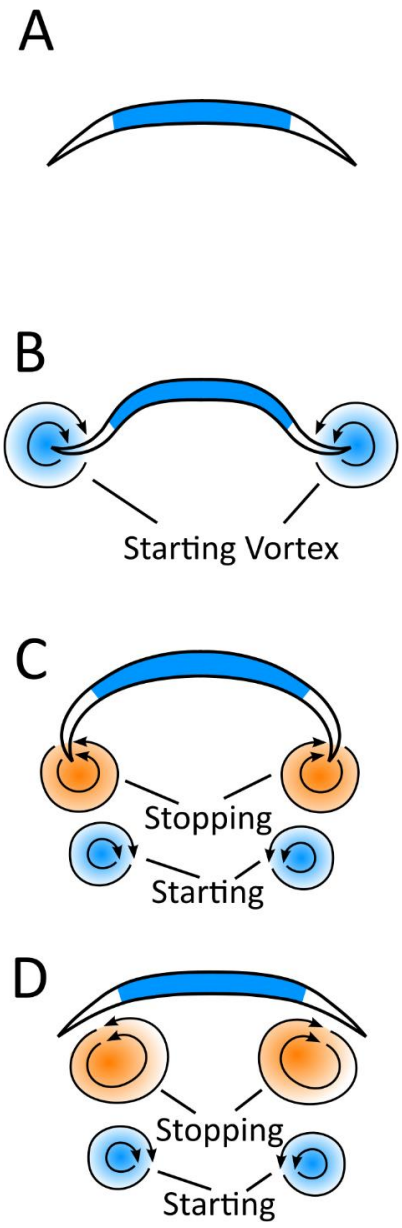


Figure 1: The state-of-the-art understanding of the swimming gait of the *Aurelia aurita*. (a) The jellyfish prepares to contract to begin an actuation cycle. (b) At the start of the contraction stage, the starting vortex forms at the tip of the bell. (c) As the contraction stage ends and the relaxation stage begins, the starting vortex is shed from the tip of the bell, and the stopping vortex begins to form. (d) The stopping vortex grows and continues to rotate beneath the inner surface of the bell, providing the thrust that generates the second period of acceleration indicative of PER.

2. Experimental Design

The soft robotic jellyfish is inspired by several different types of soft robot which has been developed recently[13][14][15]. The soft robotic jellyfish consisted of several main components: a bell structure, the soft hydraulic actuator, the pulling mechanism, and the tethering tubes. The bell structure was composed of six fin-like segments connected to a centerpiece and thin membranes which connected the segments. The soft hydraulic actuator was capable of linear expansion in one direction, and the pulling mechanism translated the actuator motion to the contraction of the bell. The tethering tubes pumped water to the actuator.

2.1. Linear Actuation and Contraction of the Bell of the Jellyfish

The hydraulic linear actuator was the key mechanism used in our robot to mimic the contraction of the bell of jellyfish. As shown in Figure 2, the pulling mechanism connected the bottom surface of the actuator and the middle of the bell structure with nylon strings. Therefore, when we inflated the actuator with water, the bottom of the actuator lowered, and the pulling mechanism pulled the bell structure toward the actuator, resulting in contraction. Relaxation was the opposing process, in which we removed the water from the soft actuator, thereby deflating it, allowing the strings to slacken, and leading to the recoil of the bell. The merit of this simple pulling mechanism was that the strings were incapable of pushing, meaning it was able to deliver the hydraulic power to the bell during contraction only. This ensured the strain energy stored inside the bell and the energy stored in the surrounding fluid were the only influences acting on the jellyfish during its relaxation phase. As the relaxation of biological jellyfish is entirely passive, our pulling mechanism

successfully mimicked this feature of real jellyfish. The tethering tubes (as shown in Supplementary Figure 3), through which water passed, inflated and deflated the soft actuator, while the unique design of the one-to-three fitting minimized the potential pushing or pulling forces that could have acted on the robot. This enabled us to acquire very accurate displacement and velocity curves and to observe fine details in these curves, such as the second period of acceleration, which indicated PER.

We constructed the bell segments from two different types of silicone rubber, Ecoflex® 30 and Mold Star® 30. Ecoflex® 30 was a very soft, white silicone rubber; it was quite yielding to water currents and pressure. We placed Ecoflex® 30 at the bottom edges of the bell segments to make them very flexible. This ensured the edges remained in contact with the stopping vortices for as long as possible and minimized the amount of turbulence present in the fluid around the robot, increasing its efficiency. Mold Star® 30 was a stiffer, blue silicone rubber, and it formed the upper parts of the fins and the centerpiece at the apex of the body of the jellyfish. This gave the jellyfish sufficient stiffness to contract when actuated and to propel itself through the water. Thin latex membranes connected the separate bell segments. They were sufficiently thin so as not to interfere with the bell segments' shape during contraction. Rather they folded to accommodate the contracted state and then reverted to their original shape in the relaxed state. These membranes ensured that when the stopping vortex drew water under the bell as the jellyfish progressed into its relaxed state, an area of high pressure formed under the bell, which lent itself to PER. The curvature of the bell segments was crucial, as it greatly contributed to the performance of the jellyfish. If the curve was too flat, it proved difficult to contain the stopping vortex that returned to the underside of the robot during the PER phase because

the water current slipped from the edge of the bell segment to avoid building up high pressure underneath the bell structure.

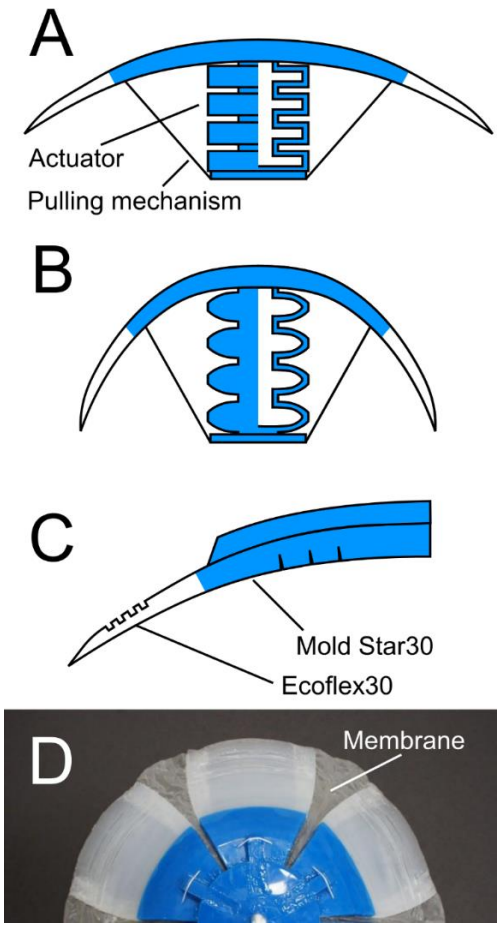


Figure 2: The detailed design of the jellyfish robot. (a) The cross sectional rendering of the robot illustrates the relaxed state of the jellyfish robot; the soft linear actuator was deflated. (b) The cross section shown with the inflated actuator depicts the pulling mechanism, which forced the bell to contract toward the lower end of the actuator. (c) The cross section of a curved bell segment is shown. The segment was made of two different materials, blue Mold Star® 30, located close to the center, and white Ecoflex® 30, positioned at the edge of the bell. The stiffness of Mold Star® 30 strengthened the upper structure, and the white Ecoflex® 30, which was much softer than Mold Star® 30, made the margin of the bell segments very flexible. (d) The top view of the bell segments shows the latex membranes, which connected the bell segments.

2.2. Fabrication of the Bladder and Bell Segments of the Jellyfish

As stated above, we constructed the soft jellyfish body structure from two types of silicone rubber, Mold Star® 30 and Ecoflex® 30. We positioned Mold Star® 30 close to the center of the jellyfish body, as it was stiffer, while the flexible Ecoflex® 30 provided the margin of the bell with the requisite softness. We made all the components using soft lithography with 3D printed molds. We made the six fins by first fabricating the top half of the fin in a partial mold using Mold Star® 30. We then placed the partial fin in the full mold for the fin and filled the remaining space with Ecoflex® 30, which bonded to Mold Star® 30 as it cured. We connected the six fins with thin latex membranes in order to form a sealed, continuous bell. We adhered the latex membranes to the fins with Sil-Poxy® and made small holes along the edge of the latex piece prior to gluing to increase the adhesion of the two dissimilar materials.

We made the hybrid (soft-hard material) linear actuator using a one-piece molding technique. The 3D printed mold assembly consisted of two half molds, which we sealed with an O-ring and fastened with screws. A 3D printed centerpiece placed within the mold prior to sealing ensured the proper formation of the hollow center. We filled the mold assembly with Mold Star® 30 and placed it in a centrifuge to remove air bubbles[16][17] from the rubber before it cured. We removed the actuator from the mold and melted the centerpiece to remove it from the actuator. A cap made of Mold Star® 30 sealed one end of the actuator. We connected three pieces of rubber tubing to the open end of the actuator at one end, connected their opposite ends to a three-to-one fitting, and connected the single tube (that began at the fitting) to the manual hydraulic system. We used nylon strings to

connect the bottom end of the actuator to the bell segments. Therefore, during the contraction step, the actuator inflated, expanded linearly, and pulled the bell toward the center of the robot. The relaxation step occurred when the actuator deflated, the strings slackened, and the bell shape reverted to its original form.

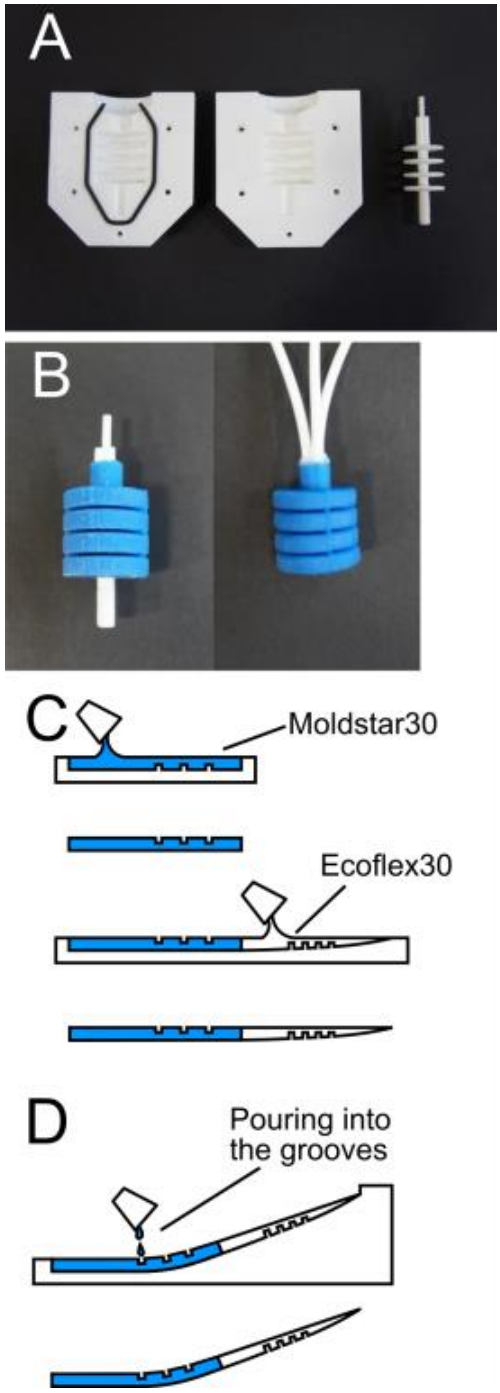


Figure 3 The fabrication of the soft, hydraulic actuator and the curved bell segments, which were composed of two materials. (a) The fabrication of the actuator involved the use of two half molds and a centerpiece. The assembly was filled with silicone rubber, fastened with screws, and placed into a centrifuge to remove air bubbles prior to curing. One half

mold had an O-ring to seal the assembly during centrifugal molding. (b) The image on the left shows the primary component of the one-piece actuator just after removal from the mold, with the centerpiece still intact. The image on the right shows a fully functional, one-piece soft linear actuator which had three pieces of tubing and one bottom cap installed. (c) The molding process of the bell segment with two different materials required two open, 2D molds. The upper image shows the molding of half the bell segment, by pouring liquid Mold Star® 30 into a 2D mold; the lower image shows the molding of the rest of the bell segment. The already-cured Mold Star® 30 piece was placed in the second mold, so that the grooves made by the first mold were face-up in this mold. (The two sets of grooves were positioned on opposite sides of the bell segment.) Ecoflex® 30 was poured into the mold to firmly bond it to the cleaned surface of the Mold Star® 30 part as it cured. (d) The gluing process which gave the curve to the 2D bell segment used Mold Star® 30 as the bonding material. Small amounts of the liquid Mold Star® 30 were poured into the set of grooves in the Mold Star® 30 part of the segment, and the bell segment cured while resting on a curved gluing stand. After the Mold Star® 30 cured completely (the application of heat made the curing process faster), the bell segment had a permanently curved profile.

2.3. Hydraulic Power Delivered to Jellyfish

The hydraulic system relied on manual actuation to power the robot. Two 10-mL syringes, silicone tubing, tube fittings and a manual valve rapidly delivered water to the soft linear actuator in the body of the jellyfish. A potentiometer, a pressure sensor, two microcontrollers, and two laptops (a motor drive chip, a power source) were used to monitor and measure the system. We had several other components, including a green LED operated by a button, used to correlate the video data and the data collected by the microcontrollers, and a buzzer, used to indicate the pace of each gait to the person who was actuating it.

2.4. Characterization and Measurement

In order to collect displacement data, we connected the jellyfish robot to the manual hydraulic system and placed it into a water tank. As we measured only vertical displacement, we forced the robot to point upward by fixing a piece of closed-cell Styrofoam to the apex of the jellyfish bell. The entire robot had a slightly negative buoyancy. We recorded (at 60 fps) the vertical swimming motion with a camcorder set on the side of the water tank and acquired the displacement data using image tracking code in MATLAB. We input the position of a specific point on the apex of the bell for each frame, and the code calculated the displacement data. We obtained the velocity data by taking the derivative of displacement with respect to time. We repeated this procedure three times to facilitate the comparison between fast, moderate, and slow speeds of actuation.

In order to perform the PIV analysis, we used a low-cost, customized setup. We introduced to the water tank twenty grams of water-neutral, plastic beads approximately 350 microns in diameter and coated with green fluorescent paint. We stacked two

projectors on the right side of the water tank and projected a vertical line of white light into the water, which illuminated a plane of particle-filled water in the tank. The movement of the fluorescent beads in this plane enabled us to visualize the fluid motion within the plane. We placed the camcorder (recording at 60 fps) facing the front of the water tank and perpendicular to the light plane, in order to visualize the 2D fluid flow field. We processed the recorded results using MATLAB with the PIVlab 1.4 toolbox[18]. We repeated this procedure three times to compare the vortex formation at each of the three speeds.

The COT metric we selected compares efficiency by measuring the power input per unit mass per distance traveled. In order to calculate the COT value for the jellyfish robot swimming under various gaits, we used the displacement and velocity measurements, the pressure sensor readings, which indicated the differential pressure value inside the tubing connected to the actuator, and the potentiometer attached to the two syringes used in actuation (see Figure 4), which indicated the position of the plungers. We found the volume of the water injected into the robot using the reading from the potentiometer, and we calculated the hydraulic energy input of the jellyfish robot from the pressure and volume data. We attached a green LED onto the front surface of the water tank and lit it prior to and following each trial. The LED served to calibrate the times corresponding to the object tracking data (acquired from video footage) and the pressure and volume data (acquired from the microcontroller). The instant the LED lit up in the video corresponded with the instant microcontroller recorded the input to light the LED. Finally, we calculated the COT for each actuation speed with the formula $COT = \frac{Energy}{Mass \times Distance} = \frac{\sum Pressure \times \Delta Volume}{Mass \times Distance}$ [1]

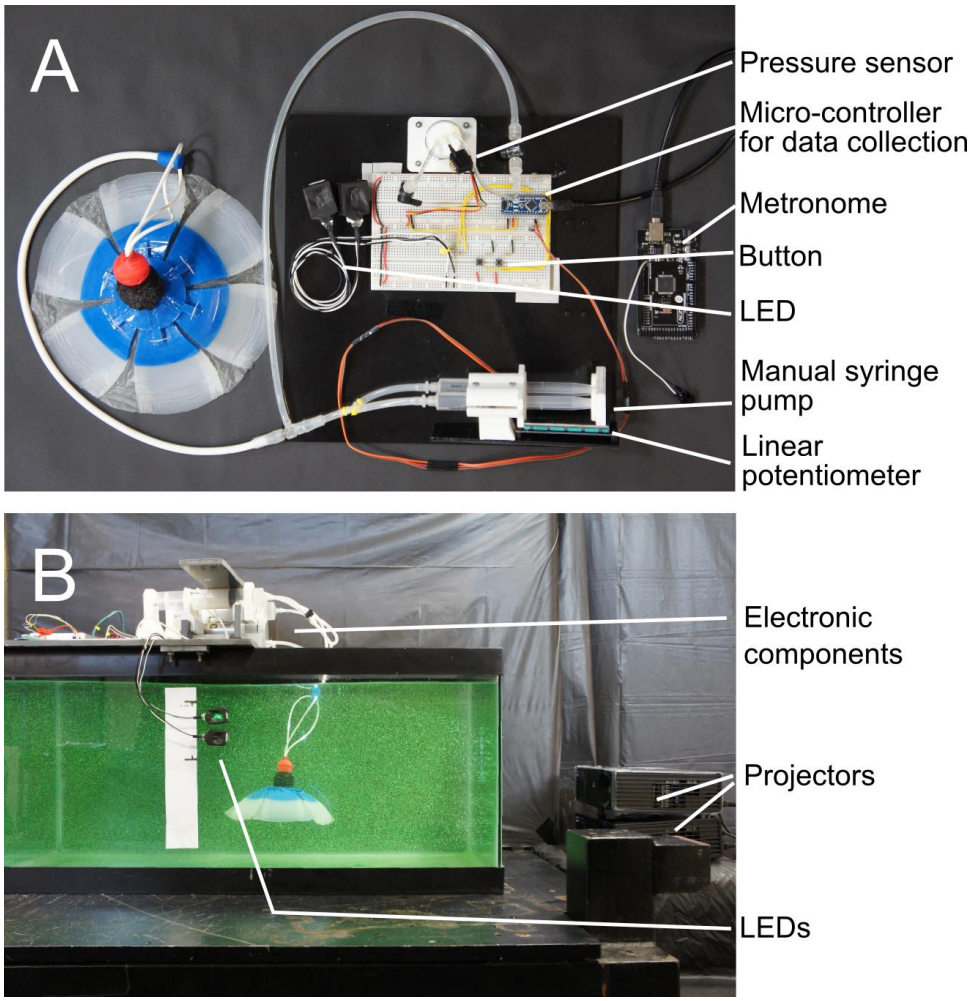


Figure 4: The complete system associated with the jellyfish robot and the experimental setup of the PIV filming. (a) The entire jellyfish robot system included the jellyfish robot itself, the manual syringe pump, a microcontroller for data collection, a metronome with is composed of a microcontroller and a buzzer, a differential pressure sensor, a linear potentiometer installed on the manual pump, a button, an LED, two projectors and two laptops. The syringes on the manual pump were attached to one another and connected to the linear potentiometer. (b) The tank used for testing is shown with the jellyfish robot placed in the particle-filled water. There were two projectors positioned on the right hand side, illuminating a single vertical plane in the tank. The green LED was mounted to the

front surface of the tank. The camera faced perpendicular to the lighted plane while recording.

3. Results and Discussion

We recorded video footage of the jellyfish robot swimming in a tank, and from this we collected displacement and velocity data. We took advantage of researching on a robotic jellyfish platform, which allowed us to freely alter the gait of the jellyfish and the speed of actuation. We investigated the potential effects of different speeds of actuation on jellyfish propulsion. We used a fast gait, a moderate gait, and a slow gait, with which we produced the displacement and velocity curves that indicated PER. We selected the three actuation speeds based on initial observations (made while operating the robot manually) and programmed a microcontroller connected to a buzzer to sound at the pace of the gait. During each actuation cycle, the first tone indicated the start of contraction, and the second tone indicated the start of relaxation. The fast gait actuated at a rate of 1.2 seconds per cycle, the moderate gait actuated at a rate of 2.0 seconds per cycle, and the slow gait actuated at a rate of 3.2 seconds per cycle.

Our jellyfish robot demonstrated the second period of acceleration (seen in the displacement and velocity plots) for the moderate and slow gaits, which indicated our robot recaptured energy stored in the surrounding fluid. Even during the first iteration of propulsion, the moderate gait exhibited PER. The presence of this trend in the first iteration was significant because it refuted other possible reasons for the behavior (e.g. momentum residual from the previous iterations of actuation). As the string mechanism, which pulled the bell to actuate the robot, was incapable of pushing, the recoil of the body could be driven only by the elastic power stored in the jellyfish bell. This signified completely passive recoil, and hence we claimed that our jellyfish robot successfully reproduced the unique PER feature found in *Aurelia aurita*.

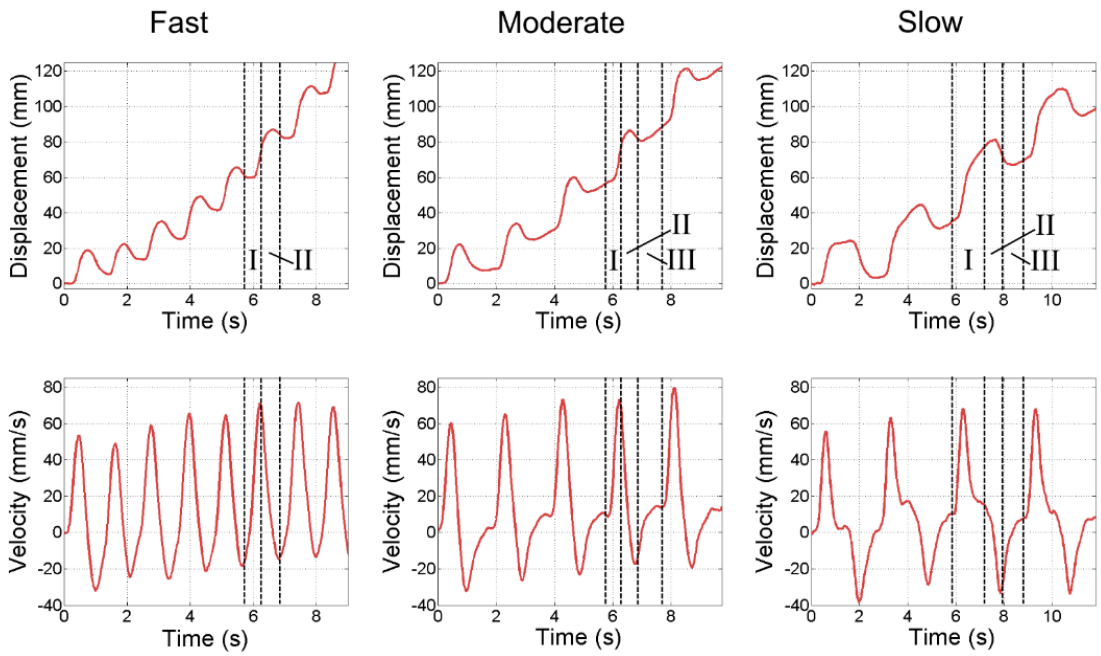


Figure 5: The displacement and velocity profiles of the jellyfish robot at three different speeds, obtained using video footage and tracking code. The displacement versus time plots for the jellyfish robot show the two or three stages of actuation: contraction, relaxation, and (for moderate and slow gaits) PER. The velocity versus time plots for the jellyfish robot show the acceleration indicative of PER at the end of each actuation cycle for only the moderate and slow gaits. (a) The velocity and displacement plots for the jellyfish robot actuated at a high speed (1.2 seconds/cycle) did not indicate the presence of PER. While this gait had the fastest actuation time and travels further in less time, it did not obtain two periods of velocity in each cycle, and the robot required more actuation cycles than the other gaits to travel the same distance. (b) The velocity and displacement plots for the jellyfish robot actuated at a moderate speed (2.0 seconds/cycle) show a very strong capability of PER. The PER trend is most apparent in the plots for moderate actuation. The presence of PER during the first iteration, though slight when compared to later iterations, was significant evidence in support of the validity of PER theory. (c) The

velocity and displacement plots for the jellyfish robot actuated at a slow speed (3.2 seconds/cycle) indicate PER, but the trend is weaker. The robot reached lower maximum velocities, but the distances travelled both forwards and backwards in each cycle were greater.

After observing the PER capability of our jellyfish robot, we focused on the flow patterns associated with each gait we tested. The formation of vortices differed greatly between the three gaits. We used PIV analysis to study these observed differences and to gain a better understanding of how fluid flows around the jellyfish robot. We found that adjusting the gait and velocity of the jellyfish robot resulted in the formation of multiple pairs of vortices during each iteration. This finding is described by organized planning of vortices (OPV). Live jellyfish and our jellyfish robot actuated at a fast pace exhibit 1-1 OPV, shedding one pair of vortices (one starting vortex and one stopping vortex) during each iteration.

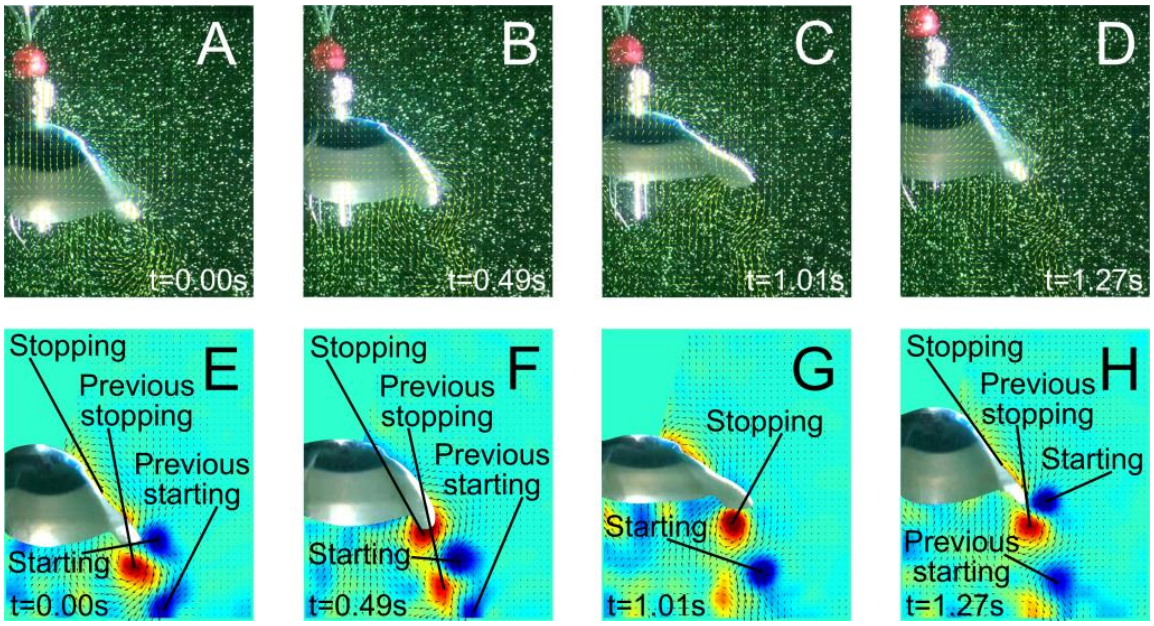


Figure 6: The vorticity associated with the fluid flow around the jellyfish robot at various stages in a fast actuation cycle (associated with a 1-1 OPV). PIVlab 1.4 in MATLAB processed the video footage and generated the vorticity plots. (a, e) The jellyfish robot began a cycle with a strong starting vortex formed by the edge of the bell. The robot shed the stopping vortex from the previous cycle from the bell, and the new stopping vortex began to form above the bell. (b, f) As contraction ended, the robot shed the starting vortex from the bell, and the stopping vortex moved below the bell. (c, g) The stopping vortex drew water under the bell as the jellyfish progressed into its relaxed state, and the starting vortex moved further away from the bell margin. (d, h) The next cycle began with the formation of new starting and stopping vortices.

As illustrated in figure 6, the robot shed a starting vortex during the contraction stage near the edge of the bell, and it interacted with the stopping vortex shed during the previous iteration of propulsion. Still during the contraction stage, a stopping vortex began to form upstream of the starting vortex, but initially it had a low vorticity. The stopping vortex moved to the underside of the bell, and the vorticity of the stopping vortex increased throughout the relaxation stage. This observation coincides with the findings stated in Gemmell's paper[19] which studied live moon jellyfish. However, slowing the gait of the jellyfish robot led to the production of 2-2 OPV, in which the robot shed two pairs of vortices (two staring and two stopping vortices) during each iteration. In addition, a moderate gait led to the production of 2-1 OPV, in which the robot shed two starting vortices and one stopping vortex during each iteration.

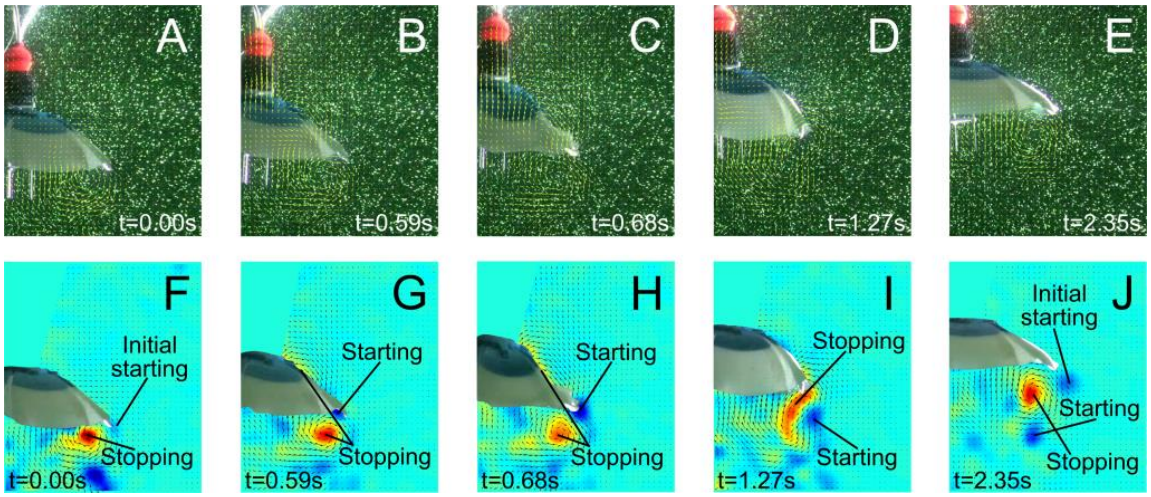


Figure 7: The vorticity associated with the fluid flow around the jellyfish robot at various stages in a moderate actuation cycle (associated with a 2-1 OPV). PIVlab 1.4 in MATLAB processed the video footage and generated the vorticity plots. (a, f) The jellyfish robot began a cycle with the formation of a weak initial starting vortex and the presence of a strong stopping vortex from the previous cycle. (b, g) As contraction began, the initial starting vortex dissipated and a starting vortex formed at the bell margin. The previous stopping vortex remained below the bell, while a new stopping vortex formed above the bell. (c, h) As contraction continued, the starting vortex increased in vorticity while the new stopping vortex began to slip down the bell toward the margin. (d, i) As relaxation began, the robot shed the starting vortex, and the previous and new stopping vortices merged to form one stopping vortex. (e, j) The next cycle began with the formation of new initial starting vortex.

As seen in figure 7, the robot shed two starting vortices and one stopping vortex when actuated at a moderate pace. Interestingly, the stopping vortex did not dissipate at the end of each cycle, but rather it was absorbed into the next stopping vortex at the start of each actuation cycle. The pace of actuation of the jellyfish robot was fast to obtain 1-1 OPV (approximately 1.2 seconds per cycle, moderate to obtain 2-1 OPV (approximately 2.0 seconds per cycle), and slow to obtain 2-2 OPV (approximately 3.2 seconds per cycle). These gaits use the same three speeds as the displacement and velocity plots in Figure 5.

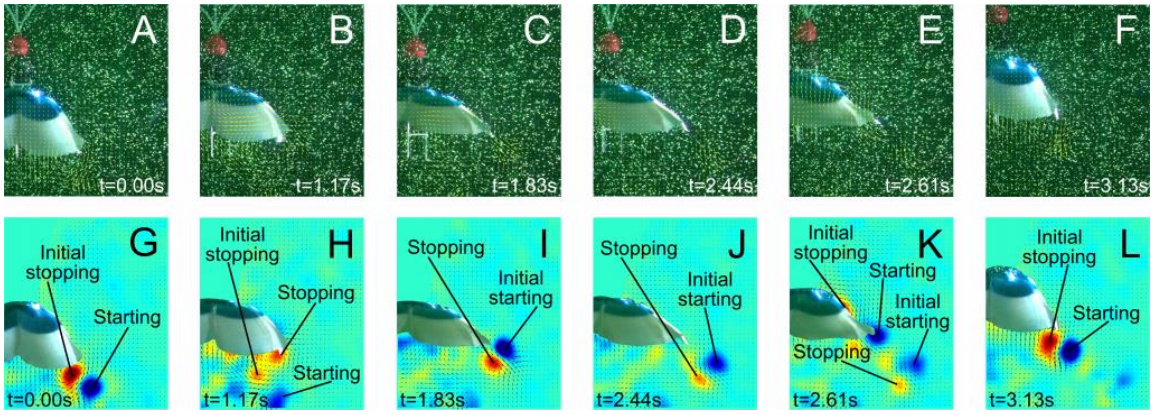


Figure 8: The vorticity associated with the fluid flow around the jellyfish robot at various stages in a slow actuation cycle (associated with a 2-2 OPV). PIVlab 1.4 in MATLAB processed the video footage and generated the vorticity plots. (a, g) The jellyfish robot began a cycle by contracting, with a starting vortex shed from the edge of the bell and an initial stopping vortex formed beneath the bell margin. (b, h) The starting vortex weakened and moved downward from the bell, and the stopping vortex formed next to the initial stopping vortex. (c, i) As the robot relaxed, the initial stopping vortex dissipated, and the initial starting vortex formed. The strong initial starting and stopping vortices rotated by the bell margin. (d, j) The robot shed the initial starting and stopping vortices from the bell margin, and they continued to rotate with a lower vorticity. (e, k) The initial starting and stopping vortices began to dissipate prior to the start of the next actuation cycle. The starting vortex formed at the bell margin, and the initial stopping vortex began to form above the bell. (f, l) The next cycle began with the completed formation of new starting and initial stopping vortices.

As seen in figure 8, the robot shed an extra set of stopping and starting vortices when actuated at a slower pace. Soon after the formation of the starting and initial stopping vortices, the robot shed the starting vortex, and the stopping vortex formed upstream of the initial stopping vortex. An initial starting vortex then formed, and the robot shed both the stopping and initial starting vortices, though they did not immediately dissipate. Then, the next cycle began with the formation new starting and initial stopping vortices.

After observing the different vortex formations, we calculated the COT values for the 1-1 OPV, 2-1 OPV, and 2-2 OPV gaits, in order to compare their efficiencies. During the experimental trials, we measured and recorded the pressure in the jellyfish actuator, the volume of water injected into the actuator, and the displacement of the apex of the jellyfish robot. Using this data, we calculated the COT values associated with the three gaits of the jellyfish robot.

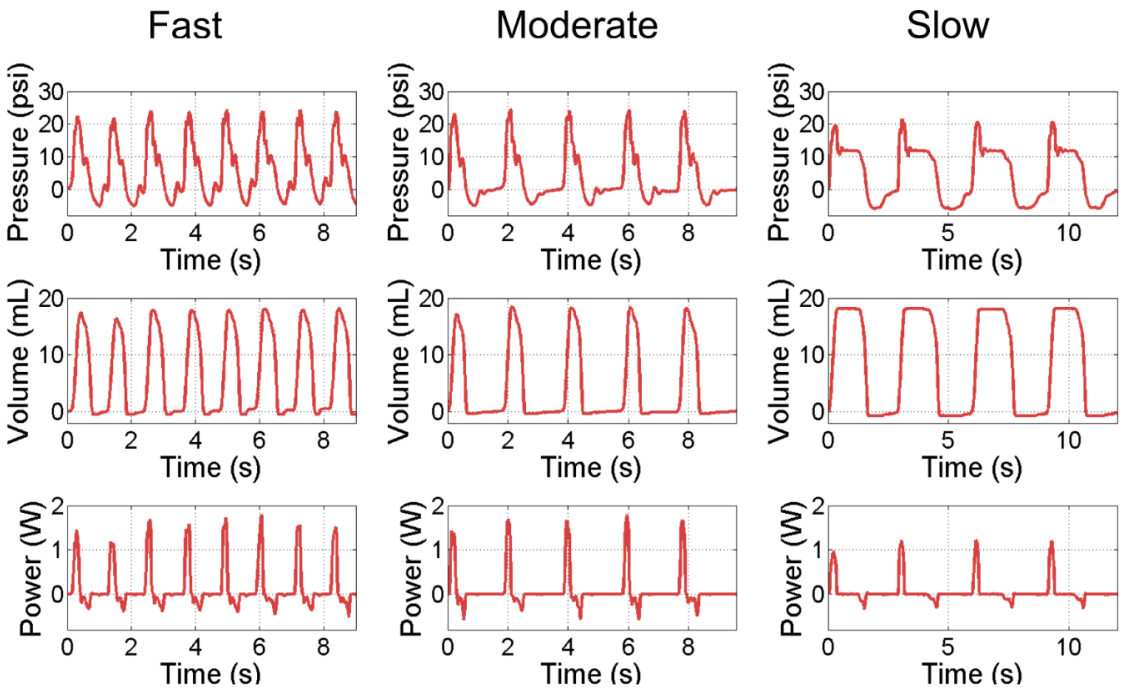


Figure 9: The plots of the pressure and volume values, measured throughout multiple cycles of jellyfish actuation, for each of the three actuation speeds. The COT values for the jellyfish robot were calculated using the data contained in these plots. (a) The fast pace of actuation yielded sharp spikes in pressure, volume, and power values and short pauses between cycles. (b) The moderate pace of actuation yielded similar sharp peaks in pressure, volume, and power values but longer pauses between cycles. (c) The slow pace of actuation yielded more sustained changes in pressure, volume, and power values and long pauses between cycles.

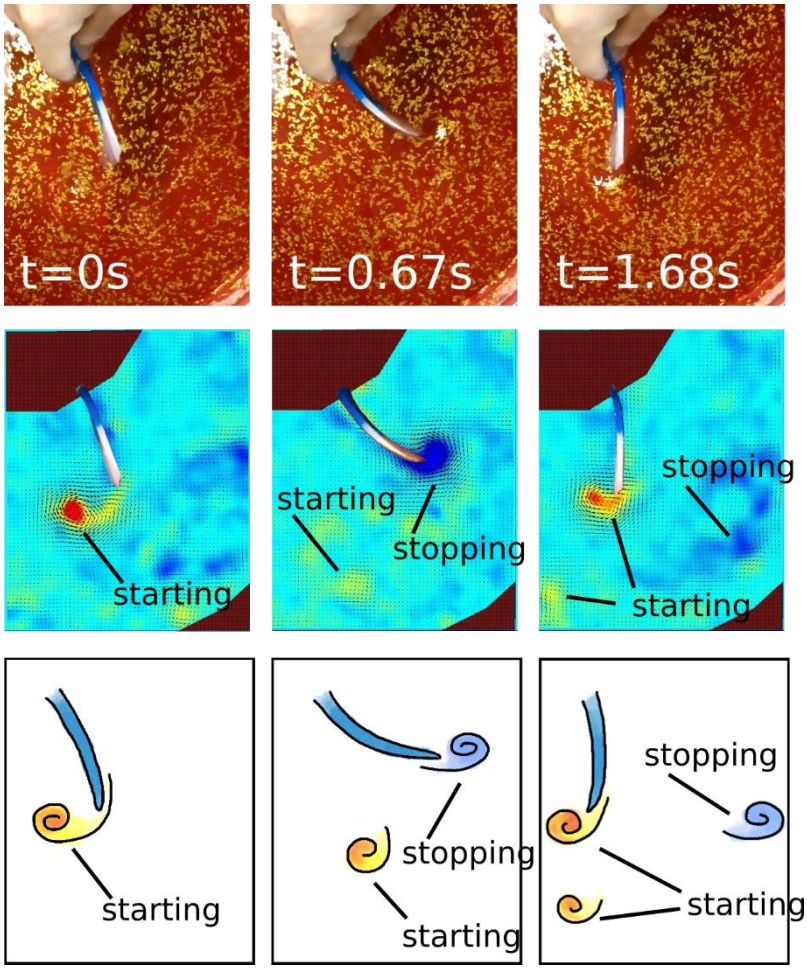
We found the COT values to be 38.49, 20.40, and 18.38 $J/(kg \cdot m)$for the fast, moderate, and slow gait, respectively, corresponding to the total energy consumed. Therefore, the gait with the highest calculated efficiency was the slow gait, and the gait with the lowest calculated efficiency was the fast gait. The efficiency of the moderate gait was in between the other two values. These findings follow logically when made analogous to physical exertion. Sprinting is faster but much more tiring than running, which is faster but more tiring than jogging.

4. Conclusions

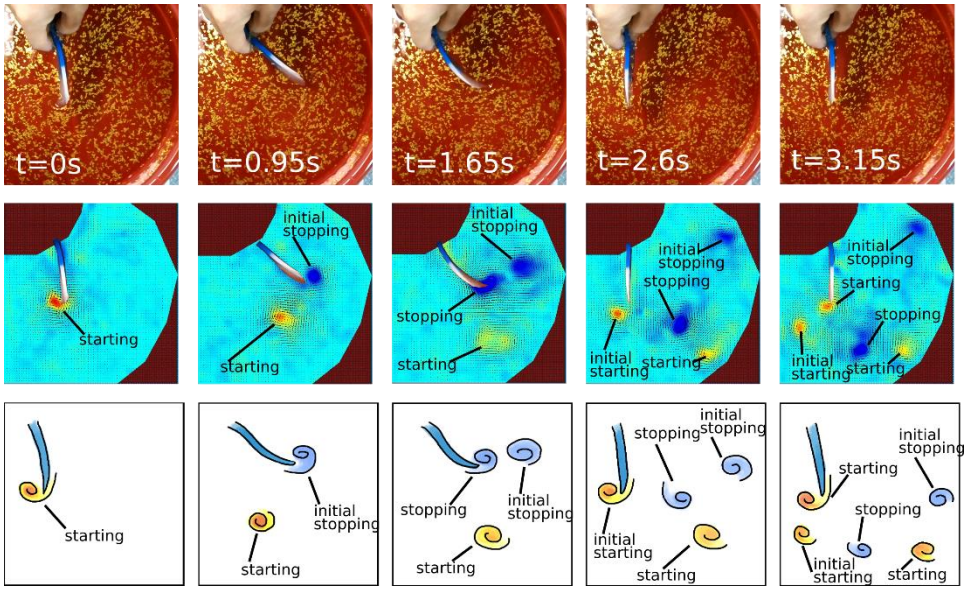
Our jellyfish robot successfully exhibited the PER capability. The design of our soft, biomimetic jellyfish robot closely matched biological jellyfish in order to replicate the natural PER phenomenon. The displacement and velocity plots of our jellyfish robot actuated at moderate and slow speeds had two, distinct periods of acceleration per cycle, which clearly indicated the presence of PER. Notably, the fast gait did not display PER. Analogous to some biological jellyfish (such as *Aurelia aurita*), the efficiency of the jellyfish robot was likely enhanced due to its PER ability. Interestingly, our jellyfish robot was able to demonstrate three different gaits: a fast 1-1 OPV, a moderate 2-1 OPV, and a slow 2-2 OPV. The COT value of the jellyfish was lowest with the 2-2 OPV, followed by the 2-1 OPV, and then the 1-1 OPV, indicating that the jellyfish was most efficient with the slow gait.

In the future, closer examination of the 1-1, 2-1, and 2-2 OPV gaits may yield a deeper understanding of the efficiency-enhancing PER mechanism and how to optimize its performance. Additional gaits can be simulated and tested to find the optimal gaits for both speed and efficiency from all possible actuation speeds. The jellyfish robot can be redesigned to be fully automated, untethered, agile, and more efficient.

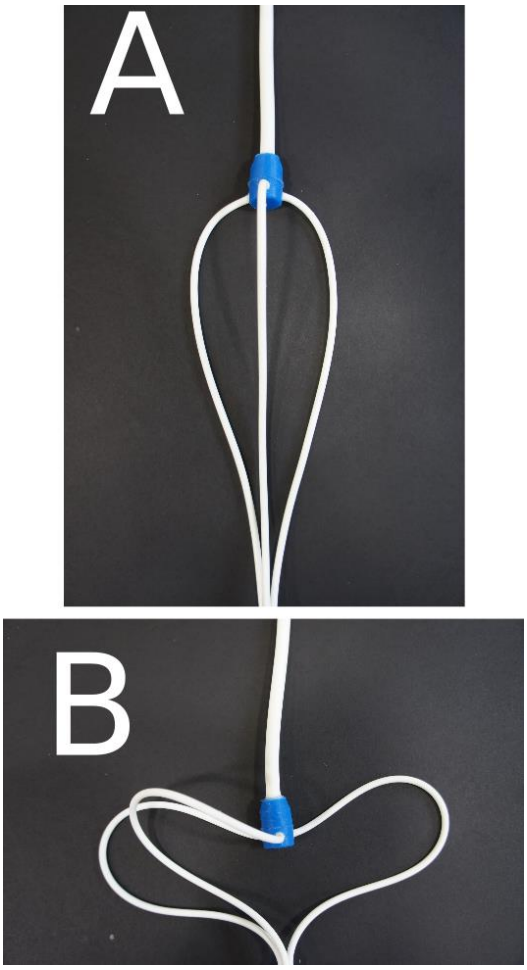
Appendix A: Supplementary Figures



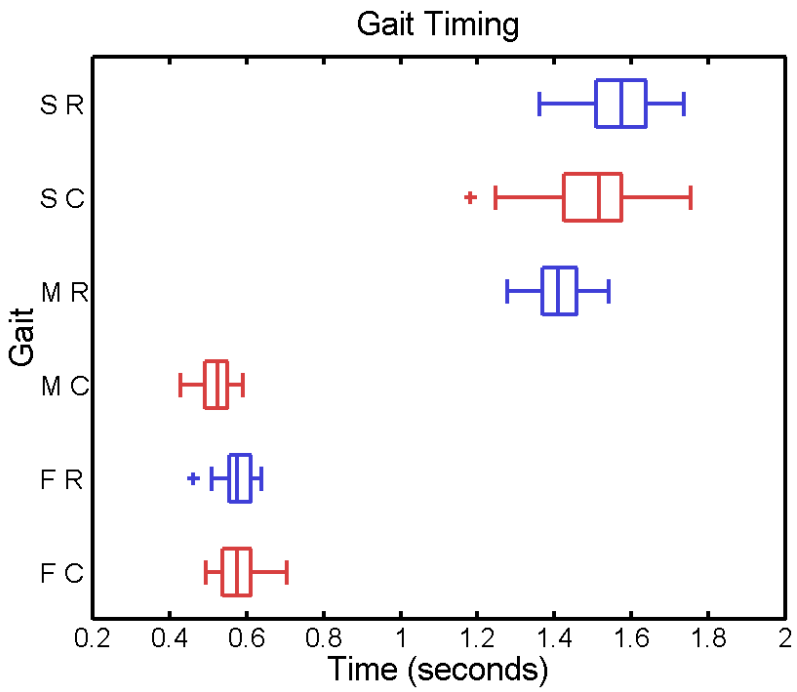
Supplementary Fig. 1: The vortex formation associated with the flapping of a single bell segment at a fast speed. The gait produced a single pair of vortices (one starting and one stopping vortex per cycle), indicative of a 1-1 OPV. The upper row shows frames taken directly from video of the actuation, the middle row shows PIV analysis of the video frames (using PIVlab 1.4 in MATLAB), and the lower row shows a pictorial representation of the vortex formation.



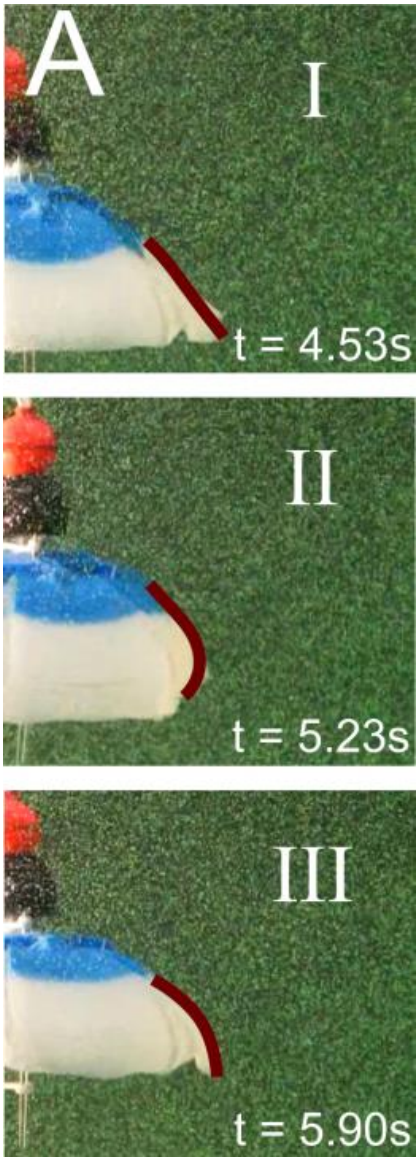
Supplementary Fig. 2: The vortex formation associated with the flapping of a single bell segment at a slow speed. The gait produced two pairs of vortices (starting, initial stopping, stopping, and initial starting) per cycle, indicative of a 2-2 OPV. The upper row shows frames taken directly from video of the actuation, the middle row shows PIV analysis of the video frames (using PIVlab 1.4 in MATLAB), and the lower row shows a pictorial representation of the vortex formation.



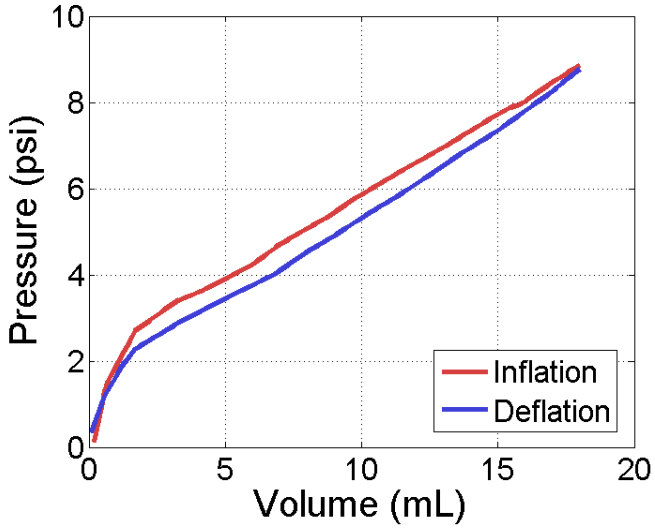
Supplementary Fig. 3: The one-to-three fitting for the rubber tubing allowed water to pass through at a sufficient flow rate, due to its fairly large intersection. The flexibility of the three narrower tubes significantly reduced potential pulling or pushing forces that may have acted on the tethered jellyfish robot.



Supplementary Fig. 4: Six box plots indicate the range of values for the contraction and relaxation time of the three different gaits. The average contraction time for the fast, moderate, and slow gaits, respectively, were 0.574, 0.525, and 1.517 seconds. The average relaxation time for the fast, moderate, and slow gaits, respectively, were 0.574, 1.410, and 1.574 seconds.



Supplementary Fig. 5: This figure corresponded to the color tracking code which produced the displacement and velocity data of the jellyfish. It shows the curvature of the bell margin at various stages of actuation is indicated with the maroon line segments. The jellyfish swam at a moderate pace. The bell tended to bend outward during the contraction stage (end of stage I), to bend inward during the relaxation stage (end of stage II), and to remain in a neutral position during the PER stage (end of stage III).



Supplementary Fig. 5: The plot shows the static pressure in the jellyfish actuator, against the volume of water injected into the actuator. The data was acquired during very slow inflation and deflation of the actuator. The data in this figure was used to calculate the total strain energy consumed within each cycle of propulsion by using the formula: $E_{strain} = \sum P_{static} \Delta V_{static}$.

Appendix B: Supplementary Table

	Dist.(mm)	Total Energy	Total Energy w/o P.L.	Total Energy w/o P.L. w/o S.E.	Positive Energy	Positive Energy w/o P.L.	Positive Energy w/o P.L. w/o S.E.
Energy(fast)	46.82	0.4145	0.4159	0.2149	0.6269	0.4251	0.2241
Energy(mid)	61.49	0.2885	0.2889	0.1549	0.4471	0.3057	0.1717
Energy(slow)	67.66	0.2860	0.2862	0.1522	0.4062	0.2679	0.1339
COT(fast)		38.49	38.62	19.96	58.47	39.48	20.81
COT(mid)		20.40	20.43	10.95	31.61	21.62	12.14
COT(slow)		18.38	18.39	9.78	26.10	17.22	8.60

Supplementary Table: Dist. stands for the distance traveled of the robot. The energy terms represent the energy consumed corresponding to the distance traveled. Total Energy includes the positive and negative energy being input into the robot, while Positive Energy only accounts for the positive. The reason of calculating the Positive Energy values is that even the negative energy represents the energy recovered, however, this portion of energy does not go back to the robot itself.

P.L. stands for pressure loss in the long tethering tube. As inflation and deflation of the actuator of the robot goes on, the high flow rate inside the tube leads to a quite significant pressure loss according to Poiseuille's law, which leads to energy loss.

S.E. stands for strain energy along with the deformation of the large deformation of the body structure and actuator of the robot, and the deformation of the soft tethering tube. Please see Supplementary Fig. 5 for details regarding the strain energy.

Appendix C: List of supplementary Movies

Supplementary Movie 1 PIV result displaying velocity vector field for 1-1 OPV (fast)

Supplementary Movie 2 PIV result displaying vorticity for 1-1 OPV (fast)

Supplementary Movie 3 PIV result displaying velocity vector field for 2-1 OPV (moderate)

Supplementary Movie 4 PIV result displaying vorticity for 2-1 OPV (moderate)

Supplementary Movie 5 PIV result displaying velocity vector field for 2-2 OPV (slow)

Supplementary Movie 6 PIV result displaying vorticity for 2-2 OPV (slow)

Supplementary Movie 7 PIV result displaying velocity vector field for 1-1 OPV (single fin)

Supplementary Movie 8 PIV result displaying vorticity for 1-1 OPV (single fin)

Supplementary Movie 9 PIV result displaying velocity vector field for 2-2 OPV (single fin)

Supplementary Movie 10 PIV result displaying vorticity for 2-2 OPV (single fin)

Appendix D: Matlab Code

Matlab code for acquiring the displacement data by using ginput

(The code below is written for the fast gait, but the code for moderate and slow gait is similar)

```
% Read, show and process videos
movobj=VideoReader('VideoName.avi');
nFrames = movobj.NumberofFrames;
Duration = movobj.Duration;
vidHeight = movobj.Height;
vidWidth = movobj.Width;
vidFPS = movobj.FrameRate;
x=[];
y=[];
frames=[];


for i=1:1:nFrames
    this_frame= read(movobj,i);
%    imshow(this_frame)
    filename=sprintf('Frame_%03d',i);
    t=Tiff([filename '.tif'],'w');
    tagstruct.ImageLength = size(this_frame,1);
    tagstruct.ImageWidth = size(this_frame,2);
    tagstruct.Photometric = Tiff.Photometric.RGB;
    tagstruct.BitsPerSample = 8;
    tagstruct.SamplesPerPixel = 3;
    tagstruct.RowsPerStrip = 16;
```

```
    tagstruct.PlanarConfiguration = Tiff.PlanarConfiguration.Chunky;
    tagstruct.Software = 'MATLAB';
    t.setTag(tagstruct);
    t.write(this_frame);
    t.close;
    [x0,y0]=ginput(1);
    x=[x;x0];
    y=[y;y0];
    frames=[frames;i];
    close all
end
```

Calculate the velocity and plot the displacement and velocity

(The code below is written for the fast gait, but the code for moderate and slow gait is similar)

```
%% calculating the overall running time
f = 1:1001;
f(852:end) = [];    f(1:292) = [];
t_min=f(1).*0.0163;     t_max=f(end).*0.0163;
t_max=t_max - t_min;
%% crop the raw velocity data
% vel(280:564)=[]; vel(1:90)=[];
% vel_s=smooth(y1, 15);


%% crop the raw displacement data
% disp1=disp;
% disp1(280:565)=[]; disp1(1:90)=[];
```

```
%% change unit
disp = disp.*100./353;              % change from pixel to mm
vel1 = vel.*(61.35).*100./353;            % change from pixel/frame to
mm/s


%% get the x axis data (time), and reset the beginning of displacement
to zero
d=size(vel1);
d=d(1);
t=linspace(0, t_max, d);


d1=disp(1);
for i=1:d
    disp(i)=disp(i)-d1;
end
%% make the plots
subplot(211);
plot(t,disp,'LineWidth',3,'color',[0.85 0.25 0.25]);
xlabel('Time (s)','FontSize',20);
ylabel('Displacement (mm)','FontSize',20);
title('Displacement vs. Time','FontSize',20);
axis([0 9 -3 125]);                     % set the axis limits
set(gca,'FontSize',14)
grid on



subplot(212);
```

```
plot(t,vel1,'LineWidth',3,'color',[0.85 0.25 0.25]);
xlabel('Time (s)','FontSize',20);
ylabel('Velocity (mm/s)','FontSize',20);
title('Velocity vs. Time','FontSize',20);
axis([0 9 -40 85]);                    % set the axis limits

set(gca,'FontSize',14)
grid on
```

Plotting the pressure, volume data; calculating and plotting the power consumed

(The code below is written for the fast gait, but the code for moderate and slow gait is similar)

```
%% type the code below in command window
% p_v = 0;
% % then import the p-v data from txt file
% % then crop the data, from 1 to 289, and then only use 536 data
points
% p_v(1:289,:) = [];
% % (851 - 293) * 0.0163 / 0.017 = 536
% p_v(537:end,:) = [];
% get rid of the 3rd column
% p_v(:,3) = [];

%% now we should have already got a well cropped "p_v" data set
p = p_v(:,1);
p1 = (p - 510) .* 15 ./ (727 - 510);        % calibrating pressure
values
```

```
p1 = p1 - p1(1);


v = p_v(:,2);
v1 = v .* 0.02;     % calibrating volume values
v1 = v1(1) - v1;      % v1 is to be printed


s = size(p);
s = s(1);
t_max = (s - 1) * 0.017;
t = linspace(0,t_max, 536);


%% make the plots for P and V
subplot(331);
plot(t,p1,'LineWidth',3,'color',[0.85 0.25 0.25]);
xlabel('Time (s)','FontSize',25);
ylabel('Pressure (psi)','FontSize',25);
% title('Pressure vs. Time','FontSize',25);
axis([0 9 -8 30]);                    % set the axis limits
set(gca,'FontSize',20)
grid on



subplot(334);
plot(t,v1,'LineWidth',3,'color',[0.85 0.25 0.25]);
xlabel('Time (s)','FontSize',25);
ylabel('Volume (mL)','FontSize',25);
% title('Volume vs. Time','FontSize',25);
axis([0 9 -2 20]);                    % set the axis limits
```

```
set(gca,'FontSize',20)
grid on

%% calculating delta_v
s2 = size(p1);
s2 = s2(1) - 1;
delta_v = zeros(s2,1);
for i = 1:1:(s2)
    delta_v(i) = v1(i+1) - v1(i);
end

t(end) = [];
%% calculate the work input
w = p(1:s2) .* delta_v(1:s2);
% change unit
w = w / 1450;

%% make the plots for p.*delta_v

subplot(337);
plot(t,w,'LineWidth',3,'color',[0.85 0.25 0.25]);
xlabel('Time (s)','FontSize',25);
ylabel('Energy (J)','FontSize',25);
% title('Energy vs. Time','FontSize',25);
axis([0 9 -1.5 1.5]);                    % set the axis limits
```

```
set(gca,'FontSize',20)
grid on
```

Plotting the static pressure against volume, in order to calculate the total strain energy consumed

```
% Static P V plot
%% inflation data
p_raw_infla = static_p_v_infla(1:10:end,1);
v_raw_infla = static_p_v_infla(1:10:end,2);
%% interpolate the relationship between static pressure and volume
% unit conversion for pressure
p_raw_infla = (p_raw_infla - 510) .* 15 ./ 217;
% unit conversion for volume
v_raw_infla = v_raw_infla ./ 50;
% interpolate
v_infla = 0:0.1:18;
p_infla = interp1(v_raw_infla, p_raw_infla, v_infla);
% plot(v_raw_infla, p_raw_infla, 'o', v_infla, p_infla, ':.');
% xlim([0 20]);


%% deflation data
p_raw_defla = static_p_v_defla(1:10:end,1);
v_raw_defla = static_p_v_defla(1:10:end,2);
%% interpolate the relationship between static pressure and volume
% unit conversion for pressure
p_raw_defla = (p_raw_defla - 510) .* 15 ./ 217;
% unit conversion for volume
v_raw_defla = v_raw_defla ./ 50;
```

```
% interpolate
v_defla = 0:0.1:18;
p_defla = interp1(v_raw_defla, p_raw_defla, v_defla);
% plot(v_raw_defla, p_raw_defla, 'o', v_defla, p_defla, ':.');
% xlim([0 20]);


%% plot
figure
plot(v_infla, p_infla,'LineWidth',3.5,'color',[0.85 0.25 0.25]);
hold on
plot(v_defla, p_defla,'LineWidth',3.5,'color',[0.25 0.25 0.85]);
legend('Inflation','Deflation')
xlabel('Volume (mL)','FontSize',25);
ylabel('Pressure (psi)','FontSize',25);
% title('Static Pressure vs. Volume','FontSize',25);
xlim([0 20]);
set(gca,'FontSize',20)
grid on
```

Creating supplementary movies with frames

```
%Write Video from Picture Files
vid=VideoWriter('PIV11OPV.avi');


vid.FrameRate=15;


open(vid);
```

```
for k=1:125
    pic=sprintf('PIVlab_out_%03d.jpg',k);
    frame=imread(pic);
    writeVideo(vid,frame);
end


close(vid);
```

Appendix E: Arduino Code

Manual control of the jellyfish robot and record the pressure and volume data at the same time

```
int pressPin = 7;          // analog pin, for pressure

int potPin = 0;            // analog pin, for linear potentiometer

int buttonPin = 12;        // digital pin

int ledPin = 7;            // digital pin



int pressure;

int potVal;                // value of potentiometer

int ledState;



void setup() {

  Serial.begin(9600);

  Serial.print("pressure");

  Serial.print("\t");

  Serial.print("potentiometer");

  Serial.print("\t");

  Serial.println("ledState");
```

```
  pinMode(buttonPin,INPUT);
  pinMode(ledPin,OUTPUT);
}

void loop() {

  if(digitalRead(buttonPin)==HIGH){
    digitalWrite(ledPin,HIGH);
    ledState = 1;
  }
  else{
    digitalWrite(ledPin,LOW);
    ledState = 0;
  }

  pressure = analogRead(pressPin);
  potVal = analogRead(potPin);
```

```
  Serial.print(pressure);
  Serial.print("\t");
  Serial.print(potVal);
  Serial.print("\t");
  Serial.println(ledState);

  delay(17);
}
```

Metronome for manual control

```
int speakerPin = 9;

int tones[] = {261, 277, 294, 311, 330, 349, 370, 392, 415, 440};
//          mid C   C#   D    D#   E    F    F#   G    G#   A

void setup(){
```

```
}


void loop(){
  tone(speakerPin, 440);          // pick a note
  delay(100);                     // the inflation timing in milli-sec
  noTone(speakerPin);                 // stop the buzzer
  delay(400);
  tone(speakerPin, 261);          // pick another note
  delay(100);                     // the deflation timing in milli-sec
  noTone(speakerPin);                 // stop the buzzer
  delay(1400);
}
```

Recording the data of static pressure and corresponding volume

```
int pressPin = A7;    // Analog pin for pressure sensor
int pressVal = 0;     // variable to store the pressure value


int potPin = A0;      // Analog pin for linear potentiometer
```

```
int potVal = 0;       // variable to store the linear potentiometer's value


void setup() {
  Serial.begin(9600);
  Serial.print("pressure");
  Serial.print("\t");
  Serial.println("potentiometer");
}


void loop() {
  pressVal = analogRead(pressPin);
  potVal = analogRead(potPin);
  Serial.print(pressVal);
  Serial.print("\t");
  Serial.println(potVal);
  delay(500);
}
```

Appendix F: Formula used in this work

Calculate velocity with displacement

$$v_k = (D_k - D_{k-1})/\Delta t$$

Finite energy difference

$$\Delta E_k = P_k \Delta V_k = P_k (V_k - V_{k-1})$$

Energy

$$E = \sum \Delta E_k$$

Pressure loss in the tube (Hagen-Poiseuille equation)

$$P_{loss_k} = \frac{128 \mu L Q_k}{\pi d^4} = \frac{128 \mu L}{\pi d^4} \frac{\Delta V_k}{\Delta t} = \frac{128 \mu L}{\pi d^4 \Delta t} \Delta V_k = k_{loss} \Delta V_k$$

Energy consumed without pressure lost in the tube

$$E' = \sum \Delta E_k' = \sum (P_k \Delta V_k - P_{loss_k} \Delta V_k) = \sum (P_k \Delta V_k - k_{loss} \Delta V_k^2)$$

Total strain energy

$$E_{strain} = \sum P_{static} \Delta V_{static}$$

Energy consumed without pressure lost in the tube, and without strain energy lost in the body structure of the robot

$$E'' = E' - E_{strain}$$

References

[1] B. J. Gemmell, J. H. Costello, S. P. Colin, C. J. Stewart, J. O. Dabiri, D. Tafti, S. Priya, *Proc. Natl. Acad. Sci.* **2013**, *110*, 17904.
[2] T. L. Daniel, *Can. J. Zool.* **1983**, *61*, 1406.
[3] T. L. Daniel, *J. Exp. Biol.* **1985**, *119*, 149.
[4] L. Kozlowski, Jellyfish a model of efficiency. *Los Angel. Times* **2010**.
[5] J. C. Nawroth, H. Lee, A. W. Feinberg, C. M. Ripplinger, M. L. McCain, A. Grosberg, J. O. Dabiri, K. K. Parker, *Nat. Biotechnol.* **2012**, *30*, 792.
[6] A. Villanueva, C. Smith, S. Priya, *Bioinspir. Biomim.* **2011**, *6*, 036004.
[7] J. O. Dabiri, S. P. Colin, J. H. Costello, M. Gharib, *J. Exp. Biol.* **2005**, *208*, 1257.
[8] J. O. Dabiri, M. Gharib, *J. Fluid Mech.* **2004**, *511*, 311.
[9] J. O. Dabiri, M. Gharib, *Proc. R. Soc. B Biol. Sci.* **2005**, *272*, 1557.
[10] J. O. Dabiri, S. P. Colin, J. H. Costello, *J. Exp. Biol.* **2006**, *209*, 2025.
[11] J. O. Dabiri, *Annu. Rev. Fluid Mech.* **2009**, *41*, 17.
[12] J. O. Dabiri, S. P. Colin, K. Katija, J. H. Costello, *J. Exp. Biol.* **2010**, *213*, 1217.
[13] F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, G. M. Whitesides, *Angew. Chem.* **2011**, *123*, 1930.
[14] R. F. Shepherd, F. Ilievski, W. Choi, S. A. Morin, A. A. Stokes, A. D. Mazzeo, X. Chen, M. Wang, G. M. Whitesides, *Proc. Natl. Acad. Sci.* **2011**, *108*, 20400.
[15] R. F. Shepherd, A. A. Stokes, J. Freake, J. Barber, P. W. Snyder, A. D. Mazzeo, L. Cademartiri, S. A. Morin, G. M. Whitesides, *Angew. Chem. Int. Ed.* **2013**, *52*, 2892.
[16] A. D. Mazzeo, M. E. Lustrino, D. E. Hardt, *Polym. Eng. Sci.* **2012**, *52*, 80.
[17] A. D. Mazzeo, D. E. Hardt, *J. Micro Nano-Manuf.* **2013**, *1*, 021001.
[18] W. Thielicke, E. J. Stamhuis, *J. Open Res. Softw.* **2014**, *2*.
[19] B. J. Gemmell, J. H. Costello, S. P. Colin, *Commun. Integr. Biol.* **2014**, *7*, e29014.