Engineering Automated Microrobotic Systems with Machine Learning-Based Control for Biomedical Applications

by

Michael Richard Behrens

B.S.E., Walla Walla University, 2016

Submitted to the Graduate Faculty of the

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2021

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Michael Richard Behrens

It was defended on

November 9, 2021

and approved by

Dr. Lance Davidson, William Kepler Whiteford Professor, Department of Bioengineering

Dr. Fabrisia Ambrosio, Associate Professor, Department of Physical Medicine and Rehabilitation

Dr. Douglas Weber, Akhtar and Bhutta Professor, Department of Mechanical Engineering, Carnegie Mellon University

Dissertation Director: Dr. Warren Ruder, Associate Professor and William Kepler Whiteford Faculty Fellow, Department of Bioengineering

Engineering Automated Microrobotic Systems with Machine Learning-Based Control for Biomedical Applications

Michael Richard Behrens

University of Pittsburgh, 2021

For many diseases and injuries, the pathological effects are local to particular areas of the body. In such cases, systemic drug delivery is nonideal treatment because drugs used to treat the pathological condition may have a deleterious effect on the remaining healthy tissue and cause damaging side effects. A similar kind of harm is done in many surgical scenarios when healthy tissue must be cut in order to give the surgeon physical access to the damaged tissue underneath. There exists an unmet clinical need to be able to deliver minimally invasive therapeutic interventions precisely within the body, while minimizing harm to healthy tissue. Towards this end, we are developing swimming microrobotic systems to traverse the fluidic pathways within the body, in order to deliver locally targeted therapeutic payloads noninvasively, and with precision, to areas which are difficult to access noninvasively using current methods.

In this dissertation, I present a control approach which uses machine learning in order to develop robust control policies for magnetically actuated microrobots navigating within small scale fluidic environments. We have developed a controller for a helical magnetic hydrogel microrobot that uses the soft actor critic reinforcement learning algorithm to autonomously derive control policies which allow the microrobot to swim through an uncharacterized biomimetic fluidic environment, actuated by time varying magnetic fields generated from a three-axis array of electromagnets. Our results show that this model-free control approach can learn and adapt to the dynamic features of the microrobot, the fluidic environment, and the electromagnetic actuator in order to successfully accomplish navigation tasks. Additionally, we demonstrate that the control policies learned by the reinforcement learning algorithm repeatedly converge to recapitulate the behavior of rationally designed controllers based on physical models of helical swimming microrobots. We also present an approach for developing biohybrid magnetic microrobots with chemical sensing capabilities based on genetically encoded surface chemistry, and explore hydrogel based magnetic microrobots as vehicles for antibiotic delivery. Together, these contributions form a suite of microrobotic technologies for autonomous navigation, sensing, and localized drug delivery within small scale fluidic environments, and are an important step towards clinical applications of microrobotic technology.

Table of Contents

List of Tables

List of Figures

Acknowledgements

This dissertation is dedicated to my dad, Dr. Richard T. Behrens (1963-2020) who continues to inspire me with his playful, inquisitive, and imaginative spirit every day.

I would like to gratefully acknowledge the unyielding support and guidance I have received from my mentor, Professor Warren Ruder. It is only through his guidance and support that I was able to begin, and to complete this research. I would also like to acknowledge the guidance I have received from the other members of my committee, Professor Lance Davidson, Professor Fabrisia Ambrosio, and Professor Doug Weber, for their willingness to help guide me through the completion of my research.

I would also like to thank the other members of my lab who have inspired me, collaborated with me, laughed with me, and commiserated with me over the last five years, including Felicia Scott, Keith Heyde, MJ Rice, Haley Fuller, Ting-Yen Wei, and all the others who have helped me along the way to completing this research.

I would like to thank my fiancé Samantha, for her love and for her patience as I have completed this work. I cannot imagine having done it without her support.

I would also like to thank all my family and friends for everything leading up to this moment, my mom Debbie and dad Rick, my sister Julie, and everyone else, too many to name, who helped me along the way.

I also acknowledge financial support from the Air Force Office of Scientific Research, the Office of Naval Research, the National Institutes of Health and the CATER training program.

1.0 Introduction

Advancements in biomedical technology are one of the primary drivers of improved health and longevity in modern society, driving increases in quality of life. However, the iatrogenic effects of medical treatment (that is, the unintended side effects and harms caused by the application of medicine) pose a significant burden on patients and society at large [1, 2]. Patients suffer from unintended side effects of treatments that are too often imprecise in their anatomical targeting and physiological effects. These harmful effects are often exacerbated by the most common drug delivery strategies, which systemically administer to the drug to perfuse throughout the whole body [3, 4]. When drugs are applied systemically, achieving an effective concentration of the drug at the target site, which is often a small region of the body such as a single organ or tissue, means that the concentration of the drug in the other systems of the body will also be high. While unhealthy tissues exposed to drugs can benefit from the effects of the drug, healthy tissues exposed to drugs rarely become healthier upon exposure to the drug. The net effect of the drug may be to improve the pathological condition which it is targeting, but this is often offset by unwanted and sometimes very severe side effects for the patient. As a salient example of this principle, consider the effect of anticancer chemotherapeutic drugs on healthy cells in the body. Chemotherapeutic drugs can be broadly cytotoxic, leading to cell death [5]. Cell death is the treatment goal when the cell exposed to the chemotherapeutic drug is a cancerous cell, but for noncancerous cells which come into contact with the drug, the results can be quite destructive [6]. In patients undergoing chemotherapy, the systemic shock of these drugs often can leave the patient feeling worse after the treatment than before, as healthy cells suffer because the chemotherapeutic drugs were not precisely directed to the cancerous cells which they are targeting.

Another major class of iatrogenic harms to patients occurs during surgical operations, in which the harms to the patient can result from mechanical insult to healthy tissue [7]. This is clearly the case when healthy skin and connective tissue is cut by the surgeon's scalpel while the surgeon is carving an access path to the site of an internal problem area, such as a torn muscle, a ruptured appendix, or a tumor to be excised. Clearly, there exists room for improvement in modern medical practice, both by increasing the effectiveness of therapeutic treatments, and at decreasing the unwanted side effects of the treatment.

The field of minimally invasive medicine seeks to minimize iatrogenic harms by developing surgical systems that allow the surgeon access to the site of injury by way of small incisions in procedures such as laparoscopic surgery or heart catheterization. Minimally invasive medical procedures can also be aided by robotic surgical aids, using precise motion control technology and advanced imaging systems to allow doctors to do more with less harm [8]. The proliferation of minimally invasive medical procedures is a boon to patients and to the healthcare system supporting them, reducing pain and discomfort on the part of the patients, and reducing financial burden on the healthcare system, which does not have to spend the additional resources to treat the side effects of the medical procedure [2].

To illustrate one area of medical science in which minimally invasive protocols could make a significant difference in the quality of treatment and reduction of harm, consider the broad category of treatments for neurological disorders. Despite significant investment, neurological disorders remain a leading cause of death in the United States [9]. One reason that addressing neurological disorders can be difficult is that the nervous system is protected by numerous physical and biochemical barriers that isolate it from the outside world and from the rest of the body. The brain is protected from without by multi-layered protection including the dura and pia mater, the cerebrospinal fluid, the skull, and the integumentary system on top of that. From within, the brain is also biochemically isolated from the rest of the body by the blood brain barrier (BBB), which tightly regulates the passage of material from the bloodstream into the brain, and excludes many drugs which can pass easily from the bloodstream into the other systems of the body [10]. This means that for many neurological conditions, not only is systemic drug delivery not ideal, but it is also often not possible because of the protected nature of neurological tissue. There is an urgent need to create a robust approach that can safely direct pharmaceutical drugs within the cerebrospinal fluid environment of the central nervous system, because once we have this ability, clinicians will be able to customize drug treatments to release at precise locations in the CNS with defined local dosage and exposure times.

The field of microrobotics shows great potential for increasing targeting specificity for drug delivery and reducing the tissue insult sustained during surgical procedures. Untethered small-scale swimming robots are being developed to travel in constrained spaces within the body which are difficult to access by conventional surgical means [11-13]. Microrobots designed to operate within the cerebrospinal fluid that could be injected intrathecally and wirelessly directed to sites of tumors, lesions, or infections in the CNS, could dramatically change the way we treat conditions including brain tumors [14], and Parkinson’s disease [15]. A significant body of research has been established exploring the possible forms and functions of clinical microrobots, which has largely converged on externally generated magnetic fields for control and power transmission [16-24]. The magnetically-powered locomotion motifs that have emerged include rolling along surfaces [25, 26], pulling by magnetic gradients [27], and helical swimming [28-30]. Therapeutic tasks for which microrobots have been evaluated include drug delivery [20], cell delivery [29, 30],delivery of guide wires [31], microinjections [24], and biofilm removal [32].

Despite this breadth of research, many microrobots have never been evaluated in a sufficiently biomimetic *in vitro* experimental platform to justify their use *in vivo* [20, 24, 30]. Furthermore, the design-build-test cycle of microrobot development suffers from several bottlenecks, such as the necessity of an experienced human operator for steering the microrobot to the target site when no automation is used in the control system [33]. Even when automation is designed into the control system using classical control methods, accurate dynamic models for the microrobot and its external environment must be derived for each new robot design, and often fail to capture the diversity of physical conditions within the body in which the robot could be expected to operate. These are unaddressed problems which will significantly slow the translation of microrobots to the clinic until they are addressed.

In this dissertation I will present our research focused on expanding the capabilities or microrobotic systems, particularly emphasizing controllability and testing in biomimetic *in vitro* systems, which will enable accelerated research and development efforts to bring microrobotic therapeutic applications to clinical use.

1.1 Dissertation Contributions

Chapter 2 presents a broad survey of the microrobotic research landscape and highlights how our research fits into the broader scope of the field.

Chapter 3 reports on our development of smart microrobotic systems, based on machine learning for model-free feedback control and navigation of microrobots in biomimetic *in vitro* environments. These smart microrobots can autonomously explore their environments and optimize their control policies in order to effectively navigate through fluidic networks.

Chapter 4 reports our development of a low-cost 3D-printed programmable microfluidic pump suitable for use in accurately mimicking the fluid dynamic conditions *in vivo* which would be encountered by a microrobot in the body. This pump is also broadly applicable for expanding access to microfluidic technologies in resource constrained settings.

Chapter 5 discusses the use of hydrogel-based magnetic microrobots as drug delivery vehicles and explores the use of these robots for delivering targeted antibiotics to spatially localized subpopulations of bacteria. This chapter also discusses the potential for using synthetic biology for expanding the therapeutic applications of magnetic microrobots by hybridizing living cells with magnetic particles via genetic modification of cell surface chemistry.

Chapter 6 presents a conclusion and discussion of the outlook and future research directions of the technology developed and presented in this dissertation.

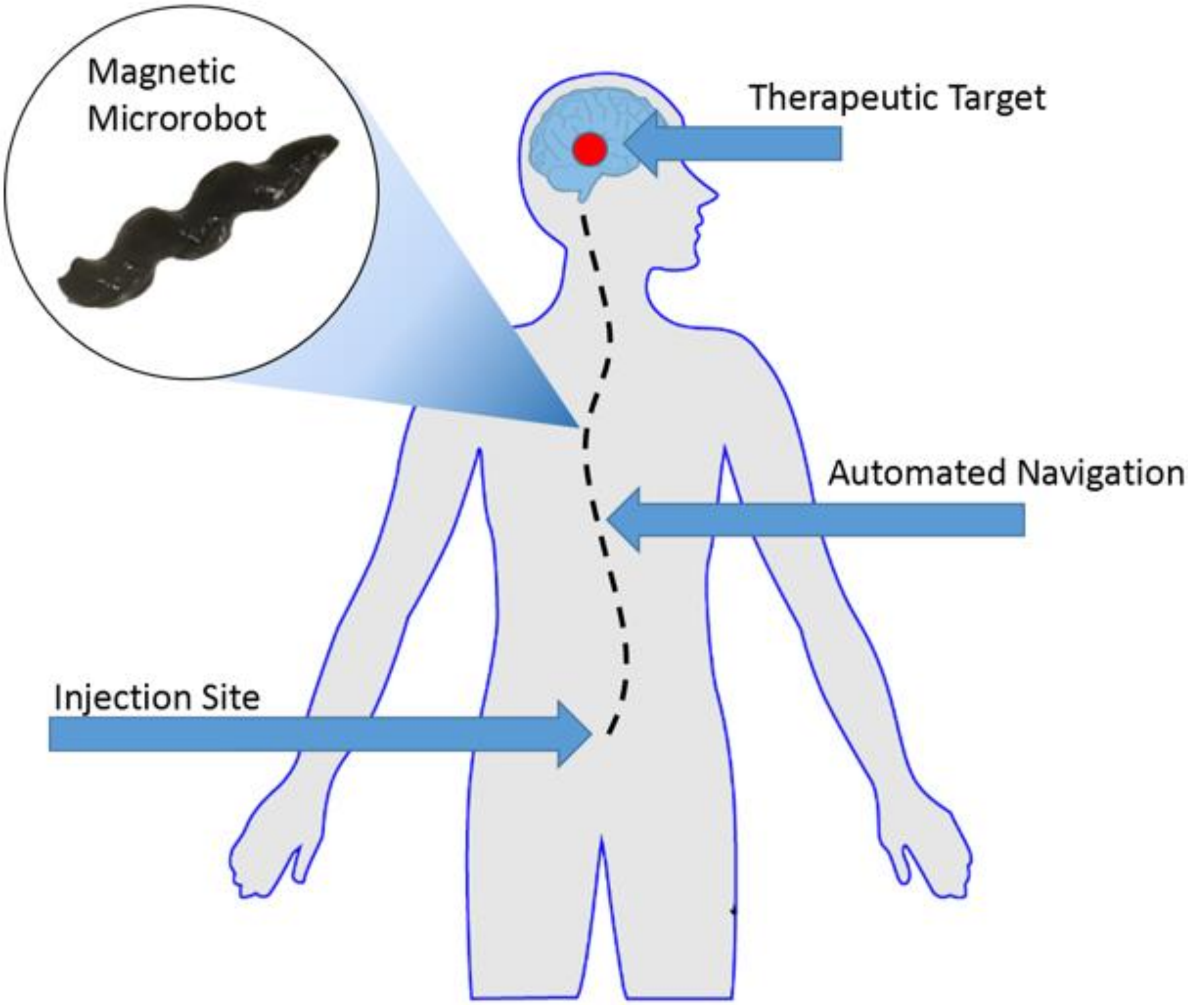


Figure 1-1. Microrobots as mobile theraputic medical devivces.

Completion of this research is an important step towards developing improved drug delivery systems by enabling delivery within the body precisely where it is needed, while minimizing the impact of surgical and drug related side effects and complications. Once robust methods of microrobot delivery and therapeutic function are established, microrobotic technology is likely to be impactful in many fields of medicine, particularly for treating difficult to access regions of the body such as the central nervous system.

2.0 Overview of Microrobotic Technology

Microrobotic systems first entered the popular consciousness through the visionary lectures of Richard Feynman, and through science fiction stories such as Fantastic Voyage [34]. In that 1960s film, a team of scientists is shrunk down in a submarine and sent on a mission into the body of a human scientist in order to repair damage to his brain. This film has apparently made such an impact on the researchers studying microrobots, that in a recent microrobotics focused conference, more than half of the talks at the conference began by pointing out the fact that the field was started by this film. Professor Feynman and his lecture “There’s plenty of room at the bottom” was implicated in probably another third of the introductions. While Fantastic Voyage is primarily an amusing and fanciful story, it did paint a useful picture of some of the difficulties that would be encountered by a swimming microscale device that was trying to operate in the body, including navigation difficulties, loss of communications with the support systems outside the body, and hostile encounters with elements of the immune system. These are some of the difficult challenges that real microrobots which are called upon to operate in the in vivo environment must reckon with, which will only be overcome with sophisticated engineering efforts.

There are a broad range of small-scale robotic devices that have been developed and presented in the literature, ranging from nanoscale molecular robotic systems [35] to milli-scale integrated MEMS robots [36]. What unites them is the ability of the systems to perform movements and actions at small scales, with varying degrees of autonomy. Some microrobotic systems incorporate sensors and actuators into the microrobot and allow the microrobot to perform perception-action sequences [37], but in most cases, the sensors, processors, and actuators of the microrobot are not microscale, but are composed of full-size computers, cameras, microscopes, electromagnetic coils, etc. [13, 38]. These elements are incorporated into microrobotic systems in order to facilitate precise movement and actuation of small scall untethered devices, as well as permit feedback control.

At this point I will make a note about the nomenclature used in this dissertation. In traditional macroscale robotics, the entire robotic system including the mobile microdevice, the sensors, the actuators, the power supply, and the processing units might be collectively referred to as the “robot”, and the small mobile element commonly called a “microrobot” would simply the end effector of the device, a subcomponent of the whole robotic system. However, it is standard practice in the field of microrobotics research to refer to the small-scale mobile element alone as a microrobot, without reference to and regardless to the size of the external hardware that enables the full system to function [39]. I shall be following the convention of the microrobotics field and referring only to the small, mobile element as a microrobot, even if the whole robotic system includes other elements. Additionally, I shall refer to all small-scale robotic systems presented in this dissertation as microrobots, whether or not all the relevant spatial dimensions of the robotic system are strictly less than 1 mm in length. For our purposes, if it looks like a microrobot, and swims like a microrobot in a low Reynolds number environment where viscous forces dominate, it will be referred to as microrobot.

The most common design feature, shared amongst most systems which are called microrobots, is mobility: the ability for the microrobot to actively move through its environment [11]. Microrobots can either contain their own on-board power source for generating motion, or they can harness energy from their environment and convert it into motion.

Microrobots driven by their own onboard power supply have been developed by incorporating living cells with active swimming elements, such as bacteria [40] or sperm with flagellar motors [41] into the structure of the microrobot [42]. Not only can living cells incorporated into a microrobot enable swimming behavior, but the cells can also act as sensors and information processing centers on the microrobot [43]. We will explore the possible applications of the exciting class of hybrid biological microrobots more in chapter 5.

Chemical energy from the environment can also be harvested by microrobots to generate motion. Janus particles are a common microrobot class which uses chemical fuel in the environment to generate motion. These particles use spatially patterned catalysts on a single hemisphere to interact with chemical fuel present in the environment (often H_2O_2) [44]. The fuel is converted by the catalytic element on one half of the particle into H_2O and O_2, and the gradient of O_2 bubbles created on one side of the particle push the particle forward. Other microrobotic systems have been developed which use surface bound enzymes such as urease or glucose oxidase to convert more biologically compatible chemical fuels, such as glucose and urea, into robot motion [45]. Self-propelled microrobots which are able to actively swim, but not necessarily be controlled with a high degree of precision, have been explored for medical applications such as active drug diffusion in luminal organs such as the bladder, which naturally possesses a high concentration of urea, a useful chemical fuel to power robots with urease-based propulsion [46]. Beyond potential applications in biology, self-propelled microparticle-based microrobots are actively being explored for applications such as environmental remediation [47, 48], where their active mobility enhances diffusion and allows them to efficiently move through a contaminated liquid for treatment [49]. These applications rely on swarm behavior, which is an active area of microrobotics research which promises to create more potent effects by using multiple microrobots collaborating to accomplish a task [25].

A more common strategy for microrobotic power and control is to generate externally controlled energy fields which interact with the robot to create motion. Microrobots based on this princple have been driven with light [50], electric fields [51], ultrasound [52], or magnetic fields [18]. Externally generated energy fields such as these can be effectively used to precisely control the movements of microrobots compared to self-propelled microrobots. Light can be patterned quite precisely by sources such as lasers and LED arrays and can be used to control robot motion with high precision [50]. Typically, the light is transduced into motion by heating a section of the microrobot, and then the heat gradient creates convective forces which the robot can harness for motion. Light can also be used to stimulate optogenetically engineered cells in biohybrid robotic systems in order to generate motion that is directed by light and powered by active cells like cardiomyocytes [53].

The use of externally generated magnetic fields to control untethered robotic movement has attracted particular attention due to the advantages of magnetic fields for navigation *in vivo* [17, 25, 30, 54, 55]. Magnetic fields can pass safely through tissue [56], and can be set up in order to steer microrobotic systems in three dimensions [57]. The biocompatibility of high strength magnetic fields is well attested to by the ubiquity of magnetic resonance imaging (MRI) for routine medical diagnostics. Magnetic microrobots, possessing ferromagnetic or paramagnetic materials which interact with magnetic fields and feel forces and torques, are a very common class of microrobots, and are likely to be highly applicable for biomedical applications [13, 18] . In chapter 3, we will explore magnetic manipulation of microrobots extensively, and explore how optimal control systems for operating magnetic microrobots can be developed with machine learning methods.

Materials selection is an important component of microrobotic system design [37, 58]. In most microrobotic systems, the material properties of the microrobot directly determine the capabilities and potential uses of the microrobot. As an example of a first order consideration, some of the material used to construct a microrobot must be magnetically susceptible if the microrobot is to be controlled by externally generated magnetic fields creating forces and torques on the microrobot. Microrobots made of composite materials can add additional elements of functionality. Magnetic microrobots with integrated soft polymer materials have been created in order to create microscale robotic grippers which are useful for biopsy and cargo carrying, for example [16, 59, 60]. Robots moving through an *in vivo* environment should be biocompatible, with soft elements to match the mechanical properties of the tissue with which they are interacting [61]. Some researchers have also demonstrated how incorporating biological materials into the design of the microrobot can be used to help camouflage the microrobot against attack from immune cells [61, 62], highlighting how microrobots that can successfully operate in the body might ultimately be quite sophisticated.

As the field of microrobotics matures, the design of microrobotic systems is trending towards the use of complex composite materials [26, 29, 63, 64], dynamic morphologies [16, 39, 65, 66], and integrated biological components [41, 42, 61, 67, 68]. This trend is driven by the desire to increase sophistication and capabilities of microrobotic systems. Additionally, microrobots are increasingly expected to operate in complex and uncharacterized environments [62, 65, 69]. All of these factors create difficulties when constructing dynamic and kinematic models of microrobotic behavior, making it especially complex and challenging to use classical feedback control systems to coordinate microrobot behaviors [18, 39, 59]. This challenge in achieving robust control has prompted calls for improvements in microrobot control methods [70] and adaptive locomotion strategies [71].

The goal of a control system for a remotely actuated microrobot is to manipulate the shape and magnitude of the actuating energy field in order to move the microrobot to achieve an intended dynamic behavior. Achieving this task usually requires an accurate dynamic model of the complete system, including the dynamics of the robot, the environment, and the actuator. Significant work has been done developing dynamic and kinematic models for different microrobots and actuators [39, 60, 72, 73]. These models are often developed by making simplifying assumptions about the system such as uniform magnetization [72], ideal shape [72], and system linearity [74, 75], which could lead to behavioral deviations between the physical system and the modeled system. The difficulty in accurately modeling the dynamics of microrobot behavior increases significantly for microrobots with complex magnetization profiles, soft material composition, or active shape-changing capabilities [20, 39, 64, 66, 76]. As the capabilities and associated complexity of microrobotic systems expand, the difficulty of creating accurate dynamic models of the system behavior must increase as well. Control system development through reinforcement learning is a promising approach to control the next generation of complex microrobotic systems accurately, and without the need for extensive modeling and system identification. Reinforcement learning is a control engineering approach that has achieved model-free control for a range of robotic tasks, and could help overcome current impediments to robust microrobotic control [77-80]. In the next chapter, we present the research we have done to create a robust control system for model-free microrobot control based on deep reinforcement learning and explore the implications of this technology for the future of microrobotics.

3.0 Creating Smart Microrobots with Reinforcement Learning

3.1 Introduction

Adaptable organisms which can learn new behaviors based on past experience can thrive in a wide range of changing environmental conditions by tailoring their behavior to suit the conditions of their environment. Systems capable of learning adaptive behavioral patterns based on past events are ubiquitous in nature and are found across all levels of biological hierarchy, including in biochemical networks [81], bacteria [81, 82], nematode worms [83], insects [84], plants [85], adaptive immune systems [86], and animal behavior [87]. Because of the wide-ranging applicability of adaption and learning to the success of living organisms, developing engineered systems that can also learn new behaviors from past experience is a longstanding goal of engineers and scientists [88]. There have been many notable successes in this field. Within the field of synthetic biology for example, learning and memory behaviors have been engineered into genetically engineered bacteria with bistable genetic toggle switches [89] and Pavlovian associative learning circuits [90]. In artificial intelligence (AI) and machine learning research domains such as deep learning [91], machines have been developed which can efficiently learn on their own how to extract patterns from large sets of data for tasks such as image recognition [92] and text classification [93], with significant economic and societal impact [94]. Reinforcement Learning (RL) is a subdomain of machine learning which is of particular interest for the creation of adaptable intelligent agents such as robots which are capable of learning and modulating their behavior. RL algorithms have achieved success in robotic control [77, 79, 80, 95], and have even been shown to exceed human level performance in complex tasks with large possible state spaces that cannot be tractably and exhaustively modeled, such as the game of Go [96].

One class of robotic system which could significantly benefit from adaptive learning behaviors is micro/nanorobots [71]. Microrobotic systems have received significant research attention for performing micromanipulation tasks and particularly for their potential therapeutic biomedical applications [12, 58]. There are several compelling reasons to explore the use of adaptive learning control algorithms for micro and nanorobotic control. First, the design of microrobotic systems is trending towards the use of complex composite materials [26, 29, 63, 64], dynamic morphologies [16, 39, 65, 66], and integrated biological components [41, 42, 61, 67, 68], all of which can create difficulties in accurately modeling the robotic system behavior. Additionally, microrobots are increasingly expected to operate in complex and uncharacterized environments [62, 65, 69]. Due to the wide range of environmental conditions encountered and the complexity involved in dynamic and kinematic modeling at the microscale, microrobots that can learn from environmental interactions and adapt their behavior with reinforcement learning could prove to be highly efficient and high-performance systems. In fact, RL has already demonstrated considerable promise for controlling microrobot behaviors. RL agents have been trained to control microrobot behavior in simulation for solving navigation and swimming challenges in heterogenous fluids [97, 98], and an early-form RL algorithm, Q-learning, has been shown to be effective for controlling the behavior of laser-driven microparticles in a discretized grid environment [78]. However, control of more kinematically complex real-world microrobots that operate in dynamic biomimetic microfluidic environments with clinically relevant magnetic actuation has not yet been reported.

In this work, we demonstrate that deep reinforcement learning based on the Soft Actor Critic algorithm [99] can be used to create smart soft helical magnetic microrobots which autonomously learn optimized swimming behaviors when actuated with non-uniform, nonlinear, and time-varying magnetic fields in a physical fluid environment. Our RL microrobots learned successful actuation policies both from state variable input and directly from raw images, without any *a priori* knowledge about the dynamics of the microrobot, the electromagnetic actuator, or the environment (Figure 3-1A). The microrobotic RL agent discovered multiple successful actuation strategies in separate learning trials, and the control policies learned by the agent all recapitulated the behavior of theoretically optimal physics approaches for actuating helical magnetic microrobots [100]. These results demonstrate the potential of reinforcement learning for developing high performance multi-input, multi-output (MIMO) controllers for microrobots without the need for explicit system modeling. This capability to autonomously learn model-free microrobot control algorithms could significantly reduce the time and resources required to develop high performance microrobotic systems.

3.2 Results

In order to create an environment where we could test the hypothesized efficacy of RL control systems for magnetic microrobots, we first designed and built a physical arena with multidimensional magnetic actuation. Next, we deployed a magnetic microrobot, whose design was inspired by the work of Kumar and colleagues [101], in our magnetic arena. In our experimental setup, a helical agar magnetic robot (HAMR) (Figure 3-1B) was tasked with swimming clockwise through a fluid filled lumen (Figure 3-1C) under control of a non-uniform rotating magnetic field generated by a three-axis array of electromagnetic coils (Magneturret) (Figure 3-1D). To complete the feedback control loop, an overhead camera was used to track the position of the HAMR in the channel. An RL agent, acting as the controller, received information about the state of the system as captured by the overhead camera, and returned electromagnetic control as output. The fundamental control problem was encapsulated by this question: how should the currents in the electromagnetic coils be modulated in order to create a magnetic field that places forces and torques on the HAMR sufficient to drive its locomotion toward a specific target?

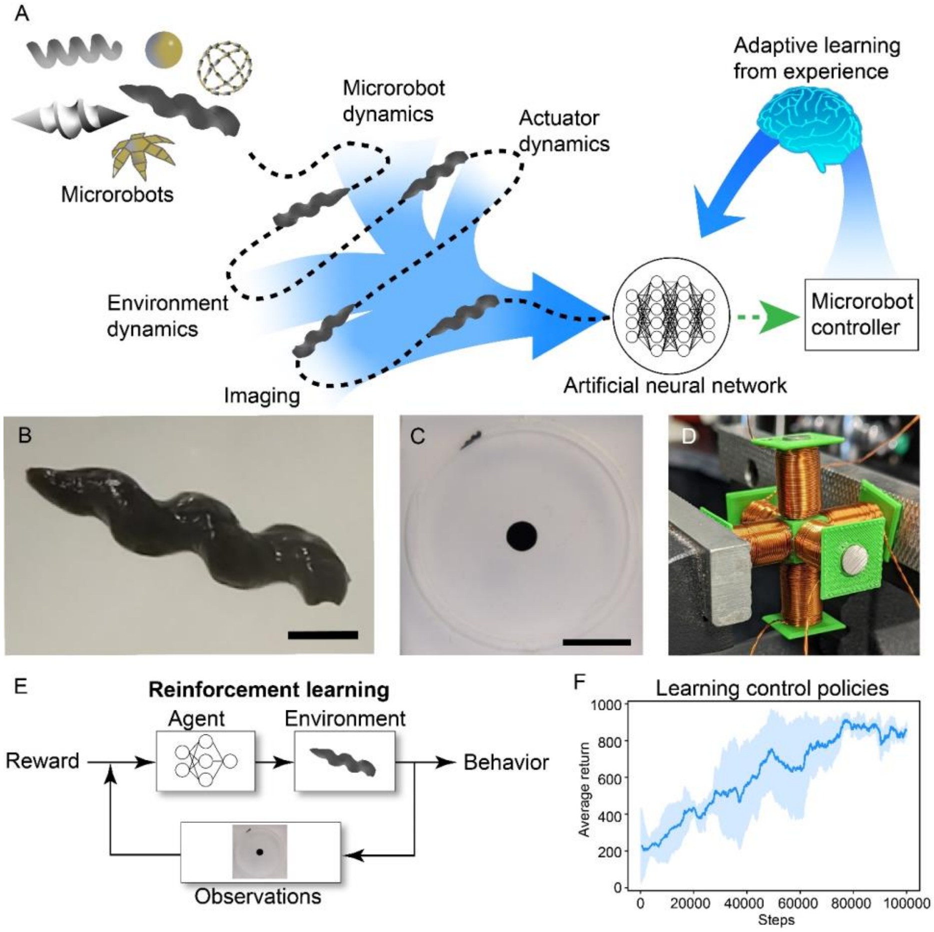


Figure 3-1. Microrobots with unknown dynamics in uncharacterized environments can be controlled with deep reinforcement learning.(A) Microrobotic systems are designed with a great variety of shapes, sizes, materials, and actuation methods. Controllers based on artificial deep neural networks trained with reinforcement learning (RL) can factor in all of these complex dynamic systems and inputs to create model-free microrobot controllers. We developed an RL-based control system to control **(B)** helical agar magnetic robots (HAMRs), within **(C)** a circular microfluidic arena using **(D)** a three-axis electromagnet (Magneturret) placed below the arena. The task for the RL agent was to move the HAMR continuously around the circle in a clockwise direction. **(E)** The controller we developed used a reward signal as a 'reference input' to give the agent information about the desired behavior. The agent then acted in the environment by manipulating the magnetic fields of the Magneturret to move the HAMR. The behavior of the HAMR was observed by an overhead camera, which fed state information back to the agent. **(F)** The RL agent learned HAMR control policies that optimized the expected future rewards, which led to performance improvement over time.

Instead of explicitly modeling the dynamics of the magnetic actuator and the HAMR within the environment and specifying a controller, we performed the much simpler task of specifying the desired behavior of the HAMR in the form of a reward signal (Figure 3-1E). The agent observed the state of the environment along with a reward signal containing information about which actions lead towards the successful completion of the task. The RL agent started without any *a priori* information about the task and had to learn to perform the task by sampling actions from the space of all possible actions and learning which actions resulted in behavior which was rewarded.

The RL controller required us to develop and formulate the task as well as the associated reward signal. At the beginning of each training episode, a target position was defined, 20° clockwise from the starting position of the HAMR in the circular channel. The objective of the RL agent was to develop an action policy, π, which maximized the total value of the rewards it would receive if it followed that policy in the future. When the environment was in state, s, the agent chose an action, a, from the policy according to $a\sim \pi(\cdot\,|s)$, probabilistically selecting from a distribution of possible actions available in that state. The agent received a reward when it selected actions that moved the HAMR clockwise through the circular lumen towards the target, and it received a negative reward when it moved the HAMR counterclockwise. If the HAMR reached the target within the allotted time, the agent was given a large bonus reward, and the target position was advanced 20°. The reward function we selected was $r(s,a) = \Delta\theta_r + 1000 \textbf{ if } (\theta_r = \theta_g)$ where θ_r is the angular position of the HAMR in the channel in degrees, θ_g is the angular position of the goal, and $\Delta\theta_r$ is the change in angular position of the HAMR as a result taking of action a in state s.

RL problems are usually formalized as Markov decision processes [102], in which at time t, the state of the system s_t is observed by the agent. The agent performs an action a_t which changes the state of the environment to $s_t{}'$, yielding a reward $r_t(s_t, a_t)$. This process continues for the duration of the task, yielding a trajectory of the form $(s_t, a_t, r_t, s_{t+1},\ a_{t+1}, r_{t+1}, s_{t+2} \ldots)$. The goal of the RL agent is to identify an optimal policy $\pi^*(a|s\ )$ for selecting actions, based on state observations, that maximize the rewards received for following the policy. Over the course of training, the agent autonomously learned a control policy by trying actions in the environment, observing the reward obtained by performing those actions, and modifying its future behavior in order to maximize the expected future return (Figure 3-1F).

The control problem for our microrobotic system was formulated as an episodic, discrete time problem with a continuous action space and continuous state space (Figure 3-2). The state space consisted of all the possible states for the system: the position of the HAMR within the channel, the speed of the HAMR, the shape of the magnetic fields, the time remaining in the episode, and relative position of the robot to the target position in the channel. The action space consisted of four continuous actions, which controlled the magnitudes and phase angles for sinusoidal currents in the Magneturret. While the current waveforms could theoretically take on an infinite number of shapes, we chose to define the applied waveforms as sinusoids to bound the space of possible actions that the agent could take. Sinusoidal currents were chosen because these can be used to generate rotating magnetic fields in other three-axis electromagnetic actuators for microrobots, such as Helmholtz coils [32].

3.2.1 Entropy Regularized Deep Reinforcement Learning Enables Continuous Microrobot Control

We selected the Soft Actor Critic RL algorithm (SAC) for this effort. SAC is a maximum entropy RL algorithm that seeks to balance the expected future rewards with the information entropy of the policy [99]. In other words, SAC learns a policy that successfully completes the task while acting as randomly as possible, which in practice often leads to robust policies that are tolerant of perturbations in environmental conditions [77]. SAC had previously proven useful for real-world robotic tasks with high-dimensional, continuous state and action spaces [77, 79], which suggested that it would be applicable our microrobotic control problem. In previously reported applications of real-world reinforcement learning with physical systems [103], SAC was demonstrated to be highly sample efficient, requiring relatively few environmental interactions in order to develop a successful policy. Sample efficiency is critical when performing reinforcement learning with real-world robotics in order to reduce wear and tear on the system, and in order to minimize the time needed to learn a policy [95].

The SAC algorithm seeks to develop an optimal stochastic policy π^*:

$$\pi^* = arg\max_{\pi} \sum_{t} \mathbb{E}_{(s_t,a_t)\sim\pi}[r(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot\,|s_t)]$$

where $\mathcal{H}$ is the information entropy of the policy and α is a temperature hyperparameter, which balances the relative impact of the policy entropy against the expected future rewards. Here, we used a version of the SAC algorithm in which the temperature is automatically tuned via gradient descent so that the entropy of the policy continually matches a target entropy, $\bar{\mathcal{H}}$, which we selected to be -4 (-Dim of the actions space) [104]. A full derivation of the soft actor critic algorithm is beyond the scope of this dissertation, but interested readers are directed to Haarnoja et al. [99]. Briefly, the SAC algorithm uses an actor, π, which is a deep neural network that takes the state of the system s_t as input and returns action a_t as output. A value function is created to rate the value of taking actions when in particular states and instantiated using two critic neural networks $Q_{1,2}(s, a)$ which take states and actions as input and return values corresponding to the relative value of taking action a_t in state s_t. Two Q networks are trained in order to reduce overestimation in the value function. Environmental transitions in the form of (s, a, r, s', d) sets are recorded in an experience replay buffer, D, where d is a done flag denoting a terminal state, set either when the microrobot has reached the goal, or the episode has timed out. The SAC algorithm learns off-policy by randomly sampling minibatches of past experiences from D, and performing stochastic gradient descent over the minibatch in order to minimize a loss function for the actor network, π, critic networks, Q_1 and Q_2, and temperature parameter, α. Over the course of learning, the parameters of the actor and critic neural networks are updated so that the behavior of the policy approaches the optimum policy, π^*. A detailed version of the algorithm is available in Appendix A.4, neural network architectures and hyperparameters used are available in Appendix Table 1.

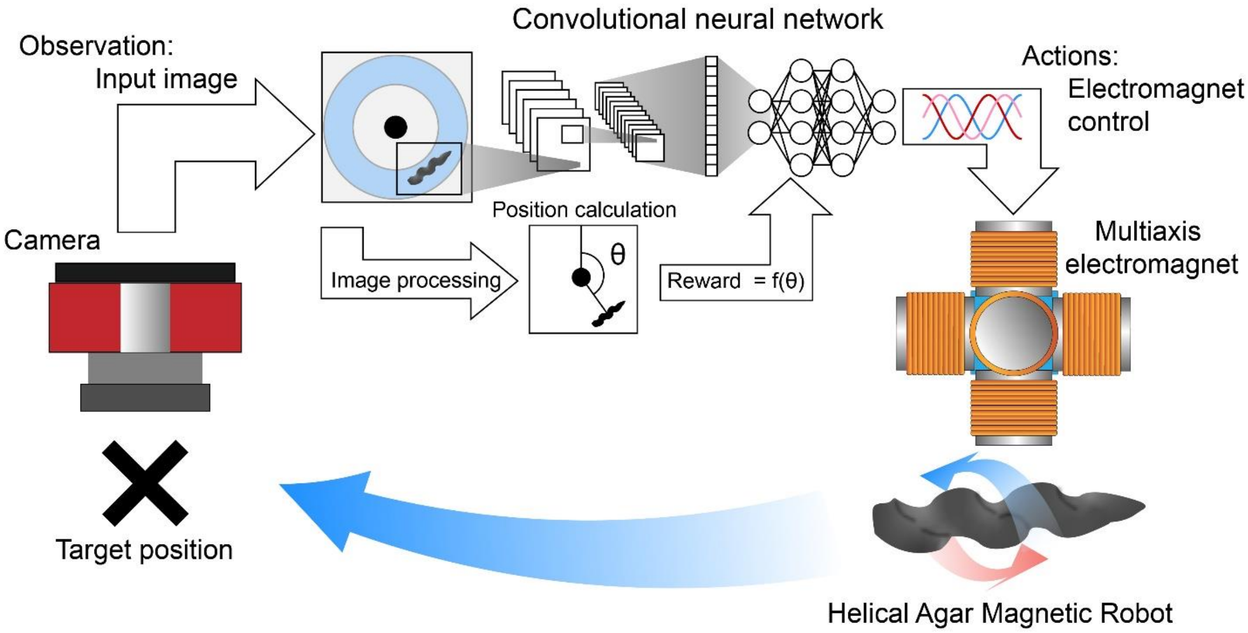


Figure 3-2. Closed loop control of magnetic helical microrobots based on deep reinforcement learning. Our system consisted of a helical agar magnetic robot (HAMR) in a circular fluidic track that was given the task of moving to a target position along the track. The movements of the HAMR were controlled by a deep reinforcement learning agent, which captured the state of the system with an overhead camera in order to detect the position of the HAMR in the channel. This state information was then fed into a neural network that returned a set of continuous actions, which were used to control the currents in a multiaxis electromagnet. Magnetic fields generated by the electromagnet induced motion in the HAMR. The deep neural network was trained with reinforcement learning by a reward signal calculated based on the position of the HAMR.

3.2.2 Hardware Implementation to Control Magnetic Microrobots with Reinforcement Learning

Magnetic fields created by electromagnetic coils are common actuators used for magnetic microrobots, and have significant potential for clinical medical applications [13, 17, 28]. Magnetic fields act on a magnetic microrobot by imparting forces and torques on the robot. For a microrobot with a magnetic moment, $\boldsymbol{m}$, in a magnetic field, $\boldsymbol{B}$, the robot experiences a force $\boldsymbol{F}$ according to

$\boldsymbol{F} = \nabla(\boldsymbol{m} \cdot \boldsymbol{B})$. In a non-uniform magnetic field (i.e., a magnetic field with a spatial gradient), a ferromagnetic or paramagnetic microrobot feels force in the direction of increasing magnetic field gradient. The magnetic microrobot also experiences a torque according to $\boldsymbol{\tau} = \boldsymbol{m} \times \boldsymbol{B}$, which acts to align the magnetic moment of the microrobot with the direction of the magnetic field. When the magnetic field is rotated so that the direction of $\boldsymbol{B}$ is constantly changing, it is possible to use this torque to impart spin to the microrobot at the frequency of the rotating magnetic field, up to the step out frequency of the robot [72]. If the spinning microrobot Is helically shaped, rotation can be transduced into forward motion so that the microrobot swims as if propelled by flagella [100]. This non-reciprocal helical swimming is efficient in low Reynolds number fluidic environments commonly encountered by microrobots [100]. Because of the efficiency of this swimming mode, and because the magnetic torque available to a microrobot decreases more slowly with distance compared to the force [18], magnetic microrobots are often helically shaped [29, 30, 63]. For this reason, we selected a helical magnetic microrobot, the HAMR, as our model system.

The HAMR that we created for this study was composed of a 2% w/v agar hydrogel, which was uniformly diffused with 10% w/v iron oxide nanopowder to form a magnetically susceptible soft polymer [101]. This magnetic agar solution was heated to melting temperature and a syringe was used to inject the liquid into a helical mold created using a stereolithography 3D printer (Figure 3-3B). The agar in the mold solidified and the robots were removed with a metal needle and stored long-term in deionized (DI) water. The HAMRs molded for this study were 4.4 mm in length, 1 mm in diameter, and asymmetrical from head to tail, with flat head and a pointed tail (Figure 3-3C,D). Microrobots formed with this technique have been previously shown to be controllable within rotating magnetic fields, and to perform biomedical functions such as cell delivery [101] and active biofilm removal in the root canal of human teeth [32]. For our application, this HAMR design had several advantages. The HAMRs were simple to manufacture at low cost with batch fabrication methods. The HAMRs were small enough to act as helical swimming robots in a flow regime with Reynolds number ~1, but large enough, about the size of a small grain of rice, to be easily manipulated and visualized without the use of microscopes or other micromanipulation tools. Because the HAMRs swim with non-reciprocal, helical motion in the presence of a rotating magnetic field, a very common motif in microrobotic research [24, 30, 63], insights gained from this study could be readily be extended to other microrobotic systems with similar characteristics. Because the HAMRs were made of soft hydrogel, they were flexible and deformable. Soft bodied robots have many favorable characteristics for *in vivo* use such as deformability to fit through irregular shaped channels and enhanced biocompatibility (e.g., by matching the elastic modulus of the biological environment) [105]. These characteristics make soft-bodied microrobots appealing for biomedical applications, but It can be more difficult to create accurate dynamic models for soft-bodied microrobots [65]. Our method of using reinforcement learning to develop control systems without explicit modeling could be particularly useful for soft microrobots due to this modeling constraint. Finally, despite being soft-bodied, the hydrogel structure of the HAMR did not experience noticeable wear over the course of several months of continuous use, thus meeting a practical reinforcement learning constraint that the system not be susceptible to significant wear and tear during extended use [103].

As an actuator for our model microrobot system, we developed a three-axis magnetic coil actuator – the Magneturret - which contained six permalloy-core magnetic coils arranged on the faces of a 3D-printed Acrylonitrile butadiene styrene (ABS) plastic cube (Figure 3-3E). The two coils on opposite sides of the central cube along each axis were wired together in series so that they both contribute to the generation of a magnetic field along their respective axis. Each of the three coils, hereafter referred to as the X, Y, and Z coils, were driven with a sinusoidal current generated by a pulse width modulated (PWM) signal created by a microcontroller and amplified in an H-bridge motor driver. The resulting magnetic field, produced by the superposition of the magnetic fields from the three coils, could be modulated by varying the frequency, amplitude, and phase angle of the sine current waves in each coil. To cool the coils and prevent thermal damage, the Magneturret was sealed with epoxy resin into a 3D printed housing and coolant was continuously pumped through while the coil was operating. The RL agent was given direct control over the magnitude and phase angles of the sinusoidal driving currents in the X and Y-axis coils of the Magneturret (Figure 3-3A). The Z-axis magnitude was calculated as the larger of the two magnitudes in X and Y, and the Z-axis phase angle was fixed. The sinusoidal currents in each axis used a fixed angular frequency of 100 rad/s (15.9 Hz). A summary of the control variables in presented in Table 3-1.

Table 3-1. Control inputs for electromagnet waveforms.

Magnetic Coil Control Parameters

Control variable	**Symbol**	**Source**	**Range**
Frequency	f	Fixed	15.9 Hz (ω=2πf=100 rad/s)
Magnitude X	M_X	RL agent	[-1,1] unitless
Magnitude Y	M_Y	RL agent	[-1,1] unitless
Magnitude Z	M_Z	$\max(\lvert M_X\rvert, \lvert M_Y\rvert)$	[0,1] unitless
Phase angle X	Φ_X	RL agent	[0,2π] radians
Phase angle Y	Φ_Y	RL agent	[0,2π] radians

Control Equations

Current in X-axis coil	$I_x = M_x \sin(ft + \varphi_x)$
Current in Y-axis coil	$I_y = M_y \sin(ft + \varphi_y)$
Current in Z-axis coil	$I_z = M_z \sin(ft)$

We chose to operate our HAMR in a circular, fluid-filled track for this study. This arena served as a simple environment, which mimics the tortuous *in vivo* luminal environments that microrobots operating in the body might encounter, while providing a simple environment for us to establish a robust proof-of-concept reinforcement learning control system (Figure 3-3F). The HAMR could swim in a complete circle within this arena, and no human intervention was required to reset the position of the robot in the environment during training, which facilitated automated learning [95]. The arena was constructed by pouring polydimethylsiloxane (PDMS) over a polyvinyl chloride ring with an outer diameter of 34 mm and a 1.7 mm x 3 mm rectangular cross section. The PDMS was then cured and plasma bonded to a second flat sheet of cured PDMS to form a rectangular lumen for the HAMR to swim. PDMS is transparent, allowing us to see the robot in the arena and to visually track it with an overhead camera. During long-term learning experiments, the PDMS arena was submerged in a petri dish filled with DI water in order to prevent the formation of air bubbles in the channel due to evaporation over the course of an experiment. This petri dish was then placed on top of the Magneturret, with the center of the Z-axis coil aligned with the center of the circular track (Figure 3-3G). A black rubber wafer was placed into the center of the arena on top of the PDMS to act as a fiduciary marker so that the center of the arena could easily be identified with image processing. A diffuse white LED backlight was positioned between the Magneturret and the PDMS arena for uniform bottom-up illumination which facilitated simple image processing by binary thresholding to identify the position of the microrobot in the channel.

We did not perform an extensive analysis to identify the shape or magnitude of the magnetic field created by the Magneturret, or to model the swimming dynamics of the HAMR in the PDMS arena. We hypothesized that we would be able to develop a high-performance control system using RL without going through the effort of developing a system model first.

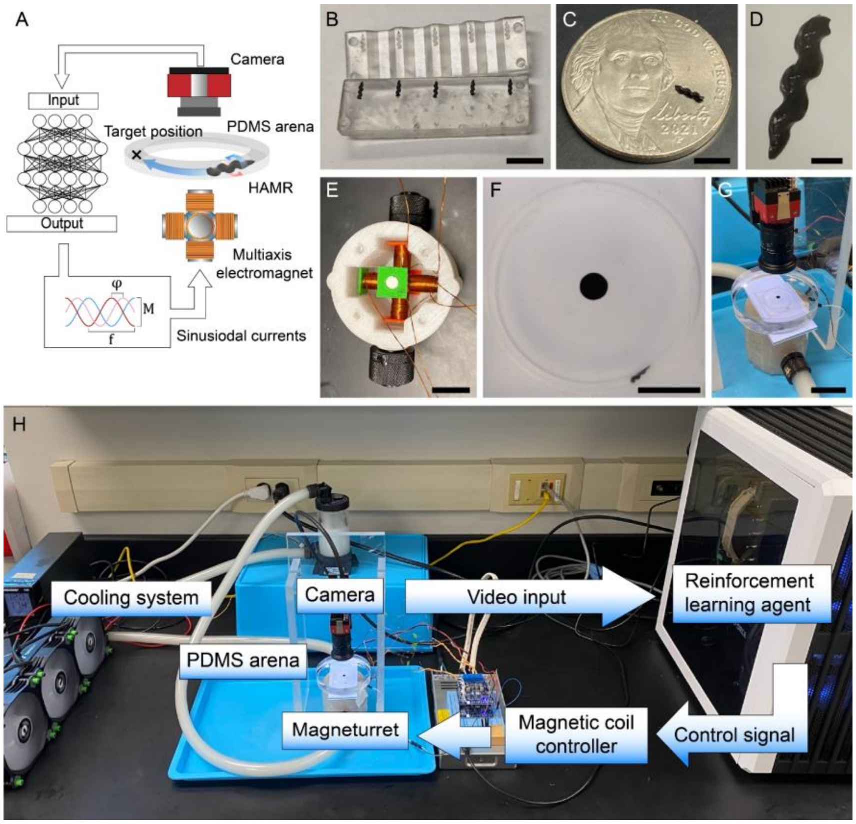


Figure 3-3. Hardware for real-world control of magnetic microrobots using reinforcement learning. Our system consisted of a helical agar magnetic robot (HAMR) in a circular microfluidic arena, controlled by a multiaxis electromagnet (Magneturret). The position of the HAMR was recorded with an overhead camera. The electromagnetic coils were driven by sinusoidal current waveforms defined by their frequency, f, phase angle, φ, and magnitude, M. **(B)** HAMRs were fabricated by molding molten hydrogel in 3D printed molds. Scale bar = 10 mm. **(C)** HAMR with a United States 5-cent coin. Scale Bar = 5 mm. **(D)** HAMRs were composed of agar hydrogel infused with iron oxide nanopowder. Scale bar = 1 mm. **(E)** The Magneturret was composed of six coils of copper magnet wire wrapped around permalloy cores, positioned on the faces of a central 3D printed cube. The Magneturret was enclosed within a 3D printed housing and sealed with epoxy, and glycerol coolant was continuously pumped through the Magneturret housing. Scale bar = 20 mm. **(F)** A circular PDMS track with a rectangular cross section used as an arena for the HAMR. A black, circular fiduciary marker indicates the center of the arena. Scale bar = 10 mm. **(G)** The PDMS arena was submerged in a water filled petri dish placed on top of the Magneturret, with an acrylic LED light sheet as a backlight for uniform bottom-up illumination. Scale bar = 30 mm. **(H)** The complete hardware system.

3.2.3 Reinforcement Learning Can Be Used to Learn Microrobot Control Policies

Reinforcement learning systems have been demonstrated that can learn to achieve tasks from a wide range of state information sources [80]. For this study, we evaluated the ability of our RL agent to learn control policies from either state vector-based inputs (Figure 3-4A) or raw images augmented with the goal position (Figure 3-4B). In state vector input mode, the angular position, θ_r, of the microrobot in the channel was calculated with image processing by binary thresholding and simple morphological operations. The camera was deliberately run with a slow shutter speed, so the image was intentionally washed out to remove noise. This simplified the task of using binary thresholding operations to identify the position of the HAMR and the center of the channel. The angular position of the HAMR in the channel was measured relative to the fiduciary marker in the center of the circular arena. This information, as well as the position of the goal, θ_g, the last action taken by the agent ($M_{x,t-1}, M_{y,t-1}, \varphi_{x,t-1}, \varphi_{y,t-1}$), and the time, t , remaining in the episode were used to create a state vector. In the second input mode, we gave the agent observations in the form of raw pixel data from the camera. The images from the overhead camera were scaled down to 64x64 pixels, and the images were augmented with a marker indicating the position of the goal θ_g, as a line radiating outward from the center of the circular track to the goal position in front of the HAMR. These images were then passed into a convolutional neural network, which is a deep neural network architecture that has been demonstrated to effectively learn to identify features in images for classification tasks [106], and has been used to control robots with raw image input using reinforcement learning [104].

Reinforcement learning is based on the mathematics of Markov decision processes, which theoretically require the full state of the system to be available to the agent in order for convergence to be guaranteed [102]. In our implementation, the velocity of the HAMR at any given time could not be determined from a single still-frame image, so the total state of the system given to the agent at each time step was composed of three concatenated sub-observations taken 0.3 seconds apart. This allowed the agent to infer the velocity of the HAMR based on differences between the three sub-observations. This technique of batching sequential observations for improving the observability of the system for RL has been used successfully in domains such as Atari video games, in which the agent learned from raw pixel data gathered from sequential screenshots of the game [107].

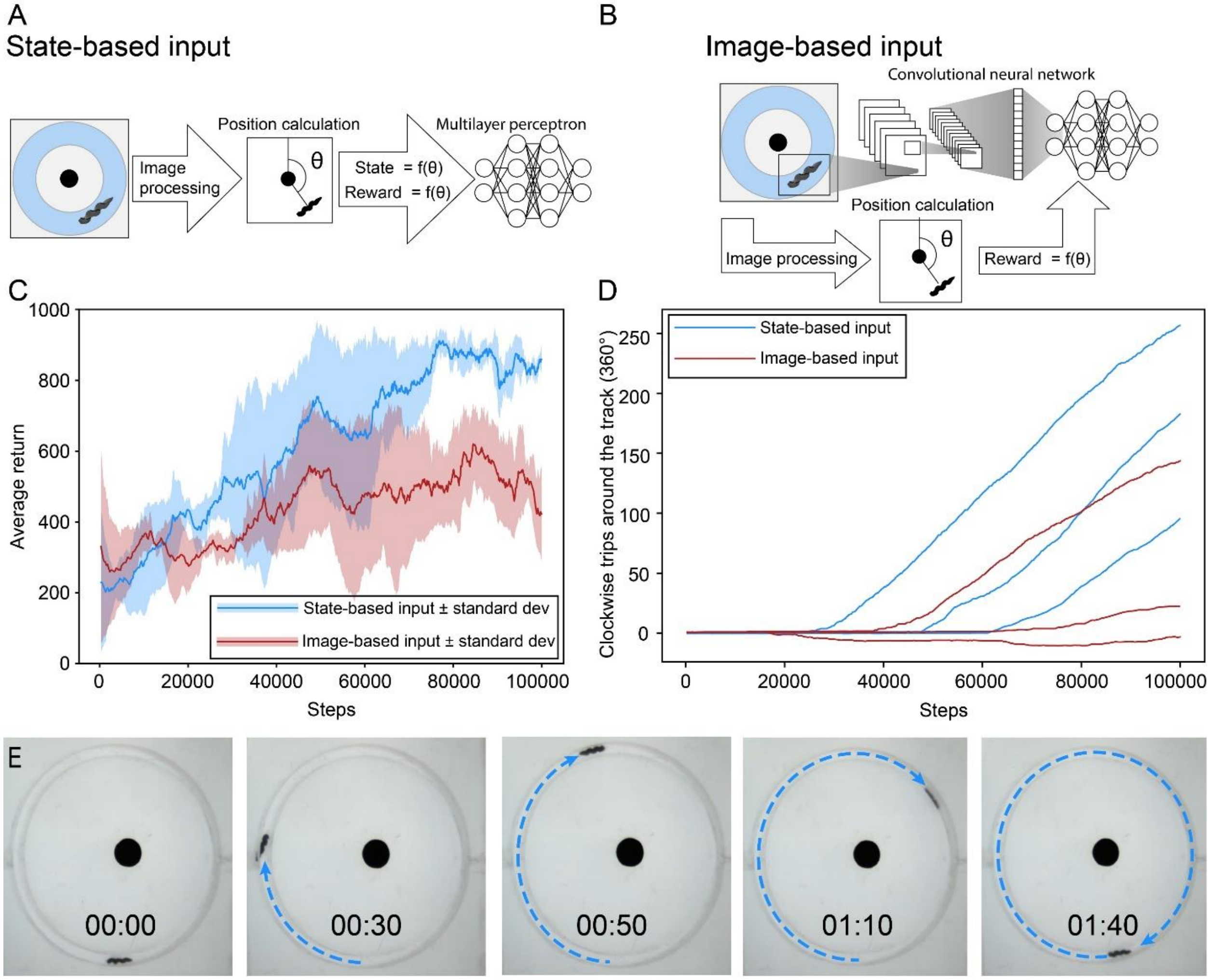


Figure 3-4. Reinforcement learning yielded successful control policies for the HAMR within 100,000 time steps, using both state vectors and images as input. (A) We used two input modes to train the RL agent. The first mode was to calculate a state vector consisting of the robot position, θ_r, the goal position, θ_g, the last action taken by the agent, $M_x, M_y, \varphi_x, \varphi_y$, and the time remaining in the episode. Three sequential observations were taken 0.3 seconds apart, and combined as a single state, s. **(B)** The second input mode used images of the microrobot in the arena as the state. Three sequential images were then passed to a convolutional neural network as the state, s. **(C)** We trained the agent for a total of 100,000 time steps per training session. Traces represent the average and standard deviation of the return from three successful training runs with each input type. **(D)** Successful training resulted in policies which, after an initial learning period, achieved consistent forward motion, going continuously around the circular arena. **(E)** The RL agent learned policies that moved the microrobot around the full circular arena with helical swimming motion.

At the beginning of each learning trial, the actor and critic neural networks' trainable parameters were randomly initialized. We allowed the neural networks to train for a maximum of 100,000 times steps, using a fixed ratio of one gradient update per environmental step, which has been shown to reduce training speed in exchange for higher training stability [108]. 100,000 environment steps were adequate time for effective actuation policies to be learned, both with state vector input and raw images (Figure 3-4C). It is commonly reported when using reinforcement learning that several million environmental steps are necessary to derive a successful policy [95, 107], so this result shows the sample efficiency of SAC, which is critically important for RL tasks that are using physical systems instead of simulations. After 100,000 steps the state vector-based policies achieved significantly higher overall level of performance, and it is likely that the raw-image-based policies could have benefited from longer training periods [104]. A time-lapse movie of the HAMR recorded during the learning process is shown in supplementary movie 1 in Appendix A.5.

Once the training sessions were complete, we evaluated the learned policies to test their performance. For evaluating policies, we used the highest performing policy parameters learned during a training session by monitoring a rolling average of the return over the last 100 episodes and saving the policy parameters each time the rolling average performance exceeded the last best performing model (Appendix Figure 1). This was done because we sometimes observed a drop in performance after the peak performance was achieved in training, possibly due to overfitting. Early stopping, or selecting a policy before performance degradation has occurred, is a common technique used to prevent overfitting in neural networks [109]. For each input type, we trained three policies that resulted in net positive (clockwise) movement from the HAMR (Figure 3-4D). We performed three learning trials with state vector input, each of which resulted in a policy able to continuously move the robot around the track. In contrast, in order to develop three successful policies using image-based input within 100k time steps, we had to repeat the experiment six times. In the three trials in which the robot did not succeed within 100,000 time steps, the agent did not successfully explore all possible states, restricting its exploration to a partial section of the circular track. Successful policies were able to move the robot indefinitely around the complete circular track (Figure 3-4E, supplementary movie 2 in Appendix A.5).

We recorded each action taken by the agent during training sessions and the resultant change in state of the microrobot. For a single training run with state-vector input, we plotted the distribution of actions as a function of the resultant change in HAMR position, $\Delta\theta_r$ (Figure 3-5A). We separated the total 100k steps into 5 bins of 20k steps each. The distribution of actions over the first 20k time steps (Figure 3-5A, i) is centered around a sharp peak of actions which result in no net movement, as we would expect from an agent with little experience randomly exploring the space of possible actions. By the second batch of 20k steps (Figure 3-5A, ii), a pattern emerged in which the action distribution shifted to a bimodal distribution in which most actions still result in no net movement, but a second peak on the positive side indicates a trend towards selecting actions which result in clockwise movement. However, during this phase of training, the net motion of the robot remained close to zero (Figure 3-5B), because of fattening of the negative tail in the action distribution. As the learning process continued, the distribution continued to shift until the average movement was clockwise, with a second peak around 5 degrees per time step, and a narrow tail representing few actions, which caused the robot to move in the counterclockwise direction (Figure 3-5A v).

The soft actor critic algorithm learns a continuous stochastic policy, π, sampling actions from the policy according to $a_t \sim \pi(\cdot\,|s_t)$, in which the actions selected during training are randomly sampled from a Gaussian distribution, and the agent learns the mean μ and the variance of this distribution over the course of training [99]. This is done in order to explore the space of possible actions during training. During training the agent seeks to balance the sum of future rewards with the information entropy of the policy by maximizing an entropy regularized objective function, and the policy entropy corresponds to the explore/exploit tradeoff the agent makes during training. However, once the policies were trained, performance during policy evaluation could be increased by selecting actions from the mean of the distribution without further stochastic exploration according to $a_t = \mu(s_t)$. This deterministic evaluation led to an increase in the proportion of actions taken by the agent which resulted in positive motion, for both state-based (Figure 3-5C) and image-based agents (Figure 3-5D). We compared the total average velocity achieved by all of the trained policies in both deterministic and stochastic action selection modes, which showed that deterministic action selection led to higher performance for both policy types, and that the state–based policies achieved notably better performance than the image-based policies after 100,000 environmental steps (Figure 3-6A).

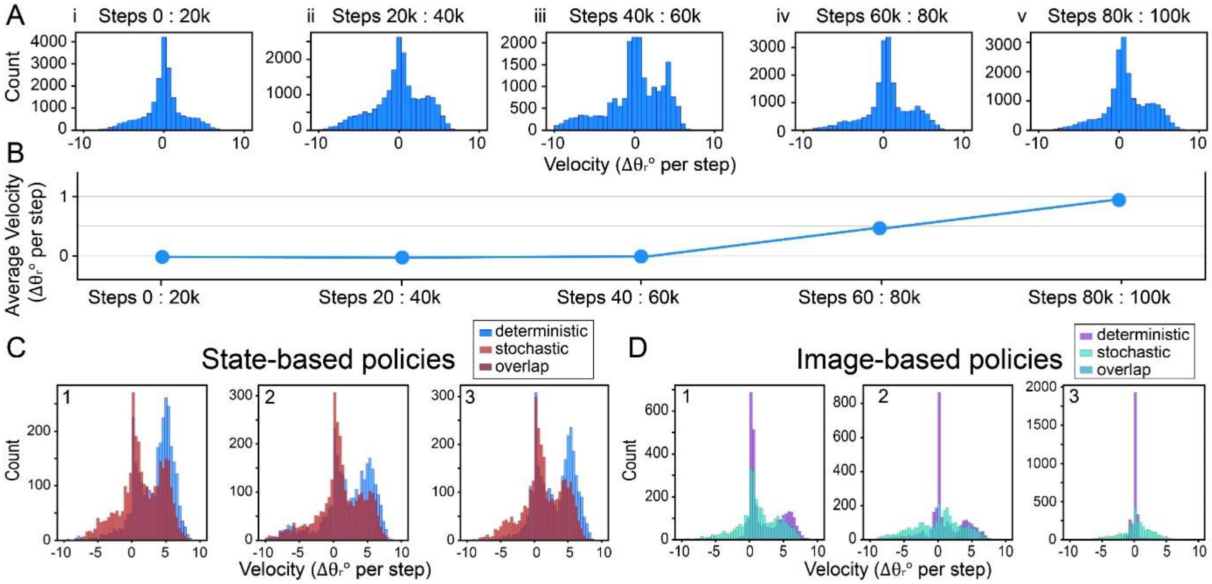


Figure 3-5. Evaluating the learning performance of the RL agent during multiple training sessions. (A) We recorded the actions and state of the agent over 100,000 training steps. The actions taken by the agent at the beginning of the training session resulted in no net forward motion **(i)**. As the agent learned, the action distribution became bimodal **(ii-v)**, with a second peak at ~5 degrees per action, indicating the agent was increasingly taking actions which resulted in forward movement. **(B)** Averaging the velocity distribution shows that by between 40k and 60k steps the agent had learned to achieve net positive movement. **(C)** Following training for each of the six RL agents, we evaluated the performance of the learned stochastic policy, π, and the deterministic policy μ for 3000 steps without additional learning. The performance increased while selecting actions deterministically from μ. **(D)** Evaluation of deterministic and stochastic action selection for image-based policies.

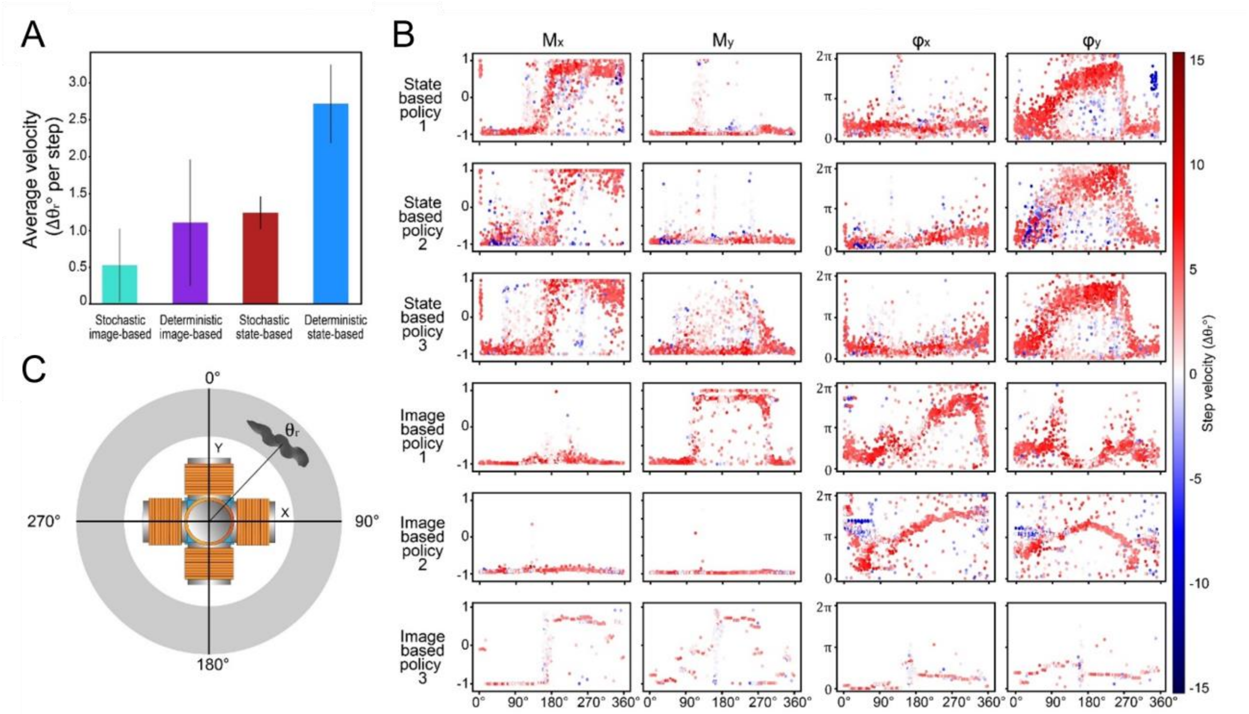


Figure 3-6. Learned policies over multiple training runs. (A) The average velocity of the HAMR during evaluation for each input type and action selection method, ± standard deviation. **(B)** Actions taken by the policies during deterministic evaluation plotted according to θ_r at the beginning of the action, and color coded according to the resultant velocity during that action. **(C)** Schematic showing the angular position of the robot in the circular channel.

We next examined the distribution of the action values chosen by the RL agent when evaluated deterministically according to $a_t = \mu(s_t)$. For each of the four actions ($\mathrm{M_x}, \mathrm{M_y}, \varphi_\mathrm{x}, \varphi_\mathrm{y}$) taken by the policy over 3000 time steps, we plotted the value of the action against the position of the HAMR, θ_r (Figure 3-6B). The plotted actions are color-coded according to $\Delta\theta_r$, with red actions indicating positive forward motion and blue actions indicating retrograde motion. Most actions taken by each of the six policies during evaluation resulted in positive motion. Each of the three policies trained using state-vector based input followed similar patterns, in which the phase angle of the X coil was held constant, and the magnitude of the X coil varied according to the position of the microrobot θ_r. The Y coil was controlled by actuating the phase angle as a function of position and holding the magnitude relatively constant. In contrast, the policies learned by the image-based agents were more heterogeneous, finding different possible ways to manipulate the 4 actions in order to produce forward motion. One pattern that is consistent in all learned policies is that the magnitudes tend to hold steady close to the maximum or minimum values of -1 and positive 1, regardless of θ_r. This would result in the largest amplitude sine waves, which we would expect because larger magnetic fields would be able to create more powerful torques on the HAMR.

3.2.4 Deriving Mathematical Control Policies from RL-Trained Policies

We observed that the policies learned by the RL agent sometimes performed actions that were obviously non-optimal (resulting in negative motion). We could likely further increase the performance of these polices by using techniques like hyperparameter tuning and longer training times [108]. However, by observing the behavior of the RL agent, we hypothesized that if we could distill the policies learned by the network into mathematical functions of the state variables, we might achieve a higher level of performance. To test this, we chose one of the state-based policies and one of the image-based policies, and fit regression models to the data in order to create continuous control signals as a function of the robot position θ_r. First, we examined policy 1, learned by the state-vector based agent (Figure 3-6B). This policy was acting by modulating the magnitude in the X coil in what appeared to be a square wave pattern, and the phase angle in the Y axis coil in what appeared closer to a sine wave. The other two actions were held approximately constant regardless of the position of the robot. We selected the subset of actions taken by policy which had resulted in a positive velocity of at least 3°, discarding the lower performing actions for this analysis. We then fit sinusoidal regression models to the M_x and φ_y action distributions and fit a square wave to M_x (Figure 3-7A). The resulting policies are shown in Figure 3-7A as solid black lines. The sine wave policy (Figure 3-7B) and the mixed sine/square wave policy (Figure 3-7C) that we developed with the regression models were then used to control the HAMR. The image-based policy that we evaluated was significantly more complex than the state-based policy (Figure 3-7D). We modeled this policy mathematically by fitting a 20th order polynomial to the data and used this polynomial policy to drive the HAMR (Figure 3-7E).

The sinusoidal policy achieved the highest level of performance (Figure 3-7F), achieving the highest average HAMR velocity of all policies tested in this study, while the square/sine policy performed slightly worse than the neural network policy on which it was based. Despite the complexity of the polynomial policy compared to the sine and square wave-based policies, the performance of this policy was approximately equal to that of the sine/square wave policy and was superior to the neural network-based policy on which it was based. Since these mathematical policies only use θ_r as the input, it is possible that we could further increase the performance of mathematically inferred policies by taking other parts of the state vector into account.

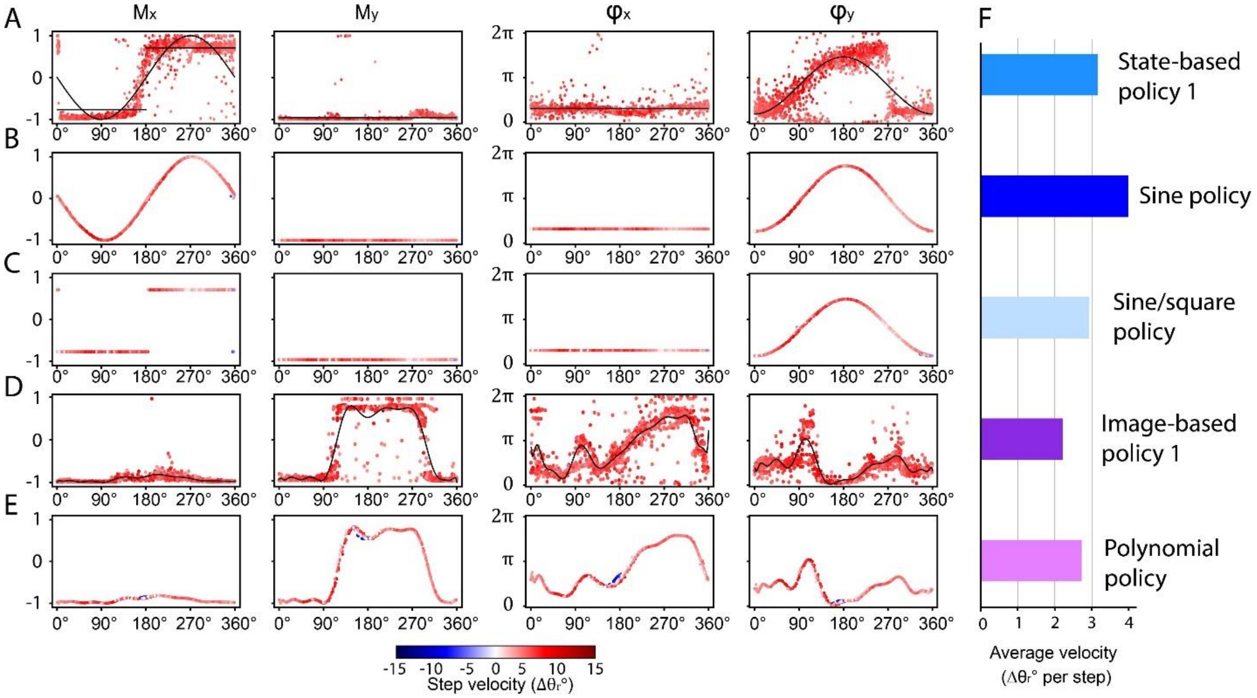


Figure 3-7. Control policies learned by the RL agent could be translated into continuous functions in order to increase performance. (A) For one of the state-based policies that the RL agent learned, the actions were plotted as a function of θ_r and mathematical functions were fit to the subset of actions which yielded HAMR velocities greater than 3 degrees per step. Sine waves and square waves were fit to the data via regression (shown in black), and those mathematical function were used to control the HAMR for 1000 time steps. **(B)** The results of running the sinusoidal policy and **(C)** the sine/square policy. **(D)** The policy learned by the imaged-based agent was fit with a 20th degree polynomial function. **(E)** The polynomial policy was used to drive the HAMR for 1000 time steps. **(F)** Comparing the average velocity of the HAMR when controlled by each policy.

3.2.5 Control Policies Learned by the RL Agent Recapitulate the Behavior of Optimal Policies based on Physical Models

Finally, we wanted to know whether the policies learned by the RL agent were modulating the magnetic field in a way that matched the control systems that human researchers have developed based on physical models. When using a uniform rotating magnetic field to steer a helical microrobot, the rotating magnetic field is usually made to rotate about the helical axis of the microrobot, which is also the direction in which the microrobot will swim (Figure 3-8A) [72]. Therefore, to drive a helical microrobot around a circular track like the one we used, we would expect that the direction of the rotating magnetic field would be tangent to the circular track at all points along the circle for the optimal policy (Figure 3-8A).

Although we did not record the magnetic field at all points along the track during policy evaluation, we can approximate the behavior of the magnetic fields based on the actions taken by the agent. The recorded actions selected by the policy while driving the HAMR around the track were used by us to estimate the magnetic fields produced during the action. To do this, we constructed a three-dimensional vector $\boldsymbol{B} = [\boldsymbol{B_x}, \boldsymbol{B_y}, \boldsymbol{B_z}]$, where $\boldsymbol{B}_n = M_n \sin{(ft + \varphi_n)}$, where the magnitude and phase angle were selected by the policy. The actions taken by the agent are run for a total of 0.9 seconds during each time step, during which time the actions are held constant and $\boldsymbol{B}$ rotates as a function of time with an angular frequency of 100 rad/s. Taking the cross product $\boldsymbol{B}_\perp = \boldsymbol{B}(t) \times \boldsymbol{B}(t+1)$ results in a vector $\boldsymbol{B}_\perp$ which points in the direction perpendicular to the plane of the rotating magnetic field (Figure 3-8A). By calculating the azimuthal angle $\theta_{B\perp} = \arctan(B_{\perp y} / B_{\perp x})$ we can approximate the direction of the rotating magnetic field during an action taken by the policy. The results of this analysis are shown by plotting an arrow with direction $\theta_{B\perp}$

at the point θ_r along the circular track for each action taken by the policy. Results are shown for the inferred mathematical policies (Figure 3-8B), the image-trained policies (Figure 3-8C), and the state trained policies (Figure 3-8D). Each policy learned by the RL agent, regardless of input type, created a rotating magnetic field nearly perpendicular to the direction of travel of the microrobot, recapitulating the behavior of the theoretical optimal policy that was based on a physical analysis of helical swimming magnetic microrobots [72].

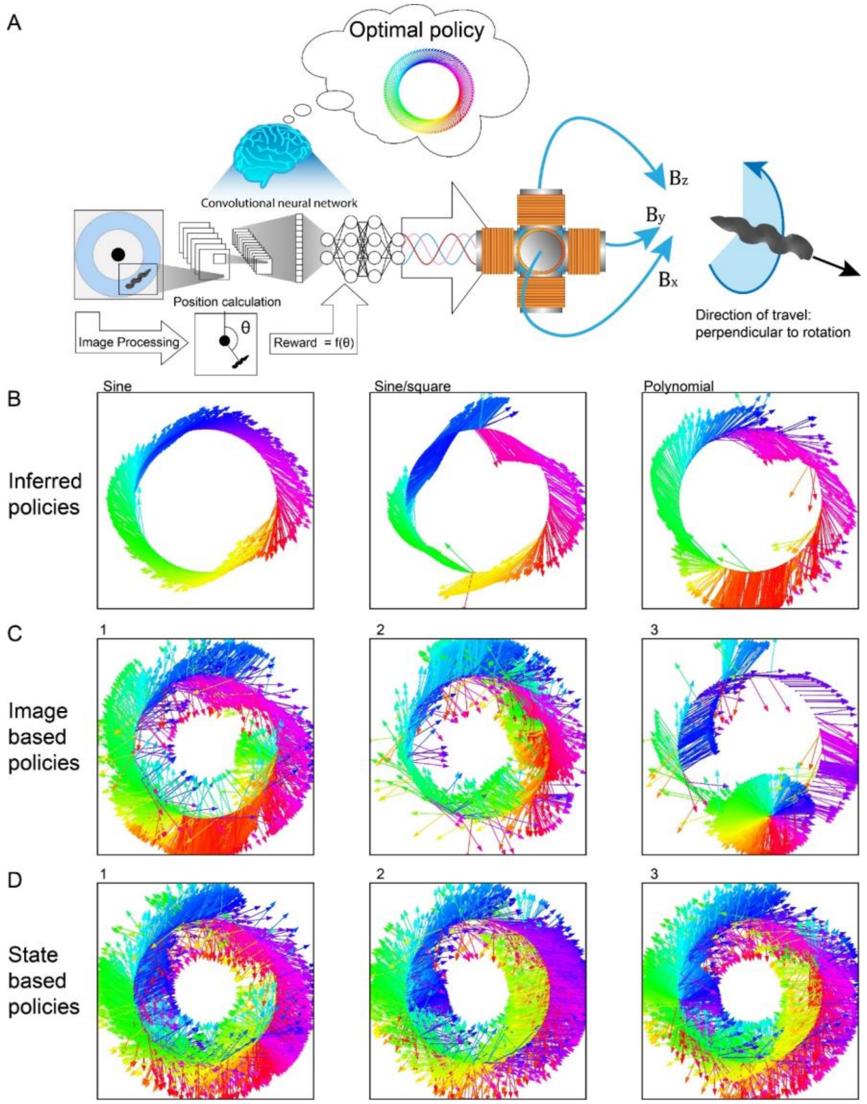


Figure 3-8. Control policies learned by the RL agent recapitulate the behavior of optimal policies based on theoretical physical models. Magnetic helical microrobots swim with propeller-like motion, transducing magnetic field rotation into torque, torque into angular velocity, and angular velocity into linear velocity in the direction of travel. The theoretical optimal policy for controlling a magnetic helical microrobot in a circular channel would be to create a rotating magnetic field perpendicular to the direction of travel of the robot at each point in the circle, so that the microrobot moves tangent to the circle at all points along the track [72]. Plotting a vector perpendicular to the calculated plane of magnetic field rotation for each action at each point along the circular track demonstrates that each policy learned by the RL agent recapitulated this theoretical optimal policy behavior. Vectors are color coded according to their azimuthal angle. **(C)** The inferred mathematical policies. **(D)** The image trained policies. **(E)** The state trained policies.

3.3 Discussion

Here, we have reported the development of a closed-loop control system for magnetic helical microrobots, which was implemented using reinforcement learning to discover control policies without the need for any dynamic system modeling. Continuous control policies for high-dimensional action spaces were represented by deep neural networks for effective control of magnetic fields to actuate a helical microrobot within a fluid-filled lumen. High-dimensional inputs including state-vector inputs and raw images were sufficient to represent the state of the microrobot. Compared with previously reported control systems for magnetic microrobots [18], we believe that the system we have presented possesses a number of key advantages. Electromagnetic actuation systems for microrobots are either air core, such as Helmholtz coils and Maxwell coils, or contain soft magnetic materials in the core which enhance the strength of the generated magnetic field, but can lead to nonlinearities when summing the combined effect of fields from multiple coils [38]. Nonlinearities make modeling the behavior of the system more difficult [110], particularly when the coils are run with high enough power to magnetically saturate the core material. Additionally, when controlling microrobots with permanent magnets, those magnets are often modeled as dipole sources for simplicity [111], and the actual behavior of the physical system may not match the idealized model behavior. Neural network-based controllers trained with RL learn control policies from observing the actual behavior of the physical system, and deep neural networks can accurately model non-linear functions [112]. Control policies learned with RL will automatically take into account the real system dynamics, and this model-free control approach can greatly simplify the job of the microrobotic engineer.

Many microrobotic systems are composed of soft, shape changing materials, which are inherently harder to model than rigid bodies [39, 59, 64-66]. Here, we have shown that our algorithm was able to control a soft helical microrobot without any dynamic modeling on the part of the control system designers. Other algorithms which have been used to control soft microrobots such as force-current mapping with PID control and path planning algorithms [18, 59, 110] could potentially be combined with reinforcement learning in order to optimize the gains in the PID controllers, and adapt to changes in environmental conditions by a process of continuous learning, or to optimize for multiple variables. Force-current mapping algorithms used to control microrobots are also often created with assumptions of linearity in magnetic field addition, which could be violated with soft magnetic cores in the driving coils [74].

While the sine and sine/square policies that we created based on analyzing the learned policies might have been arrived at by a first principles analysis of the problem, this is certainly not the case for the polynomial policy we derived from the image-based input policy, which does not map to our intuitions of how to actuate a magnetic helical microrobot. We believe that this provides strong support for the idea that a deep neural network trained to control microrobots with reinforcement learning is likely to arrive at policies that are unintuitive and could potentially uncover useful behaviors which would not be suspected or created by human engineers. Furthermore, our analysis of the direction of the rotating magnetic fields created by the learned policies strongly supports the idea that the RL agent reliably developed near-optimal behavior which matches the behavior of a rationally designed controller. This suggests that if RL were applied to a more complex microrobotic system for which no good models of optimal behavior were available, the RL agent could be able to autonomously identify the best way to control the system. This ability to detect subtle patterns from high dimensional data could ultimately lead to state-of-the-art control policies that exceed the performance of human designed policies, as has been seen with reinforcement learning algorithms in domains like Go [96], and classic Atari games [107].

In our experimentation, we found that the RL agent could learn successful polices from both state vector input, and from raw camera images. This input flexibility demonstrates that RL could be applicable for a broad class of biomedical imaging modes in which the state of the system might be represented by MRI, X-ray, ultrasound, or other biomedical imaging methods [69]. Our results are consistent with the findings of Haarnoja et al. [104], in that the use of raw images as input requires more training time in order to develop high quality policies compared to state vector input. Using higher dimension input like images has the potential to encode richer policies which respond to objects in the field of view such as obstacles which could impede the forward progress of the microrobot but would not be observable from lower dimensional feedback available in a state vector representation. In complex environments in which environmental factors such as lumen shape, fluid flow profiles, surface interactions, and biological interactions are likely to be a significant factor [12], the ability to use machine vision for state representation could significantly improve microrobot performance. All these points strongly favor the use of RL for developing the next generation of microrobot control systems.

3.4 Methods

3.4.1 Helical Agar Magnetic Robot

The HAMR was constructed based on a method published by Hunter et al. [101]. The structure of the robot was formed from a 2% w/v agar-based hydrogel (Fisher Cat. No. BP1423-500). The agar was melted to above 80 degrees Celsius and mixed with iron oxide nanopowder (Sigma Aldrich Cat. No. 637106) to a total concentration of 10% w/v. This mix was injected into a helical 3D printed mold printed on the Elegoo Mars stereolithography 3D printer to form a helical microrobot 4.4 mm in length. The microrobot was manually removed from the mold after cooling and solidifying and stored in deionized water until use. Because the yield of this batch fabrication technique was not 100%, robots used for subsequent experiments were chosen based on their morphology and responsiveness to magnetic fields.

3.4.2 Circular Swimming Arena

The PDMS swimming arena was created by molding Sylgard 184 elastomer (Sigma Aldrich Cat. No. 761036) over a thin 3mm tall section of polyvinyl chloride pipe (31 mm Inner Diameter, 34mm Outer diameter). Access holes for the microrobot were cut, and then the molded PDMS was plasma bonded to a thin uniform sheet of PDMS to close the channel, and cured overnight at 65° C.

3.4.3 Magneturret

The Magneturret was constructed by winding 6 identical coils with 400 turns each of 30-gauge magnet wire (Remington Industries Cat. No. 30H200P) around a 0.26-inch diameter permalloy core (National Electronic Alloys Cat. No. HY-MU 80 Rod .260 AS DRAWN) cut to a length of 20 mm. These coils were fixed to the sides of an Acrylonitrile butadiene styrene (ABS) 3D printed cube with quick set epoxy. The coil was enclosed in a 3D printed housing printed in Zortrax Z-glass filament with a Zortrax M200 printer and sealed with epoxy. Glycerol coolant was pumped through the housing with a liquid CPU cooling system (Thermaltake Cat. No. CL-W253-CU12SW-A). The coils were energized by creating sinusoidal currents with an Arduino STEMtera breadboard, which took serial commands from the RL agent over USB and turned them into PWM signals which were sent to two Pololu Dual G2 High-Power Motor Driver 24v14 Shields. The power supply used to power the coils and was a Madewell 24V DC power supply.

3.4.4 Overhead Camera

The overhead camera was an Alvium 1800 U-500c with a 6mm fixed focal length lens from Edmund Optics. The camera used to take images for the state was set at a long exposure so that the HAMR and the center mark were the only visible objects in the image. A second identical camera placed above the arena at a slight angle was used to simultaneously record normal exposure video of the HAMR in the arena during operation, so that the features in the image were not washed out.

3.4.5 RL Algorithm

The soft actor critic RL agent was developed in Python, using TensorFlow 2.0 for creating the neural network models. This was run on a desktop workstation from Lambda Labs. Separate processes were used for data collection and updating the neural networks so that the two operations could run in parallel. The full algorithm details are available in Supplementary Algorithm 1.

3.4.6 Supplementary Materials

The following supplementary materials are available in Appendix A:

- Fig S1: Highest performing policy parameters during training
- Appendix Table 1: Hyperparameters
- Supplementary Algorithm S1: Soft-actor-critic for microrobot control
- Movie S1: Training process time course.
- Movie S2: HAMR swimming around the track

4.0 Hardware for Creating Biomimetic *In Vitro* Environments for Microrobot Development

The tools and techniques for creating smart robotic control system presented in chapter 3 face several engineering obstacles before they can be efficiently used for therapeutic microrobot applications. So far, we have studied the use of reinforcement learning for microrobot control in a very simple model system, namely, a helical microrobot constrained in a circular track under static fluid conditions. While this system was valuable for testing and developing the use of RL for microrobot control, ultimately the microrobots controlled by the system will be asked to operate in significantly more complex environments. As the next step towards making RL control of microrobots *in vivo* a viable technology, we need to develop biomimetic environments for microrobots in vitro, where they can be developed and evaluated [113]. The RL-based control system is likely to perform better if it is trained in an environment that as closely as possible matches the environment in which operation will occur [103].

In the environments *in vivo* in which microrobots will be called upon to operate (blood vessels, cerebrospinal fluid, etc.), the fluid around the robot will not be static [12]. Therefore, one of the next critical steps towards fully developing an RL based microrobot control system is to evaluate the performance of the microrobot in the presence of dynamic fluids which closely match the properties of the fluid flows *in vivo*. To create biomimetic environments with biologically relevant fluid flow, we have developed a fully programmable pump for precisely actuating fluids in microscale in vitro environments. This pump can be produced at low cost and is highly applicable for a wide range of microfluidic applications, particularly in resource constrained settings. The rest of this chapter highlights this tool and is published as Behrens et al. "Open-source, 3D-printed peristaltic pumps for small volume point-of-care liquid handling", Scientific Reports. 2020 [114].

4.1 Introduction

Microfluidic systems are ubiquitous tools within science and engineering laboratories around the world that enable low-cost and high throughput analysis via the miniaturization and parallelization of experimental systems. To enable broader applications and lower the barrier to entry for using microfluidic technology, developing open-source and low-cost tools for handling fluids is a promising avenue of research [115-117]. Open-source microfluidic tools could be particularly impactful when used for point-of-care diagnostics in resource limited settings. The growing use of microfluidics in point-of-care diagnostic roles is driven by the desire for more personalized medical treatments that are tailored to the specific pathologies identified in the patient [118-121]. This is because assays that can be performed at the point of care dramatically improve time-to-diagnosis, leading to improvements in medical practitioner decision making and patient outcomes [122]. In order for point-of-care diagnostic devices to be widely adopted into more clinical settings so that they can effectively improve healthcare outcomes, advances must be made in key enabling technologies, including improved microfluidic device designs, and improved peripheral systems for fluid actuation and sensing [123]. Increasing the use of open-source tools for future point-of-care diagnostic tools would enable reductions in system cost and increase ease of use.

Microfluidic point-of-care diagnostic systems must include three basic components: the physical microfluidic device, systems to read the assay output, and systems to control the flow of liquid reagents (Figure 4-1A). The physical microfluidic device can be manufactured from a wide range of low-cost materials, including PDMS [124], glass [125], plastic [126], and paper [127-129], and commonly takes the form of a single use disposable cassette [118]. To read the output from the assay, sometimes expensive computer systems, sensors and microscopes are used, while some assays are designed to be read out by visual inspection [129].

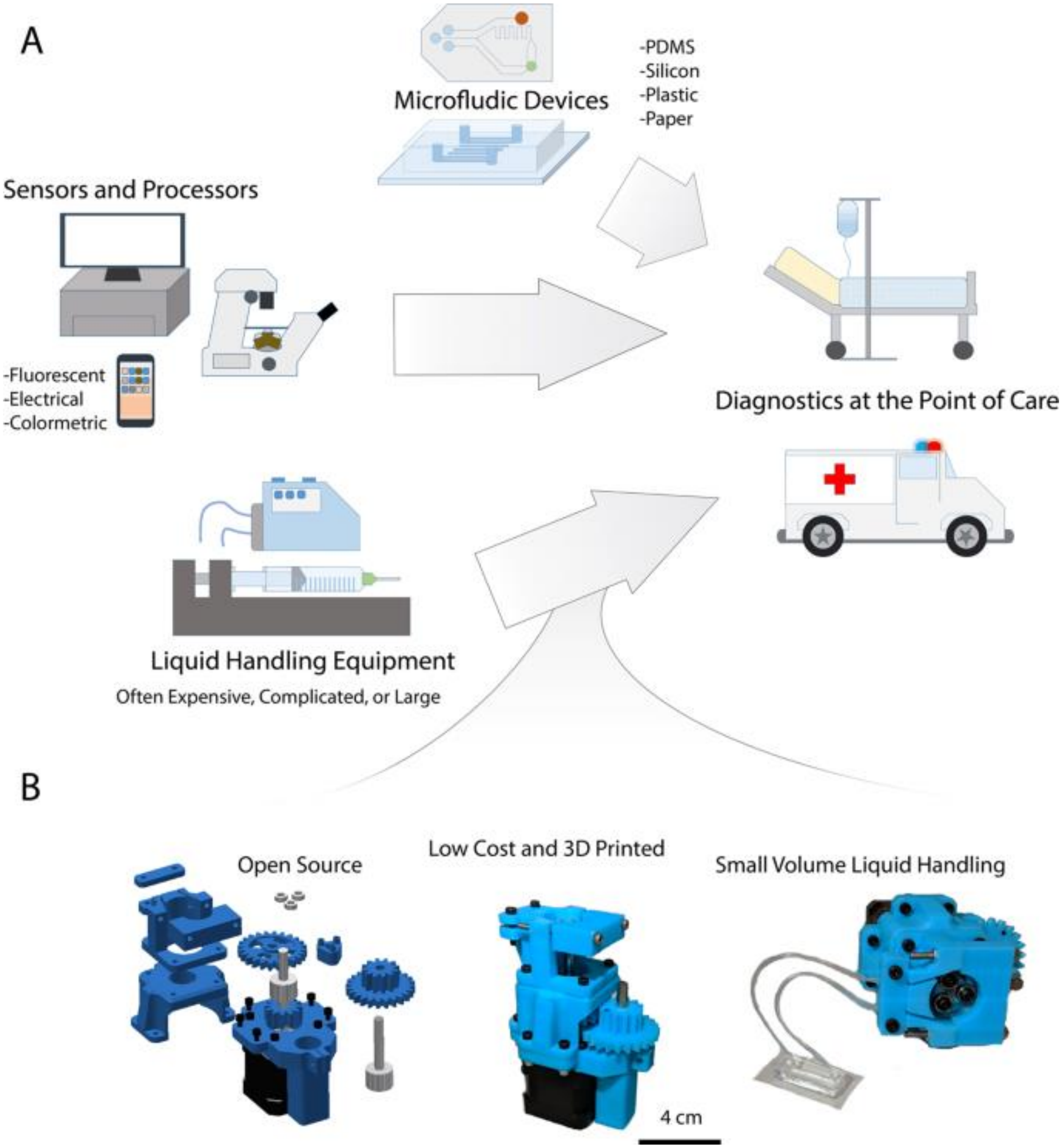


Figure 4-1. Liquid Handling for Point-of-Care Diagnostics. (A) Diagnostic devices for point-of-care applications often require integration of microfluidics with complicated and expensive peripheral equipment for signal readout and for liquid handling. (B) As a low-cost alternative to complex and expensive liquid handling equipment, we have developed an open-source, 3D-printed programmable peristaltic pump that can be used for precision low volume liquid handling and deployed with point-of-care diagnostic tools. The pump is assembled with a combination of 3D-printed parts and commonly available hardware, and is programmable via an Arduino microcontroller.

Most microfluidic systems require precisely controlled fluid flow. While there are microfluidic systems that employ sophisticated on-chip pumps for liquid handling [130-134], these devices are often complex to manufacture, expensive, or limited to low flow rates, which may limit their potential for mass adoption in point-of-care diagnostic systems. Instead, most microfluidic devices employ off-chip pumps to provide pressure and flow rate control. Many innovative off-chip pumps have been developed for microfluidics, including simple, ultra-low cost (< $10) reinforced latex balloons inflated to provide a steady pressure source [135], more expensive (< $1000), yet programmable, low volume piezoelectric pumps [136], and costly (> $1000), but sophisticated and robust syringe pumps [136, 137]. Syringe pumps, in particular, are among the most common type of pump used for laboratory microfluidics. Syringe pumps can be set up to deliver constant flow rates, or constant pressure if feedback control is employed [115]. However, syringe pumps are not without disadvantages for many applications. Syringe pumps are limited to dispensing fixed volumes of liquid limited by the syringe volume, and they are not capable of driving recirculating flow within a closed system. Additionally, common syringe pumps used in scientific laboratories, such those produced by companies like Harvard Apparatus, can be prohibitively expensive for adoption in low resource clinical settings. Peristaltic pumps provide an alternative to syringe pumps which are better suited for some applications. These devices pump liquid by cyclically squeezing a flexible tube against a rigid housing [138]. Unlike syringe pumps they can be used drive recirculating flow in a closed fluidic circuit, and they are physically isolated from the sample by the walls of the tubing, which reduces contamination and safety concerns. Additionally, peristaltic pumps are well suited for dosing and metering fluids [139], tasks for which they are widely employed for large volume liquid handling applications. Peristaltic pumps are employed in many microfluidic applications [140-142], particularly those requiring recirculating flow for cell culture [141]. Several classes of on-chip integrated microfluidic peristaltic pumps have been developed, in which channels within a PDMS microfluidic device are squeezed by external hardware such as rollers [130], magnets [133, 134], or by adjacent pressurized microfluidic channels [124, 143] . While integrated pumps allow more precise control of small liquid volumes, integrated pump designs constrain the geometry of the microfluidic device. Additionally, integrated pumps often require complex fabrication procedures when compared to microfluidic devices powered by off-chip peristaltic pumps. However, commercial, off-chip peristaltic pumps designed for microscale liquid handling, similar to commercial syringe pumps, can be prohibitively expensive for many point-of-care diagnostic systems.

Here, we present the design of a low-cost (~$120) open-source, 3D-printed peristaltic pump that can be manufactured using common tools and hardware, that is designed for microliter-scale liquid handling and amenable to deployment with diagnostic microfluidic systems (Figure 4-1B). This pump operates by peristaltic action of rolling ball bearings applying pressure to the outside of small diameter tubing. It is modular, able to accept a range of tubing diameters by quickly swapping out 3D-printed parts. The pump is driven by a stepper motor and power is transmitted to the pump through a 3D-printed 4:1 gearbox to increase the available torque. The motor is programmable via an open-source microcontroller (Arduino) to precisely control flow rate and direction. Here, we characterize the performance of this pump and demonstrate its applicability for a variety of small volume precision liquid handling applications.

4.2 Results

4.2.1 Pump Design

The system architecture of the pump is subdivided into two basic subsystems: The electrical system, containing the power supply, microcontroller and motor driver, and the mechanical pump, based on a stepper motor and acrylonitrile butadiene styrene (ABS) 3D-printed components, and assembled with basic hardware. A complete bill of materials and assembly instruction set is provided in the supplementary materials (Appendix Table 2. Bill of materials, Appendix Figure 8. Peristaltic pump assembly instructions.). An Arduino microcontroller acts as the onboard central processing unit of the pump and can be programmed using the free Arduino integrated development environment (IDE) via a USB connection to a personal computer. The microcontroller is responsible for sending step and direction commands to a EasyDriver stepper motor driver, which controls the rotation speed and direction of the pump rotor. Power is transmitted to the pump rotor via a 3D-printed 4:1 gearbox that is mounted to the stepper motor. The pump mounted on top of the gearbox is composed of two key components: the rotor, and the stator. The rotor body is 3D-printed, and three ball bearings are secured to the rotor in a triangular configuration. These form the rollers which provide force on the tubing during pumping. The stator is composed of three 3D-printed pieces: a base which is specific to the tubing diameter being used, and two clamps that secure the tubing in place. The tubing is clamped between the rotor and the stator, so that when the rotor rotates the ball bearings squeeze the tubing against the stator. Peristalsis is achieved by cyclical compression of the tubing by the rotor, which provides force which drives fluid through the tubing (Figure 4-2A).

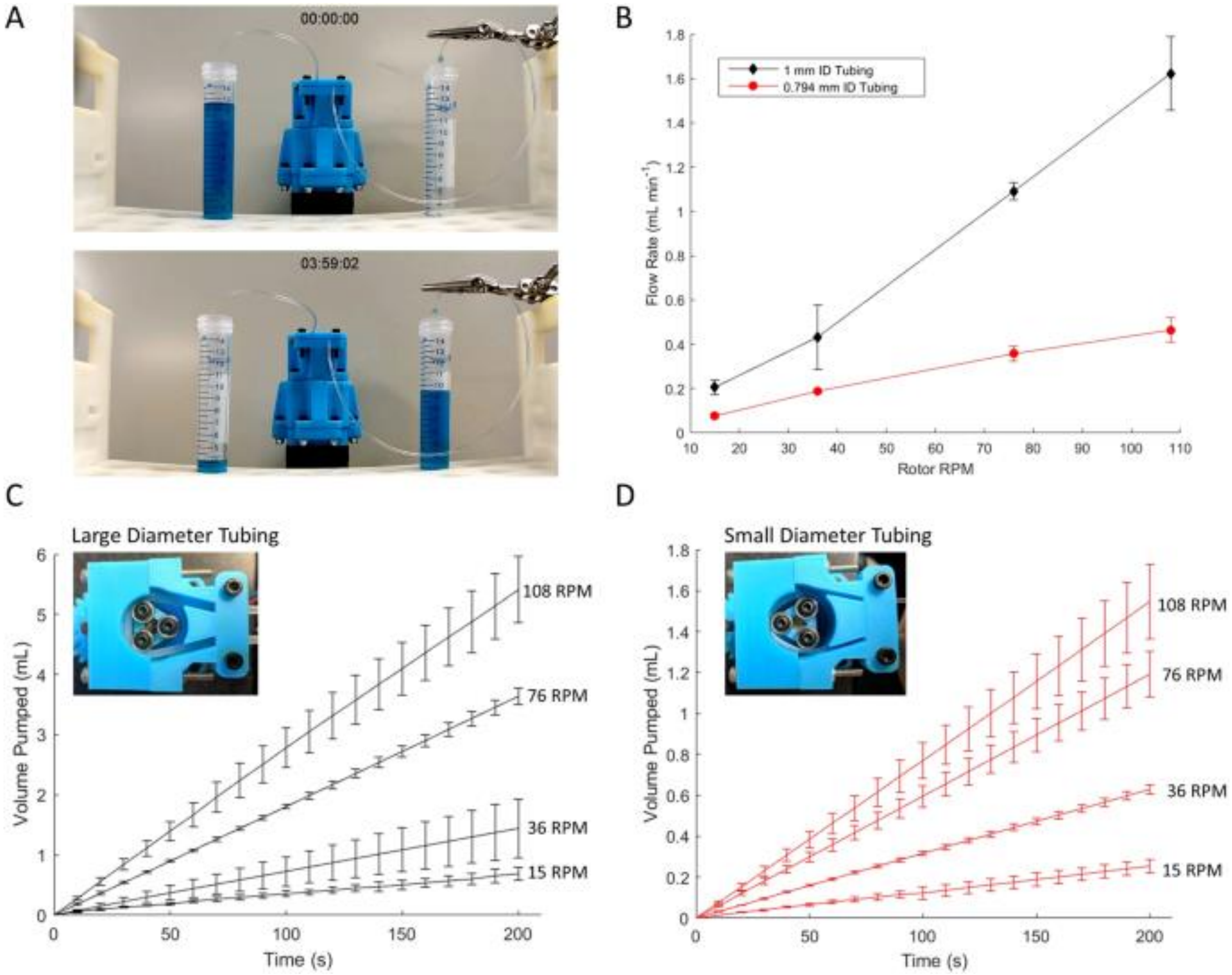


Figure 4-2. Flow rate. (**A**) The pump uses peristaltic motion of rotating ball bearings with silicone tubing to transfer fluid. (**B**) Flow rate is controlled by varying the rotation speed of the pump, or by varying the diameter of the tubing. (**C**) Total volume pumped over time with 3 mm OD, 1 mm ID tubing, at four different pump rotation speeds. (**D**) Total volume pumped over time with 1.59 mm OD, 0.79 mm ID tubing. Traces represent the average results for three independent trials, with error bars representing one standard deviation above and below the mean.

4.2.2 Flow Rate

The rate of fluid flow through the pump is controlled by angular velocity of the rotor, which is encoded into the commands delivered to the motor controller by the Arduino, and by the diameter of tubing used. The pump was tested for two different sizes of silicone tubing: 1.59 mm (0.0625”) outer diameter (OD), 0.79 mm (0.03125”) inner diameter (ID) tubing, and 3 mm OD/1 mm ID tubing. Each tube had a durometer hardness rating of 50 A, designed to be used with peristaltic pumps. To switch between the two sizes of tubing, two different sizes of stator were designed and 3D-printed and can be quickly swapped out to modulate the pump to accommodate different tube diameters.

To test the pump rate as a function of rotation speed, the pump was set up to pump deionized water onto a precision analytical balance. The balance was programmed to record real time weight measurements once per second. The pump rotation speed was determined by counting the number of rotations over a period of 2 min. Results of this experiment show that the pump rate is dependent on both the tubing diameter, and on the rotation speed of the pump (Figure 4-2B). For a given rotation speed and tubing size, the pump maintains a constant average pump rate, resulting in a linear increase in pumped fluid volume over time (Figure 4-2C, D). Friction constraints arising from manufacturing tolerances limit the practical rotational speed to approximately 108 revolutions per minute (RPM), at which speed the pump achieves 0.46 mL min−1 with the 1.59 mm tubing, and 1.62 mL min−1 with the 3 mm tubing. Variation in pump rate can be observed when tubing is clamped into the stator with differing levels of tension applied to the tubing, which accounts for the majority of variation across trials. Once tubing is securely clamped in place, the pump rate is highly uniform (Appendix Figure 3). Furthermore, since the hydrodynamic pressure drop in microfluidic channels is generally very large, we characterized the flow rate vs pressure drop generated by the system, and also calculated the power consumption and efficiency of our system. These performance benchmarks are depicted in Appendix Figure 6 and Appendix Figure 7.

4.2.3 Pressure Generation for Microfluidics

This pump was designed for use in microfluidic applications to provide a low-cost tool that can be used for diagnostic purposes. One major application in microfluidic systems is control over fluid pressure within microfluidic channels. Applications of pressure driven flow include fluid based digital logic [124], actuation of on-chip valves [131, 143], and control of laminar flow interfaces between fluids in a channel [115]. In order to test the ability of this pump to generate pressure for microfluidic applications, we used a Y-channel microfluidic device to track the position of a laminar flow interface over time. Controlling the position of a laminar flow interface between two co-flowing fluids in a single channel is a technique commonly used for applications such as controlling molecular diffusion [144], patterning domains of small molecules within cells [145], and for microfabrication within capillaries [146]. We tested the ability of our pump to generate stable laminar flow interfaces, and tested the dynamic pressure produced by the pump. The pump was attached to the inlet of one channel in a Y-channel PDMS microfluidic device, and a syringe suspended above the microfluidic device was attached to the other inlet (Figure 4-3A). The pressure at the syringe inlet can be calculated based upon the height of the liquid in the syringe above the microfluidic device according to the equation $P = \rho g h$, where P is the pressure at the inlet, ρ is the density of the liquid, g is the acceleration due to gravity, and h is the height of the reservoir. An electric circuit analogy for microfluidic circuits was employed to calculate the pressure generated by the pump given the known pressure at the other inlet [147]. For fluid dynamics in low Reynolds number regimes such as those within a microfluidic channel, a simple relationship between flow rate and pressure can be derived from Hagen–Poiseuille's law: $\Delta P = QR_H$, where ΔP is the change in pressure over distance within a microfluidic channel, Q is the volumetric flow rate in the channel, and R_H is the hydraulic resistance of the channel, which can be calculated from the channel geometry. For a rectangular microfluidic channel where the height of the channel is much less than the width, R_H can be approximated by $R_H = 12\eta L/wh3$, where η is the viscosity of the fluid, L is the channel length, w is the channel width, and h is the channel height. The simplified equation relating pressure and flow rate in microfluidic chips is an analog to Ohm's Law: $\Delta V = IR$, which is extensively used in electrical circuit analysis. Accordingly, the mathematics of circuit analysis, including Kirchhoff's voltage and current laws, can be applied to microfluidic systems to determine pressures and flow rates within the system [147]. Change in pressure is analogous to change in voltage, volumetric flow rate is analogous to current, and hydraulic resistance is analogous to electrical resistance. Applying this analogy, an equivalent circuit model for the microfluidic device used in this experiment is presented in Figure 4-3A. By observing the position of the laminar flow interface of two parallel fluid flows within the channel, it is possible to determine the relative flow rates of the two liquids according to the following relationship: $\frac{w_1}{w_2} = \frac{Q_1}{Q_2}$ [147]. Employing this relationship in the circuit analysis of the system allows calculation of the unknown pressure generated by the pump, when compared to the known pressure due to gravity at the other inlet. The average pressure generated by the pump is a function of pump rotational speed (Figure 4-3B). As the pump rotation speed is increased, the fraction of the flow in the microfluidic channel downstream of the Y-junction that is supplied by the pump increases, while the fraction of the flow supplied by the elevated fluid reservoir decreases (Figure 4-3C). While the average pressure generated by the pump for a given rotational speed is constant, the dynamic pressure generated by the pump is not steady over time, but exhibits a cyclical fluctuation. As the rotor applies pressure to different parts of the tubing when turning, this generates a cyclical fluctuation in fluid pressure, which can be observed by tracking the position of the laminar flow interface in the channel over time (Figure 4-3D). The dominant frequency of this oscillation can be calculated as angular frequency of the rotor multiplied by the number of rollers on the rotor (Appendix Figure 4. The frequency of pressure oscillations from the pump is a function of pump RPM.).

4.2.4 Programmable Precision Liquid Handling

As microfluidic systems expand into a growing number of application areas, the requirements for fluid pumping profiles will certainly expand as well. This pump was designed to be open-source and reprogrammable, in order to simply and rapidly modify the characteristics of the driven flow to meet operator-defined requirements (Figure 4-4A). One application for which peristaltic pumps are commonly employed is liquid metering, so we tested the ability of our pump to precisely and repeatedly dispense aliquots of liquid. First, a calibration test was performed by clamping a 1.59 mm tube into the pump and observing the flow rate by pumping deionized water onto a precision analytical balance at a constant rotor RPM. The resultant pump rate was used to calculate the time required to dispense a given volume of liquid. Tests were performed by pumping 100 aliquots each of 10 μL, 30 μL, and 50 μL onto the analytical balance. Additionally, 50 μL was individually dispensed into 8 PCR tubes to visually confirm aliquot volume (Figure 4-4B).

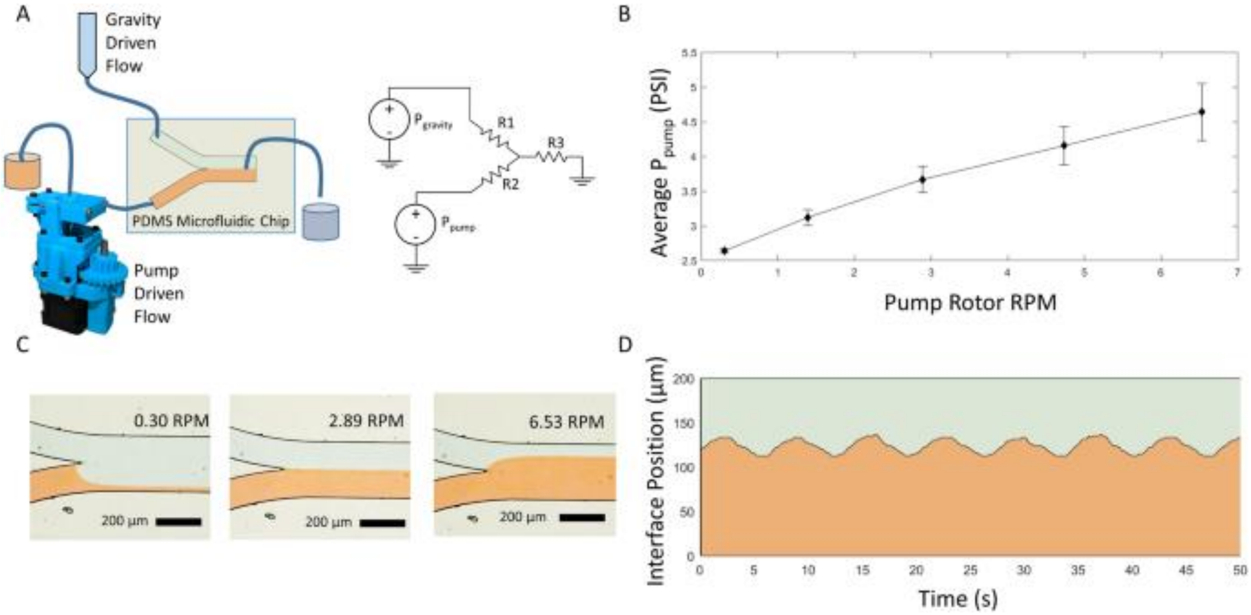


Figure 4-3. Pressure driven flow for microfluidics. (A) The peristaltic pump was set up to pump water supplemented with red food coloring through a PDMS microfluidic device with a Y-channel configuration. A syringe with water and blue food coloring was attached to the other inlet of the Y channel, and suspended 2.47 m above the microfluidic device to create a constant pressure source. Using an electrical circuit analysis analogy for microfluidics, the unknown pressure generated by the pump was calculated by observing the position of a laminar flow interface, and the known pressure from the gravity driven flow. (B) Pressure supplied by the pump is a function of rotor RPM. Data represent the results of three independent trials, and error bars represent one standard deviation above and below the mean. (C) The position of a laminar flow interface between the water from the pump and the water from the syringe was tracked to calculate the pressure generated by the pump. (D) The position of the laminar flow interface within the channel oscillates over time due to the cyclical application of force to the silicone tubing by the rotor.

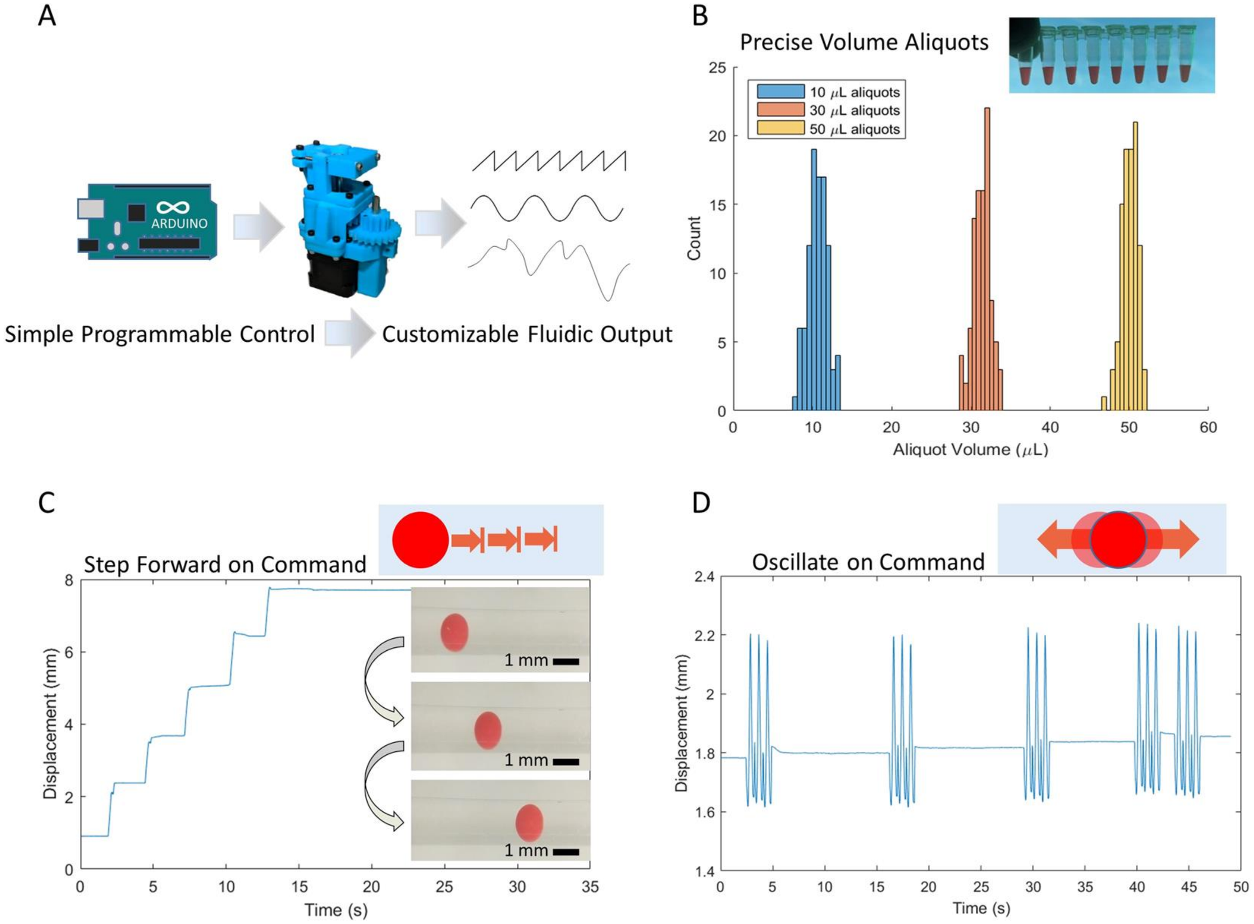


Figure 4-4. Programmable functions for custom liquid handling. (**A**) The pump is programmed with an Arduino microcontroller, which allows for user-defined arbitrary pumping behaviors. (**B**) The pump can be programmed to reliably and repeatedly dispense precise volumes of liquid. Histograms represents the volumes for 100 aliquots of deionized water at three different aliquot sizes. (**C**) The pump was programmed to drive a set volume of water through a 1 mm glass capillary under control of a user-operated switch. A cellulose acetate (CA) plastic sphere was placed in the capillary to aid in visualization of the moving fluid column, and the position of the sphere was tracked visually. (**D**) The pump was set to oscillate the column of water back and forth under user-defined control, and the movement of a CA sphere within the capillary was visually tracked.

Next, we expanded the versatility of our pump, the pump was reprogrammed to perform customized tasks under control of user input via rocker switches installed in the electronics box. Two custom programs were implemented: one to step forward, pumping fluid a specified amount before coming to rest (Figure 4-4C), and one to oscillate the fluid back and forth (Figure 4-4D). Example programs to run the pump are included in the Supplementary Materials (Appendix B.4). To test these programs, the pump tubing was attached to a water filled glass capillary and a spherical cellulose acetate bead was placed in the capillary to visualize movement of the liquid column. The capillary was placed under a stereo microscope and the bead was observed with a camera as the operator toggled the switch on the electronics box to trigger the programmed response from the pump. The movement of the cellulose acetate bead within the channel was used to visualize the flow of liquid within the channel. Within the glass capillary, the low Reynolds number (≈1) means that viscous forces dominate over inertial forces, so the movement of the bead within the fluid closely tracks the movement of the column of fluid.

4.2.5 Endothelial Cell Structural Response under Laminar Fluid Shear Stress

We also tested the pumps' ability to function as a tool for liquid handling in biological applications, we used the pump to generate laminar fluid shear stress on human vascular endothelial cells (HUVECs). Endothelial cells have been well documented to be responsive to fluid shear stress [148]. In fact, their alignment and change in morphology under laminar fluid shear stress is believed to protect the endothelium from many cardiovascular diseases and in general be barrier protective. For this study, we exposed a monolayer of HUVECs within a PDMS microfluidic channel to laminar fluid shear stress of 0.1498 Pa for 24 h. The endothelial cells exhibited a non-preferred orientation at the beginning of the experiment (Figure 4-5A–C), while endothelial cells began to exhibit a more elongated shape and alignment along the direction of fluid shear at 24 h (Figure 4-5D–F).

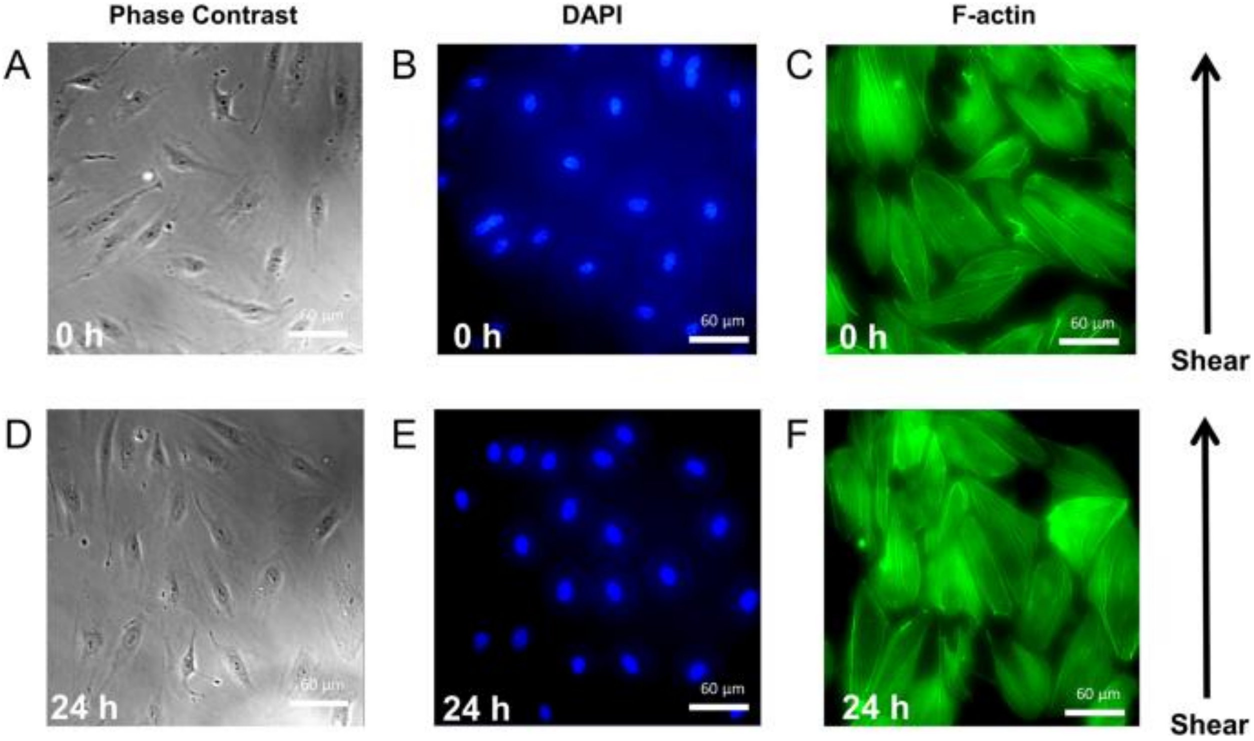


Figure 4-5. Laminar fluid shear stress causes alignment of endothelial cells. Cells in a monolayer within a microfluidic channel were exposed to shear stress from laminar flow generated by the pump, at 0.1498 Pa. Phase contrast and fluorescent images of Nucleus (DAPI) and F-actin at 0 h (**A**–**C**) and 24 h (**D**–**F**) of fluid shear stress.

4.3 Discussion

We have presented an open-source peristaltic pump built from 3D-printed plastic parts and common hardware and demonstrated its applicability for small volume liquid handling in microfluidic applications, including generation of laminar flow interfaces and controlled sample movement under operator control. This pump represents a simple and inexpensive alternative to commercial pumping systems for microfluidics, with a number of key advantages. Unlike many of the comparable inexpensive peristaltic pumps on the market, this pump represents a complete liquid handling system, with its own power and control systems. The total price of the physical pump is approximately half the cost of the entire system, ≈$65, which is the number to compare to microfluidic pumps on the market with comparable performance specifications, such as the microfluidic peristaltic pump produced by Dolomite Microfluidics, which retails at $220 without power and control hardware, or the mp6 piezoelectric pumps produced by Bartels Mikrotechnik and sold through Servoflo, which costs $950 for a starter package. Although our pump is not the lowest cost option that has been developed for fluid handling in microfluidics (e.g., $2 reinforced latex balloons were developed by Thurgood et al. to provide a constant pressure source [135]), our design strikes a balance between reducing cost and enabling programmable flow profiles that are suitable for a wide range of applications. Furthermore, the open-source design allows for simple modifications to adapt the pump to application specific constraints such as tubing diameter, which affords this pump greater application flexibility in comparison to similarly priced commercial systems. Additionally, the 4:1 gearbox developed for this pump accepts attachments for a NEMA17 specified stepper motor, so it can be adopted for any other projects which require additional torque from a stepper motor. Programmability via the Arduino IDE allows for fine control over flow profiles, and integrated programmable switches in the electronics box can be easily programmed to control user-specified pumping routines, including stepping and oscillating flow. Fluids propelled by the pump are insulated from the hardware by the tubing, so sterility and contamination concerns are minimized in this system. Additionally, the peristaltic pump can be used to drive recirculating flow in a closed system, or to draw fluid from an arbitrarily large reservoir, which are key advantages over syringe pumps, one of the most common methods of microfluidic fluid control. Previously, our lab has presented a design for a 3D-printed feedback-controlled syringe pump that can act as a stable pressure source which may be better suited for applications which require precisely controlled pressure driven flow [115]. Together, these two pumps form a range of fluid handling technologies that are applicable to a wide range of point-of-care microfluidic applications.

We demonstrated the pump's ability to be used for biological applications, and thus its feasibility as a point-of-care diagnostic system, by deploying our pump in conjunction with a microfabricated PDMS microchannel to exert a laminar fluid shear stress on a confluent HUVEC monolayer for 24 h. Our results revealed that (1) our pump can be used in conjunction with standard devices (such as a flow chamber) used in the biomedical field to study biological cells for a prolonged period of time, and (2) our pump was able to induce a laminar fluid shear stress-induced alignment of endothelial cells. Although, other groups have demonstrated the latter before, the uniqueness of this portion of our study may be found in the fact that the results we present here can be executed at a considerably lower operational cost and in the fact that this experiment utilizes materials that are much more accessible. We therefore believe our device will allow others who may be interested in running similar experiments to do so in previously inaccessible resource limited settings.

4.4 Methods

4.4.1 Design and Construction of the Peristaltic Pump

All 3D-printed components were designed in 3D-computer aided design (CAD) software (Autodesk Inventor Professional 2016). The designs were exported as STL files and prepared for 3D-printing with Z-Suite (Z-SUITE Ver. 2.11.1.0, Zortrax), and parts were printed in Z-ABS plastic using a 3D-printer (Zortrax M200). The pump was assembled with a combination of commonly available hardware and 3D-printed components, using common tools. All 3D-printed parts are made available as STL files in Appendix B.5, and a full bill of materials outlining the required hardware is also supplied in Appendix Table 2. Bill of materials A detailed set of instructions for assembly is also included in the Supplementary Slide Deck in Appendix B.3, and an electrical schematic is included in Appendix Figure 5.

4.4.2 Design and Construction of Microfluidic Chips

Microfluidic devices were prepared by defining the channel geometry in 2D CAD software (Autodesk AutoCAD). These designs were then used as a mask for soft lithography. The CAD design was transferred to a maskless aligner (Heidelberg MLA100 Direct Write Lithographer), and lithographically patterned into SU-8 photoresist that was spin-coated onto a silicon wafer. Poly dimethlysiloxane (PDMS) (Sylgard 184, Dow Corning) was mixed at a 10:1 base to curing agent ratio and poured over the wafer. The PDMS was degassed in a desiccator under vacuum pressure and cured in an oven at 65 °C for 1 h. The PDMS was then removed from the wafer and sliced into individual microfluidic devices. Inlet and outlet holes were punched in the devices using a blunt tipped stainless steel dispensing needle (Cat. No. 75165A675, McMaster Carr Supply Company). The devices were then oxidized in a plasma cleaner (PDC-32G, Harrick Plasma) for 30 s and plasma bonded to glass cover slips (22 × 40 mm). The devices were then incubated overnight at 65 °C to ensure a tight bond between the glass and the PDMS.

4.4.3 Fluid Flow Rate Determination

Flow rate was calculated by pumping deionized water into a weigh boat on an analytical balance (Adventurer AX, Ohaus) through silicone rubber tubing. Two sizes of tubing were tested: 1.59 mm OD tubing (Cat. No. 5236K204, McMaster Carr Supply Company), and 3 mm OD tubing (Cat. No. 5054K304, McMaster Carr Supply Company). Data was recorded via serial connection to a personal computer and logged using SPDC Data Collection V2.03 (Ohaus). Weight measurements were recorded once per second. Data were analyzed in MATLAB. The rotor rotation speed was measured by counting revolutions over a period of 2 min.

4.4.4 Laminar Flow Interface in Microfluidic Channel

Silicone rubber 1.59 mm OD tubing (Cat. No. 5236K204, McMaster Carr Supply Company) was inserted into the pump. One end of the tubing was placed in a reservoir containing water supplemented with red food coloring (Red Food Color, McCormick Culinary), and the other end of the tubing was attached to one of the Y-inlets on the PDMS microfluidic device. The other inlet of the microfluidic device was connected to tubing running to a 10 mL syringe placed 2.47 meters above the microfluidic device, containing water and blue food coloring (Blue Food Color, McCormick Culinary). The outlet of the device was connected to tubing leading to an open waste collection container. Fluid was allowed to drain from the syringe through the device under gravitational power, while the speed of the pump was adjusted to vary the flow of fluid from the pump. The microfluidic device was placed onto the stage of a Nikon Eclipse Ts2 microscope and viewed through a 4X objective lens. Video was captured with a camera in the eyepiece of the microscope (Celestron Digital Microscope Imager HD 5MP). The position of the laminar flow interface was calculated using ImageJ digital image processing software by binary thresholding and tracking the position of the boundary of a binary object representing the flow.

4.4.5 Precise Volume Aliquoting

The pump was prepared with 1.59 mm OD silicone rubber tubing (Cat. No. 5236K204, McMaster Carr Supply Company) and set to run at 30 RPM. Flow rate was calculated by pumping deionized water into a weigh boat on an Ohaus Adventurer analytical balance through silicone rubber tubing, and the flow rate was used to calculate the time required to aliquot a specified volume of water. The pump was then programmed to run 100 cycles for the calculated length of time, pausing for 5 s between cycles. Weight data was recorded via serial connection to a personal computer, and data was logged using SPDC Data Collection V2.03 (Ohaus). Weight measurements were recorded once per second. Data were analyzed in MATLAB, and forward difference approximation was used to automatically identify intervals during which the pump was not running. The difference in weight between pauses was used to calculate the total amount of fluid pumped during the cycle. For pumping precise aliquots into individual PCR tubes, the pump was programmed to run for the specified interval when a switch was manually flipped. The dispensing end of the silicone tubing was placed in a new PCR tube for each aliquot.

4.4.6 Visualization of Operator-Controlled Functions

The pump was programmed with the Arduino IDE to generate custom responses to an operator-controlled switch. Fluid flow was visualized by placing a cellulose acetate 0.91 mm diameter red sphere (Cat. No. CAS-RED-1.3 0.91+/−0.05 mm-100, Cospheric) into a glass capillary with an inner diameter of 1 mm (Cat No. 13-678-20A, Fisher Scientific). The capillary was then connected to silicone rubber 1.59 mm OD tubing (Cat. No. 5236K204, McMaster Carr Supply Company), and water was pumped through the capillary. The capillary was placed under a stereo microscope (Carolina Biological Supply Company) at 2X magnification, and video was captured with a phone camera (Pixel 2, Google) viewing through the eyepiece. Movement of the cellulose bead was analyzed in ImageJ by converting the video to grayscale, thresholding to create a binary object, and tracking the coordinates of the centroid of the object over time. Small binary objects were filtered out to ensure that only a single object was tracked in each frame.

4.4.7 Cell Culture

Human umbilical vein endothelial cells (HUVECs) were cultured in medium 200 supplemented with large vessel endothelial supplement and 1% penicillin-streptomycin on 0.1% gelatin-coated flasks at 37 °C and 5% CO2.

4.4.8 Fluid Shear Stress

The PDMS microchannel was coated with 0.1% collagen I and incubated at 37 °C for 4 h. After this time, Collagen I was flushed out of the chamber with PBS. HUVECs were subsequently seeded into the chamber, which was placed into a tissue culture incubator at 37 °C and 5% CO2 for at least 12 h before the experiment. Experimentation consisted of HUVECs being exposed to a fluid shear stress of 0.1498 Pa for 24 h in an incubator.

4.4.9 DAPI-Phalloidin Staining

HUVECs were first fixed by slowly pipetting 4% formaldehyde into the microchannel and then incubated at 37 °C for 15 min, followed by cell permeabilization with 0.2% Triton-X 100 for 5 min at 37 °C. Alexa Fluor 488 phalloidin (1:20) was next added to stain for f-actin for 2 h at 37 °C. After this time, fluoromount-G with DAPI was pipetted into the microchannel and HUVECs were then imaged using a Zeiss Inverted microscope with a 40x objective.

5.0 Biological Applications of Microrobots: Targeted Drug Delivery and Biohybrid Microrobots

5.1 Targeted Drug Delivery with Magnetic Microrobots

Untethered swimming magnetic robots have undergone significant research and development as potential platforms to carry out therapeutic tasks within difficult-to-access fluidic regions of the body [11, 13]. One of the most enticing benefits of developing untethered magnetic robots for *in vivo* therapy is the potential to reduce unintended consequences and collateral damage patients suffer because of more invasive forms of therapy. One common treatment with unintended consequences is antibiotic therapy for treating bacterial infections. Antibiotics are one of the most successful innovations of modern medicine, and have saved countless lives by preventing and treating bacterial infections [149]. However, the liberal use of antibiotics common in today's medical practice is driving the emergence of antibiotic resistant bacteria [150] and can negatively affect the beneficial bacteria of the microbiome [3]. New classes of antibiotics are slow and difficult to develop, and so novel solutions that reduce the negative effects of current antibiotic treatments while retaining clinical efficacy is of great interest for medical practitioners and their patients.

Many antibiotic treatments prescribed today are designed to achieve a systemic effect throughout the entire body. While this approach is effective at eliminating infections, the collateral damage on the commensal microbiome is considerable, which can lead to further health complications [151]. Some methods for locally administering antibiotics have been developed, such as applying ointments for topical infections [152] or biodegradable materials that can be implanted at a wound site [153]. However, achieving a localized effect on infections deep inside the body remains difficult. Untethered magnetic robots operating *in vivo* are a promising technical solution to address this treatment gap. A robotic chassis incorporating magnetic materials could be loaded with antibiotics and magnetically driven through fluid-filled lumens within the body such as blood vessels [154], where the therapeutic cargo can be released precisely on target [55]. Towards this end, we have explored the use of the HAMR for Targeted Infection Mitigation and Elimination therapy (HAMR TIME therapy) which could reduce the damaging effects of systemic antibiotic treatment on the microbiome.

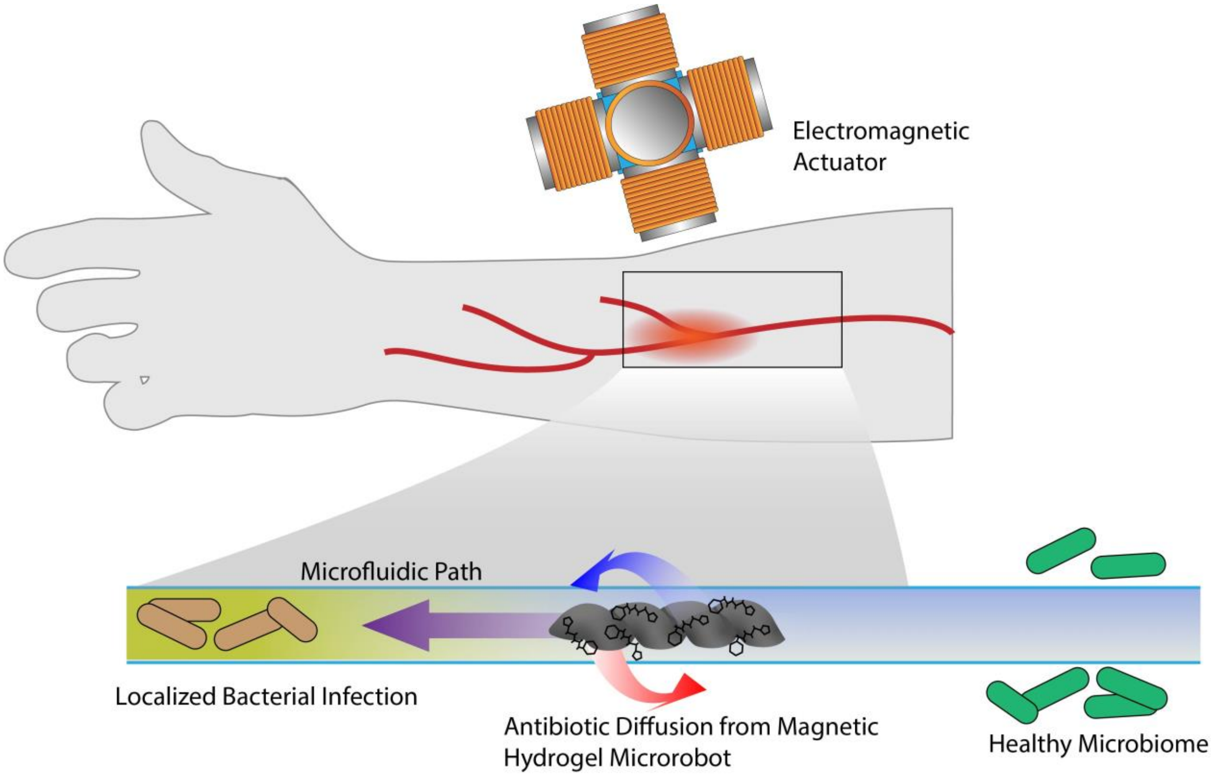


Figure 5-1. Targeted antibiotic delivery with magnetic microrobots. This conceptual diagram shows a possible way that magnetic microrobots could be used in a therapeutic context to delivery antibiotic to targeted regions in the body, where the drug can treat a localized bacterial infection, while reducing damage to the commensal microbiome.

5.1.1 Assessing Drug Loading and Delivery Capability of Agar Microrobots

Development of precise drug delivery systems is not a new area of research [4]. Significant advancements in drug delivery have come from the development of new materials for encapsulating drugs, and enabling spatial targeting and controlled release profiles [155]. Hydrogels, polymeric materials which exhibit significant levels of porosity and hold a large content of water, are an attractive material for localized drug delivery for several reasons. First and foremost, it is possible to create biocompatible hydrogels which are well accepted in the body without significant immune response [156]. Hydrogels, being composed of long chain organic molecules, can be tuned significantly to change their material properties such as stiffness, biodegradation rate, and drug release kinetics. Additionally, hydrogels can be composed of hybrid materials, incorporating elements such as magnetically susceptible iron nanopowder into their structure, which facilitates their use as magnetic microrobotic materials [101]. The HAMR robots originally presented in chapter 3 of this dissertation are composed of a magnetic hydrogel made of agar. Agar is a biopolymer derived from algae, and forms a biocompatible hydrogel, and is used for applications such as bacterial culture media. Agar is also used as a food addative in some parts of the world.

We have evaluated the use of the HAMR robots for use in antibiotic delivery applications, by loading the hydrogel structure of the robots with concentrated antibiotic and evaluating the release of the drug for preventing local bacterial growth. The first experiment we performed, to assess the feasibility of the HAMR as a drug delivery mechanism, was to test the ability of the HAMR to diffuse kanamycin into an Agar culture plate and locally inhibit bacterial growth (Figure 5-2). First, a diluted solution of MG 1655 wild type E. coli bacteria in LB media were uniformly applied to the surface of a nutrient agar plate with no antibiotics. The HAMR was incubated in a highly concentrated kanamycin antibiotic solution ((100mg/ml) for 2 hours under light shaking, in order to load the HAMR with antibiotic. Then HAMR was then removed, the excess liquid was removed from the surface of the HAMR, and it was placed on the petri dish with the E. coli. Outgrowth of E coli was observed after incubation overnight. The results of this experiment show that the antibiotic passively diffused from the hydrogel structure of the HAMR and created a zone around the HAMR where bacterial outgrowth was prevented (Figure 5-2B).

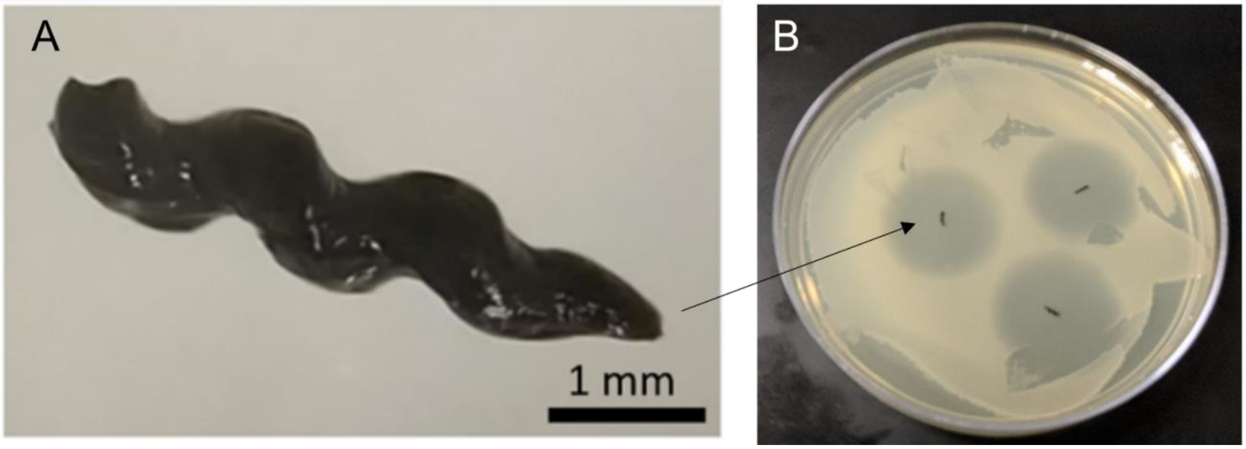


Figure 5-2. Antibiotic delivery with HAMR supresses growth of *E. Coli.* HAMRs were stored long term in DI water. Prior to antibiotic delivery tests they were incubated with concentrated antibiotic solution for 2 hours by placing the HAMR in a 1000x stock solution of kanamycin (100mg/ml) under light shaking. (B) HAMRs were then removed from the antibiotic solution, dried with a clean laboratory wipe to remove excess liquid, and deposited onto a freshly streaked agar plate with MG1655 wild type E. Coli. Results were recorded after overnight outgrowth of the bacteria at 37°C.

Having established the proof of concept that the HAMR can passively diffuse antibiotics and locally inhibit bacterial growth, we are currently undergoing experiments to quantify the release kinetics of antibiotics from the HAMR. These efforts are being led by Haley Fuller, a member of our laboratory. Results from preliminary studies to establishing a tetracycline release assay are presented in Appendix Figure 9.

5.1.2 Microfluidic Experiment for Antibiotic Delivery with Microrobots

We next evaluated the HAMR for use in HAMR TIME therapy in a biomimetic microfluidic model of bacterial infection (Figure 5-3A, B). For this analysis, we developed a magnetic actuator based on a rotating permanent magnet fixed to a mobile cart which could translate along a curved 3D printed track (Figure 5-3C).

In work led by Haley Fuller, we developed a U-shaped 3D-printed microfluidic device which allows us to create two fluidically linked bacterial colonies in agar motility media (Figure 5-3A). This model system allows us to recapitulate some aspects of bacterial infection, where the motility media can act as a biomimetic biofilm, and we can test the ability of the HAMR to penetrate the motility media under magnetic actuation. The two colonies of bacteria are inoculated in the same manner, by stabbing the agar with a bacteria laden needle, using MG 1655 wild type *E. coli* as the model infectious agent. The HAMR is the placed into the center of the U-shaped channel and is directed to swim towards one end of the U-shaped channel, and into the motility media. The two bacterial colonies are fluidically linked with an LB media filled channel.

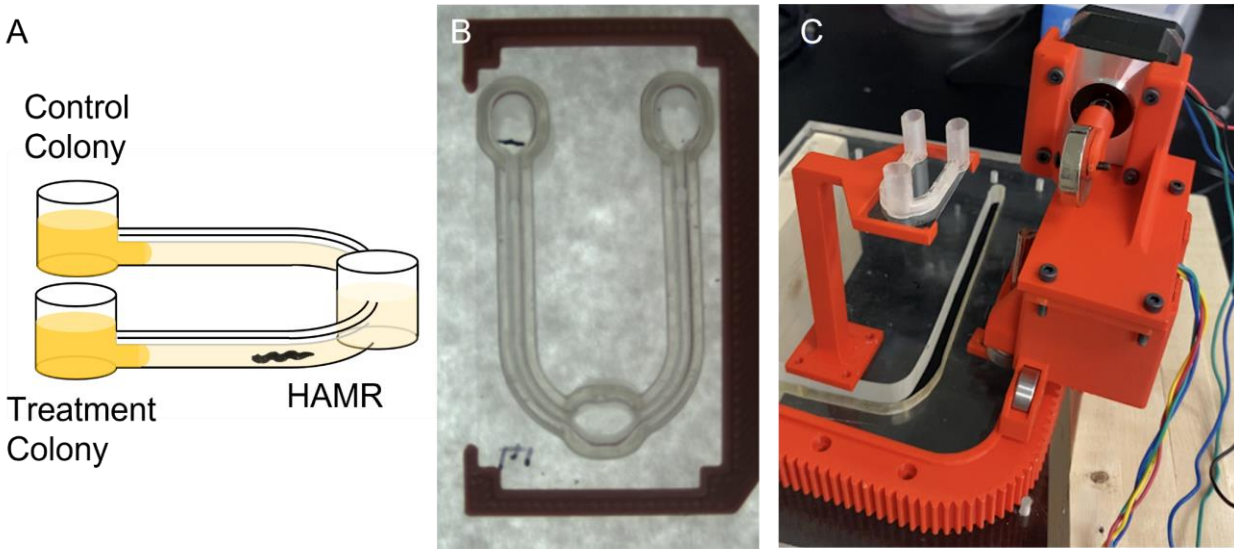


Figure 5-3. Biomimetic localized infection model and permanent magnetic actuator. To evaluate the HAMR for delivering targeted antibiotic therapy to a bacterial colony which is fluidically linked to a control colony, we developed a U-shaped microfluidic culture device. Two culture chambers in each end of the U are loaded with 0.25% agar in LB broth (motility media). The remaining middle section of the channel is filled with LB media. Cultures of wild-type MG1655 *E. coli* cells were inoculated in the culture chambers at each end by stabing a bacteria-laden needle into the solidified agar media. The HAMR is then loaded into the center of U-shaped channel, where it can be magnetically actuated to swim toward the treatment colony. (B) Photograph of the U-shaped culture chamber. (C) A permanent magnet is used as an actuator for the HAMR for these experiments. The permanent Neodymium magnet is rotated by a stepper motor to create a rotating magnetic field which induces the HAMR to swim. The rotating magnet can be moved along a 3D-printed track in order to control the position of the rotating magnetic field relative to the U-shaped channel.

Permanent magnetic actuators have been evaluated for use in magnetic microrobot control and possess a number of advantages that electromagnets do not [18, 21, 111]. The first advantage is power efficiency. An electromagnet requires a continuous supply of power in order to generate magnetic fields, whereas a permanent magnet does not require power. This aspect of permanent magnetic control of microrobots could be significant, particularly for applications where heat dissipation and mobility are important design considerations. In our application for testing HAMR TIME therapy in a biomimetic infection model, the permanent magnetic actuator (Figure 5-3C) provides simplicity, robustness, and repeatability in experimental design, requiring significantly less electrical hardware and software support for reliable operation than the Magneturret developed in chapter 3. For the HAMR TIME therapeutic experiments the permanent magnetic control system was run in an open loop configuration, with the magnet following a preprogrammed path without receiving position or velocity feedback from the HAMR.

We evaluated HAMR TIME therapy for preventative and delayed antimicrobial activity (Figure 5-4). HAMRs were incubated for 1 hour with Tetracycline [5mg/mL]. the culture chambers at the ends of the U-channel microfluidic device were filled with warm 0.25% agar which was allowed to cool and solidify. The center of the U-channel was then filled with LB media. Bacteria stabs were then made in each culture chamber with MG 1655 wild type *E. coli.* For preventative treatment (Figure 5-4A), the tetracycline loaded HAMR was placed into the center chamber and immediately actuated to swim into one of the ends of the U-channel to embed itself into the motility media. In the delayed treatment model (Figure 5-4B), the HAMR was added to the center of the U channel and actuated to swim to the bacterial culture chamber after 4.5 hours of outgrowth at 37°C. Images were then recorded periodically while the microfluidic devices were incubated, up to 25.5 hours post inoculation. In the preventative treatment, no visible bacterial outgrowth was present after 25.5 hours in the culture chamber into which the HAMR swam. In contrast, the fluidically linked chamber at the other end of the U supported visible bacterial outgrowth, showing that the HAMR's antibiotic effect was localized to one end of the channel. In the delayed treatment, a small visible bloom of bacterial outgrowth is visible in the chamber next to the HAMR, which grew before the HAMR was moved into position, but growth in this chamber was halted when the HAMR was moved into position.

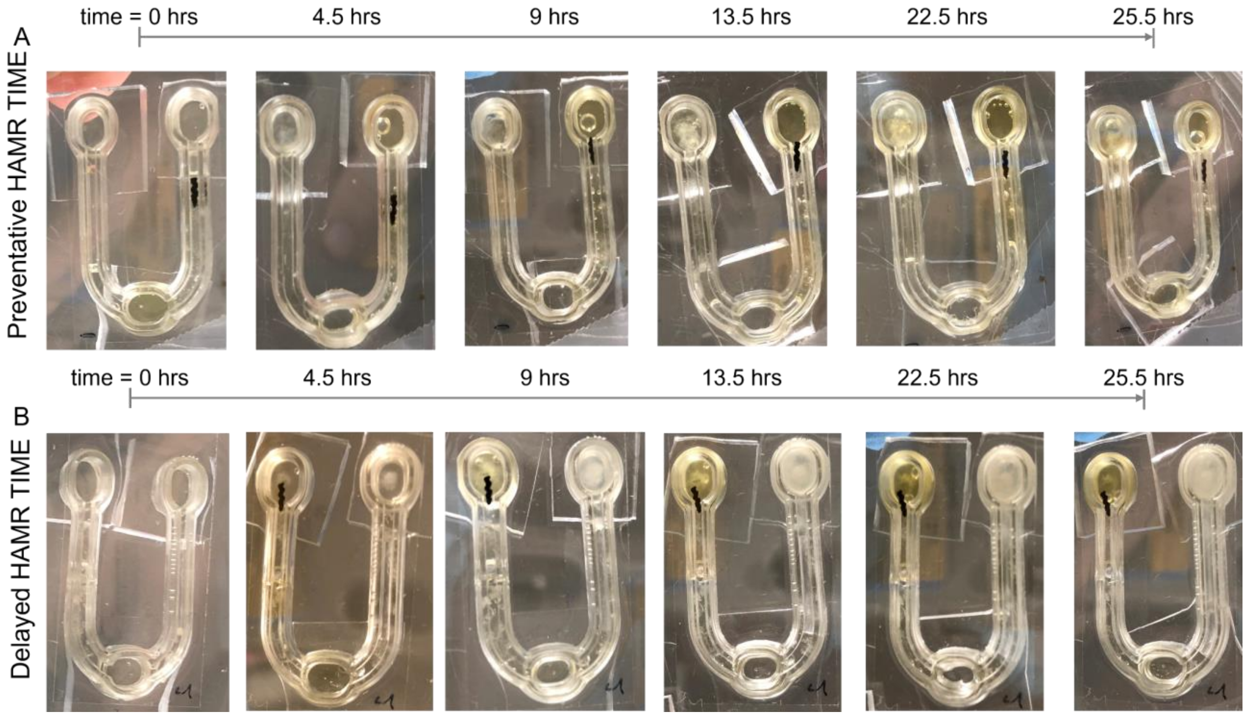


Figure 5-4. HAMR TIME therapy in a localized bacterial infection model. (A) Delayed HAMR TIME therapy in which the bacteria culture was inoculated and then the HAMR was moved into position after 4.5 hours of incubation, and (B) Preventative HAMR TIME therapy, in which the bacteria culture was inoculated and then the HAMR was immediately moved into position. Preventative HAMR TIME therapy completely eliminated any visible colony formation, while delayed HAMR TIME therapy resulted in a small initial visible colony, with arrested growth after 4.5 hours. In both preventative and delayed HAMR TIME therapy, the fluidically connected control colony experienced proliferative bacterial growth, indicating the the HAMR was successful in locally concentrating the effect of the antibiotic.

In this section, we have discussed our experiments with the HAMR for targeted infection treatment and mitigation therapy, using antibiotics as a drug payload. We have demonstrated that the HAMR is able to store a supply of antibiotic, swim to a target site, and locally inhibit bacterial growth, while allowing bacterial growth in a fluidically linked distal culture chamber. This is the first step towards therapeutic functionality for magnetic microrobotics and could have a significant impact on human health if we can reduce the use of systemic antibiotics for treatment of localized bacterial infections.

5.2 Towards Biohybrid Magnetic Genetically Programmed Multicellular Microrobots

Synthetic biology is a field primed to significantly enhance the capabilities of magnetic biohybrid microrobots by engineering new sensing and actuation capabilities into the cells onboard the robot. Cells, in contrast to non-living microsystems, excel at detecting changes in local biochemistry and environmental conditions and orchestrating complex and sophisticated responses to these changes. With synthetic biological tools, cells can be genetically programmed in order to customize cellular interactions with the environment. Cells have been programmed to sense many classes of signals including light [53, 157-159], quorum signals [160], toxic metals [160], temperature [161], metabolites [162], antibiotics [163, 164], and cancer markers [165, 166]. Cells have also been programmed to respond to these signals with sophisticated behaviors including producing fluorescent proteins [163], secreting therapeutics [167, 168], programming surface chemistry [169], generating electrical signals [170], synthesis of biomaterials [171], and attacking cancer cells [166, 172]. Programmed cells have also been incorporated into soft robotic grippers as chemical sensors [173]. The array of possible programmable cellular behaviors is highly advantageous for the development of sophisticated microrobotic systems with complex autonomous behaviors.

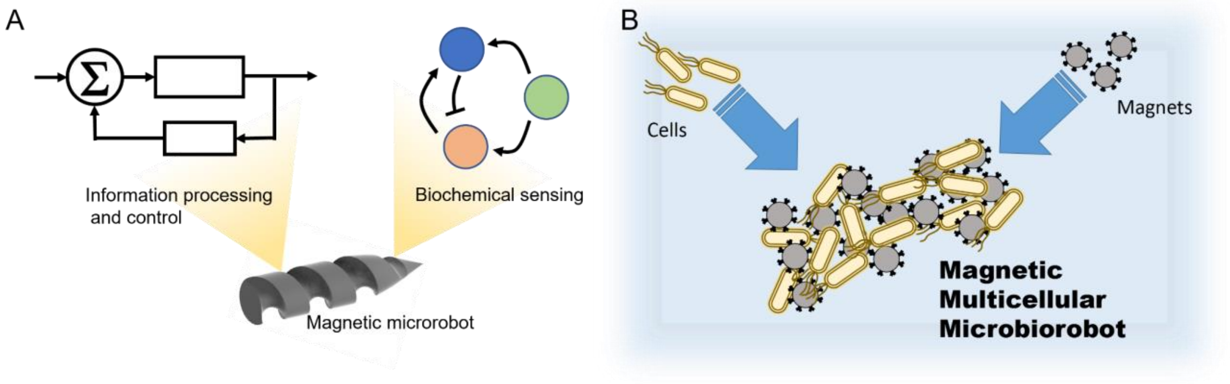


Figure 5-5. Enhancing the capabilities of magnetic microrobots with integrated living cells. (A) Magnetic microrobots such as the HAMR are capable of performing sophisticated navigation and passive drug delivery, but do not have onboard sensing and information processing capabilities. (B) Microrobots created by hybridizing living cells with magnetic microparticles can display elements from both materials. Living cells can act as sensors and information processors, while the magnetic particles can act as drug delivery and mobilization elements in a biohybrid magnetic multicellular microrobot.

In this section, we report on our efforts to develop a hybrid biorobotic platform that uses genetically engineered Escherichia coli cells to sense a chemical environmental cue and respond by self-assembling into multicellular biomagnetic helical biorobots. This work is based on hybrid biomaterial synthesis work led by Dr. Felicia Scott, a former member of our lab [174]. The system was designed to recapitulate the function of microfabricated magnetic helices that can be used to magnetically navigate and deliver payloads (such as the HAMR) (Figure 5-5). In this system, cells have been engineered to use chemical receptor proteins as environmental sensors which activate the production of engineered downstream genes. Upon exposure to the inducer molecule, the engineered cells express a hybrid surface-displayed protein, displaying a cargo protein on the surface of the cell. When induced activated cells are mixed with magnetic microparticles that have been functionalized to bind to the surface-displayed cargo protein, the cells act as crosslinkers between the magnetic microparticles, forming a multicellular biomagnetic polymeric material (Figure 5-5 B).

When placed into a spherical channel and exposed to a rotating magnetic field, the biohybrid magnetic material adopts a semi-helical conformation and can be propelled through the channel by rotating it with a rotating magnetic field. This biorobotic platform can be propelled in a frequency dependent manner with an externally rotating magnetic field, can be used for micromanipulation and delivery of miniature payloads, and can be used to activate localized fluid mixing. This biorobotic platform fundamentally advances the field of hybrid biorobotics by demonstrating that a self-assembling biohybrid magnetic microrobot architecture can be genetically programmed, and chemically activated. In this section I will discuss the architecture of the system, the capabilities we have demonstrated, and the ongoing work to improve this system with new crosslinking mechanisms.

5.2.1 Biohybrid Microrobot Assembly Based on mCherry Surface Display

Genetically engineered cells perform a dual role in our biorobotic system as both the decentralized processing units sensing and responding to external chemical cues, and as the physical glue that causes to robots to form out of a diffuse cloud of suspended biorobot components. In order turn cells into inducible crosslinkers for the biorobot, we used modified *E. coli* cells that respond to the presence of arabinose by modulating their surface chemistry. Wild type K12 MG1655 *E. coli* cells contain the ara operon, containing the arabinose metabolizing araBAD genes under control of the P_{BAD} promoter [175]. We used the strain K12 MG1655 *ΔlacIΔaraBAD*, which lacks the araBAD gene cluster, preventing the cell from metabolizing arabinose so that arabinose can be repurposed as a signaling molecule. We transformed these cells with the plasmid pFYS_019, encoding a fusion protein lpp-ompA-mCherry under the control of the P_{BAD} promoter (Appendix Figure 10). The plasmid also contains a β-lactamase gene, allowing for selective growth using β-lactam antibiotics, such as Ampicillin. When these cells are induced with arabinose, arabinose binds to the repressor protein araC, which releases repression of the P_{BAD} promoter and causes the cells to express the fusion protein. The lpp-ompA fusion protein system we used allows C-terminal expression of a protein of interest on the outer membrane of gram negative bacteria, such as *E. coli* [176]. By fusing the red fluorescent protein mCherry to the end of the lpp-ompA, mCherry was targeted to the outer surface of the cell where it can interact with the external environment (Figure 5-6A).

Once induced to express mCherry on the surface, the cells are ready to act as crosslinkers within magnetic biorobots. To achieve this, superparamagnetic fluorescent magnetic microparticles functionalized with reactive carboxyl groups were conjugated to anti-mCherry monoclonal antibodies through a carbodiimide protein crosslinking reaction. When arabinose-induced cells are mixed with the antibody functionalized microparticles, the mCherry on the surface of the cells binds to the anti-mCherry antibodies on the particles through an immunological binding reaction (Figure 5-6B). After incubation, the cells and magnetic particles coalesce into a hybrid magnetic-biological multicellular robot (Figure 5-6C).

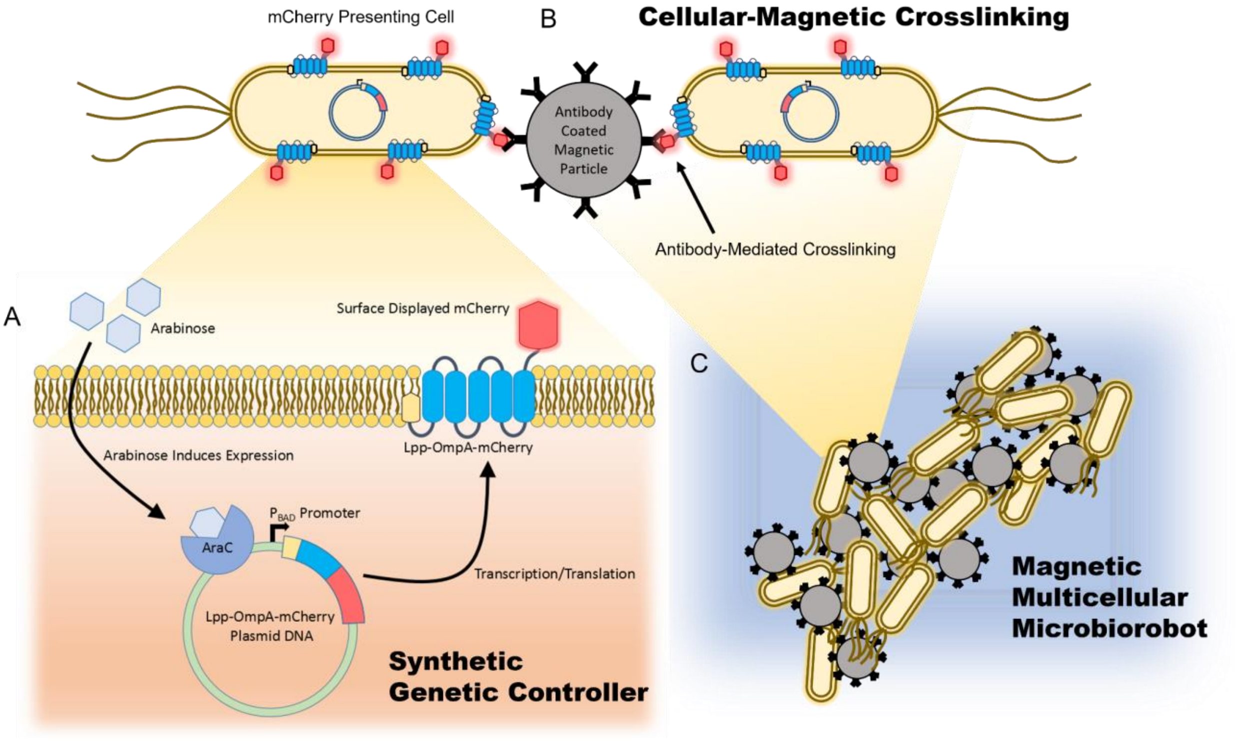


Figure 5-6. Magnetic multicellular microrobot self assembly scheme. (A) In the prototype system we have developed, a synthetic genentic controller in the form of an engineered plasmid is transformed into the chassis cell, a genetically modified MG1655 ΔLacIΔAraBAD E. coli cell. The plasmid expresses a fusion protein when the cell is exposed to an inducer chemical, arabinose. The fusion protein, lpp-OmpA-mCherry, is expressed upon induction with arabiose, and displayed on the outer memberane surface of the cell. (B) when these induced cells presenting mCHerry on their surface are mixed with anti-mCherry coated magnetic microparticles, the two units self assemble into a biohubrid magnetic multicellular material (C) , which can be magneticlly actuated as a microrobot.

To determine the appropriate concentration of arabinose to achieve full induction of the cells, a dose response curve was generated by maintaining cells in log phase growth in LB supplemented with ampicillin at 37° C along with varying concentrations of arabinose. Fluorescence measurements were performed with a plate reader, and it was determined that 5×10^{-3} % arabinose was sufficient for full induction (Figure 5-7A). 5×10^{-3}% arabinose was used to induce robot self-assembly in all subsequent experiments. Flow cytometry analysis of cells induced with 5×10^{-3} % arabinose confirmed a greater than 100-fold increase in red fluorescence intensity compared to uninduced cells (Figure 5-7B). The presence of mCherry on the surface of the cells was confirmed with fluorescent microscopy, by observing a ring of concentrated red fluorescence outlining the edge of the *E. coli* cells (Figure 5-7C).

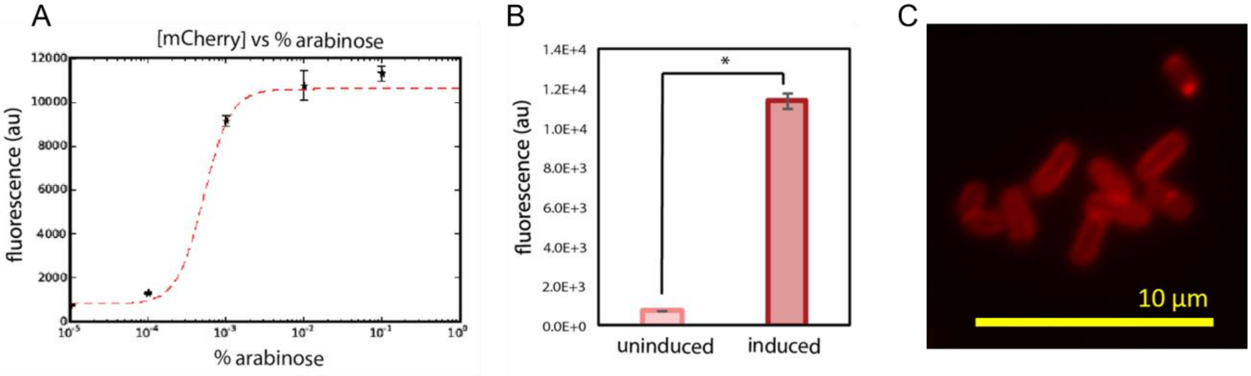


Figure 5-7. Surface-displayed mCherry induction with arabinose. (A) An arabinose dose response was conducted in order to determine the level of fluorescence output for a given arabinose concentration. (B) Cells induced at 5×10^{-3} % arabinose display over 100-fold the red fluorescent output of uninduced cells. (C) Fluorescent microscopy confirms the present of mCherry on the cell surface.

In order to form biorobots, cells were grown in shaking liquid culture for 24 hours in LB + ampicillin, both with and without arabinose, pelleted by centrifugation, and resuspended in PBS to an OD_{600} of 1. Next, 10 μL of cells were mixed with 10 μL of functionalized magnetic microparticles suspended in PBS at a density 2.5% w/v and shaken at 400 rpm for one hour in microcentrifuge tubes. After 1-hour incubation, the magnetic particles were separated out of suspension with a magnetic separation rack containing a permanent neodymium magnet. For fluorescent microscopy analysis of the self-assembled biorobots, the mass of magnetic particles was then transferred by pipet into a viewing chamber made of polydimethylsiloxane (PDMS) molded over a glass capillary to form a cylindrical channel. Fluorescent microscopy analysis confirmed that the robots prepared with induced cells were comprised of both fluorescent magnetic particles and cells displaying mCherry (Figure 5-8B). In contrast, uninduced cells did not display mCherry fluorescence and failed to catalyze biorobot aggregation (Figure 5-8A).

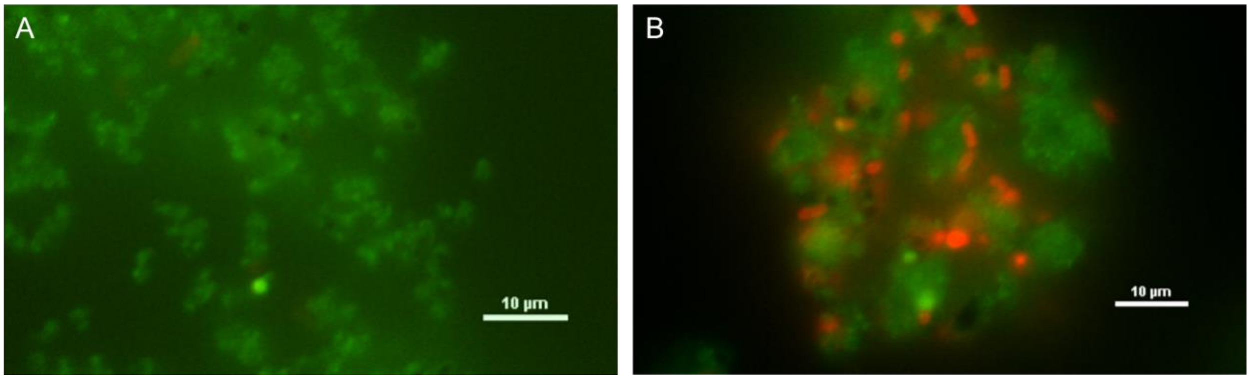


Figure 5-8. Fluorescent microscopy of self assembled biohybrid microrobots. (A) uninduced cells do not express mCherry, and so when particles are removed from suspension after mixing with uninduced cells, no cells remain, and the particles do not adhere into a uniform mass. (Particles are green fluorescent, cells are red fluorecent) (B) When cells are induced and express mCherry, they aggregate toghether and form a biohybrid magnetic material, with both particles and cells in the strucutre of the material.

5.2.2 Biorobot Self-Assembly and Magnetic Actuation

The genetic circuit we engineered allows us to induce the self-assembly of biorobots out of cells and microparticles under control of the inducer molecule arabinose, by turning the cells into selectively activatable crosslinkers. When the cells within the biorobot were induced with arabinose prior to mixing with the magnetic microparticles, the self-assembled robot assumed a slug-like shape and rotated in synchrony with the magnetic field (Figure 5-9A). In contrast, uninduced cells were not able to self-assemble, and the cloud of magnetic particles does not form a cohesive material (Figure 5-9B).

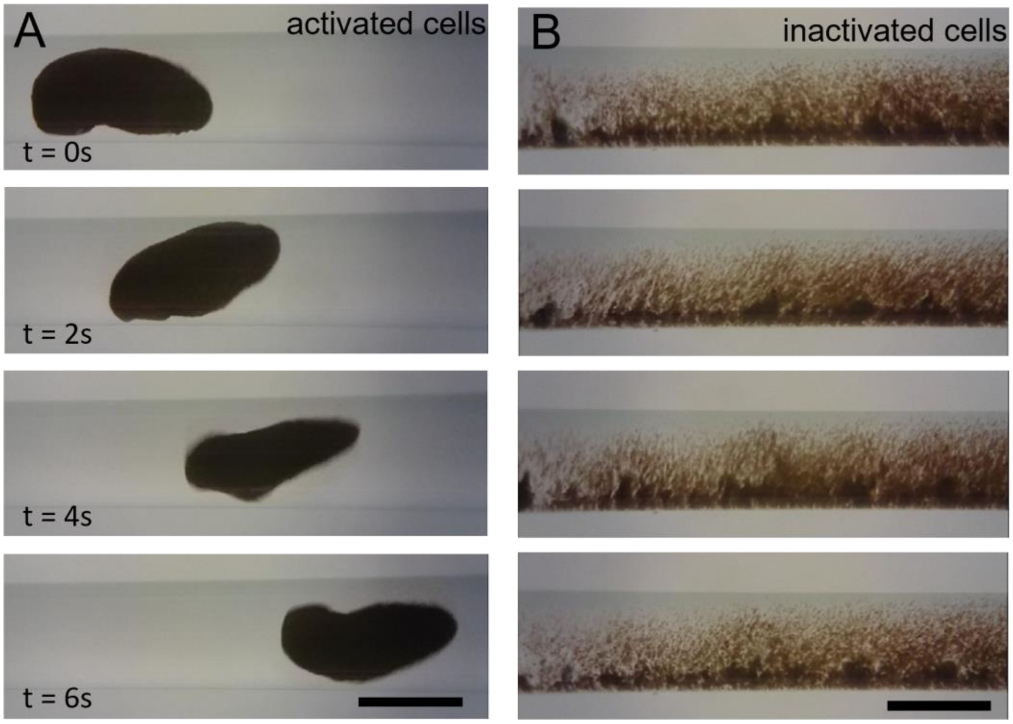


Figure 5-9. Biohybrid microrobots form with inducible surface display. (A) When cells are induced with arabionose prior to mixing with the functionalized microparticles, they form cohesive material which can be actuated by a rotating permanent magnet. When the magnet is rotated at 2 Hz and translated axially parallel to the glass capillary contraining the microrobot, the microrobot rotates and swims with a helical motion. (B) When the cells are not induced with arabionose prior to mixing with the microparticles, no microrobots form. (Scale bar = 1mm).

One of the key features of our biorobot is directed mobility within aqueous environments. Our biorobots were designed in order to move under power from an externally generated rotating magnetic field, which generates a torque on the magnetic microparticles within the robot. For this study, we developed a magnetic actuator based on a rotating permanent neodymium magnet. The magnet was affixed transverse to the axis of the rotor of a NEMA17 stepper motor, which was controlled by an Arduino microcontroller and powered by a benchtop power supply. The stepper motor was rotated at a defined frequency in order to set up a rotating magnetic field. The rotating magnet and stepper motor were then affixed to the end of a motorized linear actuator, and translated parallel to the axis of the capillary containing the biorobot, at a constant distance of 2.5 cm. At this distance, the maximum magnetic force applied to the robot by the rotating magnetic field was 8.4 mT.

To assess biorobot swimming velocity under power from the rotating magnetic field, biorobots were prepared and placed in glass capillaries as previously described. When exposed to the rotating magnetic field, the biorobot adopted a quasi-helical structure, and asymmetry in the shape of the rotating biorobot produces a propulsive driving force against the fluid media, which enables the biorobot to swim with a non-reciprocal helical motion (Figure 5-9A). In contrast, the uninduced cells and magnetic particles do not exhibit coordinated swimming behavior when exposed to the same magnetic field (Figure 5-9B).

When the magnet was rotated, the robot rotated in synchrony with the magnet due to the torque on the magnetic dipole of the biorobot, according to the equation $\boldsymbol{\tau} = \boldsymbol{m} \times \boldsymbol{B}$, where $\boldsymbol{\tau}$ is the torque exerted on the dipole, $\boldsymbol{m}$ is the magnetic dipole moment, and $\boldsymbol{B}$ is the magnetic field vector. Because the magnetic field from a permanent magnet is not a spatially uniform, there is a gradient component of the magnetic field which creates a force on the magnetic biorobot according to the equation $\boldsymbol{F} = \nabla(\mathbf{m} \cdot \mathbf{B})$, where $\boldsymbol{F}$ is the force generated, $\boldsymbol{m}$ is the magnetic moment of the biorobot, and $\boldsymbol{B}$ is the magnetic field vector. The gradient component of the rotating magnetic field creates a force which draws the robot towards the area of greatest magnetic flux density, along the axis of the permanent magnet. The result of this force is the creation of an equilibrium position directly in front of the rotating magnet towards which the robot will tend to swim, and then spin in place once it has reached equilibrium. Using this permanent magnetic setup, the biorobot could be accurately guided to a fixed position by translating the rotating magnet with the linear actuator to the desired location.

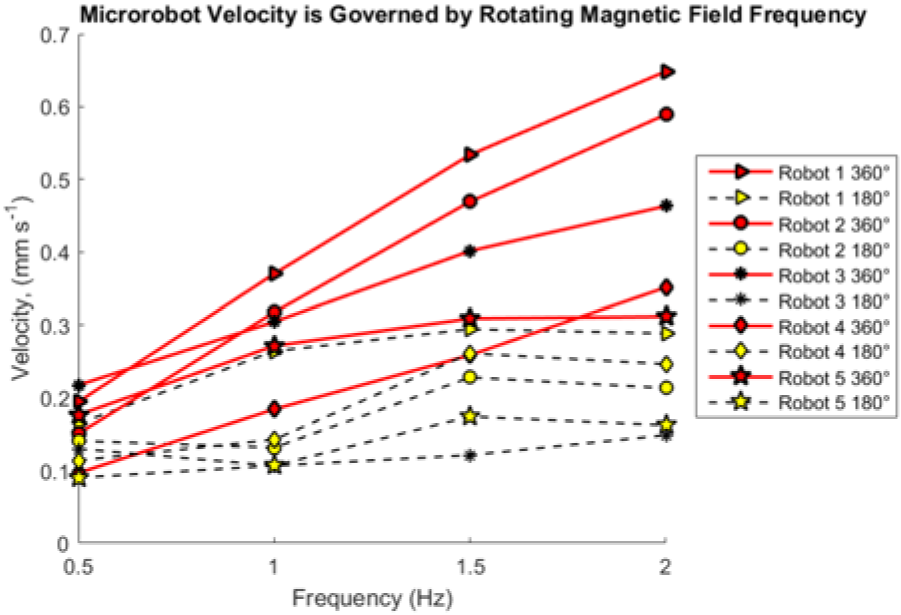


Figure 5-10. Biohybrid magnetic micororobot swimming speed. The biohybrid microrobots exhibit magnetic field frequency dependent swimming velocity. We recorded the swimming speed of 5 separate biohybrid microrobots at varying magnetic field rotation speeds (in red). Each robot exhibited an increase in speed when the rotation frequency was increased. To test whether this observed swimming behavior was the result of propulsive forces generated by continuous rotation, rather than the translational motion of the permanent magnet parallel to the glass capillary, we configured the magnet to rotate 180° clockwise, then 180° counterclockwise, alternating directions rather than spinning continuously, and measured the speed of the robots under this magnetic field actuation. Without continuous spinning, the robots were all significantly slower than when they were swimming continuously, suggesting that propulsive forces generated by the shape of the robots were contributing to the swimming speed.

To investigate whether the swimming behavior observed in the robot was due primarily to propulsive force generated by non-reciprocal helical swimming, or by attractive forces generated by the magnetic field gradient, we observed the biorobots swimming under two different magnetic field rotational modes, full 360° rotation, and oscillatory rotation in which the magnet rotates 180°, and then reverses rotation direction and returns 180°. Under full 360° rotation, the biorobots exhibited frequency dependent swimming velocity, with velocity increasing as a function of rotation frequency. Under and oscillatory 180° rotation, the frequency-dependent increase in velocity was significantly attenuated, suggesting that propulsive forces generated as a result of helical motion are helping to drive the biorobot through the capillary (Figure 5-10).

5.2.3 Micromanipulation with Biohybrid Magnetic Microrobots

The suitability of our biorobotic platform for micromanipulation and payload delivery was also investigated. To do this, the biorobot was formed and placed into a glass capillary along with a cellulose acetate sphere 0.91 mm in diameter which acted as a payload. The biorobot was driven through the capillary under power from the external rotating magnetic field, and guided to swim towards the spherical payload, carrying it through the channel upon docking. When the cells were induced and the biorobot self-assembled, the robot was able to successfully swim towards, dock with, and carry the payload through the capillary (Figure 5-11A). However, when uninduced cells and particles were added to the capillary and exposed to the rotating magnetic field, the diffuse cloud of magnetic particles was incapable of generating enough concentrated force to move the payload (Figure 5-11B).

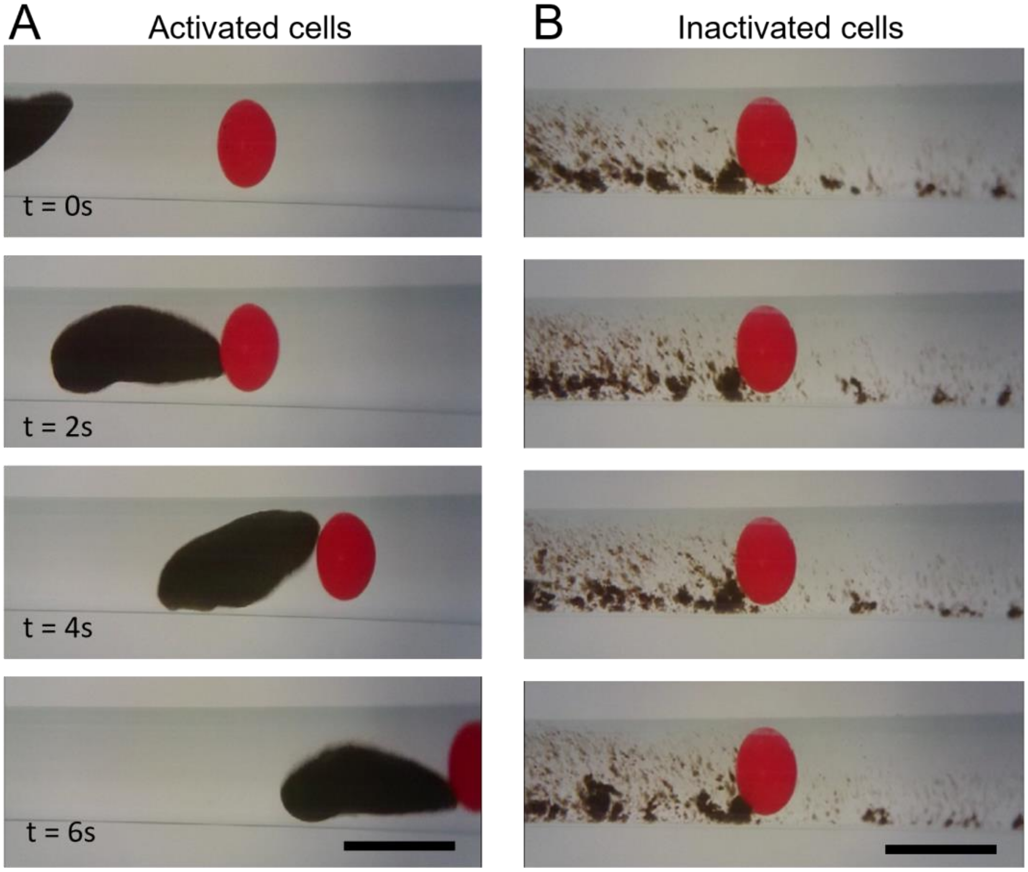


Figure 5-11. Micromanipulation with biohybrid magnetic microrobots. (A) The biohybrid magnetic microrobots are capable of moving large payloads within fluidic channels by applying force to them while swimming. They payload is a red cellulose acetate microsphere. (B) Inactivated cells do not create microrobots which are capable of moving large cargo. (Scale bar = 1 mm).

5.2.4 Undetermined Crosslinking Mechanism in mCherry-Based Self-Assembly

Despite the reliable and repeatable nature of the microrobot assembly method based on mCherry surface display that we have presented and analyzed, further experimentation showed that our understanding of the crosslinking physics involved in the assembly process is incomplete. When arabinose induced *E. coli* cells transformed with the pFYS_019 plasmid are mixed with magnetic microparticles which have not been conjugated with anti-mCherry antibodies, the biohybrid robot still forms into a cohesive structure (Figure 5-12A). However, when uninduced *E. coli* cells are mixed with the unconjugated magnetic microparticles, no microrobot forms (Figure 5-12B). In other words, the reaction occurs when the cells are induced, but not because of the presence of mCherry antibodies on the surface of the cells.

Based on this result, future work in on this project is underway in order to determine the crosslinking mechanism at play, as well as to create new crosslinking mechanisms based upon inducible surface display, which form crosslinks with functionalized magnetic particles based on our proposed crosslinking scheme. Towards this end we are actively developing new surface display plasmids to test, including HA, His, and myc protein tags display with the lpp-OmpA surface display system and with the intimin N-terminus display system [177]. Plasmids containing these new surface display systems were developed by my colleague Ting-Yen Wei, and can be found in Appendix C.3.

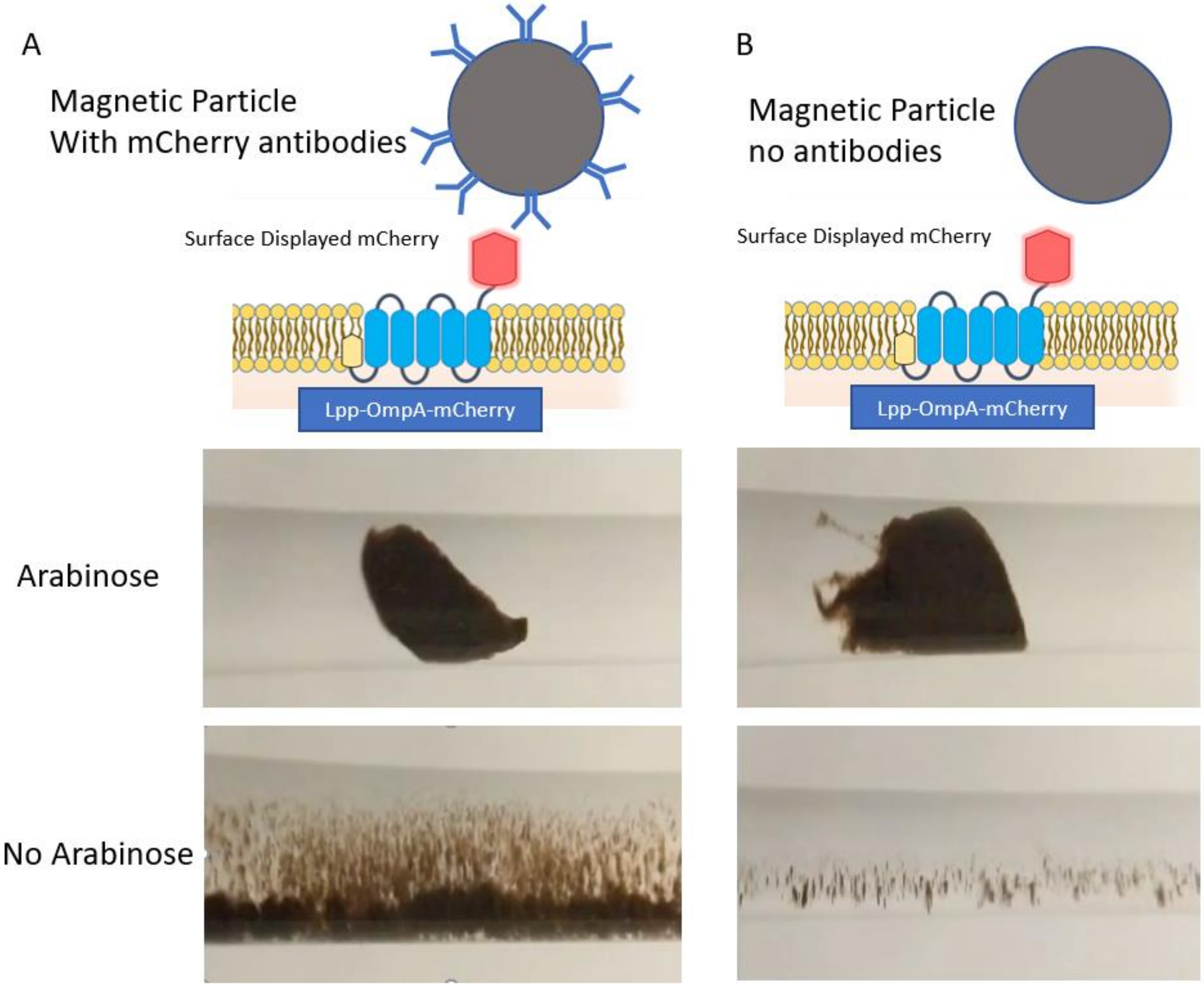


Figure 5-12. mCherry antibodies are not necessary for biohybrid microrobot self assembly. (A) When induced cells expressing mCherry on the surface are mixed with magnetic particles conjugated with anti-mCherry antibodies, they form microrobots. No microrobots form when the cells are uninduced. (B) When induced cells are mixed with magnetic particles without antibodies preesnt, microrobots still form. Microrobots do not form when the cells are uninduced however. This result suggests that the crosslinking mechanism is determined by an effect of the arabinose, but not becasue mCherry on the cell surface is binding to its cognate antibody.

5.2.5 Materials and Methods

5.2.5.1 Bacterial Chassis and Cell Culture

The *E. coli* strain K12 MG1655 *ΔlacIΔaraBAD* used in this study was obtained from the Coli Genetic Stock Center. This strain was chosen so that arabinose could persist in the cell and act as a signal for gene expression.

5.2.5.2 Genetic Circuit Construction

The methods used to construct the mCherry surface-display plasmid pFYS_019 were based on a "plug-and-play" methodology developed by Litcofsky et al. [178]. The Plasmid was modified from a plasmid used in a previous study, pFYS_018, by replacing the SNAP tag with mCherry at the C-terminal end of the fusion protein [174]. Plasmid DNA was extracted using Epoch® Biolabs Nucleic Acid Kits (Epoch Cat. No. 2160250). Test cuts with restriction enzymes and Sanger sequencing were performed to verify constructs. Touchdown PCR was used for all gene amplification. The plasmid map for pFYS_019 can be found in Appendix Figure 10. pFYS_019..

5.2.5.3 Flow Cytometry Analysis

Flow cytometry analysis was performed with a BD Accuri™ C6 flow cytometer and FL4-H laser. Collection limits were set to 10,000 events, slow fluidics, and a threshold of FSC-H < 1,000. Data were collected with the BD CSampler software and analyzed with FlowJo™ software.

5.2.5.4 Biorobot Assembly

MG1655 *ΔlacIΔaraBAD E. coli* cells transformed with the pFYS_019 plasmid were grown in 5 mL LB (Fisher LB Broth, Miller Cat. No. BP1426-500) with Ampicillin overnight at 37°C with agitation, in round bottom culture tubes. The cells were then induced by diluting the culture 1:1000 into LB with 0.005% L-arabinose (Acros Organics Cat. No. 104980250) and grown for 24 hours with agitation at 225 rpm at 37°C. Cells were also diluted 1:1000 into LB without arabinose as negative controls. 24 hours post induction, the cells were spun down by centrifuging at 10,000 rpm for 1 minute and the cells were resuspended in PBS to an OD630 between 0.8 and 1.2. mCherry protein expression was confirmed through fluorescent signal detection with a BioTek Synergy HT plate reader (590/645 excitation/emission). The cells and antibody-conjugated magnetic microparticles particles were combined at a 1:1 ratio by adding 10 μL of cells in PBS to 10 μL of antibody-conjugated microparticles. The cells and particles were mixed by gently pipetting up and down several times and incubated in microcentrifuge tubes for 1 hour at room temperature with agitation at 400 rpm. During this incubation, cells expressing surface displayed mCherry adhere to the antibody-conjugated microparticles and form the magnetic multicellular biorobots. Next, the tubes were placed on a magnetic rack (NEB.com, Cat. No. S1506S) for 30 seconds to aggregate the particles onto the side of the microcentrifuge tube. 200 μL of PBS was gently added to the microcentrifuge tube, and the tube was removed from the magnetic rack without disturbing the biorobot lodged on the side of the tube. Magnetic force from a permanent neodymium magnet was used to dislodge the biorobot from the side of the tube, by gently pulling the biorobot directly away from the wall of the tube, until suspended in the PBS. To transfer the biorobot to a glass capillary for subsequent experiments, the end was cut off of the tip of a 1000 μL pipet tip to form an opening approximately 2mm in diameter, and the biorobot was gently pipetted into the cut pipet tip and transferred to a glass Pasteur pipet (Fisher Cat. No. 13-678-20A) that was filled with PBS and capped on the narrow end with a plug of 10:1 cured PDMS (Sylgard 184 Silicone Elastomer kit, Sigma Aldrich Cat. No. 761036). The biorobot was then drawn into the narrow capillary section of the pipet with gravitational force.

5.2.5.5 Fluorescent Microscopy and Viewing Chamber Preparation

Samples were placed into a PDMS viewing chamber which was loaded onto the stage of a Nikon Ti epifluorescent microscope. An 100X oil emersion objective lens was used, and Differential Interference Contrast (DIC), TxRed (570/20Ex, 625/50Em) and FITC (495/15Ex, 530/30Em) filter sets were used to observe the sample. Images were recorded with a ANDOR Zyla sCMOS camera. The PDMS viewing chambers used in this study were prepared with a Sylgard 184 Silicone Elastomer kit (Sigma Aldrich Cat. No. 761036). PDMS, mixed at a 1:10 ratio with crosslinker, was carefully poured into a clean, new petri dish containing smooth glass rods 1.65 mm in diameter, broken from the tips of glass Pasteur pipettes (Fisher Cat. No. 13-678-20A), which were used as molds for the PDMS in order to form cylindrical chambers. The mold was then degassed and placed in an oven at 70° C for 1 hour. After curing, the PDMS was removed from the petri dish and cut into separate devices with a razor blade. The razor blade was also used to cut the PDMS along the length of the glass rod before removal using small, precision tweezers. Inlet and outlet holes were punched in either end of the channels using a 3 mm biopsy punch and the resulting devices were cleaned, dried, and plasma cleaned in a Harrick Plasma PDc-32G plasma cleaner along with glass cover slips for 30 seconds before sealing the chips. Sealed chips were placed overnight in the oven at 70° C and then stored at room temperature until use.

5.2.5.6 Antibody Conjugation to Microparticles

Carboxyl-coated fluorescent magnetic microparticles (Spherotech cat# FCM-0552-2H) were conjugated to Rat anti-mCherry monoclonal antibodies (Fisher Cat. No. M11217) though a carbodiimide crosslinking scheme. 1-Ethyl-3-(3-dimethylaminopropyl) carbodiimide (EDAC) and coupling buffer from a Polylink Coupling Kit (Bangs Laboratories Cat. No. PL01N) were used to conjugate antibodies to the microparticles. 500μL of well mixed microparticles (0.5% w/v) were added to a microcentrifuge tube, and centrifuged at 10,000 rpm for 5 minutes, and then resuspended in coupling buffer. Separately, 2mg of EDAC powder (Thermo Scientific Cat. No. 22980) was added to 1ml of coupling buffer immediately before use, at room temperature, and vortexed until well mixed and EDAC was fully dissolved. 500 μL of the EDAC solution was aliquoted into the microcentrifuge tube with the microparticles. 50 μL of anti-mCherry monoclonal antibodies were added to the microcentrifuge tube, with the EDAC and microparticles, and the tube was vortexed for 10s before being placed in a shaker at 400 rpm for 2 hours at room temperature. After 2 hours, the tube was centrifuged at 5000g for 5 minutes, and the microparticles were resuspended in 1000 μL PBS (Gibco Cat No. 79911-044). The tube was then centrifuged again at 5000g for 5 minutes, and the microparticles were resuspended in 200 μL PBS. The Antibody-conjugated microparticles were stored at 4°C and vortexed before every use.

5.2.5.7 Magnetic Manipulation of Biorobot

A stack of 6 neodymium disk magnets (Master Magnetics Inc. Part No. 07046) was affixed with lab tape transversely to the axis of a stepper motor (Sparkfun Cat. No. ROB-09238). The stepper motor was controlled by an Arduino Uno microcontroller and powered through a Sparkfun Easy Driver (Cat. No. ROB-12779) and a laboratory benchtop power supply. For testing rotational control of the biorobot, the motor was programmed to rotate at between 0.5 and 2 Hz and placed 2.5cm from the biorobot which was suspended in glass capillary. The maximum magnetic field strength experienced by the robot was 8.4mT. The motor was rotated along an axis parallel to the major axis of the capillary, 2.5 cm away from the capillary. The motor and magnet were affixed to the end of a custom linear actuator which could be moved 1.5cm back and forth parallel to the glass capillary in order to drive the robot through the channel. Magnetic field strength was measured with a Sparkfun 9DOF Sensor Stick (Sparkfun Cat. No. Sen-13944 ROHS).

5.2.5.8 Micromanipulation and Cargo Transport

0.91mm diameter red cellulose acetate spheres (Cospheric Cat. No. CAS-RED-1.3 0.91+/-0.05mm-100) were placed into the glass capillary with the biorobot to act as a payload. The biorobots were driven towards the payload with a rotating magnetic field at 2 Hz. Particle tracking of the payload was performed by color thresholding and particle analysis in ImageJ.

6.0 Conclusion and Outlook

In this dissertation I have presented a suite of microrobotic technologies, based on the long-term goal of developing therapeutic strategies for delivering minimally invasive care deep within the human body. We believe that we have made significant advances beyond the state of the art in several key areas of research. To the best of my knowledge, the reinforcement learning based microrobot control system presented in chapter 3 is the first reported system for controlling magnetic microrobot locomotion with machine learning. While the system presented here represents the initial phase of the development of machine learning based control for microrobots, there is reason to believe that this technology could have a significant impact on microrobotics research moving forward. Some of the most advanced robotic systems developed to date, such as the self-driving systems develops by companies such as Tesla, Waymo, and Uber rely heavily on machine learning to integrate high level sensor observations and making control decisions in highly complex environmental conditions [179]. Factory and warehouse robots are also increasingly employing machine learning techniques, particularly for perception problems [180]. The success of machine learning in these complex unstructured domains in which classical robotic control methods have stalled suggests that in the highly complex domain of soft, flexible, and biohybrid microrobots operating inside the human body, machine learning is likely to lead to the most successful control methods.

It is noteworthy that some of the most complex and sophisticated robot behaviors (such as the dancing robots of Boston Dynamics: https://youtu.be/fn3KWM1kuAw) are developed with classical control methods for underactuated robots [181]. Machine learning methods have not so far been able to demonstrate the agility and sophisticated coordinated movements of these robots based on dynamic physics simulations and classical control methods. It is possible that the best approach for microrobot control in the future might be a combination of machine learning based methods and classical control methods. For example, machine learning could be employed for image recognition, extracting useful semantic information from an image of a microrobot from a biomedical imaging device The information processed by the machine learning algorithms could then be transferred to a classical control system for generating motion primitives based on the high-level commands from the machine learning system.

Machine learning for microrobot control is likely to be significantly impactful in cases when it is infeasible to develop accurate dynamic models of the microrobot. The biohybrid magnetic microrobots presented in chapter 5, and biological hybrid microrobots in general, would prove to be exceedingly difficult to accurately model, because of the dynamic nature of the biological systems and their interactions with the environment. For example, the self-assembled microrobots developed in chapter 5 mimic helical swimming behavior, but they are irregularly shaped, with highly complex material properties based on living hybrid materials. In such cases, developing a robust physical model of the microrobot behavior from first principles could prove infeasible. Machine learning based control, however, could be employed effectively in this situation. Given sufficient training time for the reinforcement learning agent to observe the dynamics of the microrobot, the neural networks could be trained to accurately represent the system dynamics, approximating the dynamic model based on real world data.

The reinforcement learning control system that have developed was tasked with solving microrobot locomotion in an extremely simple microfluidic system, in two dimensions, without branching, in a circular arena. There are good reasons that we started with this simple environment for initially developing the control system, but in order to increase the utility of reinforcement learning for microrobot control, additional features of the in vivo environment should be progressively added back to the microrobot environment. Features such as branched networks, and dynamic fluidic conditions, should be developed. In order to create microfluidic environments with biomimetic fluid flow, we developed a 3D-printed peristaltic pump for microfluidic applications. With this tool, we have demonstrated that endothelial cells exposed to fluid shear stress generated by the pumped flow will exhibit a classical response to shear stress and align their cytoskeletal fibers in the direction of the flow, supporting the idea that we can generate highly accurate biomimetic flow profiles in microfluidic devices. Therefore, the pump that we have developed will be useful in future studies for incorporating accurate fluid dynamic conditions into the microfluidic environment for evaluating future iterations of our reinforcement learning based control system.

Another step towards creating realistic biomimetic conditions for studying reinforcement learning based microrobot control methods will be to incorporate living cells into the microfluidic environment, creating an organ-on-a-chip system in which the microrobot can be evaluated (Figure 6-1). In an effort spearheaded by my colleague Haley Fuller, we have expanded significantly on the possible uses for Organ-on-a-chip technology as a proving ground for micro-bio-robotic technologies in “Fuller, Behrens et al. The Future Application of Organ-on-a-Chip Technologies as Proving Grounds for MicroBioRobots. Micromachines (2021)” [113]. Organ-on-a-chip technologies seek to mimic the *in vivo* environment as closely as possible in an *in vitro* context, making them ideal systems for expanding the capabilities of machine learning based microrobot control.

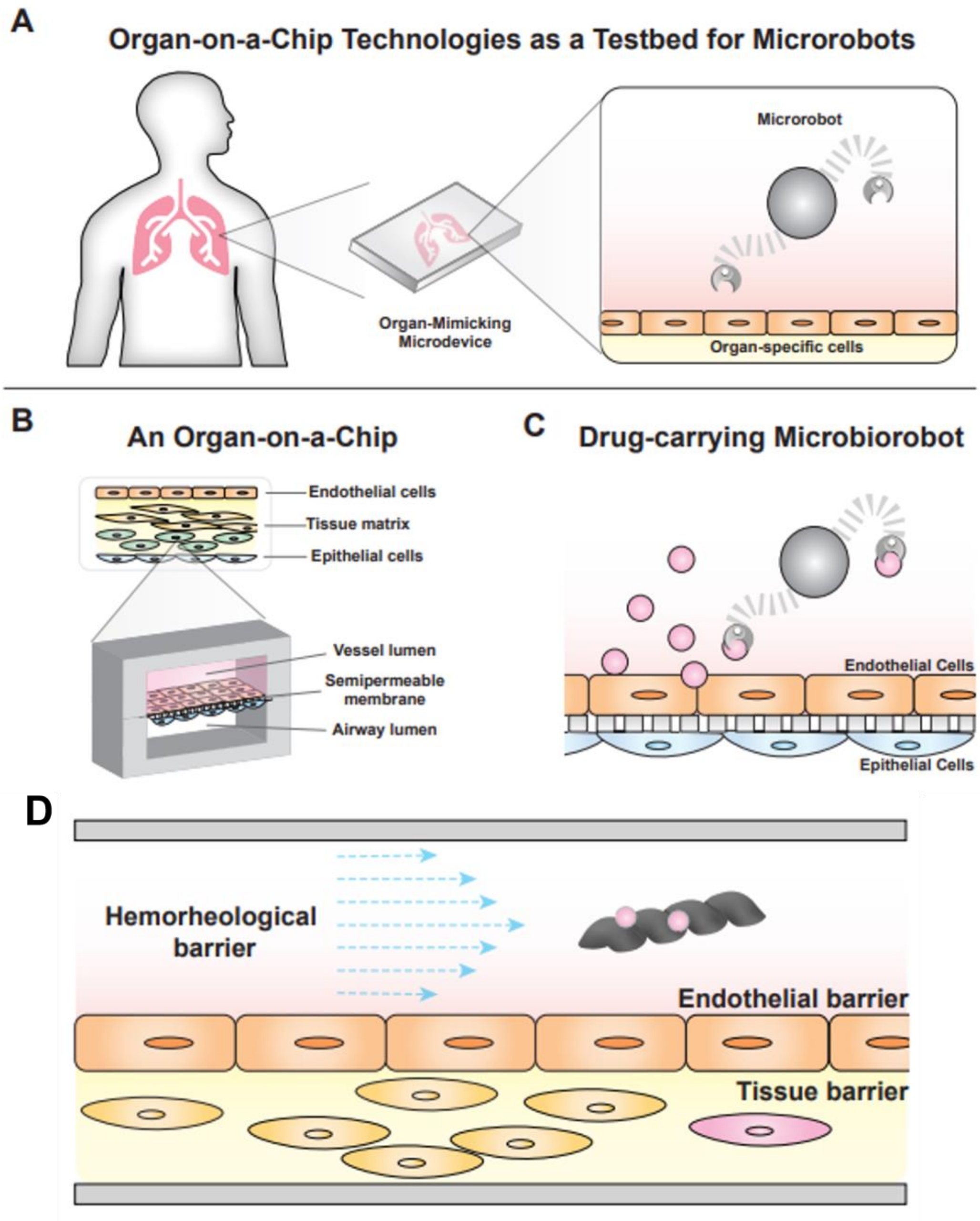


Figure 6-1. Organ-on-a-Chip systems for developing microrobot technologies. One of the key challenges for developing a reinforcement learning based microrobot control system is the ability to train the system in a highly accurate simulation of the *in vivo* environment. Organ-on-a-chip systems which include 3D cell cultures and accurate dynamic flow regimes are highly promising systems for further developing the microrobotics technologies described in this dissertation, including microrobot control, drug delivery, and biohybrid self-assembly. Originally published in Fuller, Behrens et al. Micromachines (2021).

Organ-on-a-chip systems will likely also be useful for evaluating drug delivery with microrobotic systems, such as the antibiotic therapy presented in chapter 5. Organ-on-a-chip systems combined with microrobotic technologies will enable researchers to gain better understanding of the interactions between microrobots and living cells prior to *in vivo* research, to accelerate the development cycle and bring microrobotic technologies to clinical use sooner rather than later. The technological developments presented in this dissertation represent a significant step forward towards this future of sophisticated microrobotics for minimally invasive medicine.

Appendix A Supplementary Materials for Chapter 3

Appendix A.1 Contents

Supplementary Figure 1: Highest performing policy parameters during training

Supplementary Figure 2: HAMR assembly process

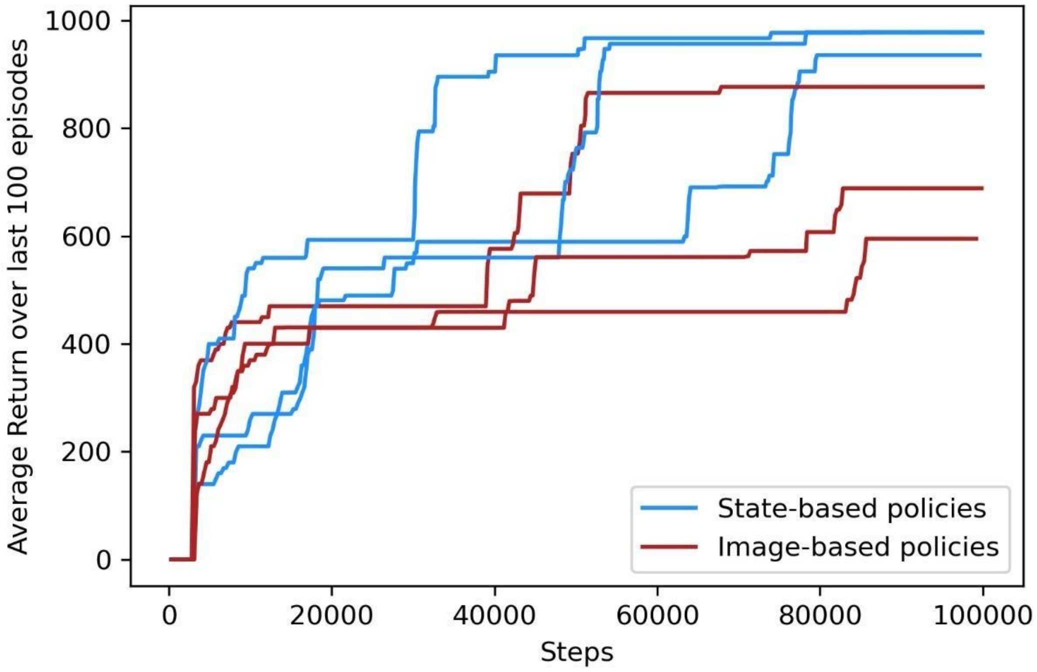
Supplementary Table 1: Hyperparameters

Supplementary Algorithm 1: Soft actor-critic for microrobot control

Supplementary Movie 1: Training process time course

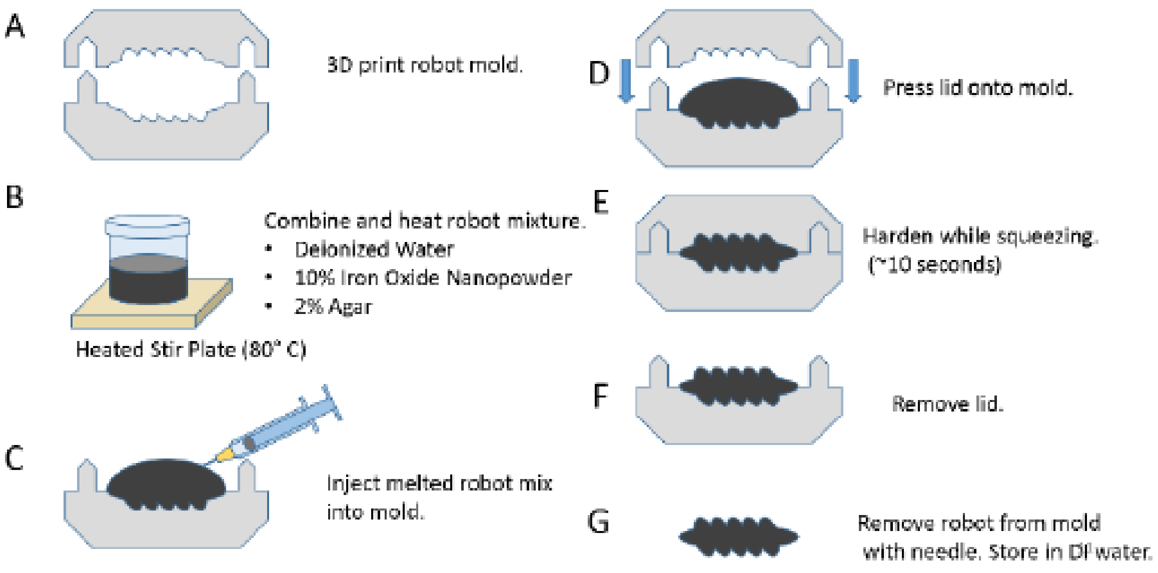
Supplementary Movie 2: HAMR swimming around the track

Appendix A.2 Supplementary Figures



Appendix Figure 1. Selecting the highest performing policy parameters during training.

During training, the average return from the policy was continually monitored with a rolling average of the last 100 episodes. Whenever the average return during the last 100 episodes was higher than at any point previously recorded during training, the parameters (i.e. the weights and biases) of the actor network, π, were saved. For evaluating the performance of the policy after learning, the highest performing parameters saved during the training session were used, rather than the policy parameters of the network at the end of the training session. This was done because we sometimes observed a decrease in the performance of the policy at the end of training. This figure shows the results from the three state-based policies, and the three image-based policies.



Appendix Figure 2. HAMR assembly process.

Appendix A.3 Hyperparameters

Appendix Table 1. Hyperparameters used for soft actor critic algorithm and neural networks.

State actor network architecture	1. Input Shape: (7,3) 2. (1)→Dense layer with 256 neurons. activation: Relu 3. (2)→Dropout (0.2) 4. (3)→Dense layer with 256 neurons. Activation: Relu 5. (4)→Dropout(0.2) 6. (5)→Output shape: (4)
State critic network architecture	1. State Input Shape: (7,3) 2. (1)→Dense layer with 16 neurons, activation: Relu 3. Action Input Shape: (4) 4. (3)→Dense layer with 16 neurons, activation: Relu 5. (2,4)→Concatenate Sate and Action Input 6. (5)→Dense layer with 256 neurons. activation: Relu 7. (6)→Dropout (0.2) 8. (7)→Dense layer with 256 neurons. Activation: Relu 9. (8)→Dropout(0.2) 10. (9)→Output shape: (1)

Appendix Table 1 (continued).

Convolutional actor network architecture	1. Input Shape: (64,64,3) 2. (1)→2D convolutional layer, 16 filters, 3x3 kernel, activation: Relu 3. (2)→Max pooling(2,2) 4. (3)→2D convolutional layer, 32 filters, 3x3 kernel, activation: Relu 5. (4)→Max pooling(2,2) 6. (5)→2D convolutional layer, 64 filters, 3x3 kernel, activation: Relu 7. (6)→Max pooling(2,2) 8. (7)→Flatten 9. (8)→Dense layer with 64 neurons. activation: Relu 10. (9)→Dense layer with 256 neurons. activation: Relu 11. (10)→Dropout (0.2) 12. (11)→Dense layer with 256 neurons. Activation: Relu 13. (12)→Dropout(0.2) 14. (13)→Output shape: (4)
Convolutional critic network architecture	1. State Input Shape: (64,64,3) 2. (1)→2D convolutional layer, 16 filters, 3x3 kernel, activation: Relu 3. (2)→Max pooling(2,2) 4. (3)→2D convolutional layer, 32 filters, 3x3 kernel, activation: Relu 5. (4)→Max pooling(2,2) 6. (5)→2D convolutional layer, 64 filters, 3x3 kernel, activation: Relu 7. (6)→Max pooling(2,2) 8. (7)→Flatten 9. (8)→Dense layer with 64 neurons. activation: Relu 10. Action Input Shape: (4) 11. (10)→Dense layer with 16 neurons, activation: Relu 12. (9,11)→Concatenate Sate and Action Inputs 13. (12)→Dense layer with 256 neurons. activation: Relu 14. (13)→Dropout (0.2) 15. (14)→Dense layer with 256 neurons. Activation: Relu 16. (15)→Dropout(0.2) 17. Output Shape: (1)

Appendix Table 1 (Continued).

Experience replay buffer size:	100,000
Batch Size	256
Learning Rate	0.0003
Gamma	0.99
Tau	0.005
Number of Actions	4
Target Entropy	-4 (negative Dim of Number of Actions)
Policy Update Period	1 minute
Total Step Duration	0.9 Seconds (three concatenated observations spaced 0.3 seconds apart)
Goal Distance	20 Degrees
Episode Length	33 Steps
Gradient update steps per environmental step:	1

Appendix A.4 Supplementary Algorithm: Soft Actor-Critic for Microrobot Control

For this work we adapted the soft actor critic algorithm reported by Haarnoja et al. [104]. Soft-actor critic uses an actor neural network to define the policy, π, with trainable parameters σ, two critic neural networks to estimate the action-value function, Q_1,Q_2, parameterized by ω_1,ω_2, and two target critic networks, $Q_{1\ \mathrm{target}}$,$Q_{2\ \mathrm{target}}$, which are initialized with parameters ω_1,ω_2, and slowly updated with Polyak averaging as the critic networks are updated in order to stabilize the target estimates used to calculate the a critic loss function. Soft actor critic learns off-policy by storing transitions in a replay buffer and randomly sampling minibatches from the replay buffer to perform policy and value function updates.

The details of our SAC implementation, including all the equations for computing the gradient updates for the policy, the action-value function, and the temperature are those of Haarnoja et al. Our algorithm differs in that it includes specific functions and steps for interfacing the SAC algorithm with our physical system. We used two separate processes, which ran independently of each other, for data collection and neural network training.

The data collection process used π to select actions while observing the environment, using the most recently updated policy parameters σ. Each action was taken in the environment for a total of 0.9 seconds, and during this time, three sequential observations were recorded 0.3 seconds apart, which were then concatenated together to for a single state observation. The (s,a,r,s’,d) transitions collected at each step were then stored in a first-in first-out (FIFO) replay buffer, which was shared between the data collection and training processes.

Concurrent with the data collection process, the training process sampled minibatches for the replay buffer and updated the actor and critic networks with gradient descent. The training process was limited to one gradient step per environmental step, in order to prevent overfitting. Periodically, the latest parameters of π were sent to the data collection process.

Supplementary Algorithm S1: Soft actor-critic for microrobot control

```
Initialize policy parameters $\sigma$, Q-function parameters $\omega_1, \omega_2$, and empty FIFO replay buffer $D$
Set target Q-function parameters equal to main parameters $\omega_{target,n} \leftarrow \omega_n$
Initialize $\bar{\mathcal{H}} = -$ number of actions, $\alpha = 1$
Observe initial state $s_{t=0}$, and calculate $\theta_{robot} \in (0,360°)$
Set $\theta_{goal} \leftarrow \theta_{robot} + 20°$, $\theta_{goal} \in (0,360°)$
Data Collection Process: repeat
    If new $\pi_\sigma$ is available, update
    For t steps in range (33)
        select action $a_t \sim \pi_\sigma(\cdot\,|s_t)$
        Execute $a_t$ in the environment
        For j in range (3)
            Wait 0.3 seconds
            Observe next state $s'_j$, reward $r_j(s'_j)$, and $d$ (1 If $\theta_{robot} = \theta_{goal}$ $\boldsymbol{Or}$ $t = 33$ $\boldsymbol{Else}$ 0)
        End For
        Set $s'_t \leftarrow (s'_{j=1}, s'_{j=2}, s'_{j=3})$, $r_t \leftarrow \sum_j r_j$
        Store transition $(s_t, a_t, r_t, s'_t, d)$ in replay buffer $D$
        Set $s_t = s'_t$
        If done: set $\theta_{goal} \leftarrow \theta_{robot} + 20°$, end For
    End For
    Set $\theta_{goal} \leftarrow \theta_{robot} + 20°$
Training Process: repeat
    If number of updates < number of transitions in $D$
        Randomly sample batch of transitions, $B = N(s, a, r, s', d)$ from $D$
        Compute targets for the Q functions:
        $y(r, s', d) = r + \gamma(1-d)\left(\min_{i=1,2} Q_{\omega_{target,i}}(s', \tilde{a}') - \alpha \log \pi_\sigma(\tilde{a}'|s')\right)$, $\tilde{a}' \sim \pi_\sigma(\cdot\,|s')$
        Update Q-functions using
        $\nabla_{\omega_i} \frac{1}{|B|} \sum_{(s,a,r,s',d) \in B} \left(Q_{\omega_i}(s,a) - y(r,s',d)\right)^2$ for $i = 1,2$
        Update Policy using
        $\nabla_\sigma \frac{1}{|B|} \sum_{s \in B} \left(\min_{i=1,2} Q_{\omega_i}(s, \tilde{a}_\sigma(s)) - \alpha \log \pi_\sigma(\tilde{a}_\sigma(s)|s)\right)$, $\tilde{a}_\sigma \sim \pi_\sigma(\cdot\,|s)$
        Update temperature $\alpha$ using
        $\nabla_\sigma \frac{1}{|B|} \sum_{s \in B} (-\alpha \log \pi_\sigma(a|s) - \alpha\bar{\mathcal{H}})$, $a \sim \pi_\sigma(\cdot\,|s)$
        Update the target Q-functions using
        $\omega_{target,n} \leftarrow \omega_{target,n} + (1-\tau)\omega_n$ for $n = 1,2$
    End If
    Send latest $\pi_\sigma$ to data collection process every minute
Until convergence
```

Appendix A.5 Supplementary Movies

Supplementary Movie S1: Training process time course. This movie shows an accelerated time course of the HAMR's movements during training. The agent starts out randomly exploring actions, and as training progresses the HAMR begins to move continuously in a clockwise direction around the arena.

Supplementary Movie S2: HAMR swimming around the track. This movie shows the swimming action of the HAMR in real time, for one complete trip around the circular track during evaluation of state-based policy No. 1.

Appendix B Supplementary Materials for Chapter 4

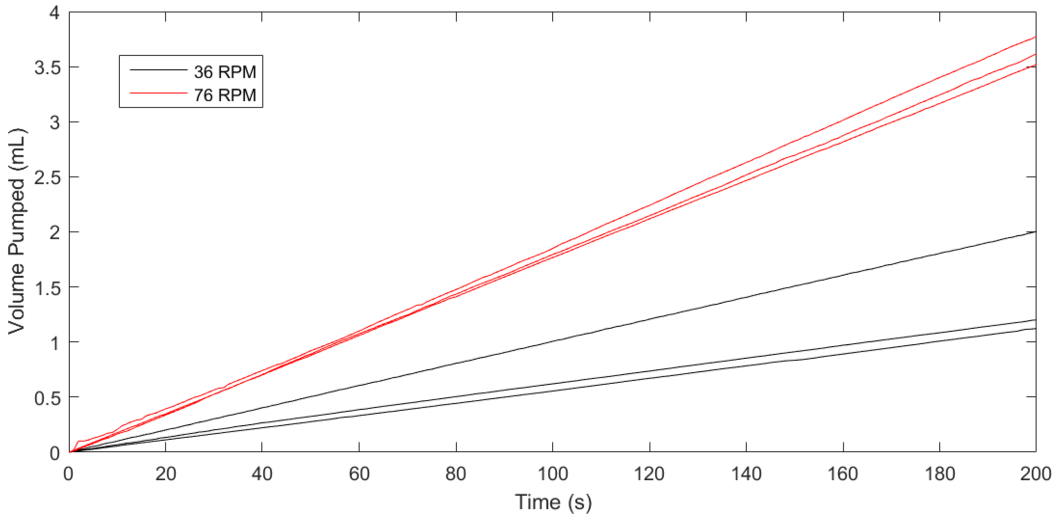
Appendix B.1 Contents

7. Peristaltic Pump Stator (3 mm)
8. Peristaltic Pump Lid (1.5 mm)
9. Peristaltic Pump Lid (3 mm)
10. Peristaltic Pump Clamp
11. Peristaltic Pump Rotor
12. Electronics Lid
13. Electronics Case

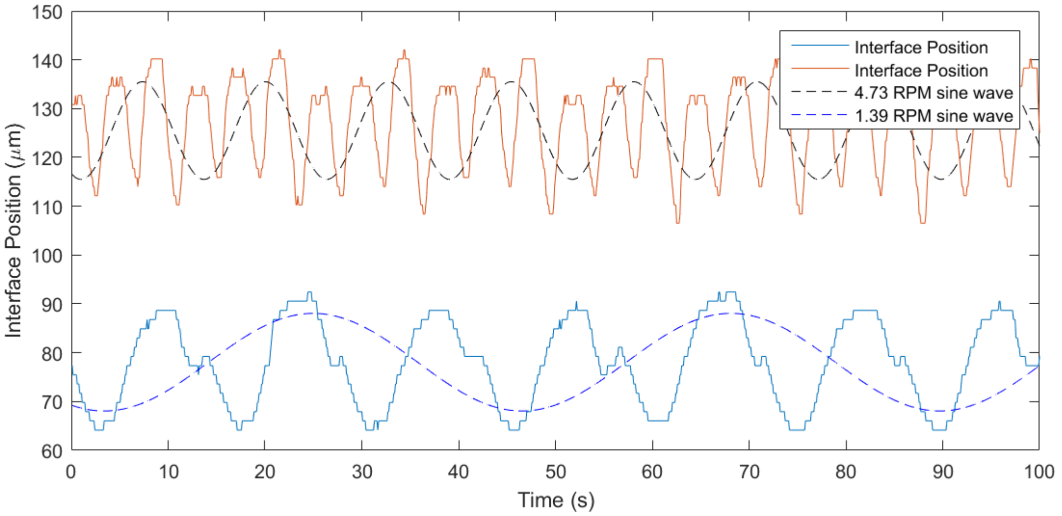
Appendix B.2 Supplementary Figures and Tables

Appendix Table 2. Bill of materials.

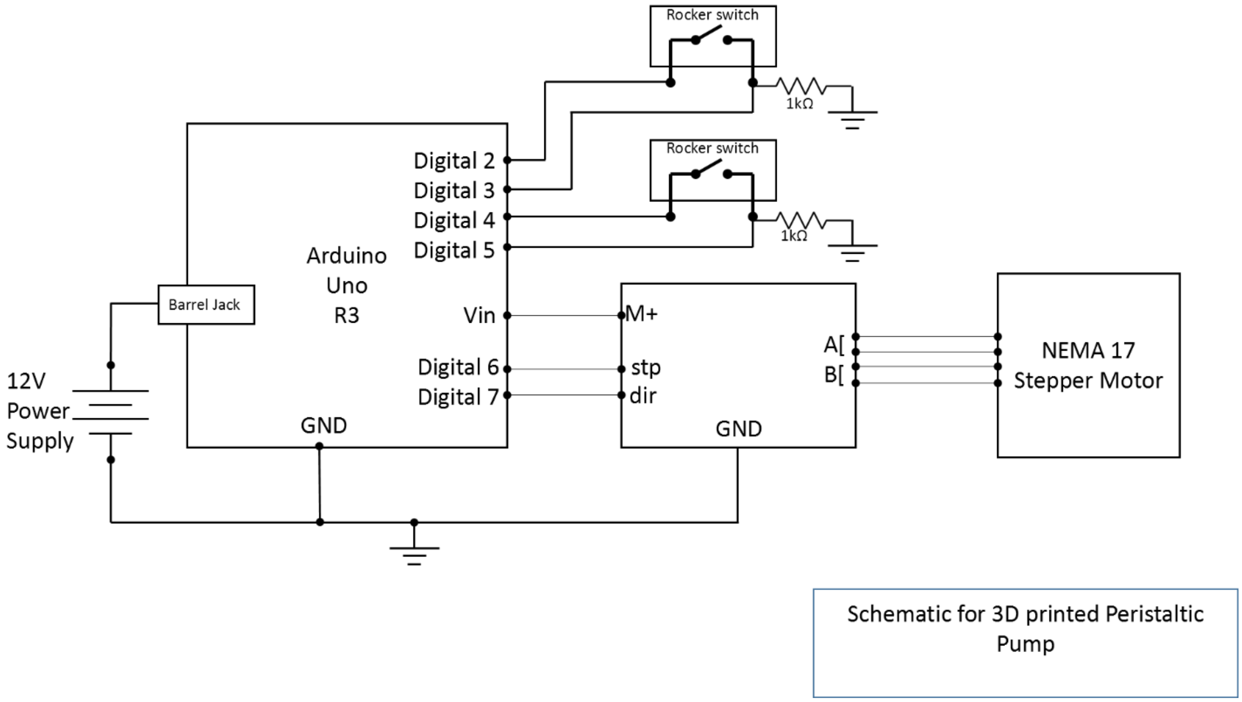
Item	**Price**	**Quantity**	**Total Cost**
NEMA 17 40mm 1.8° stepper motor	$ 10.00	1	$ 10.00
Hex Socket Machine Screws M3-0.5 x 16mm	$ 0.07	29	$ 2.03
17x6x6 mm ball bearings	$ 5.44	4	$ 21.76
6mmx100mm steel shaft	$ 1.13	1	$ 1.13
Zinc M3-0.5 nut	$ 0.14	20	$ 2.80
0.1875" ID 0.3125" OD 0.125" Width ball bearings	$ 6.92	3	$ 20.76
Hex Socket Machine Screws M3-0.5 x 40mm	$ 0.15	2	$ 0.30
Easydriver Stepper Motor Driver	$ 14.95	1	$ 14.95
Arduino Uno R3	$ 22.95	1	$ 22.95
1kΩ resistors	$ 0.04	2	$ 0.08
wires	$ 0.05	1	$ 0.05
header	$ 0.75	1	$ 0.75
rocker switches	$ 0.95	2	$ 1.90
12V power supply	$ 8.70	1	$ 8.70
ABS plastic	$0.05/g	200g	$ 10.00
		Total	**$ 118.16**



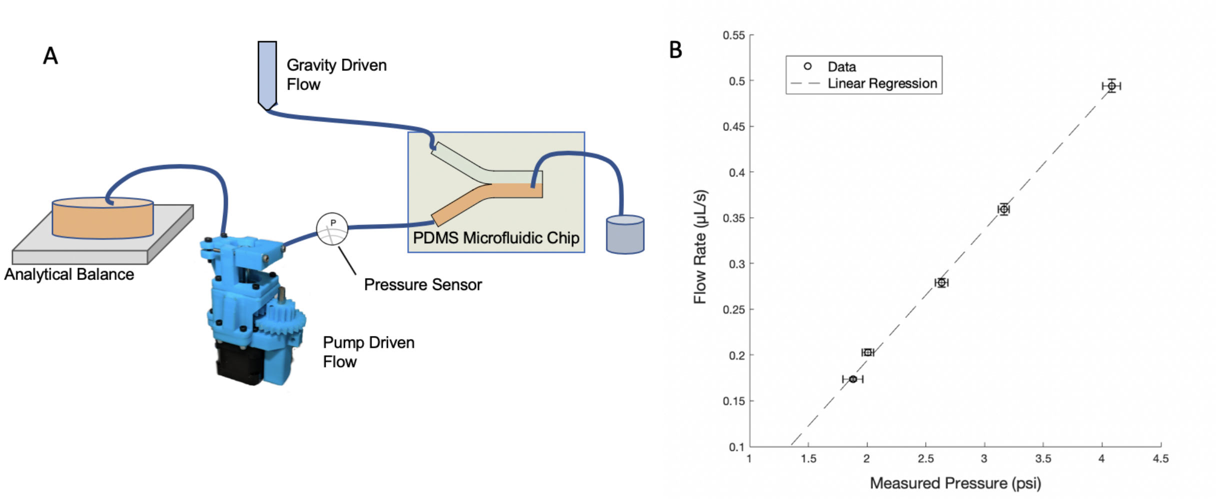
Appendix Figure 3. Individual flow rate trials. Flow rate is constant for a constant RPM and tubing installation. The variation is introduced by varying the tension of the screws clamping the tubing into place within the pump, and by varying the tension of the tubing against the rotor. When the tubing is replaced the pump rate may not be exactly the same as it was previously. The three traces at each RPM here represent three separate tube installations in the pump.



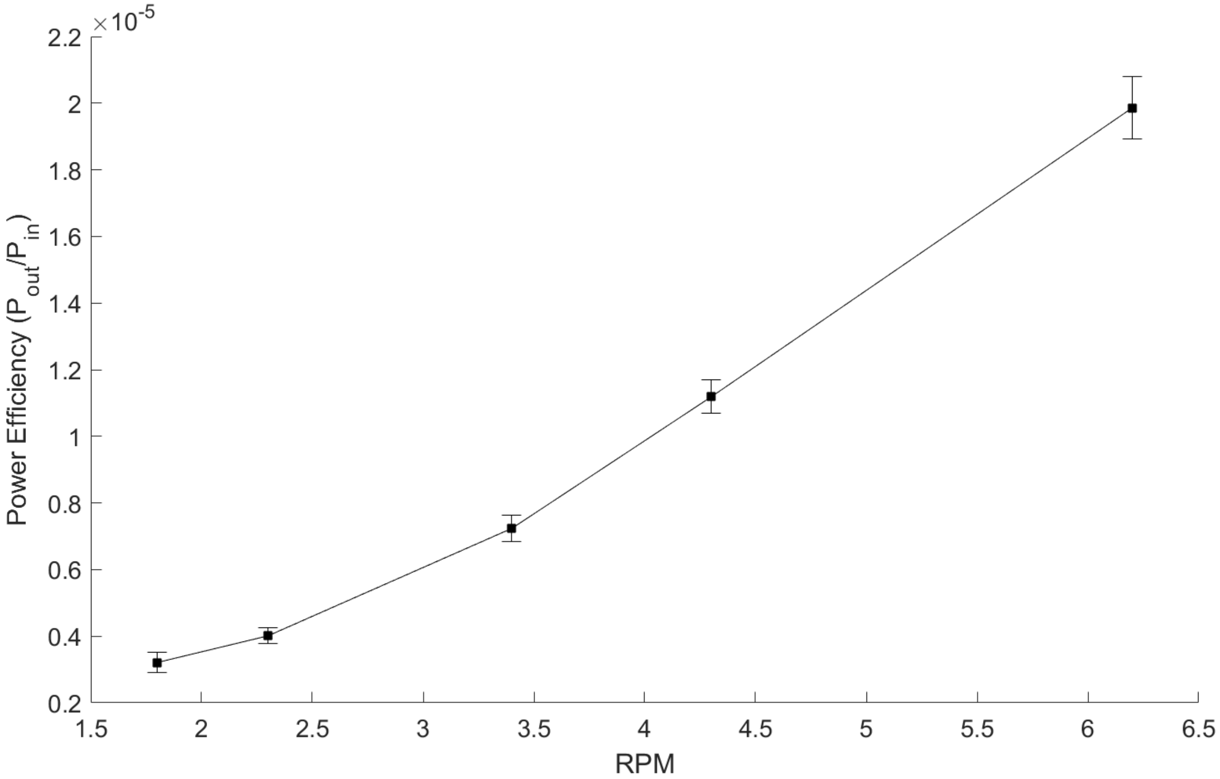
Appendix Figure 4. The frequency of pressure oscillations from the pump is a function of pump RPM. The frequency of oscillation of 3X the frequency of the pump, because of the three ball bearings on the rotor that apply pressure to the tubing.



Appendix Figure 5. Electrical schematic.



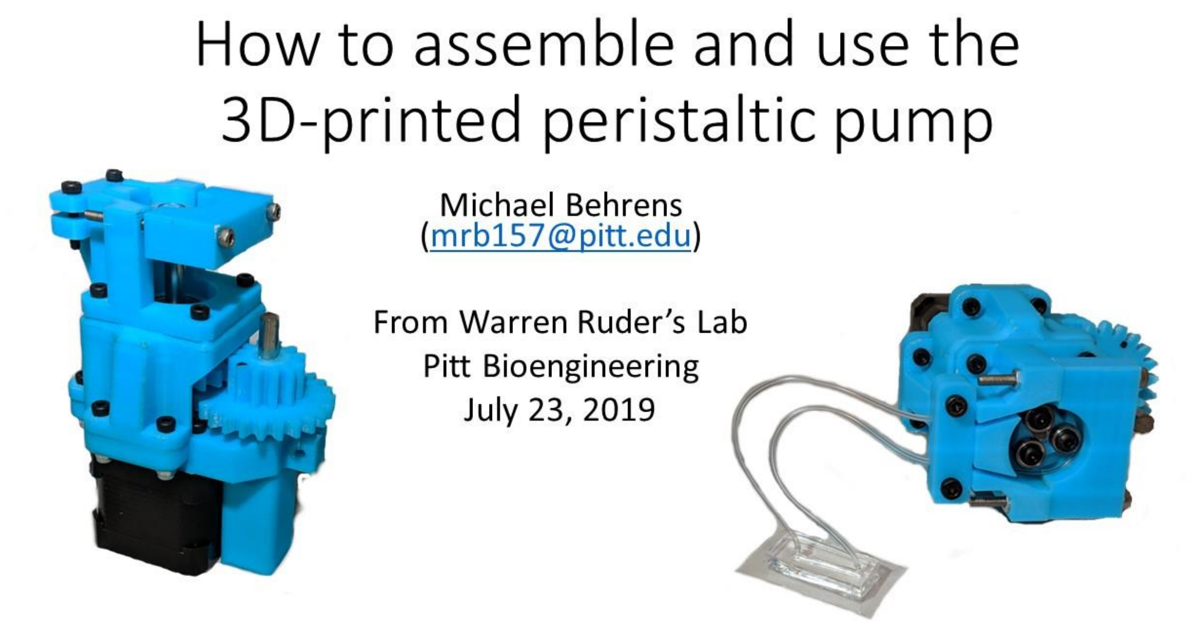
Appendix Figure 6. Pressure versus flow rate. A) Pressure drop across the microfluidic device was measured by recording the output from a pressure sensor placed in line between the pump and the microfluidic channel inlet. The flow rate from the pump was measured by drawing the fluid off of a reservoir upon an analytical balance (Ohaus Adventurer) and reading out the real time weight measurements with a computer. To obtain each measurement, the speed of the pump was set by programming the Arduino microcontroller, and pressure and flow rate were recorded over a time period of two minutes. The average values over two minutes were plotted. The experiment was repeated on three identical microfluidic devices. B) Measured pressure vs flow rate is shown. Error bars are representative of one standard deviation around the mean value taken at the same speed setting across three identical microfluidic devices.



Appendix Figure 7. Power efficency of the peristaltic pump. Pump efficiency as a function of pump rotations per minute is shown. Power consumption was determined by powering the pump from a laboratory power supply and multiplying the voltage by the current to obtain the electrical power drawn by the pump. The efficiency of the pump in converting electrical power to mechanical fluid power can also be calculated as P_{out} / P_{in} where P_{out} is calculated as output pressure multiplied by flow rate, and P_{in} is calculated as input voltage multiplied by input current. The input power was calculated to be 0.7 W, regardless of the RPM, while the output power was variable. Error bars represent one standard deviation.

Appendix B.3 Assembly Instructions: How to Assemble and Use the Peristaltic Pump

Appendix Figure 8. Peristaltic pump assembly instructions. The following slides document the assembly steps for the 3D printed peristaltic pump for microfluidic applications that is featured in chapter 4, and are published in the supplementary materials of Behrens et al. Scientific Reports, 2020.

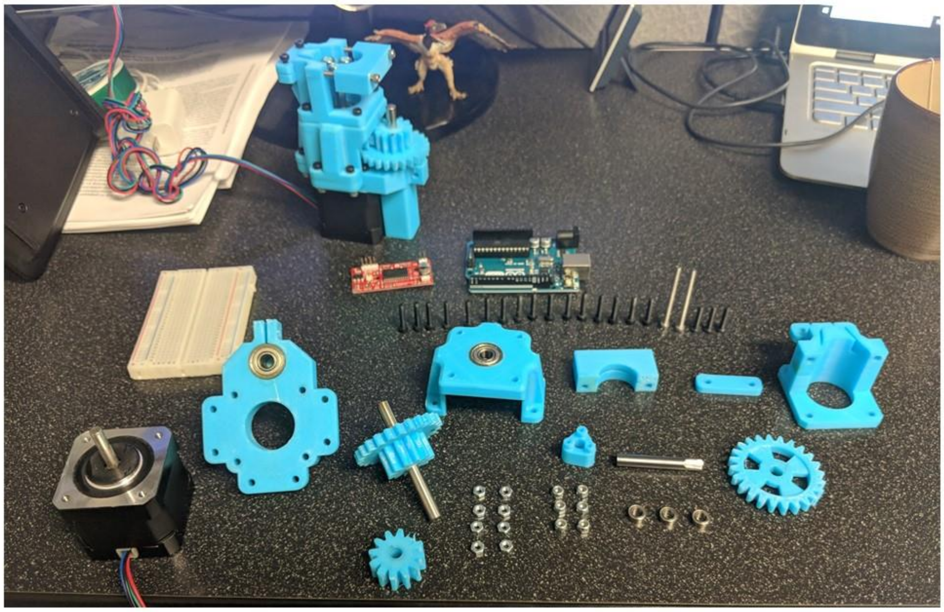


How to assemble and use the
3D-printed peristaltic pump

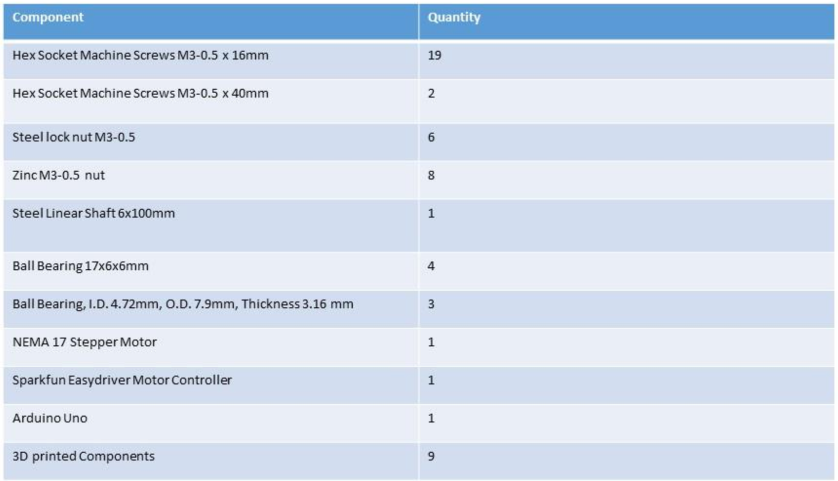
Michael Behrens
(mrb157@pitt.edu)

From Warren Ruder’s Lab
Pitt Bioengineering
July 23, 2019

Parts



Parts

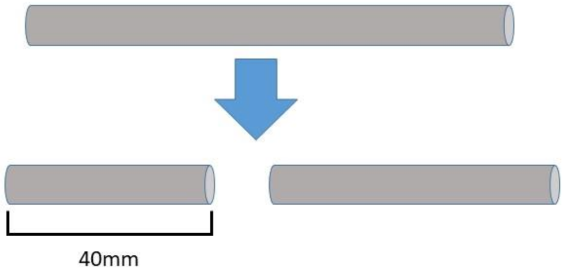


Component	Quantity
Hex Socket Machine Screws M3-0.5 x 16mm	19
Hex Socket Machine Screws M3-0.5 x 40mm	2
Steel lock nut M3-0.5	6
Zinc M3-0.5 nut	8
Steel Linear Shaft 6x100mm	1
Ball Bearing 17x6x6mm	4
Ball Bearing, I.D. 4.72mm, O.D. 7.9mm, Thickness 3.16 mm	3
NEMA 17 Stepper Motor	1
Sparkfun Easydriver Motor Controller	1
Arduino Uno	1
3D printed Components	9

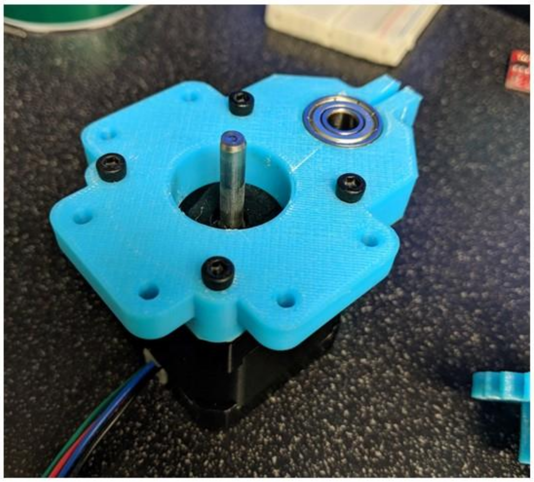
3D printed components

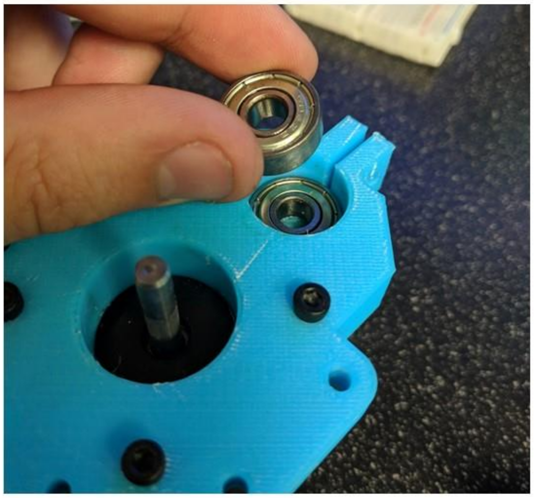
- Gear Box Base
- Small Gear
- Stacked Gear
- Gear box upper case
- Large gear
- Pump stator (comes in 2 flavors, one for 1.47mm tube, one for 3mm tube)
- Pump lid (comes in 2 flavors, one for 1.47mm tube, one for 3mm tube)
- Pump clamp
- Pump rotor

Cut 6mm shaft into 2 pieces. One piece should be 40mm, the other doesn't matter.

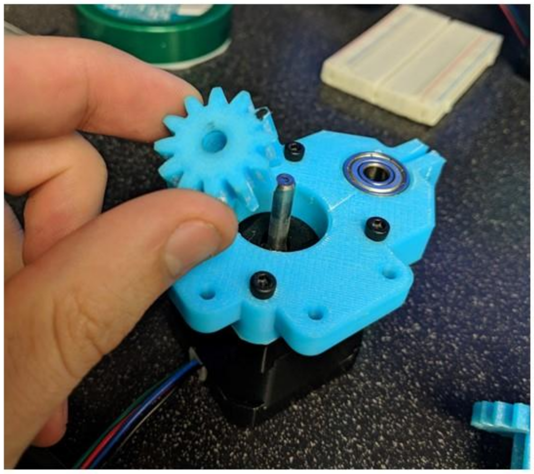


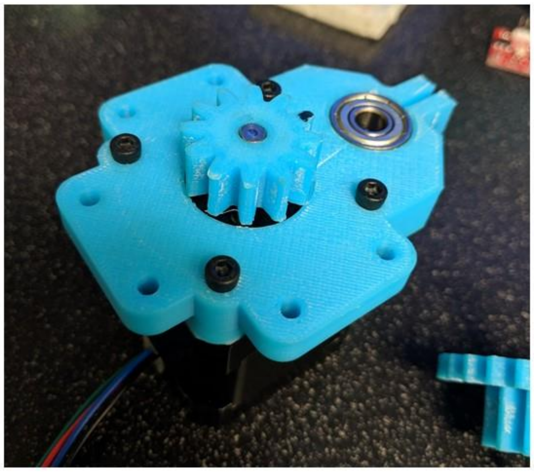
Attach baseplate to stepper motor with hex screws and insert bearings



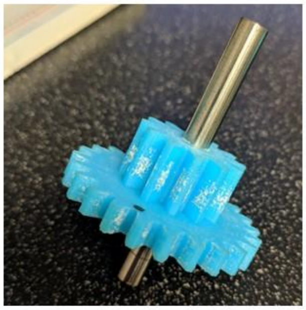


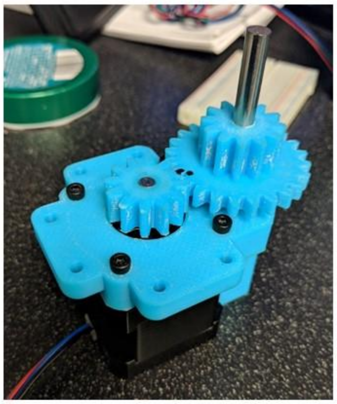
Press fit small gear onto stepper motor shaft

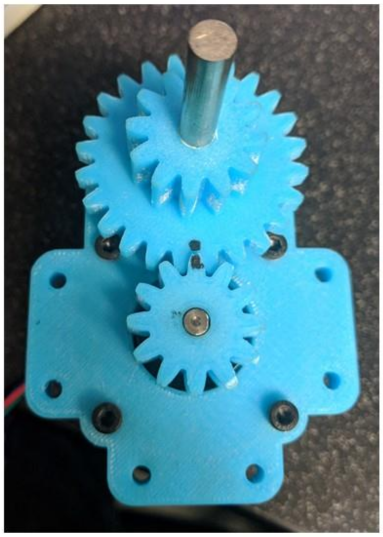




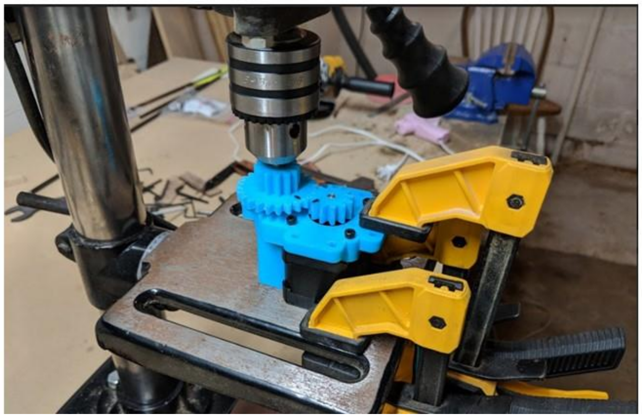
Press stacked gear onto long end of cut shaft, so that it meshes with the gear on the motor





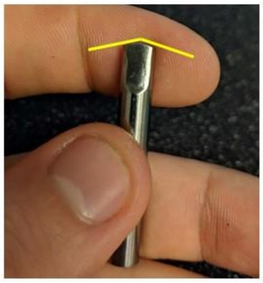


Depending on the accuracy of your 3D printer, the gears may not mesh smoothly at first, causing them to lock up. This can be fixed by driving the shaft with a drill until the small burrs and imperfections have worn away on the gears. This took me about 30 minutes on a drill press, going at a relatively slow speed to avoid overheating.

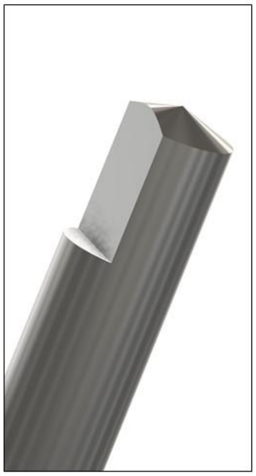


Insert ball bearings and nuts into top casing

Cut shaft to 40mm, mill side of shaft with dremel tool until it fits it’s gear, and create a very shallow point on end of shaft. This is so that it spins freely on it’s end like a top.

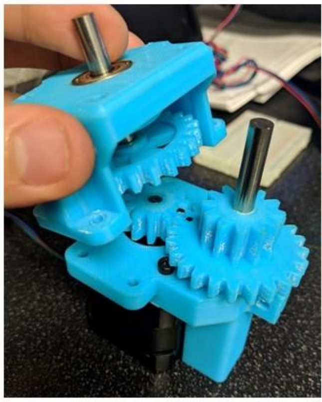
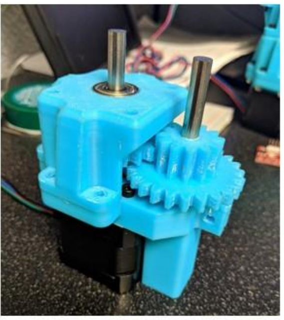


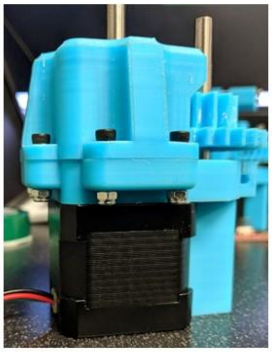
Tip: the best way to make this shallow point is by mounting the shaft in a drill or drill press, and spinning it while grinding with a dermal. Or use a lathe if you have one.



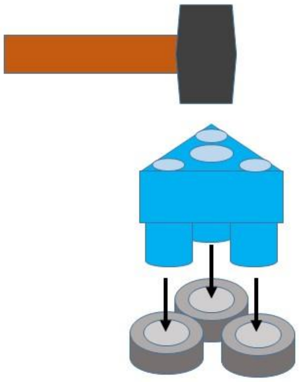
Insert machined shaft into casing, and attach fitted gear

Mount top casing assembly to bottom casing, screw together

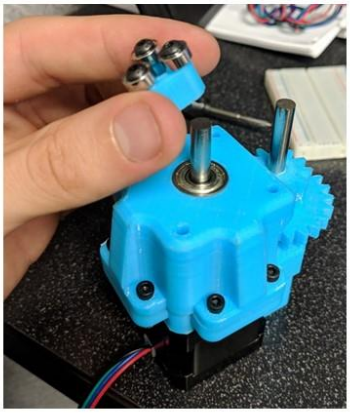





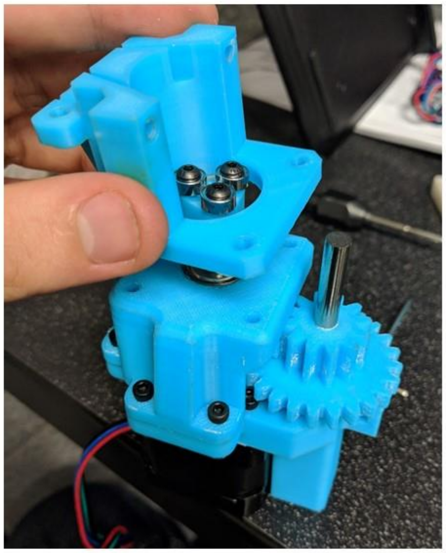
Insert small bearings onto rotor. This is best done by placing the bearings on the table, placing the rotor on top of them, and then giving a few light taps with a hammer. Then tap the holes for the screws with an M3-0.5 thread tap, and screw together.

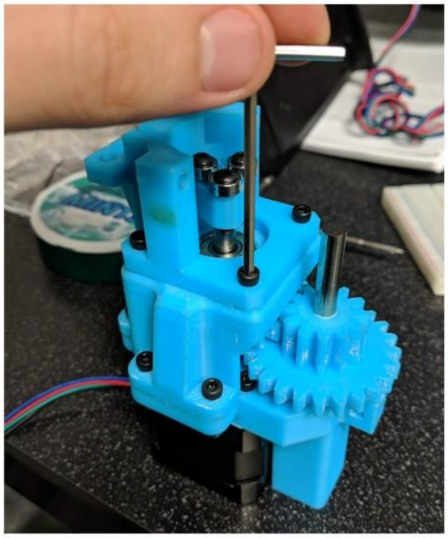




Press rotor onto shaft. The precise position of the rotor on the shaft must be adjusted in a later step in order to align the ball bearings with the Tygon tubing. For this step, just press it down part way onto the shaft.

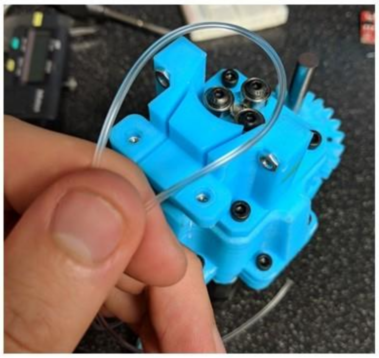


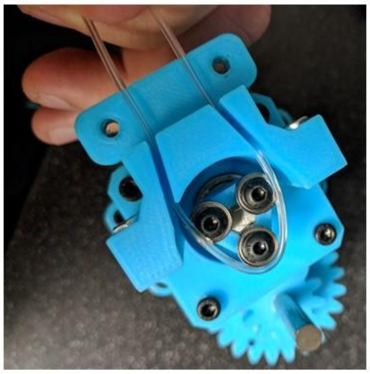
Mount Peristaltic Pump Stator



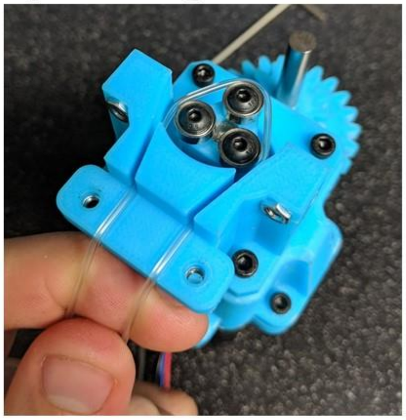


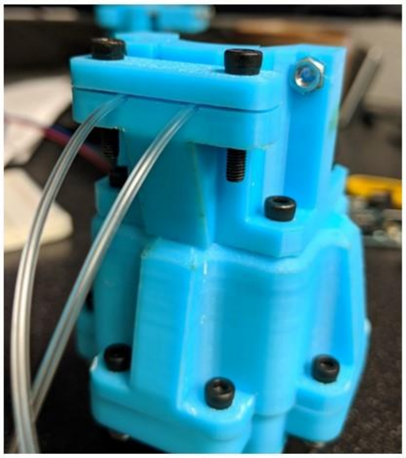
Form loop in tygon tubing and insert into slots in stator, and press against rotor



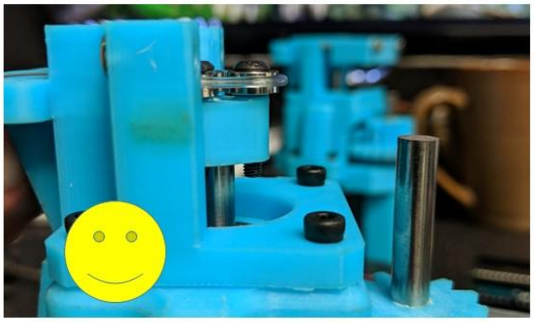


Apply tension to the tubing against the rotor, and clamp in place. Do not over tighten the clamp, it should just be enough to hold the tubing in place without pinching it shut.

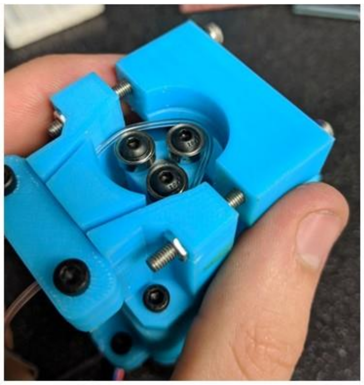


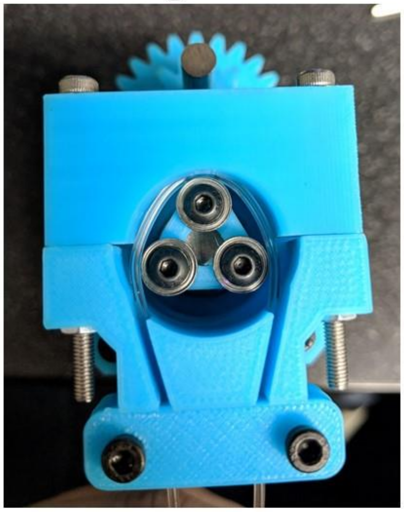


Adjust the height of the rotor on the shaft so that the tubing is straight, and sits squarely on the bearings.

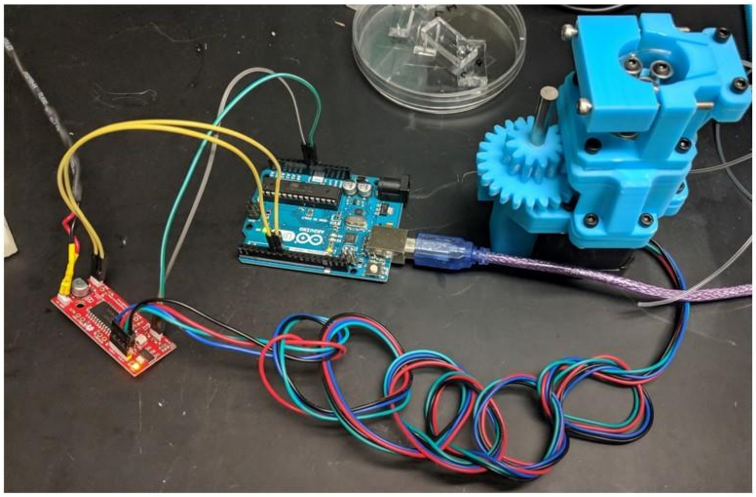


Add second clamp. This should be fully tightened for use, and loosened when the pump is not running, to prevent damage to the tubing.





Hook up the electronics.



Stp goes to digital pin 6 and dir goes to digital pin 7. 10V power should be connected to M+ on the motor driver. Ground should be connected to Gnd on both boards. +5V from the motor driver goes to Vin on the Arduino. The leads from the stepper motor should be attached to the A and B pins on the motor driver.

Program the Arduino

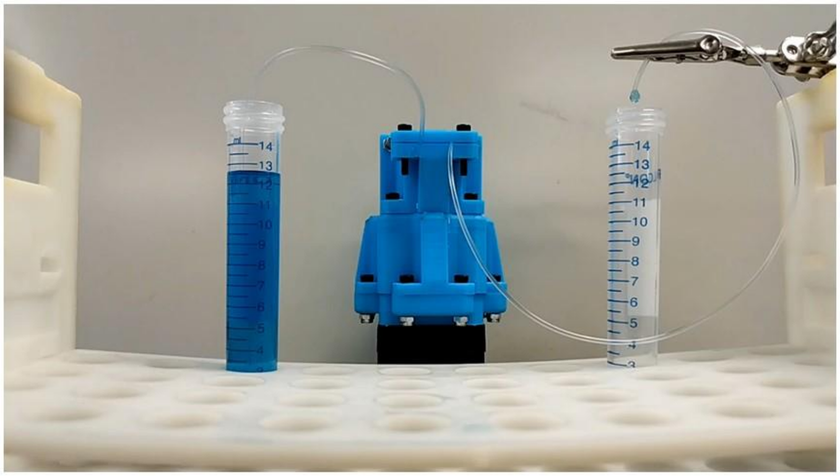
This is the simplest program to control the pump. It just spins the rotor in a single direction at a set speed. The speed is controlled by altering the length of the delays in the main loop.

```
#define stp 6
#define dir 7

void setup() {
   pinMode(stp, OUTPUT);
   pinMode(dir, OUTPUT);
}

//Main loop
void loop() {
   digitalWrite(stp,HIGH); //Trigger one step forward
   delay(.1);
   digitalWrite(stp,LOW); //Pull step pin low so it can be triggered again
   delay(1);
}
```

Pump Liquid.



Appendix B.4 Arduino Code for the Pump

Appendix B.4.1 Program 1: Run Pump at Constant Speed

```
      /*
       * this program is to control the peristaltic pump to run at a fixed
speed, controlled by the frequency variable in the program.
       */

      //Input the Frequency. Do not exceed 2. (is not actually in Hz. To verify
RMP, the speed should be manually tested by counting revolutions)

      float frequency = 1.8;


      //*****************************************************************
*********************************************************************
********************
      float loopTime = (639.6/frequency-287.8);
      float delayTime = loopTime/2;
      unsigned long timeElapsed = 0;
      int stp = 6;
      int dir = 7;
```

```
      int initialized = 0;
      int power = 1;
      int onSwitch = 0;
      int sensor = 2;
      int powerSwitch = 3;
      int sensorRead = 0;
      int directionSwitch = 5;
      int dirSensor = 4;
      int dirSensorRead = 0;
      int previousDir = 0;

      long previousMillis = 0;
      long interval = 2000;
      int a = 60;

      // the setup function runs once when you press reset or power the board
      void setup() {
        // initialize digital pin 13 as an output.
        pinMode(stp, OUTPUT);
        pinMode(dir,OUTPUT);
        pinMode(powerSwitch,OUTPUT);
        pinMode(directionSwitch, OUTPUT);
        pinMode(dirSensor,INPUT);
        pinMode(sensor, INPUT);
        pinMode(valve,OUTPUT);
        digitalWrite(dir,HIGH);
        digitalWrite(powerSwitch,HIGH);
        digitalWrite(directionSwitch,HIGH);

        Serial.begin(250000);
        while (!Serial) {
          ;
        }
      }

      // the loop function runs over and over again forever
      void loop() {
        if (onSwitch == 1)
        {
          if (initialized == 0)
          {
            for (int i = 500; i > 0; i--)
            {
              digitalWrite(stp, HIGH);   // turn the LED on (HIGH is the voltage
level)
              delayMicroseconds(delayTime+i);              // wait for a second
              digitalWrite(stp, LOW);    // turn the LED off by making the
voltage LOW
              delayMicroseconds(delayTime+i);              // wait for a second
            }
            initialized = 1;
          }

          digitalWrite(stp, HIGH);   // turn the LED on (HIGH is the voltage
level)
          delayMicroseconds(delayTime);              // wait for a second
```

```
          digitalWrite(stp, LOW);    // turn the LED off by making the voltage
LOW
          delayMicroseconds(delayTime);              // wait for a second
        }
        takeReading();

      }

      void takeReading()
      {
        sensorRead = digitalRead(sensor);
         if(sensorRead == HIGH)
        {
          onSwitch = 1;
        }
        else
        {
          onSwitch = 0;
          initialized = 0;
        }

        dirSensorRead = digitalRead(dirSensor);
        if (dirSensorRead != previousDir)
        {
          initialized = 0;
        }
        previousDir = dirSensorRead;
        if(dirSensorRead == HIGH)
        {
          digitalWrite(dir,HIGH);
        }
        else
        {
          digitalWrite(dir,LOW);
        }
      }
```

Appendix B.4.2 Program 2: Oscillate

```
      /*
       * this program is to control the peristlatic pump to oscillate fluids
back and forth on command from a switch
       */

      //Input the Frequency. Do not exceed 2. Note that this is not accurate in
Hz, to verify the speed of the pump, manually check the RPM.

      float frequency = 2; //2 to 1.66 to 1.33 to 1

      //**********************************************************************
******************************************************************************
********************
      float loopTime = (639.6/frequency-287.8);
```

```
float delayTime = loopTime/2;
unsigned long timeElapsed = 0;
int stp = 6;
int dir = 7;
int initialized = 0;
int power = 1;
int onSwitch = 0;
int sensor = 3;
int powerSwitch = 2;
int sensorRead = 0;
int directionSwitch = 4;
int dirSensor = 5;
int dirSensorRead = 0;
int previousDir = 0;

int valve = 8;
long j = 0;
int valveState = 1;

long previousMillis = 0;
long interval = 2000;
int a = 60;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(stp, OUTPUT);
  pinMode(dir,OUTPUT);
  pinMode(powerSwitch,OUTPUT);
  pinMode(directionSwitch, OUTPUT);
  pinMode(dirSensor,INPUT);
  pinMode(sensor, INPUT);
  pinMode(valve,OUTPUT);
  digitalWrite(dir,HIGH);
  digitalWrite(powerSwitch,HIGH);
  digitalWrite(directionSwitch,HIGH);
  digitalWrite(valve,HIGH);

  Serial.begin(250000);
  while (!Serial) {
    ;
  }
}

// the loop function runs over and over again forever
void loop() {

  takeReading();


}

void takeReading()
{
  sensorRead = digitalRead(sensor);
   if(sensorRead == HIGH)
  {
```

```
          onSwitch = 1;
        }
        else
        {
          onSwitch = 0;
          initialized = 0;
        }

        dirSensorRead = digitalRead(dirSensor);
        if (dirSensorRead != previousDir)
        {
          initialized = 0;
        }
        previousDir = dirSensorRead;
        if(dirSensorRead == HIGH)
        {
          //digitalWrite(dir,HIGH);
          for (int j = 0; j<6; j++) //change direction 6 times
          {
            for (int i = 1000; i > 0; i--) //run 1000 steps in one direction
            {
              digitalWrite(stp, HIGH);   // turn the LED on (HIGH is the voltage
level)
              delayMicroseconds(200);              // wait for a second
              digitalWrite(stp, LOW);     // turn the LED off by making the
voltage LOW
              delayMicroseconds(200);              // wait for a second
            }
            digitalWrite(dir,!digitalRead(dir));
          }

          while (dirSensorRead == HIGH)
          {
            dirSensorRead = digitalRead(dirSensor);
          }
        }
        else
        {
          //digitalWrite(dir,LOW);
        }


      }
```

Appendix B.4.3 Program 3. Step Forward

```
      /*
       * this program is to control the peristaltic pump to step the fluid a
set amount forward on command from a switch
       */
```

```
      //Input the Frequency . Do not exceed 2.

      float frequency = 1


      //*********************************************************************
*****************************************************************************
********************
      float loopTime = (639.6/frequency-287.8);
      float delayTime = loopTime/2;
      unsigned long timeElapsed = 0;
      int stp = 6;
      int dir = 7;
      int initialized = 0;
      int power = 1;
      int onSwitch = 0;
      int sensor = 2;
      int powerSwitch = 3;
      int sensorRead = 0;
      int directionSwitch = 5;
      int dirSensor = 4;
      int dirSensorRead = 0;
      int previousDir = 0;

      int valve = 8;
      long j = 0;
      int valveState = 1;

      long previousMillis = 0;
      long interval = 2000;
      int a = 60;

      // the setup function runs once when you press reset or power the board
      void setup() {
        // initialize digital pin 13 as an output.
        pinMode(stp, OUTPUT);
        pinMode(dir,OUTPUT);
        pinMode(powerSwitch,OUTPUT);
        pinMode(directionSwitch, OUTPUT);
        pinMode(dirSensor,INPUT);
        pinMode(sensor, INPUT);
        pinMode(valve,OUTPUT);
        digitalWrite(dir,HIGH);
        digitalWrite(powerSwitch,HIGH);
        digitalWrite(directionSwitch,HIGH);
        digitalWrite(valve,HIGH);

        Serial.begin(250000);
        while (!Serial) {
          ;
        }
      }

      // the loop function runs over and over again forever
      void loop() {

        takeReading();
```

```
      }

      void takeReading()
      {
        sensorRead = digitalRead(sensor);
         if(sensorRead == HIGH)
        {
          onSwitch = 1;
        }
        else
        {
          onSwitch = 0;
          initialized = 0;
        }

        dirSensorRead = digitalRead(dirSensor);
        if (dirSensorRead != previousDir)
        {
          initialized = 0;
        }
        previousDir = dirSensorRead;
        if(dirSensorRead == HIGH)
        {
          //digitalWrite(dir,HIGH);
          for (int i = 4000; i > 0; i--)
            {
              digitalWrite(stp, HIGH);   // turn the LED on (HIGH is the voltage
level)
              delayMicroseconds(100);              // wait for a second
              digitalWrite(stp,  LOW);     //  turn  the  LED  off  by  making  the
voltage LOW
              delayMicroseconds(100);              // wait for a second
            }
          while (dirSensorRead == HIGH)
          {
            dirSensorRead = digitalRead(dirSensor);
          }
        }
        else
        {
          //digitalWrite(dir,LOW);
        }


      }
```

Appendix B.5 3D STL Files for Peristaltic Pump

The STL files for printing the peristaltic pump presented in Behrens et al. Scientific Reports (2020) are available for download at https://www.nature.com/articles/s41598-020-58246-6. Note: the published files were rendered in centimeters.

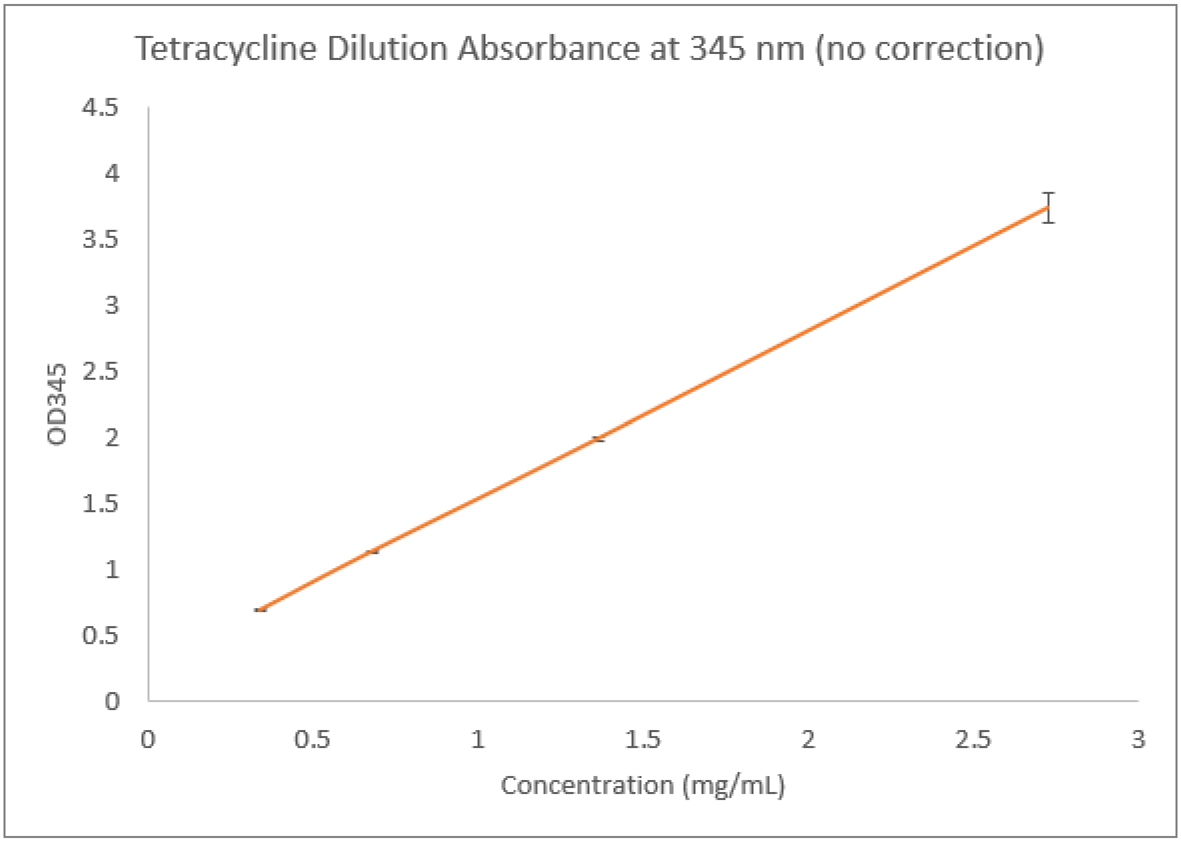
Appendix C Supplementary Information for Chapter 5

Appendix C.1 Contents

Supplementary Figures

Plasmid Maps

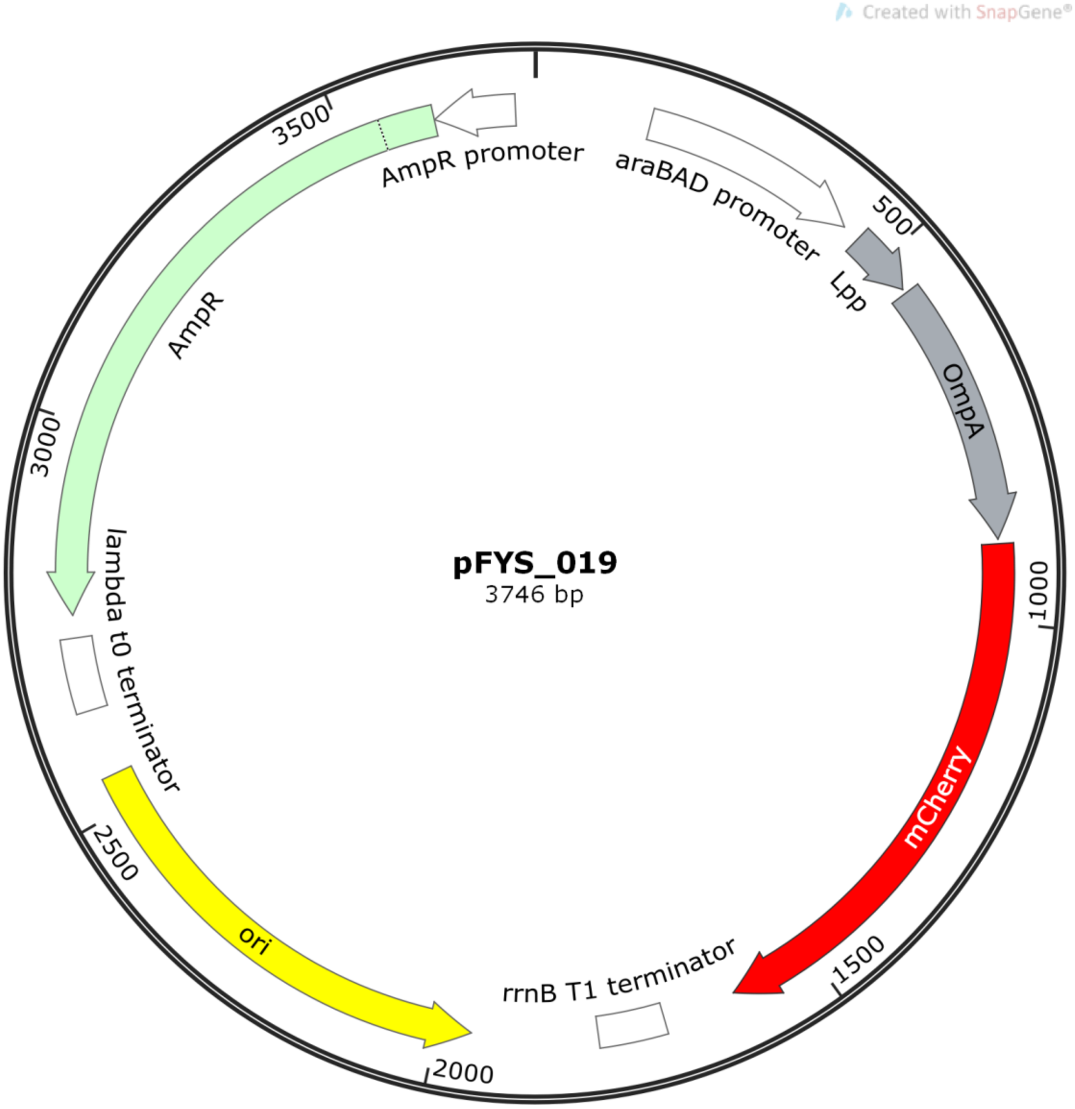
Appendix C.2 Supplementary Figures



Appendix Figure 9. Tetracycline concentration assay. This calibration assay was performed by creating solutions at varying tetracycline concentrations and measuring the optical absorbance of 200 µL of fluid in a 96 well culture plate at 345nm. This calibration curve will enable us to study the release kinetics of tetracycline from the HAMR by loading the HAMR with an initial concentration of tetracycline and placing it in an aqueous solution and monitoring the tetracycline concentration in solution over time.

Appendix C.3 Plasmid Maps

Appendix C.3.1 pFYS_019



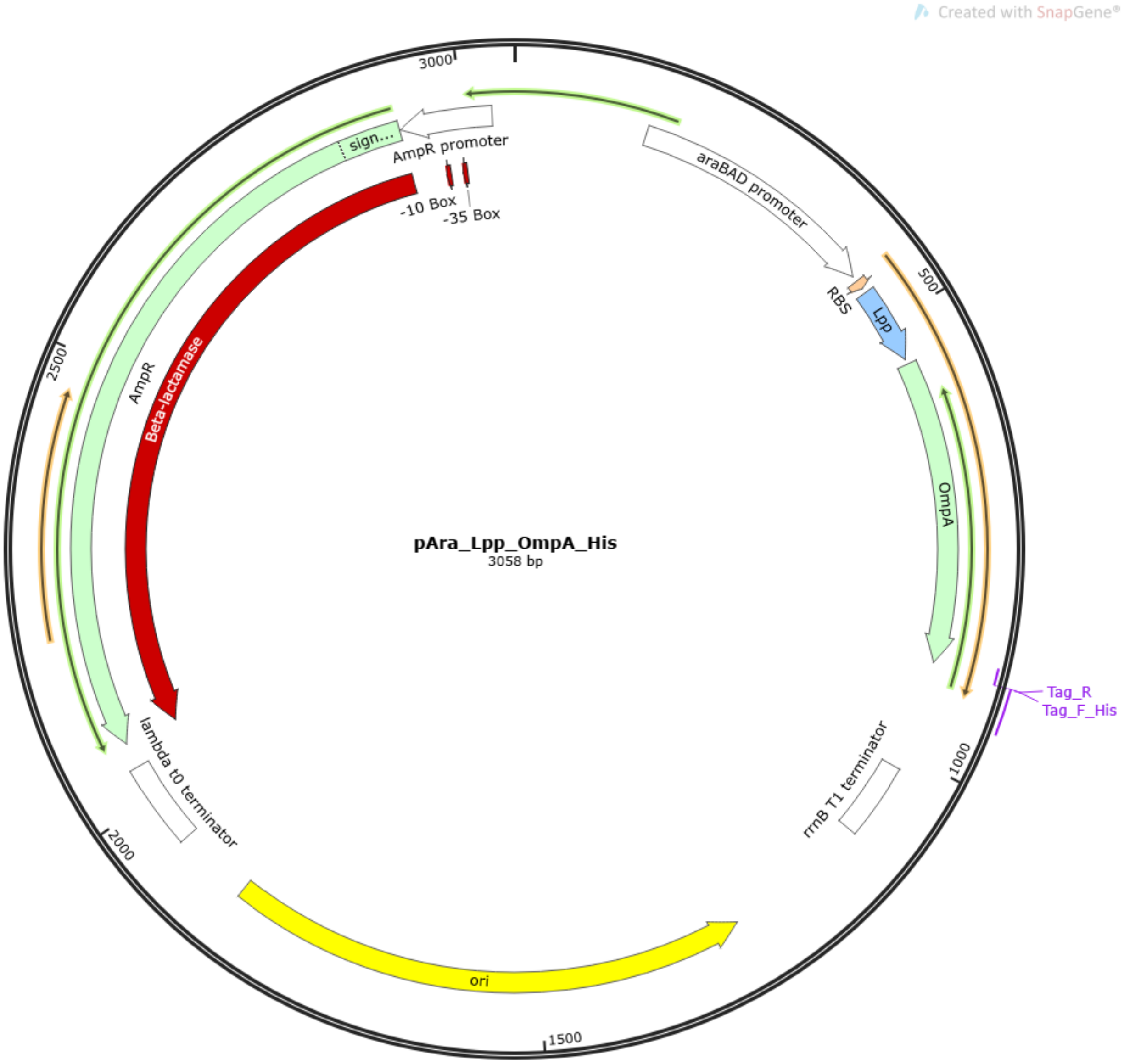
Appendix Figure 10. pFYS_019. Lpp-OmpA-mCherry under control of araBAD promoter, for arabinose inducible surface display of mCherry in gram negative bacteria.

```
  1 gacgtctgtg caagtactac tgttctgcag tcacttgagt tcgataccca gctgggtgga
 61 gtgcaccaag gagcatgcga aggaaacgtt tcgcagaagc ttccgcaagg taccactttg
121 ccgcggagta tttgtacatt tgaaggatcc aagaaaccaa tagtccatat tgcatcagac
181 attgccgtca ctgcgtcttt tactggctct tctcgctaac caaaccggta accccgctta
```

```
 241 ttaaaagcat tctgtaacaa agcgggacca aagccatgac aaaagcgcgt aacaaaagtg
 301 tctataatca cggcagaaaa gtccacattg attatttgca cggcgtcaca ctttgctatg
 361 ccatagcatt tttatccata agattagcgg atcatacctg acgcttttta tcgcaactct
 421 ctactgtttc tccatccatg gaggaaggag gagtcgacat gaaagctact aaactggtac
 481 tgggcgcggt aatcctgggt tctactctgc tggcaggttg ctccagcaac gctaaaatcg
 541 atcaggaatt caacccgtat gttggctttg aaatgggtta cgactggtta ggtcgtatgc
 601 cgtacaaagg cagcgttgaa aacggtgcat acaaagctca gggcgttcaa ctgaccgcta
 661 aactgggtta cccaatcact gacgacctgg acatctacac tcgtctgggt ggcatggtat
 721 ggcgtgcaga cactaaatcc aacgtttatg gtaaaaacca cgacaccggc gtttctccgg
 781 tcttcgctgg cggtgttgag tacgcgatca ctcctgaaat cgctacccgt ctggaatacc
 841 agtggaccaa caacatcggt gacgcacaca ccatcggcac tcgtccggac aacgagctca
 901 tggtgagcaa gggcgaggag gataacatgg ccatcatcaa ggagttcatg cgcttcaagg
 961 ttcacatgga gggctccgtg aacggccacg agttcgagat cgagggcgag ggcgagggcc
1021 gcccctacga gggcacccag accgccaagc tgaaggtgac caagggtggc cccctgccct
1081 tcgcctggga catcctgtcc cctcagttca tgtacggctc caaggcctac gtgaagcacc
1141 ccgccgacat ccccgactac ttgaagctgt ccttccccga gggcttcaag tgggagcgcg
1201 tgatgaactt cgaggacggc ggcgtggtga ccgtgaccca ggactcctcc ctgcaagacg
1261 gcgagttcat ctacaaggtg aagctgcgcg gcaccaactt cccctccgac ggccccgtaa
1321 tgcagaagaa gactatgggc tgggaggcct cctccgagcg gatgtacccc gaggacggcg
1381 cgctgaaggg cgagatcaag cagaggctga agctgaagga cggcggccac tacgacgctg
1441 aggtcaagac cacctacaag gccaagaagc ccgtgcaact gcccggcgcg tacaacgtca
1501 acatcaagtt ggacatcacc tcccacaacg aggactacac catcgtggaa cagtacgaac
1561 gcgccgaggg ccgccactcc accggcggca tggacgagct gtataagtaa cccgggacta
1621 cacaattgtc cccggcgcc aggggttgata tctatcgccc tagggaccgt ctcgagagaa
1681 tcaatattaa tccaacgcgt ggcatcaaat aaaacgaaag gctcagtcga aagactgggc
1741 ctttcgtttt atctgttgtt tgtcggtgaa cgctctcctg agtaggacaa atccgccgcc
1801 ctagacttag gcgttcggct gcggcgagcg gtatcagctc actcaaaggc ggtaatacgg
1861 ttatccacag aatcagggga taacgcagga aagaacatgt gagcaaaagg ccagcaaaag
1921 gccaggaacc gtaaaaaggc cgcgttgctg gcgtttttcc ataggctccg cccccctgac
1981 gagcatcaca aaaatcgacg ctcaagtcag aggtggcgaa acccgacagg actataaaga
2041 taccaggcgt ttccccctgg aagctccctc gtgcgctctc ctgttccgac cctgccgctt
2101 accggatacc tgtccgcctt tctcccttcg ggaagcgtgg cgctttctca tagctcacgc
2161 tgtaggtatc tcagttcggt gtaggtcgtt cgctccaagc tgggctgtct gcacgaaccc
2221 cccgttcagc ccgaccgctg cgccttatcc ggtaactatc gtcttgagtc caacccggta
2281 agacacgact tatcgccact ggcagcagcc actggtaaca ggattagcag agcgaggtat
2341 gtaggcggtg ctacagagtt cttgaagtgg tggcctaact acggctacac tagaaggaca
2401 gtatttggta tctgcgctct gctgaagcca gttaccttcg gaaaaagagt tggtagctct
2461 tgatccggca aacaaaccac cgctggtagc ggtggttttt ttgtttgcaa gcagcagatt
2521 acgcgcagaa aaaaaggatc tcaagaagat cctttgatct tttctacggg gtctgacgct
2581 cagtggaacg aaaactcacg ttaagggatt ttggtcatgg ctagtgcttg gattctcacc
2641 aataaaaaac gcccggcggc aaccgagcgt tctgaacaaa tccagatgga gttctgaggt
2701 cattactgga tctatcaaca ggagtccaag ccaattcgta aacttggtct gacagttacc
2761 aatgcttaat cagtgaggca cctatctcag cgatctgtct atttcgttca tccatagttg
2821 cctgactccc cgtcgtgtag ataactacga tacgggaggg cttaccatct ggccccagtg
2881 ctgcaatgat accgcgagac ccacgctcac cggctccaga tttatcagca ataaaccagc
2941 cagccggaag ggccgagcgc agaagtggtc ctgcaacttt atccgcctcc atccagtcta
3001 ttaattgttg ccgggaagct agagtaagta gttcgccagt taatagtttg cgcagcgttg
3061 ttgccattgc tacaggcatc gtggtgtcac gctcgtcgtt tggtatggct tcattcagct
3121 ccggttccca acgatcaagg cgagttacat gatcccccat gttgtgcaaa aaagcggtta
3181 gctccttcgg tcctccgatc gttgtcagaa gtaagttggc cgcagtgtta tcactcatgg
3241 ttatggcagc actgcataat tctcttactg tcatgccatc cgtaagatgc ttttctgtga
3301 ctggtgagta ttcaaccaag tcattctgag aatagtgtat gcggcgaccg agttgctctt
3361 gcccggcgtc aatacgggat aataccgcgc cacatagcag aactttaaaa gtgctcatca
3421 ttggaaagcg ttcttcgggg cgaaaactct caaggatctt accgctgttg agatccagtt
3481 cgatgtaacc cactcgcgca cccaactgat cttcagcatc ttttactttc accagcgttt
3541 ctgggtgagc aaaaacagga aggcaaaatg ccgcaaaaaa gggaataagg gcgacacgga
3601 aatgttgaat actcatactc ttcctttttc agtattattg aagcatttat cagggttatt
```

3661 gtctcatgag cggatacata tttgaatgta tttagaaaaa taaacaaata ggggttccgc
3721 gcacatttcc ccgaaaagtg ccacct

Appendix C.3.2 pAra-lpp-OmpA-His



Appendix Figure 11. pAra-lpp-OmpA-His. Arabinose-inducible surface displayed His tag on the C-terminal end of lpp-OmpA.

1 gacgtctgtg caagtactac tgttctgcag tcacttgagt tcgataccca gctgggtgga

61 gtgcaccaag gagcatgcga aggaaacgtt tcgcagaagc ttccgcaagg taccactttg

121 ccgcggagta tttgtacatt tgaaggatcc aagaaaccaa tagtccatat tgcatcagac

181 attgccgtca ctgcgtcttt tactggctct tctcgctaac caaaccggta accccgctta

241 ttaaaagcat tctgtaacaa agcgggacca aagccatgac aaaagcgcgt aacaaaagtg

301 tctataatca cggcagaaaa gtccacattg attatttgca cggcgtcaca ctttgctatg

361 ccatagcatt tttatccata agattagcgg atcatacctg acgcttttta tcgcaactct

421 ctactgtttc tccatccatg gaggaaggag gagtcgacat gaaagctact aaactggtac

481 tgggcgcggt aatcctgggt tctactctgc tggcaggttg ctccagcaac gctaaaatcg

541 atcaggaatt caacccgtat gttggctttg aaatgggtta cgactggtta ggtcgtatgc

601 cgtacaaagg cagcgttgaa aacggtgcat acaaagctca gggcgttcaa ctgaccgcta

661 aactgggtta cccaatcact gacgacctgg acatctacac tcgtctgggt ggcatggtat

721 ggcgtgcaga cactaaatcc aacgtttatg gtaaaaacca cgacaccggc gtttctccgg

781 tcttcgctgg cggtgttgag tacgcgatca ctcctgaaat cgctacccgt ctggaatacc

841 agtggaccaa caacatcggt gacgcacaca ccatcggcac tcgtccggac aacgagctca

901 atcatcacca tcaccatcac taaccgggac tacacaattg tcccccggcg ccagggttga

961 tatctatcgc cctagggacc gtctcgagag aatcaatatt aatccaacgc gtggcatcaa

1021 ataaaacgaa aggctcagtc gaaagactgg gcctttcgtt ttatctgttg tttgtcggtg

1081 aacgctctcc tgagtaggac aaatccgccg ccctagactt aggcgttcgg ctgcggcgag

1141 cggtatcagc tcactcaaag gcggtaatac ggttatccac agaatcaggg gataacgcag

1201 gaaagaacat gtgagcaaaa ggccagcaaa aggccaggaa ccgtaaaaag gccgcgttgc

1261 tggcgttttt ccataggctc cgcccccctg acgagcatca caaaaatcga cgctcaagtc

1321 agaggtggcg aaacccgaca ggactataaa gataccaggc gtttccccct ggaagctccc

1381 tcgtgcgctc tcctgttccg accctgccgc ttaccggata cctgtccgcc tttctccctt

1441 cgggaagcgt ggcgctttct catagctcac gctgtaggta tctcagttcg gtgtaggtcg

1501 ttcgctccaa gctgggctgt ctgcacgaac cccccgttca gcccgaccgc tgcgccttat

1561 ccggtaacta tcgtcttgag tccaacccgg taagacacga cttatcgcca ctggcagcag

1621 ccactggtaa caggattagc agagcgaggt atgtaggcgg tgctacagag ttcttgaagt

1681 ggtggcctaa ctacggctac actagaagga cagtatttgg tatctgcgct ctgctgaagc

1741 cagttacctt cggaaaaaga gttggtagct cttgatccgg caaacaaacc accgctggta

1801 gcggtggttt ttttgtttgc aagcagcaga ttacgcgcag aaaaaaagga tctcaagaag

1861 atcctttgat cttttctacg gggtctgacg ctcagtggaa cgaaaactca cgttaaggga

1921 ttttggtcat ggctagtgct tggattctca ccaataaaaa acgcccggcg gcaaccgagc

1981 gttctgaaca aatccagatg gagttctgag gtcattactg gatctatcaa caggagtcca

2041 agccaattcg taaacttggt ctgacagtta ccaatgctta atcagtgagg cacctatctc

2101 agcgatctgt ctatttcgtt catccatagt tgcctgactc cccgtcgtgt agataactac

2161 gatacgggag ggcttaccat ctggccccag tgctgcaatg ataccgcgag acccacgctc

2221 accggctcca gatttatcag caataaacca gccagccgga agggccgagc gcagaagtgg

2281 tcctgcaact ttatccgcct ccatccagtc tattaattgt tgccgggaag ctagagtaag

2341 tagttcgcca gttaatagtt tgcgcagcgt tgttgccatt gctacaggca tcgtggtgtc

2401 acgctcgtcg tttggtatgg cttcattcag ctccggttcc caacgatcaa ggcgagttac

2461 atgatccccc atgttgtgca aaaaagcggt tagctccttc ggtcctccga tcgttgtcag

2521 aagtaagttg gccgcagtgt tatcactcat ggttatggca gcactgcata attctcttac

2581 tgtcatgcca tccgtaagat gcttttctgt gactggtgag tattcaacca agtcattctg

2641 agaatagtgt atgcggcgac cgagttgctc ttgcccggcg tcaatacggg ataataccgc

2701 gccacatagc agaactttaa aagtgctcat cattggaaag cgttcttcgg ggcgaaaact

2761 ctcaaggatc ttaccgctgt tgagatccag ttcgatgtaa cccactcgcg cacccaactg

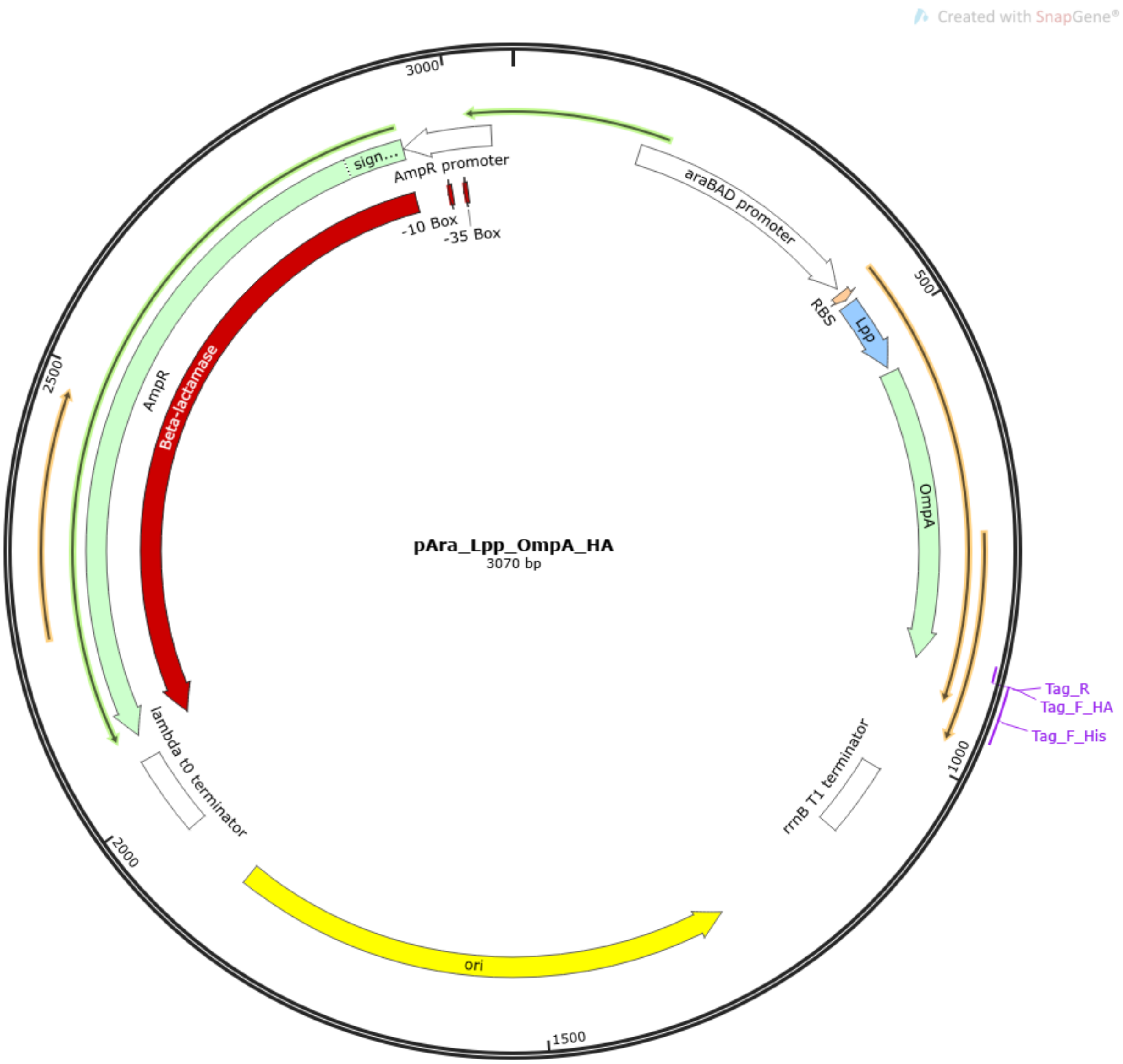
2821 atcttcagca tcttttactt tcaccagcgt ttctgggtga gcaaaaacag gaaggcaaaa

2881 tgccgcaaaa aagggaataa gggcgacacg gaaatgttga atactcatac tcttcctttt

2941 tcagtattat tgaagcattt atcagggtta ttgtctcatg agcggataca tatttgaatg

3001 tatttagaaa aataaacaaa taggggttcc gcgcacattt ccccgaaaag tgccacct

Appendix C.3.3 pAra-lpp-OmpA-HA



Appendix Figure 12. pAra-Lpp-OmpA-HA. Arabinose inducible surface displayed HA tag on the C-terminal end of lpp-OmpA.

```
  1 gacgtctgtg caagtactac tgttctgcag tcacttgagt tcgataccca gctgggtgga
 61 gtgcaccaag gagcatgcga aggaaacgtt tcgcagaagc ttccgcaagg taccactttg
121 ccgcggagta tttgtacatt tgaaggatcc aagaaaccaa tagtccatat tgcatcagac
181 attgccgtca ctgcgtcttt tactggctct tctcgctaac caaaccggta accccgctta
```

241 ttaaaagcat tctgtaacaa agcgggacca aagccatgac aaaagcgcgt aacaaaagtg

301 tctataatca cggcagaaaa gtccacattg attatttgca cggcgtcaca ctttgctatg

361 ccatagcatt tttatccata agattagcgg atcatacctg acgcttttta tcgcaactct

421 ctactgtttc tccatccatg gaggaaggag gagtcgacat gaaagctact aaactggtac

481 tgggcgcggt aatcctgggt tctactctgc tggcaggttg ctccagcaac gctaaaatcg

541 atcaggaatt caacccgtat gttggctttg aaatgggtta cgactggtta ggtcgtatgc

601 cgtacaaagg cagcgttgaa aacggtgcat acaaagctca gggcgttcaa ctgaccgcta

661 aactgggtta cccaatcact gacgacctgg acatctacac tcgtctgggt ggcatggtat

721 ggcgtgcaga cactaaatcc aacgtttatg gtaaaaacca cgacaccggc gtttctccgg

781 tcttcgctgg cggtgttgag tacgcgatca ctcctgaaat cgctacccgt ctggaatacc

841 agtggaccaa caacatcggt gacgcacaca ccatcggcac tcgtccggac aacgagctca

901 ccatgtaccc atacgatgtt ccagattacg cttaaccggg actacacaat tgtcccccgg

961 cgccagggtt gatatctatc gccctaggga ccgtctcgag agaatcaata ttaatccaac

1021 gcgtggcatc aaataaaacg aaaggctcag tcgaaagact gggcctttcg ttttatctgt

1081 tgtttgtcgg tgaacgctct cctgagtagg acaaatccgc cgccctagac ttaggcgttc

1141 ggctgcggcg agcggtatca gctcactcaa aggcggtaat acggttatcc acagaatcag

1201 gggataacgc aggaaagaac atgtgagcaa aaggccagca aaaggccagg aaccgtaaaa

1261 aggccgcgtt gctggcgttt ttccataggc tccgcccccc tgacgagcat cacaaaaatc

1321 gacgctcaag tcagaggtgg cgaaacccga caggactata aagataccag gcgtttcccc

1381 ctggaagctc cctcgtgcgc tctcctgttc cgaccctgcc gcttaccgga tacctgtccg

1441 cctttctccc ttcgggaagc gtggcgcttt ctcatagctc acgctgtagg tatctcagtt

1501 cggtgtaggt cgttcgctcc aagctgggct gtctgcacga accccccgtt cagcccgacc

1561 gctgcgcctt atccggtaac tatcgtcttg agtccaaccc ggtaagacac gacttatcgc

1621 cactggcagc agccactggt aacaggatta gcagagcgag gtatgtaggc ggtgctacag

1681 agttcttgaa gtggtggcct aactacggct acactagaag gacagtattt ggtatctgcg

1741 ctctgctgaa gccagttacc ttcggaaaaa gagttggtag ctcttgatcc ggcaaacaaa

1801 ccaccgctgg tagcggtggt ttttttgttt gcaagcagca gattacgcgc agaaaaaaag

1861 gatctcaaga agatcctttg atcttttcta cggggtctga cgctcagtgg aacgaaaact

1921 cacgttaagg gattttggtc atggctagtg cttggattct caccaataaa aaacgcccgg

1981 cggcaaccga gcgttctgaa caaatccaga tggagttctg aggtcattac tggatctatc

2041 aacaggagtc caagccaatt cgtaaacttg gtctgacagt taccaatgct taatcagtga

2101 ggcacctatc tcagcgatct gtctatttcg ttcatccata gttgcctgac tccccgtcgt

2161 gtagataact acgatacggg agggcttacc atctggcccc agtgctgcaa tgataccgcg

2221 agacccacgc tcaccggctc cagatttatc agcaataaac cagccagccg gaagggccga

2281 gcgcagaagt ggtcctgcaa ctttatccgc ctccatccag tctattaatt gttgccggga

2341 agctagagta agtagttcgc cagttaatag tttgcgcagc gttgttgcca ttgctacagg

2401 catcgtggtg tcacgctcgt cgtttggtat ggcttcattc agctccggtt cccaacgatc

2461 aaggcgagtt acatgatccc ccatgttgtg caaaaaagcg gttagctcct tcggtcctcc

2521 gatcgttgtc agaagtaagt tggccgcagt gttatcactc atggttatgg cagcactgca

2581 taattctctt actgtcatgc catccgtaag atgcttttct gtgactggtg agtattcaac

2641 caagtcattc tgagaatagt gtatgcggcg accgagttgc tcttgcccgg cgtcaatacg

2701 ggataatacc gcgccacata gcagaacttt aaaagtgctc atcattggaa agcgttcttc

2761 ggggcgaaaa ctctcaagga tcttaccgct gttgagatcc agttcgatgt aacccactcg

2821 cgcacccaac tgatcttcag catcttttac tttcaccagc gtttctgggt gagcaaaaac

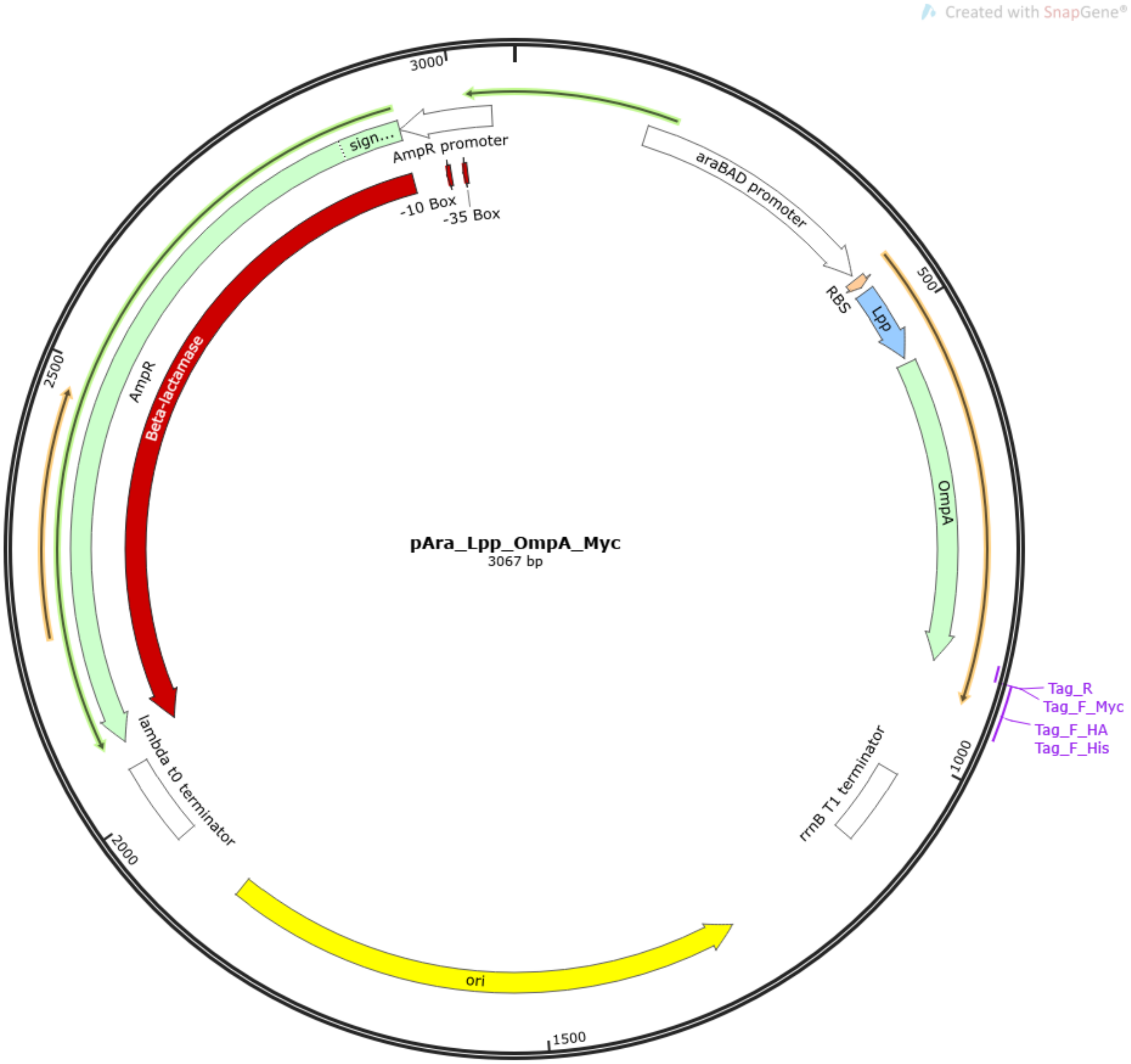
2881 aggaaggcaa aatgccgcaa aaaagggaat aagggcgaca cggaaatgtt gaatactcat

2941 actcttcctt tttcagtatt attgaagcat ttatcagggt tattgtctca tgagcggata

3001 catatttgaa tgtatttaga aaaataaaca aataggggtt ccgcgcacat ttccccgaaa

3061 agtgccacct

Appendix C.3.4 pAra-Lpp-OmpA-myc



Appendix Figure 13. pAra-Lpp-OmpA-myc. Arabinose inducible surface displayed myc tag on the C-terminal end of lpp-OmpA.

```
  1 gacgtctgtg caagtactac tgttctgcag tcacttgagt tcgataccca gctgggtgga
 61 gtgcaccaag gagcatgcga aggaaacgtt tcgcagaagc ttccgcaagg taccactttg
121 ccgcggagta tttgtacatt tgaaggatcc aagaaaccaa tagtccatat tgcatcagac
181 attgccgtca ctgcgtcttt tactggctct tctcgctaac caaaccggta accccgctta
```

```
 241 ttaaaagcat tctgtaacaa agcgggacca aagccatgac aaaagcgcgt aacaaaagtg

 301 tctataatca cggcagaaaa gtccacattg attatttgca cggcgtcaca ctttgctatg

 361 ccatagcatt tttatccata agattagcgg atcatacctg acgcttttta tcgcaactct

 421 ctactgtttc tccatccatg gaggaaggag gagtcgacat gaaagctact aaactggtac

 481 tgggcgcggt aatcctgggt tctactctgc tggcaggttg ctccagcaac gctaaaatcg

 541 atcaggaatt caacccgtat gttggctttg aaatgggtta cgactggtta ggtcgtatgc

 601 cgtacaaagg cagcgttgaa aacggtgcat acaaagctca gggcgttcaa ctgaccgcta

 661 aactgggtta cccaatcact gacgacctgg acatctacac tcgtctgggt ggcatggtat

 721 ggcgtgcaga cactaaatcc aacgtttatg gtaaaaacca cgacaccggc gtttctccgg

 781 tcttcgctgg cggtgttgag tacgcgatca ctcctgaaat cgctacccgt ctggaatacc

 841 agtggaccaa caacatcggt gacgcacaca ccatcggcac tcgtccggac aacgagctcg

 901 aacaaaaact catctcagaa gaggatctgt aaccgggact acacaattgt cccccggcgc

 961 cagggttgat atctatcgcc ctagggaccg tctcgagaga atcaatatta atccaacgcg

1021 tggcatcaaa taaaacgaaa ggctcagtcg aaagactggg cctttcgttt tatctgttgt

1081 ttgtcggtga acgctctcct gagtaggaca aatccgccgc cctagactta ggcgttcggc

1141 tgcggcgagc ggtatcagct cactcaaagg cggtaatacg gttatccaca gaatcagggg

1201 ataacgcagg aaagaacatg tgagcaaaag gccagcaaaa ggccaggaac cgtaaaaagg

1261 ccgcgttgct ggcgtttttc cataggctcc gcccccctga cgagcatcac aaaaatcgac

1321 gctcaagtca gaggtggcga aacccgacag gactataaag ataccaggcg tttccccctg

1381 gaagctccct cgtgcgctct cctgttccga ccctgccgct taccggatac ctgtccgcct

1441 ttctcccttc gggaagcgtg gcgctttctc atagctcacg ctgtaggtat ctcagttcgg

1501 tgtaggtcgt tcgctccaag ctgggctgtc tgcacgaacc ccccgttcag cccgaccgct

1561 gcgccttatc cggtaactat cgtcttgagt ccaacccggt aagacacgac ttatcgccac

1621 tggcagcagc cactggtaac aggattagca gagcgaggta tgtaggcggt gctacagagt

1681 tcttgaagtg gtggcctaac tacggctaca ctagaaggac agtatttggt atctgcgctc

1741 tgctgaagcc agttaccttc ggaaaaagag ttggtagctc ttgatccggc aaacaaacca

1801 ccgctggtag cggtggtttt tttgtttgca agcagcagat tacgcgcaga aaaaaaggat

1861 ctcaagaaga tcctttgatc ttttctacgg ggtctgacgc tcagtggaac gaaaactcac

1921 gttaagggat tttggtcatg gctagtgctt ggattctcac caataaaaaa cgcccggcgg
```

1981 caaccgagcg ttctgaacaa atccagatgg agttctgagg tcattactgg atctatcaac

2041 aggagtccaa gccaattcgt aaacttggtc tgacagttac caatgcttaa tcagtgaggc

2101 acctatctca gcgatctgtc tatttcgttc atccatagtt gcctgactcc ccgtcgtgta

2161 gataactacg atacgggagg gcttaccatc tggccccagt gctgcaatga taccgcgaga

2221 cccacgctca ccggctccag atttatcagc aataaaccag ccagccggaa gggccgagcg

2281 cagaagtggt cctgcaactt tatccgcctc catccagtct attaattgtt gccgggaagc

2341 tagagtaagt agttcgccag ttaatagttt gcgcagcgtt gttgccattg ctacaggcat

2401 cgtggtgtca cgctcgtcgt ttggtatggc ttcattcagc tccggttccc aacgatcaag

2461 gcgagttaca tgatccccca tgttgtgcaa aaaagcggtt agctccttcg gtcctccgat

2521 cgttgtcaga agtaagttgg ccgcagtgtt atcactcatg gttatggcag cactgcataa

2581 ttctcttact gtcatgccat ccgtaagatg cttttctgtg actggtgagt attcaaccaa

2641 gtcattctga gaatagtgta tgcggcgacc gagttgctct tgcccggcgt caatacggga

2701 taataccgcg ccacatagca gaactttaaa agtgctcatc attggaaagc gttcttcggg

2761 gcgaaaactc tcaaggatct taccgctgtt gagatccagt tcgatgtaac ccactcgcgc

2821 acccaactga tcttcagcat cttttacttt caccagcgtt tctgggtgag caaaaacagg

2881 aaggcaaaat gccgcaaaaa agggaataag ggcgacacgg aaatgttgaa tactcatact

2941 cttccttttt cagtattatt gaagcattta tcagggttat tgtctcatga gcggatacat

3001 atttgaatgt atttagaaaa ataaacaaat aggggttccg cgcacatttc cccgaaaagt

3061 gccacct

Appendix C.3.5 pTetR-Int-His

Appendix Figure 14. pTetR-Int-His. Tetracycline inducible surface-displayed His tag fused to the N-terminus domain of Intimin.

```
  1 tggtgcaaaa cctttgcggt atggcatgat agcgcctact agagaaagag gagaaatact

 61 agatgtccag attagataaa agtaaagtga ttaacagcgc attagagctg cttaatgagg

121 tcggaatcga aggtttaaca acccgtaaac tcgcccagaa gctaggtgta gagcagccta
```

181 cattgtattg gcatgtaaaa aataagcggg ctttgctcga cgccttagcc attgagatgt

241 tagataggca ccatactcac ttttgccctt tagaagggga aagctggcaa gatttttttac

301 gtaataacgc taaaagtttt agatgtgctt tactaagtca tcgcgatgga gcaaaagtac

361 atttaggtac acggcctaca gaaaaacagt atgaaactct cgaaaatcaa ttagcctttt

421 tatgccaaca aggtttttca ctagagaatg cattatatgc actcagcgct gtggggcatt

481 ttactttagg ttgcgtattg gaagatcaag agcatcaagt cgctaaagaa gaaagggaaa

541 cacctactac tgatagtatg ccgccattat tacgacaagc tatcgaatta tttgatcacc

601 aaggtgcaga gccagccttc ttattcggcc ttgaattgat catatgcgga ttagaaaaac

661 aacttaaatg tgaaagtggg tcctaataat actagagcca ggcatcaaat aaaacgaaag

721 gctcagtcga aagactgggc ctttcgtttt atctgttgtt tgtcggtgaa cgctctctac

781 tagagtcaca ctggctcacc ttcgggtggg cctttctgcg tttatatact agagtcccta

841 tcagtgatag agattgacat ccctatcagt gatagagata ctgagcacta ctagagaaag

901 aggagaaata ctagatgatt actcatggtt gttatacccg gacccggcac aagcataagc

961 taaaaaaaac attgattatg cttagtgctg gtttaggatt gtttttttat gttaatcaga

1021 actcatttgc aaatggtgaa aattatttta aattgggttc ggattcaaaa ctgttaactc

1081 atgatagcta tcagaatcgc ctttttata cgttgaaaac tggtgaaact gttgccgatc

1141 tttctaaatc gcaagatatt aatttatcga cgatttggtc gttgaataag catttataca

1201 gttctgaaag cgaaatgatg aaggccgcgc ctggtcagca gatcattttg ccactcaaaa

1261 aacttccctt tgaatacagt gcactaccac ttttaggttc ggcacctctt gttgctgcgg

1321 gtggtgttgc tggtcacacg aataaactga ctaaaatgtc cccggacgtg accaaaagca

1381 acatgaccga tgacaaggca ttaaattatg cggcacaaca ggcggcgagt ctcggtagcc

1441 agcttcagtc gcgatctctg aacggcgatt acgcgaaaga taccgctctt ggtatcgctg

1501 gtaaccaggc ttcgtcacag ttgcaggcct ggttacaaca ttatggaacg gcagaggtta

1561 atctgcaaag tggtaataac tttgacggta gttcactgga cttcttatta ccgttctatg

1621 attccgaaaa aatgctggca tttggtcagg tcggagcgcg ttacattgac tcccgcttta

1681 cggcaaattt aggtgcgggt cagcgttttt tccttcctgc aaacatgttg ggctataacg

1741 tcttcattga tcaggatttt tctggtgata atacccgttt aggtattggt ggcgaatact

1801 ggcgagacta tttcaaaagt agcgttaacg gctatttccg catgagcggc tggcatgagt

1861 catacaataa gaaagactat gatgagcgcc cagcaaatgg cttcgatatc cgttttaatg

1921 gctatctacc gtcatatccg gcattaggcg ccaagctgat atatgagcag tattatggtg

1981 ataatgttgc tttgtttaat tctgataagc tgcaatcgaa tcctggtgcg gcgaccgttg

2041 gtgtaaacta tactccgatt cctctggtga cgatggggat cgattaccgt catggtacgg

2101 gtaatgaaaa tgatctcctt tactcaatgc agttccgtta tcagtttgat aaatcgtggt

2161 ctcagcaaat tgaaccacag tatgttaacg agttaagaac attatcaggc agccgttacg

2221 atctggttca gcgtaataac aatattattc tggagtacaa gaagcaggat attctttctc

2281 tgaatattcc gcatgatatt aatggtactg aacacagtac gcagaagatt cagttgatcg

2341 ttaagagcaa atacggtctg gatcgtatcg tctgggatga tagtgcatta cgcagtcagg

2401 gcggtcagat tcagcatagc ggaagccaaa gcgcacaaga ctaccaggct attttgcctg

2461 cttatgtgca aggtggcagc aatatttata aagtgacggc tcgcgcctat gaccgtaatg

2521 gcaatagctc taacaatgta cagcttacta ttaccgttct gtcgaatggt caagttgtcg

2581 accaggttgg ggtaacggac tttacggcgg ataagacttc ggctaaagcg gataacgccg

2641 ataccattac ttataccgcg acggtgaaaa agaatggggt agctcaggct aatgtccctg

2701 tttcatttaa tattgtttca ggaactgcaa ctcttggggc aaatagtgcc aaaacggatg

2761 ctaacggtaa ggcaaccgta acgttgaagt cgagtacgcc aggacaggtc gtcgtgtctg

2821 ctaaaaccgc ggagatgact tcagcactta atgccagtgc ggttatattt tttgatggtg

2881 cgactagaca tcaccatcac catcactaag atatctccgg caaaaaaggg caaggtgtca

2941 ccaccctgcc ctttttcttt aaaaccgaaa agattacttc gcgttatgca ggcttcctcg

3001 ctcactgact cgctgcgctc ggtcgttcgg ctgcggcgag cggtatcagc tcactcaaag

3061 gcggtaatct cgagtcccgt caagtcagcg taatgctctg ccagtgttac aaccaattaa

3121 ccaattctga ttagaaaaac tcatcgagca tcaaatgaaa ctgcaattta ttcatatcag

3181 gattatcgat accatatttt tgaaaaagcc gtttctgtaa tgaaggagaa aactcaccga

3241 ggcagttcca taggatggca agatcctggt atcggtctgc gattccgact cgtccaacat

3301 caatacaacc tattaatttc ccctcgtcaa aaataaggtt atcaagtgag aaatcaccat

3361 gagtgacgac tgaatccggt gagaatggca aaagcttatg catttctttc cagacttgtt

3421 caacaggcca gccattacgc tcgtcatcaa aatcactcgc atcaaccaaa ccgttattca

3481 ttcgtgattg cgcctgagcg agacgaaata cgcgatcgct gttaaaagga caattacaaa

3541 caggaatcga atgcaaccgg cgcaggaaca ctgccagcgc atcaacaata ttttcacctg

3601 aatcaggata ttcttctaat acctggaatg ctgttttccc ggggatcgca gtggtgagta

3661 accatgcatc atcaggagta cggataaaat gcttgatggt cggaagaggc ataaattccg

3721 tcagccagtt tagtctgacc atctcatctg taacatcatt ggcaacgcta cctttgccat

3781 gtttcagaaa caactctggc gcatcgggct tcccatacaa tcgatagatt gtcgcacctg

3841 attgcccgac attatcgcga gcccatttat acccatataa atcagcatcc atgttggaat

3901 ttaatcgcgg cctggagcaa gacgtttccc gttgaatatg gctcataaca ccccttgtat

3961 tactgtttat gtaagcagac agttttattg ttcatgatga tatattttta tcttgtgcaa

4021 tgtaacatca gagattttga gacacaacgt ggctttgttg aataaatcga acttttgctg

4081 agttgaagga tcagatcacg catcttcccg acaacgcaga ccgttccgtg gcaaagcaaa

4141 agttcaaaat caccaactgg tccacctaca acaaagctct catcaaccgt ggctccctca

4201 ctttctggct ggatgatggg gcgattcagg cctggtatga gtcagcaaca ccttcttcac

4261 gaggcagacc tcagcgctag cggagtgtat actggcttac tatgttggca ctgatgaggg

4321 tgtcagtgaa gtgcttcatg tggcaggaga aaaaaggctg caccggtgcg tcagcagaat

4381 atgtgataca ggatatattc cgcttcctcg ctcactgact cgctacgctc ggtcgttcga

4441 ctgcggcgag cggaaatggc ttacgaacgg ggcggagatt tcctggaaga tgccaggaag

4501 atacttaaca gggaagtgag agggccgcgg caaagccgtt tttccatagg ctccgccccc

4561 ctgacaagca tcacgaaatc tgacgctcaa atcagtggtg gcgaaacccg acaggactat

4621 aaagatacca ggcgtttccc cctggcggct ccctcgtgcg ctctcctgtt cctgcctttc

4681 ggtttaccgg tgtcattccg ctgttatggc cgcgtttgtc tcattccacg cctgacactc

4741 agttccgggt aggcagttcg ctccaagctg gactgtatgc acgaaccccc cgttcagtcc

4801 gaccgctgcg ccttatccgg taactatcgt cttgagtcca acccggaaag acatgcaaaa

4861 gcaccactgg cagcagccac tggtaattga tttagaggag ttagtcttga agtcatgcgc

4921 cggttaaggc taaactgaaa ggacaagttt tggtgactgc gctcctccaa gccagttacc

4981 tcggttcaaa gagttggtag ctcagagaac cttcgaaaaa ccgccctgca aggcggtttt

5041 ttcgttttca gagcaagaga ttacgcgcag accaaaacga tctcaagaag atcatcttat

5101 taaggggtct gacgctcagt ggaacgaaaa ctcacgttaa gggattttgg tcatgagatt

5161 atcaaaaagg atcttcacct agatcctttt aaattaaaaa tgaagtttta aatcaatcta

5221 aagtatatat gagtaaactt ggtctgacag ttaccaatgc ttaatcagtg aggcacctat

5281 ctcagcgatc tgtctatttc gttcatccat agttgcctga ctccccgtcg tgtagataac

5341 tacgatacgg gagggcttac catctggccc cagtgctgca atgataccgc gagacccacg

5401 ctcaccggct ccagatttat cagcaataaa ccagccagcc ggaagggccg agcgcagaag

5461 tggtcctgca actttatccg cctccatcca gtctattcca tggtgccacc tgacgtctaa

5521 gaaaccatta ttatcatgac attaacctat aaaaataggc gtatcacgag gcagaatttc

5581 agataaaaaa aatccttagc tttcgctaag gatgatttct ggaattcgcg gccgcttcta

5641 gag

Appendix C.3.6 pTetR-Int-HA

Appendix Figure 15. pTetR-Int-HA. Tetracycline inducible surface-displayed HA tag fused to the N-terminus domain of Intimin.

1 tggtgcaaaa cctttgcggt atggcatgat agcgcctact agagaaagag gagaaatact

61 agatgtccag attagataaa agtaaagtga ttaacagcgc attagagctg cttaatgagg

121 tcggaatcga aggtttaaca acccgtaaac tcgcccagaa gctaggtgta gagcagccta

181 cattgtattg gcatgtaaaa aataagcggg ctttgctcga cgccttagcc attgagatgt

241 tagataggca ccatactcac ttttgccctt tagaaggga aagctggcaa gattttttac

301 gtaataacgc taaaagtttt agatgtgctt tactaagtca tcgcgatgga gcaaaagtac

361 atttaggtac acggcctaca gaaaaacagt atgaaactct cgaaaatcaa ttagcctttt

421 tatgccaaca aggtttttca ctagagaatg cattatatgc actcagcgct gtggggcatt

481 ttactttagg ttgcgtattg gaagatcaag agcatcaagt cgctaaagaa gaaagggaaa

541 cacctactac tgatagtatg ccgccattat tacgacaagc tatcgaatta tttgatcacc

601 aaggtgcaga gccagccttc ttattcggcc ttgaattgat catatgcgga ttagaaaaac

661 aacttaaatg tgaaagtggg tcctaataat actagagcca ggcatcaaat aaaacgaaag

721 gctcagtcga aagactgggc ctttcgtttt atctgttgtt tgtcggtgaa cgctctctac

781 tagagtcaca ctggctcacc ttcgggtggg cctttctgcg tttatatact agagtcccta

841 tcagtgatag agattgacat ccctatcagt gatagagata ctgagcacta ctagagaaag

901 aggagaaata ctagatgatt actcatggtt gttatacccg gacccggcac aagcataagc

961 taaaaaaaac attgattatg cttagtgctg gtttaggatt gtttttttat gttaatcaga

1021 actcatttgc aaatggtgaa aattatttta aattgggttc ggattcaaaa ctgttaactc

1081 atgatagcta tcagaatcgc ctttttata cgttgaaaac tggtgaaact gttgccgatc

1141 tttctaaatc gcaagatatt aatttatcga cgatttggtc gttgaataag catttataca

1201 gttctgaaag cgaaatgatg aaggccgcgc ctggtcagca gatcattttg ccactcaaaa

1261 aacttccctt tgaatacagt gcactaccac ttttaggttc ggcacctctt gttgctgcgg

1321 gtggtgttgc tggtcacacg aataaactga ctaaaatgtc cccggacgtg accaaaagca

1381 acatgaccga tgacaaggca ttaaattatg cggcacaaca ggcggcgagt ctcggtagcc

1441 agcttcagtc gcgatctctg aacggcgatt acgcgaaaga taccgctctt ggtatcgctg

1501 gtaaccaggc ttcgtcacag ttgcaggcct ggttacaaca ttatggaacg gcagaggtta

1561 atctgcaaag tggtaataac tttgacggta gttcactgga cttcttatta ccgttctatg

1621 attccgaaaa aatgctggca tttggtcagg tcggagcgcg ttacattgac tcccgcttta

1681 cggcaaattt aggtgcgggt cagcgttttt tccttcctgc aaacatgttg ggctataacg

1741 tcttcattga tcaggatttt tctggtgata atacccgttt aggtattggt ggcgaatact

1801 ggcgagacta tttcaaaagt agcgttaacg gctatttccg catgagcggc tggcatgagt

1861 catacaataa gaaagactat gatgagcgcc cagcaaatgg cttcgatatc cgttttaatg

1921 gctatctacc gtcatatccg gcattaggcg ccaagctgat atatgagcag tattatggtg

1981 ataatgttgc tttgtttaat tctgataagc tgcaatcgaa tcctggtgcg gcgaccgttg

2041 gtgtaaacta tactccgatt cctctggtga cgatggggat cgattaccgt catggtacgg

2101 gtaatgaaaa tgatctcctt tactcaatgc agttccgtta tcagtttgat aaatcgtggt

2161 ctcagcaaat tgaaccacag tatgttaacg agttaagaac attatcaggc agccgttacg

2221 atctggttca gcgtaataac aatattattc tggagtacaa gaagcaggat attctttctc

2281 tgaatattcc gcatgatatt aatggtactg aacacagtac gcagaagatt cagttgatcg

2341 ttaagagcaa atacggtctg gatcgtatcg tctgggatga tagtgcatta cgcagtcagg

2401 gcggtcagat tcagcatagc ggaagccaaa gcgcacaaga ctaccaggct attttgcctg

2461 cttatgtgca aggtggcagc aatatttata aagtgacggc tcgcgcctat gaccgtaatg

2521 gcaatagctc taacaatgta cagcttacta ttaccgttct gtcgaatggt caagttgtcg

2581 accaggttgg ggtaacggac tttacggcgg ataagacttc ggctaaagcg gataacgccg

2641 ataccattac ttataccgcg acggtgaaaa agaatggggt agctcaggct aatgtccctg

2701 tttcatttaa tattgtttca ggaactgcaa ctcttggggc aaatagtgcc aaaacggatg

2761 ctaacggtaa ggcaaccgta acgttgaagt cgagtacgcc aggacaggtc gtcgtgtctg

2821 ctaaaaccgc ggagatgact tcagcactta atgccagtgc ggttatattt tttgatggtg

2881 cgactagata cccatacgat gttccagatt acgcttaaga tatctccggc aaaaaagggc

2941 aaggtgtcac caccctgccc tttttcttta aaaccgaaaa gattacttcg cgttatgcag

3001 gcttcctcgc tcactgactc gctgcgctcg gtcgttcggc tgcggcgagc ggtatcagct

3061 cactcaaagg cggtaatctc gagtcccgtc aagtcagcgt aatgctctgc cagtgttaca

3121 accaattaac caattctgat tagaaaaact catcgagcat caaatgaaac tgcaatttat

3181 tcatatcagg attatcgata ccatattttt gaaaaagccg tttctgtaat gaaggagaaa

3241 actcaccgag gcagttccat aggatggcaa gatcctggta tcggtctgcg attccgactc

3301 gtccaacatc aatacaacct attaatttcc cctcgtcaaa aataaggtta tcaagtgaga

3361 aatcaccatg agtgacgact gaatccggtg agaatggcaa aagcttatgc atttctttcc

3421 agacttgttc aacaggccag ccattacgct cgtcatcaaa atcactcgca tcaaccaaac

3481 cgttattcat tcgtgattgc gcctgagcga gacgaaatac gcgatcgctg ttaaaaggac

3541 aattacaaac aggaatcgaa tgcaaccggc gcaggaacac tgccagcgca tcaacaatat

3601 tttcacctga atcaggatat tcttctaata cctggaatgc tgttttcccg gggatcgcag

3661 tggtgagtaa ccatgcatca tcaggagtac ggataaaatg cttgatggtc ggaagaggca

3721 taaattccgt cagccagttt agtctgacca tctcatctgt aacatcattg gcaacgctac

3781 ctttgccatg tttcagaaac aactctggcg catcgggctt cccatacaat cgatagattg

3841 tcgcacctga ttgcccgaca ttatcgcgag cccatttata cccatataaa tcagcatcca

3901 tgttggaatt taatcgcggc ctggagcaag acgtttcccg ttgaatatgg ctcataacac

3961 cccttgtatt actgtttatg taagcagaca gttttattgt tcatgatgat atatttttat

4021 cttgtgcaat gtaacatcag agattttgag acacaacgtg gctttgttga ataaatcgaa

4081 cttttgctga gttgaaggat cagatcacgc atcttcccga caacgcagac cgttccgtgg

4141 caaagcaaaa gttcaaaatc accaactggt ccacctacaa caaagctctc atcaaccgtg

4201 gctccctcac tttctggctg gatgatgggg cgattcaggc ctggtatgag tcagcaacac

4261 cttcttcacg aggcagacct cagcgctagc ggagtgtata ctggcttact atgttggcac

4321 tgatgagggt gtcagtgaag tgcttcatgt ggcaggagaa aaaaggctgc accggtgcgt

4381 cagcagaata tgtgatacag gatatattcc gcttcctcgc tcactgactc gctacgctcg

4441 gtcgttcgac tgcggcgagc ggaaatggct tacgaacggg gcggagattt cctggaagat

4501 gccaggaaga tacttaacag ggaagtgaga gggccgcggc aaagccgttt ttccataggc

4561 tccgcccccc tgacaagcat cacgaaatct gacgctcaaa tcagtggtgg cgaaacccga

4621 caggactata aagataccag gcgtttcccc ctggcggctc cctcgtgcgc tctcctgttc

4681 ctgcctttcg gtttaccggt gtcattccgc tgttatggcc gcgtttgtct cattccacgc

4741 ctgacactca gttccgggta ggcagttcgc tccaagctgg actgtatgca cgaacccccc

4801 gttcagtccg accgctgcgc cttatccggt aactatcgtc ttgagtccaa cccggaaaga

4861 catgcaaaag caccactggc agcagccact ggtaattgat ttagaggagt tagtcttgaa

4921 gtcatgcgcc ggttaaggct aaactgaaag gacaagtttt ggtgactgcg ctcctccaag

4981 ccagttacct cggttcaaag agttggtagc tcagagaacc ttcgaaaaac cgccctgcaa

5041 ggcggttttt tcgttttcag agcaagagat tacgcgcaga ccaaaacgat ctcaagaaga

5101 tcatcttatt aaggggtctg acgctcagtg gaacgaaaac tcacgttaag ggattttggt

5161 catgagatta tcaaaaagga tcttcaccta gatcctttta aattaaaaat gaagttttaa

5221 atcaatctaa agtatatatg agtaaacttg gtctgacagt taccaatgct taatcagtga

5281 ggcacctatc tcagcgatct gtctatttcg ttcatccata gttgcctgac tccccgtcgt

5341 gtagataact acgatacggg agggcttacc atctggcccc agtgctgcaa tgataccgcg

5401 agacccacgc tcaccggctc cagatttatc agcaataaac cagccagccg gaagggccga

5461 gcgcagaagt ggtcctgcaa ctttatccgc ctccatccag tctattccat ggtgccacct

5521 gacgtctaag aaaccattat tatcatgaca ttaacctata aaaataggcg tatcacgagg

5581 cagaatttca gataaaaaaa atccttagct ttcgctaagg atgatttctg gaattcgcgg

5641 ccgcttctag ag

Appendix C.3.7 pTetR-Int-myc

Appendix Figure 16. pTetR-Int-Myc. Tetracycline inducible surface-displayed myc tag fused to the N-terminus domain of Intimin.

1 tggtgcaaaa cctttgcggt atggcatgat agcgcctact agagaaagag gagaaatact

61 agatgtccag attagataaa agtaaagtga ttaacagcgc attagagctg cttaatgagg

121 tcggaatcga aggtttaaca acccgtaaac tcgcccagaa gctaggtgta gagcagccta

181 cattgtattg gcatgtaaaa aataagcggg ctttgctcga cgccttagcc attgagatgt

241 tagataggca ccatactcac ttttgccctt tagaagggga aagctggcaa gatttttttac

301 gtaataacgc taaaagtttt agatgtgctt tactaagtca tcgcgatgga gcaaaagtac

361 atttaggtac acggcctaca gaaaaacagt atgaaactct cgaaaatcaa ttagcctttt

421 tatgccaaca aggtttttca ctagagaatg cattatatgc actcagcgct gtggggcatt

481 ttactttagg ttgcgtattg gaagatcaag agcatcaagt cgctaaagaa gaaagggaaa

541 cacctactac tgatagtatg ccgccattat tacgacaagc tatcgaatta tttgatcacc

601 aaggtgcaga gccagccttc ttattcggcc ttgaattgat catatgcgga ttagaaaaac

661 aacttaaatg tgaaagtggg tcctaataat actagagcca ggcatcaaat aaaacgaaag

721 gctcagtcga aagactgggc ctttcgtttt atctgttgtt tgtcggtgaa cgctctctac

781 tagagtcaca ctggctcacc ttcgggtggg cctttctgcg tttatatact agagtcccta

841 tcagtgatag agattgacat ccctatcagt gatagagata ctgagcacta ctagagaaag

901 aggagaaata ctagatgatt actcatggtt gttatacccg gacccggcac aagcataagc

961 taaaaaaaac attgattatg cttagtgctg gtttaggatt gtttttttat gttaatcaga

1021 actcatttgc aaatggtgaa aattatttta aattgggttc ggattcaaaa ctgttaactc

1081 atgatagcta tcagaatcgc cttttttata cgttgaaaac tggtgaaact gttgccgatc

1141 tttctaaatc gcaagatatt aatttatcga cgatttggtc gttgaataag catttataca

1201 gttctgaaag cgaaatgatg aaggccgcgc ctggtcagca gatcattttg ccactcaaaa

1261 aacttccctt tgaatacagt gcactaccac ttttaggttc ggcacctctt gttgctgcgg

1321 gtggtgttgc tggtcacacg aataaactga ctaaaatgtc cccggacgtg accaaaagca

1381 acatgaccga tgacaaggca ttaaattatg cggcacaaca ggcggcgagt ctcggtagcc

1441 agcttcagtc gcgatctctg aacggcgatt acgcgaaaga taccgctctt ggtatcgctg

1501 gtaaccaggc ttcgtcacag ttgcaggcct ggttacaaca ttatggaacg gcagaggtta

1561 atctgcaaag tggtaataac tttgacggta gttcactgga cttcttatta ccgttctatg

1621 attccgaaaa aatgctggca tttggtcagg tcggagcgcg ttacattgac tcccgcttta

1681 cggcaaattt aggtgcgggt cagcgttttt tccttcctgc aaacatgttg ggctataacg

1741 tcttcattga tcaggatttt tctggtgata atacccgttt aggtattggt ggcgaatact

1801 ggcgagacta tttcaaaagt agcgttaacg gctatttccg catgagcggc tggcatgagt

1861 catacaataa gaaagactat gatgagcgcc cagcaaatgg cttcgatatc cgttttaatg

1921 gctatctacc gtcatatccg gcattaggcg ccaagctgat atatgagcag tattatggtg

1981 ataatgttgc tttgtttaat tctgataagc tgcaatcgaa tcctggtgcg gcgaccgttg

2041 gtgtaaacta tactccgatt cctctggtga cgatggggat cgattaccgt catggtacgg

2101 gtaatgaaaa tgatctcctt tactcaatgc agttccgtta tcagtttgat aaatcgtggt

2161 ctcagcaaat tgaaccacag tatgttaacg agttaagaac attatcaggc agccgttacg

2221 atctggttca gcgtaataac aatattattc tggagtacaa gaagcaggat attctttctc

2281 tgaatattcc gcatgatatt aatggtactg aacacagtac gcagaagatt cagttgatcg

2341 ttaagagcaa atacggtctg gatcgtatcg tctgggatga tagtgcatta cgcagtcagg

2401 gcggtcagat tcagcatagc ggaagccaaa gcgcacaaga ctaccaggct attttgcctg

2461 cttatgtgca aggtggcagc aatatttata aagtgacggc tcgcgcctat gaccgtaatg

2521 gcaatagctc taacaatgta cagcttacta ttaccgttct gtcgaatggt caagttgtcg

2581 accaggttgg ggtaacggac tttacggcgg ataagacttc ggctaaagcg gataacgccg

2641 ataccattac ttataccgcg acggtgaaaa agaatggggt agctcaggct aatgtccctg

2701 tttcatttaa tattgtttca ggaactgcaa ctcttggggc aaatagtgcc aaaacggatg

2761 ctaacggtaa ggcaaccgta acgttgaagt cgagtacgcc aggacaggtc gtcgtgtctg

2821 ctaaaaccgc ggagatgact tcagcactta atgccagtgc ggttatattt tttgatggtg

2881 cgactagaga acaaaaactc atctcagaag aggatctgta agatatctcc ggcaaaaaag

2941 ggcaaggtgt caccaccctg cccttttct ttaaaaccga aaagattact tcgcgttatg

3001 caggcttcct cgctcactga ctcgctgcgc tcggtcgttc ggctgcggcg agcggtatca

3061 gctcactcaa aggcggtaat ctcgagtccc gtcaagtcag cgtaatgctc tgccagtgtt

3121 acaaccaatt aaccaattct gattagaaaa actcatcgag catcaaatga aactgcaatt

3181 tattcatatc aggattatcg ataccatatt tttgaaaaag ccgtttctgt aatgaaggag

3241 aaaactcacc gaggcagttc cataggatgg caagatcctg gtatcggtct gcgattccga

3301 ctcgtccaac atcaatacaa cctattaatt tcccctcgtc aaaaataagg ttatcaagtg

3361 agaaatcacc atgagtgacg actgaatccg gtgagaatgg caaaagctta tgcatttctt

3421 tccagacttg ttcaacaggc cagccattac gctcgtcatc aaaatcactc gcatcaacca

3481 aaccgttatt cattcgtgat tgcgcctgag cgagacgaaa tacgcgatcg ctgttaaaag

3541 gacaattaca aacaggaatc gaatgcaacc ggcgcaggaa cactgccagc gcatcaacaa

3601 tattttcacc tgaatcagga tattcttcta atacctggaa tgctgtttc ccggggatcg

3661 cagtggtgag taaccatgca tcatcaggag tacggataaa atgcttgatg gtcggaagag

3721 gcataaattc cgtcagccag tttagtctga ccatctcatc tgtaacatca ttggcaacgc

3781 tacctttgcc atgtttcaga aacaactctg gcgcatcggg cttcccatac aatcgataga

3841 ttgtcgcacc tgattgcccg acattatcgc gagcccattt atacccatat aaatcagcat

3901 ccatgttgga atttaatcgc ggcctggagc aagacgtttc ccgttgaata tggctcataa

3961 caccccttgt attactgttt atgtaagcag acagttttat tgttcatgat gatatatttt

4021 tatcttgtgc aatgtaacat cagagatttt gagacacaac gtggctttgt tgaataaatc

4081 gaacttttgc tgagttgaag gatcagatca cgcatcttcc cgacaacgca gaccgttccg

4141 tggcaaagca aaagttcaaa atcaccaact ggtccaccta caacaaagct ctcatcaacc

4201 gtggctccct cactttctgg ctggatgatg gggcgattca ggcctggtat gagtcagcaa

4261 caccttcttc acgaggcaga cctcagcgct agcggagtgt atactggctt actatgttgg

4321 cactgatgag ggtgtcagtg aagtgcttca tgtggcagga gaaaaaaggc tgcaccggtg

4381 cgtcagcaga atatgtgata caggatatat tccgcttcct cgctcactga ctcgctacgc

4441 tcggtcgttc gactgcggcg agcggaaatg gcttacgaac ggggcggaga tttcctggaa

4501 gatgccagga agatacttaa cagggaagtg agagggccgc ggcaaagccg tttttccata

4561 ggctccgccc ccctgacaag catcacgaaa tctgacgctc aaatcagtgg tggcgaaacc

4621 cgacaggact ataaagatac caggcgtttc cccctggcgg ctccctcgtg cgctctcctg

4681 ttcctgcctt tcggtttacc ggtgtcattc cgctgttatg gccgcgtttg tctcattcca

4741 cgcctgacac tcagttccgg gtaggcagtt cgctccaagc tggactgtat gcacgaaccc

4801 cccgttcagt ccgaccgctg cgccttatcc ggtaactatc gtcttgagtc caacccggaa

4861 agacatgcaa aagcaccact ggcagcagcc actggtaatt gatttagagg agttagtctt

4921 gaagtcatgc gccggttaag gctaaactga aaggacaagt tttggtgact gcgctcctcc

4981 aagccagtta cctcggttca aagagttggt agctcagaga accttcgaaa aaccgccctg

5041 caaggcggtt ttttcgtttt cagagcaaga gattacgcgc agaccaaaac gatctcaaga

5101 agatcatctt attaaggggt ctgacgctca gtggaacgaa aactcacgtt aagggatttt

5161 ggtcatgaga ttatcaaaaa ggatcttcac ctagatcctt ttaaattaaa aatgaagttt

5221 taaatcaatc taaagtatat atgagtaaac ttggtctgac agttaccaat gcttaatcag

5281 tgaggcacct atctcagcga tctgtctatt tcgttcatcc atagttgcct gactccccgt

5341 cgtgtagata actacgatac gggagggctt accatctggc cccagtgctg caatgatacc

5401 gcgagaccca cgctcaccgg ctccagattt atcagcaata aaccagccag ccggaagggc

5461 cgagcgcaga agtggtcctg caactttatc cgcctccatc cagtctattc catggtgcca

5521 cctgacgtct aagaaaccat tattatcatg acattaacct ataaaaatag gcgtatcacg

5581 aggcagaatt tcagataaaa aaaatcctta gctttcgcta aggatgattt ctggaattcg

5641 cggccgcttc tagag

Bibliography

1. World Health, O., *Medication without harm*. 2017, World Health Organization.

2. Healy, M.A., et al., *Hospital and Payer Costs Associated With Surgical Complications.* JAMA Surgery, 2016. **151**(9): p. 823-830.

3. Blaser, M., *Stop the killing of beneficial bacteria.* Nature, 2011. **476**(7361): p. 393-394.

4. Langer, R., *Drug delivery and targeting.* Nature, 1998. **392**(6679 Suppl): p. 5-10.

5. Chabner, B.A. and T.G. Roberts, *Chemotherapy and the war on cancer.* Nature Reviews Cancer, 2005. **5**(1): p. 65-72.

6. Schirrmacher, V., *From chemotherapy to biological therapy: A review of novel concepts to reduce the side effects of systemic cancer treatment (Review).* Int J Oncol, 2019. **54**(2): p. 407-419.

7. Robinson, T.N. and G.V. Stiegmann, *Minimally invasive surgery.* Endoscopy, 2004. **36**(01): p. 48-51.

8. Palep, J.H., *Robotic assisted minimally invasive surgery.* Journal of minimal access surgery, 2009. **5**(1): p. 1.

9. Gooch, C.L., E. Pracht, and A.R. Borenstein, *The burden of neurological disease in the United States: A summary report and call to action.* Annals of neurology, 2017. **81**(4): p. 479-484.

10. Pardridge, W.M., *Blood–brain barrier delivery.* Drug discovery today, 2007. **12**(1-2): p. 54-61.

11. Nelson, B.J., I.K. Kaliakatsos, and J.J. Abbott, *Microrobots for Minimally Invasive Medicine.* Annual Review of Biomedical Engineering, 2010. **12**(1): p. 55-85.

12. Ceylan, H., et al., *Translational prospects of untethered medical microrobots.* Progress in Biomedical Engineering, 2019. **1**(1): p. 012002.

13. Sitti, M., et al., *Biomedical Applications of Untethered Mobile Milli/Microrobots.* Proceedings of the IEEE, 2015. **103**(2): p. 205-224.

14. Chertok, B., et al., *Iron oxide nanoparticles as a drug delivery vehicle for MRI monitored magnetic targeting of brain tumors.* Biomaterials, 2008. **29**(4): p. 487-496.

15. Garbayo, E., E. Ansorena, and M.J. Blanco-Prieto, *Drug development in Parkinson's disease: From emerging molecules to innovative drug delivery systems.* Maturitas, 2013. **76**(3): p. 272-278.

16. Breger, J.C., et al., *Self-Folding Thermo-Magnetically Responsive Soft Microgrippers.* ACS Applied Materials & Interfaces, 2015. **7**(5): p. 3398-3405.

17. Jeon, S., et al., *Magnetically actuated microrobots as a platform for stem cell transplantation.* Science Robotics, 2019. **4**(30): p. eaav4317.

18. Xu, T., et al., *Magnetic Actuation Based Motion Control for Microrobots: An Overview.* Micromachines, 2015. **6**: p. 1346-1364.

19. Carlsen, R.W., et al., *Magnetic steering control of multi-cellular bio-hybrid microswimmers.* Lab Chip, 2014. **14**(19): p. 3850-9.

20. Li, H., et al., *Magnetic actuated pH-responsive hydrogel-based soft micro-robot for targeted drug delivery.* Smart Materials and Structures, 2016. **25**(2): p. 027001.

21. Mahoney, A.W. and J.J. Abbott, *Generating Rotating Magnetic Fields With a Single Permanent Magnet for Propulsion of Untethered Magnetic Devices in a Lumen.* IEEE Transactions on Robotics, 2014. **30**(2): p. 411-420.

22. Abbott, J.J., *Parametric design of tri-axial nested Helmholtz coils.* Review of Scientific Instruments, 2015. **86**(5): p. 054701.

23. Marino, H., C. Bergeles, and B.J. Nelson, *Robust Electromagnetic Control of Microrobots Under Force and Localization Uncertainties.* IEEE Transactions on Automation Science and Engineering, 2014. **11**(1): p. 310-316.

24. Lee, S., et al., *Microrobots: A Needle-Type Microrobot for Targeted Drug Delivery by Affixing to a Microtissue (Adv. Healthcare Mater. 7/2020).* Advanced Healthcare Materials, 2020. **9**(7): p. 2070019.

25. Xie, H., et al., *Reconfigurable magnetic microrobot swarm: Multimode transformation, locomotion, and manipulation.* Science Robotics, 2019. **4**(28): p. eaav8006.

26. Alapan, Y., et al., *Multifunctional surface microrollers for targeted cargo delivery in physiological blood flow.* Science Robotics, 2020. **5**(42): p. eaba5726.

27. Yesin, K.B., K. Vollmers, and B.J. Nelson, *Modeling and Control of Untethered Biomicrorobots in a Fluidic Environment Using Electromagnetic Fields.* The International Journal of Robotics Research, 2006. **25**(5-6): p. 527-536.

28. Yan, X., et al., *Multifunctional biohybrid magnetite microrobots for imaging-guided therapy.* Science Robotics, 2017. **2**(12): p. eaaq1155.

29. Dong, M., et al., *3D-Printed Soft Magnetoelectric Microswimmers for Delivery and Differentiation of Neuron-Like Cells.* Advanced Functional Materials, 2020. **30**(17): p. 1910323.

30. Medina-Sánchez, M., et al., *Cellular Cargo Delivery: Toward Assisted Fertilization by Sperm-Carrying Micromotors.* Nano Letters, 2016. **16**(1): p. 555-561.

31. Jeon, S., et al., *A Magnetically Controlled Soft Microrobot Steering a Guidewire in a Three-Dimensional Phantom Vascular Network.* Soft Robotics, 2018. **6**(1): p. 54-68.

32. Hwang, G., et al., *Catalytic antimicrobial robots for biofilm eradication.* Science Robotics, 2019. **4**(29): p. eaaw2388.

33. Belharet, K., D. Folio, and A. Ferreira. *Control of a magnetic microrobot navigating in microfluidic arterial bifurcations through pulsatile and viscous flow*. in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012.

34. Sitti, M., *Voyage of the microrobots.* Nature, 2009. **458**(7242): p. 1121-1122.

35. Shirai, Y., et al., *Directional Control in Thermally Driven Single-Molecule Nanocars.* Nano Letters, 2005. **5**(11): p. 2330-2334.

36. Liao, T., et al. *Data-efficient Learning of Morphology and Controller for a Microrobot*. in *2019 International Conference on Robotics and Automation (ICRA)*. 2019.

37. Soto, F., et al., *Smart Materials for Microrobots.* Chemical Reviews, 2021.

38. Kummer, M.P., et al., *OctoMag: An Electromagnetic System for 5-DOF Wireless Micromanipulation.* IEEE Transactions on Robotics, 2010. **26**(6): p. 1006-1017.

39. Hu, W., et al., *Small-scale soft-bodied robot with multimodal locomotion.* Nature, 2018. **554**(7690): p. 81-85.

40. Ma, Q., et al., *Construction and operation of a microrobot based on magnetotactic bacteria in a microfluidic chip.* Biomicrofluidics, 2012. **6**(2): p. 024107.

41. Xu, H., et al., *Sperm Micromotors for Cargo Delivery through Flowing Blood.* ACS Nano, 2020. **14**(3): p. 2982-2993.

42. Ricotti, L., et al., *Biohybrid actuators for robotics: A review of devices actuated by living cells.* Science Robotics, 2017. **2**(12): p. eaaq0495.

43. Behrens, M., Ruder, WC, *Synthetic biology and cell engineering for bio-enabled nano/microrobots.* Springer Encyclopedia of Robotics, 2021.

44. Villa, K., et al., *Cooperative Multifunctional Self-Propelled Paramagnetic Microrobots with Chemical Handles for Cell Manipulation and Drug Delivery.* Advanced Functional Materials, 2018. **28**(43): p. 1804343.

45. Schmidt, C.K., et al., *Engineering microrobots for targeted cancer therapies from a medical perspective.* Nature Communications, 2020. **11**(1): p. 5618.

46. Tang, S., et al., *Enzyme-powered Janus platelet cell robots for active and targeted drug delivery.* Science Robotics, 2020. **5**(43): p. eaba6137.

47. Eskandarloo, H., A. Kierulf, and A. Abbaspourrad, *Nano-and micromotors for cleaning polluted waters: focused review on pollutant removal mechanisms.* Nanoscale, 2017. **9**(37): p. 13850-13863.

48. Gao, W. and J. Wang, *The Environmental Impact of Micro/Nanomachines: A Review.* ACS Nano, 2014. **8**(4): p. 3170-3180.

49. Soler, L., et al., *Self-Propelled Micromotors for Cleaning Polluted Water.* ACS Nano, 2013. **7**(11): p. 9611-9620.

50. Xu, L., et al., *Light-driven micro/nanomotors: from fundamentals to applications.* Chemical Society Reviews, 2017. **46**(22): p. 6905-6926.

51. Kim, H. and M.J. Kim, *Electric Field Control of Bacteria-Powered Microrobots Using a Static Obstacle Avoidance Algorithm.* IEEE Transactions on Robotics, 2016. **32**(1): p. 125-137.

52. Leal-Estrada, M., et al., *Engineering Ultrasound Fields to Power Medical Micro/Nanorobots.* Current Robotics Reports, 2021. **2**(1): p. 21-32.

53. Park, S.-J., et al., *Phototactic guidance of a tissue-engineered soft-robotic ray.* Science, 2016. **353**(6295): p. 158.

54. Li, J., et al., *Development of a magnetic microrobot for carrying and delivering targeted cells.* Science Robotics, 2018. **3**(19).

55. Fusco, S., et al., *An Integrated Microrobotic Platform for On-Demand, Targeted Therapeutic Interventions.* Advanced Materials, 2013. **26**(6): p. 952-957.

56. Schenck, J.F., *Safety of strong, static magnetic fields.* J Magn Reson Imaging, 2000. **12**(1): p. 2-19.

57. Martel, S., et al., *Flagellated Magnetotactic Bacteria as Controlled MRI-trackable Propulsion and Steering Systems for Medical Nanorobots Operating in the Human Microvasculature.* The International journal of robotics research, 2009. **28**(4): p. 571-582.

58. Wang, B., et al., *Trends in Micro-/Nanorobotics: Materials Development, Actuation, Localization, and System Integration for Biomedical Applications.* Advanced Materials, 2020. **n/a**(n/a): p. 2002047.

59. Scheggi, S., et al. *Magnetic motion control and planning of untethered soft grippers using ultrasound image feedback*. in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017.

60. Zhang, J., et al., *Reliable Grasping of Three-Dimensional Untethered Mobile Magnetic Microgripper for Autonomous Pick-and-Place.* IEEE Robotics and Automation Letters, 2017. **2**(2): p. 835-840.

61. Alapan, Y., et al., *Soft erythrocyte-based bacterial microswimmers for cargo delivery.* Science Robotics, 2018. **3**(17): p. eaar4423.

62. Yasa, I.C., et al., *Elucidating the interaction dynamics between microswimmer body and immune system for medical microrobots.* Science Robotics, 2020. **5**(43).

63. Wang, X., et al., *3D Printed Enzymatically Biodegradable Soft Helical Microswimmers.* Advanced Functional Materials, 2018. **28**(45): p. 1804107.

64. Zhang, J., et al., *Voxelated three-dimensional miniature magnetic soft machines via multimaterial heterogeneous assembly.* Science Robotics, 2021. **6**(53): p. eabf0112.

65. Medina-Sánchez, M., et al., *Swimming Microrobots: Soft, Reconfigurable, and Smart.* Advanced Functional Materials, 2018. **28**(25): p. 1707228.

66. Xu, T., et al., *Millimeter-scale flexible robots with programmable three-dimensional magnetization and motions.* Science Robotics, 2019. **4**(29): p. eaav4494.

67. Zhang, H., et al., *Dual-responsive biohybrid neutrobots for active target delivery.* Science Robotics, 2021. **6**(52): p. eaaz9519.

68. Wei, T.-Y. and W.C. Ruder, *Engineering control circuits for molecular robots using synthetic biology.* APL Materials, 2020. **8**(10): p. 101104.

69. Aziz, A., et al., *Medical Imaging of Microrobots: Toward In Vivo Applications.* ACS Nano, 2020. **14**(9): p. 10865-10893.

70. Medina-Sánchez, M. and O.G. Schmidt, *Medical microbots need better imaging and control.* Nature News, 2017. **545**(7655): p. 406.

71. Tsang, A.C.H., et al., *Roads to Smart Artificial Microswimmers.* Advanced Intelligent Systems, 2020. **2**(8): p. 1900137.

72. Wang, X., et al., *Dynamic Modeling of Magnetic Helical Microrobots.* IEEE Robotics and Automation Letters, 2021: p. 1-1.

73. Charreyron, S.L., et al., *Modeling Electromagnetic Navigation Systems.* IEEE Transactions on Robotics, 2021: p. 1-13.

74. Denasi, A. and S. Misra. *A robust controller for micro-sized agents: The prescribed performance approach*. in *2016 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*. 2016.

75. Pawashe, C., et al., *Two-Dimensional Autonomous Microparticle Manipulation Strategies for Magnetic Microrobots in Fluidic Environments.* IEEE Transactions on Robotics, 2012. **28**(2): p. 467-477.

76. Huang, H.-W., et al., *Soft micromachines with programmable motility and morphology.* Nature Communications, 2016. **7**(1): p. 12263.

77. Haarnoja, T., et al., *Learning to walk via deep reinforcement learning.* arXiv preprint arXiv:1812.11103, 2018.

78. Muiños-Landin, S., et al., *Reinforcement learning with artificial microswimmers.* Science Robotics, 2021. **6**(52): p. eabd9285.

79. Turan, M., et al., *Learning to Navigate Endoscopic Capsule Robots.* IEEE Robotics and Automation Letters, 2019. **4**(3): p. 3075-3082.

80. Kober, J., J.A. Bagnell, and J. Peters, *Reinforcement learning in robotics: A survey.* The International Journal of Robotics Research, 2013. **32**(11): p. 1238-1274.

81. Stock, J.B., A.J. Ninfa, and A.M. Stock, *Protein phosphorylation and regulation of adaptive responses in bacteria.* Microbiological reviews, 1989. **53**(4): p. 450-490.

82. Tagkopoulos, I., Y.-C. Liu, and S. Tavazoie, *Predictive Behavior Within Microbial Genetic Networks.* Science, 2008. **320**(5881): p. 1313-1317.

83. Ardiel, E.L. and C.H. Rankin, *An elegant mind: learning and memory in Caenorhabditis elegans.* Learning & memory, 2010. **17**(4): p. 191-201.

84. Giurfa, M., *Learning and cognition in insects.* WIREs Cognitive Science, 2015. **6**(4): p. 383-395.

85. Calvo Garzón, P. and F. Keijzer, *Plants: Adaptive behavior, root-brains, and minimal cognition.* Adaptive Behavior, 2011. **19**(3): p. 155-171.

86. Bonilla, F.A. and H.C. Oettgen, *Adaptive immunity.* Journal of Allergy and Clinical Immunology, 2010. **125**(2, Supplement 2): p. S33-S40.

87. Dickinson, A., *Contemporary animal learning theory*. Vol. 1. 1980: CUP Archive.

88. Minsky, M., *Steps toward Artificial Intelligence.* Proceedings of the IRE, 1961. **49**(1): p. 8-30.

89. Gardner, T.S., C.R. Cantor, and J.J. Collins, *Construction of a genetic toggle switch in Escherichia coli.* Nature, 2000. **403**(6767): p. 339-342.

90. Zhang, H., et al., *Programming a Pavlovian-like conditioning circuit in Escherichia coli.* Nature Communications, 2014. **5**(1): p. 3102.

91. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning.* Nature, 2015. **521**(7553): p. 436-444.

92. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *Imagenet classification with deep convolutional neural networks.* Advances in neural information processing systems, 2012. **25**: p. 1097-1105.

93. Brown, T.B., et al., *Language models are few-shot learners.* arXiv preprint arXiv:2005.14165, 2020.

94. Jacques Bughin, J.S., James Manyika, Michael Chui, Raoul Joshi *Notes from the AI Frontier: Modeling the Impact of AI on the World Economy*, in *Mckinsey Global Institute*. 2018, Mckinsey&Company.

95. Zhu, H., et al., *The Ingredients of Real-World Robotic Reinforcement Learning.* arXiv preprint arXiv:2004.12570, 2020.

96. Silver, D., et al., *Mastering the game of Go without human knowledge.*

97. Colabrese, S., et al., *Flow navigation by smart microswimmers via reinforcement learning.* Physical review letters, 2017. **118**(15): p. 158004.

98. Yang, Y., M.A. Bevan, and B. Li, *Efficient Navigation of Colloidal Robots in an Unknown Environment via Deep Reinforcement Learning.* Advanced Intelligent Systems, 2020. **2**(1): p. 1900106.

99. Haarnoja, T., et al. *Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor*. 2018. PMLR.

100. Abbott, J.J., et al., *How Should Microrobots Swim?* The International Journal of Robotics Research, 2009. **28**(11-12): p. 1434-1447.

101. Hunter, E.E., et al., *Toward Soft Micro Bio Robots for Cellular and Chemical Delivery.* IEEE Robotics and Automation Letters, 2018. **3**(3): p. 1592-1599.

102. Sutton, R.S. and A.G. Barto, *Introduction to reinforcement learning*. Vol. 135. 1998: MIT press Cambridge.

103. Ibarz, J., et al., *How to train your robot with deep reinforcement learning: lessons we have learned.* The International Journal of Robotics Research, 2021. **40**(4-5): p. 698-721.

104. Haarnoja, T., et al., *Soft actor-critic algorithms and applications.* arXiv preprint arXiv:1812.05905, 2018.

105. Hu, C., S. Pané, and B.J. Nelson, *Soft Micro- and Nanorobotics.* Annual Review of Control, Robotics, and Autonomous Systems, 2018. **1**(1): p. 53-75.

106. He, J., et al., *Hookworm Detection in Wireless Capsule Endoscopy Images With Deep Learning.* IEEE Transactions on Image Processing, 2018. **27**(5): p. 2379-2392.

107. Mnih, V., et al., *Human-level control through deep reinforcement learning.* Nature, 2015. **518**(7540): p. 529-533.

108. Fu, J., et al. *Diagnosing bottlenecks in deep q-learning algorithms*. PMLR.

109. Caruana, R., S. Lawrence, and L. Giles, *Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping.* Advances in neural information processing systems, 2001: p. 402-408.

110. Ongaro, F., et al., *Design of an Electromagnetic Setup for Independent Three-Dimensional Control of Pairs of Identical and Nonidentical Microrobots.* IEEE Transactions on Robotics, 2019. **35**(1): p. 174-183.

111. Mahoney, A.W. and J.J. Abbott, *Five-degree-of-freedom manipulation of an untethered magnetic device in fluid using a single permanent magnet with application in stomach capsule endoscopy.* The International Journal of Robotics Research, 2015. **35**(1-3): p. 129-147.

112. Arulkumaran, K., et al., *Deep Reinforcement Learning: A Brief Survey.* IEEE Signal Processing Magazine, 2017. **34**(6): p. 26-38.

113. Fuller, H.C., et al., *The Future Application of Organ-on-a-Chip Technologies as Proving Grounds for MicroBioRobots.* Micromachines, 2020. **11**(10): p. 947.

114. Behrens, M.R., et al., *Open-source, 3D-printed Peristaltic Pumps for Small Volume Point-of-Care Liquid Handling.* Scientific Reports, 2020. **10**(1): p. 1543.

115. Lake, J.R., K.C. Heyde, and W.C. Ruder, *Low-cost feedback-controlled syringe pressure pumps for microfluidics applications.* PLoS ONE, 2017. **12**(4): p. e0175089.

116. Kong, D.S., et al., *Open-source, community-driven microfluidics with Metafluidics.* Nature Biotechnology, 2017. **35**: p. 523.

117. Bhamla, M.S., et al., *Hand-powered ultralow-cost paper centrifuge.* Nature Biomedical Engineering, 2017. **1**(1): p. 0009.

118. Jung, W., et al., *Point-of-care testing (POCT) diagnostic systems using microfluidic lab-on-a-chip technologies.* Microelectronic Engineering, 2015. **132**: p. 46-57.

119. Laksanasopin, T., et al., *A smartphone dongle for diagnosis of infectious diseases at the point of care.* Science Translational Medicine, 2015. **7**(273): p. 273re1.

120. Millet, L.J., et al., *Modular microfluidics for point-of-care protein purifications.* Lab on a Chip, 2015. **15**(8): p. 1799-1811.

121. Ulep, T.-H., et al., *Interfacial Effect-Based Quantification of Droplet Isothermal Nucleic Acid Amplification for Bacterial Infection.* Scientific Reports, 2019. **9**(1): p. 9629.

122. Lewandrowski, K., *Point-of-care testing: an overview and a look to the future (circa 2009, United States).* Clin Lab Med, 2009. **29**(3): p. 421-32.

123. Chin, C.D., V. Linder, and S.K. Sia, *Commercialization of microfluidic point-of-care diagnostic devices.* Lab on a Chip, 2012. **12**(12): p. 2118-2134.

124. Devaraju, N.S.G.K. and M.A. Unger, *Pressure driven digital logic in PDMS based microfluidic devices fabricated by multilayer soft lithography.* Lab on a Chip, 2012. **12**(22): p. 4809-4815.

125. Grover, W.H., et al., *Monolithic membrane valves and diaphragm pumps for practical large-scale integration into glass microfluidic devices.* Sensors and Actuators B: Chemical, 2003. **89**(3): p. 315-323.

126. Bhattacharjee, N., et al., *The upcoming 3D-printing revolution in microfluidics.* Lab on a Chip, 2016. **16**(10): p. 1720-1742.

127. Takahashi, M.K., et al., *A low-cost paper-based synthetic biology platform for analyzing gut microbiota and host biomarkers.* Nature Communications, 2018. **9**(1): p. 3347.

128. Yetisen, A.K., M.S. Akram, and C.R. Lowe, *Paper-based microfluidic point-of-care diagnostic devices.* Lab on a Chip, 2013. **13**(12): p. 2210-2251.

129. Li, X., D.R. Ballerini, and W. Shen, *A perspective on paper-based microfluidics: Current status and future trends.* Biomicrofluidics, 2012. **6**(1): p. 011301.

130. Skafte-Pedersen, P., et al., *Multi-channel peristaltic pump for microfluidic applications featuring monolithic PDMS inlay.* Lab on a Chip, 2009. **9**(20): p. 3003-3006.

131. Guevara-Pantoja, P.E., et al., *Pressure-actuated monolithic acrylic microfluidic valves and pumps.* Lab on a Chip, 2018. **18**(4): p. 662-669.

132. Han, J.Y., et al., *Screw-actuated displacement micropumps for thermoplastic microfluidics.* Lab on a Chip, 2016. **16**(20): p. 3940-3946.

133. Tang, S.-Y., et al., *Versatile Microfluidic Platforms Enabled by Novel Magnetorheological Elastomer Microactuators.* Advanced Functional Materials, 2018. **28**(8): p. 1705484.

134. Liu, J.F., et al., *Microfluidic diafiltration-on-chip using an integrated magnetic peristaltic micropump.* Lab on a Chip, 2017. **17**(22): p. 3796-3803.

135. Thurgood, P., et al., *Self-sufficient, low-cost microfluidic pumps utilising reinforced balloons.* Lab on a Chip, 2019. **19**(17): p. 2885-2896.

136. Mohammed, M., et al., *Studying the Response of Aortic Endothelial Cells under Pulsatile Flow Using a Compact Microfluidic System.* Analytical Chemistry, 2019. **91**(18): p. 12077-12084.

137. Warkiani, M.E., et al., *Ultra-fast, label-free isolation of circulating tumor cells from blood using spiral microfluidics.* Nature Protocols, 2016. **11**(1): p. 134-148.

138. Esser, F., T. Masselter, and T. Speck, *Silent Pumpers: A Comparative Topical Overview of the Peristaltic Pumping Principle in Living Nature, Engineering, and Biomimetics.* Advanced Intelligent Systems, 2019. **1**(2): p. 1900009.

139. Bouchard, A.P., D.A. Duford, and E.D. Salin, *Non-contact Addition, Metering, and Distribution of Liquids into Centrifugal Microfluidic Devices in Motion.* Analytical Chemistry, 2010. **82**(20): p. 8386-8389.

140. Wang, J., J. Heo, and S.Z. Hua, *Spatially resolved shear distribution in microfluidic chip for studying force transduction mechanisms in cells.* Lab on a Chip, 2010. **10**(2): p. 235-239.

141. Ota, H., et al., *Three-dimensional spheroid-forming lab-on-a-chip using micro-rotational flow.* Sensors and Actuators B: Chemical, 2010. **147**(1): p. 359-365.

142. Sung, J.H. and M.L. Shuler, *A micro cell culture analog (µCCA) with 3-D hydrogel culture of multiple cell lines to assess metabolism-dependent cytotoxicity of anti-cancer drugs.* Lab on a Chip, 2009. **9**(10): p. 1385-1394.

143. Unger, M.A., et al., *Monolithic microfabricated valves and pumps by multilayer soft lithography.* Science, 2000. **288**(5463): p. 113-6.

144. Jandik, P., et al., *Initial study of using a laminar fluid diffusion interface for sample preparation in high-performance liquid chromatography*. Journal of Chromatography A, 2002. **954**(1): p. 33-40.

145. Takayama, S., et al., *Subcellular positioning of small molecules.* Nature, 2001. **411**(6841): p. 1016-1016.

146. Kenis, P.J.A., R.F. Ismagilov, and G.M. Whitesides, *Microfabrication Inside Capillaries Using Multiphase Laminar Flow Patterning.* Science, 1999. **285**(5424): p. 83.

147. Oh, K.W., et al., *Design of pressure-driven microfluidic networks using electric circuit analogy*. Lab Chip, 2012. **12**(3): p. 515-45.

148. Steward, R., Jr., et al., *Fluid shear, intercellular stress, and endothelial cell alignment.* Am J Physiol Cell Physiol, 2015. **308**(8): p. C657-64.

149. Aminov, R., *A Brief History of the Antibiotic Era: Lessons Learned and Challenges for the Future.* Frontiers in Microbiology, 2010. **1**: p. 134.

150. Ventola, C.L., *The Antibiotic Resistance Crisis: Part 1: Causes and Threats.* Pharmacy and Therapeutics, 2015. **40**(4): p. 277-283.

151. Blaser, M.J., *Antibiotic use and its consequences for the normal microbiome.* Science, 2016. **352**(6285): p. 544.

152. Drucker, C.R., *Update on topical antibiotics in dermatology.* Dermatologic Therapy, 2012. **25**(1): p. 6-11.

153. Stebbins, N.D., M.A. Ouimet, and K.E. Uhrich, *Antibiotic-containing polymers for localized, sustained drug delivery.* Advanced Drug Delivery Reviews, 2014. **78**: p. 77-87.

154. Jeon, S.M., et al., *Magnetic navigation system for the precise helical and translational motions of a microrobot in human blood vessels.* Journal of Applied Physics, 2012. **111**(7): p. 07E702.

155. Peppas, N.A., et al., *Hydrogels in Biology and Medicine: From Molecular Principles to Bionanotechnology.* Advanced Materials, 2006. **18**(11): p. 1345-1360.

156. Naahidi, S., et al., *Biocompatibility of hydrogel-based scaffolds for tissue engineering applications.* Biotechnology Advances, 2017. **35**(5): p. 530-544.

157. Levskaya, A., et al., *Engineering Escherichia coli to see light.* Nature, 2005. **438**(7067): p. 441-442.

158. Tabor, J.J., et al., *A Synthetic Genetic Edge Detection Program.* Cell, 2009. **137**(7): p. 1272-1281.

159. Deisseroth, K., *Optogenetics.* Nature methods, 2011. **8**(1): p. 26.

160. Prindle, A., et al., *A sensing array of radically coupled genetic 'biopixels'.* Nature, 2011. **481**: p. 39.

161. Boorsma, M., et al., *A temperature-regulated replicon-based DNA expression system.* Nature Biotechnology, 2000. **18**(4): p. 429-432.

162. Lutz, R. and H. Bujard, *Independent and tight regulation of transcriptional units in Escherichia coli via the LacR/O, the TetR/O and AraC/I1-I2 regulatory elements.* Nucleic Acids Res, 1997. **25**(6): p. 1203-10.

163. Gardner, T.S., C.R. Cantor, and J.J. Collins, *Construction of a genetic toggle switch in Escherichia coli.* Nature, 2000. **403**: p. 339.

164. Weber, W., et al., *Macrolide-based transgene control in mammalian cells and mice.* Nat Biotechnol, 2002. **20**(9): p. 901-7.

165. Xie, Z., et al., *Multi-Input RNAi-Based Logic Circuit for Identification of Specific Cancer Cells.* Science, 2011. **333**(6047): p. 1307.

166. Roybal, Kole T., et al., *Precision Tumor Recognition by T Cells With Combinatorial Antigen-Sensing Circuits.* Cell, 2016. **164**(4): p. 770-779.

167. Ye, H., et al., *Self-adjusting synthetic gene circuit for correcting insulin resistance.* Nature Biomedical Engineering, 2016. **1**: p. 0005.

168. Kojima, R., et al., *Designer exosomes produced by implanted cells intracerebrally deliver therapeutic cargo for Parkinson's disease treatment.* Nature Communications, 2018. **9**(1): p. 1305.

169. Zhang, R., et al., *Programming Surface Chemistry with Engineered Cells.* ACS Synthetic Biology, 2016. **5**(9): p. 936-941.

170. Weber, W., et al., *A synthetic mammalian electro-genetic transcription circuit.* Nucleic Acids Research, 2009. **37**(4): p. e33-e33.

171. Chen, A.Y., et al., *Synthesis and patterning of tunable multiscale materials with engineered cells.* Nature Materials, 2014. **13**: p. 515.

172. Wu, C.-Y., et al., *Remote control of therapeutic T cells through a small molecule–gated chimeric receptor.* Science, 2015. **350**(6258).

173. Justus, K.B., et al., *A biosensing soft robot: Autonomous parsing of chemical signals through integrated organic and inorganic interfaces.* Science Robotics, 2019. **4**(31): p. eaax0765.

174. Scott, F.Y., et al., *Engineering a living biomaterial via bacterial surface capture of environmental molecules.* Synthetic Biology, 2018. **3**(1).

175. Schleif, R., *Regulation of the l-arabinose operon of Escherichia coli.* Trends in Genetics, 2000. **16**(12): p. 559-565.

176. Earhart, C.F., *[30] Use of an Lpp-OmpA fusion vehicle for bacterial surface display*, in *Methods in Enzymology*. 2000, Academic Press. p. 506-516.

177. Glass, D.S. and I.H. Riedel-Kruse, *A synthetic bacterial cell-cell adhesion toolbox for programming multicellular morphologies and patterns.* Cell, 2018. **174**(3): p. 649-658.

178. Litcofsky, K.D., et al., *Iterative plug-and-play methodology for constructing and modifying synthetic gene networks.* Nat Methods, 2012. **9**(11): p. 1077-80.

179. Daily, M., et al., *Self-Driving Cars.* Computer, 2017. **50**(12): p. 18-23.

180. Lázaro-Gredilla, M., et al., *Beyond imitation: Zero-shot task transfer on robots by learning concepts as cognitive programs.* Science Robotics, 2019. **4**(26): p. eaav3150.

181. He, B., S. Wang, and Y. Liu, *Underactuated robotics: A review.* International Journal of Advanced Robotic Systems, 2019. **16**(4): p. 1729881419862164.