

Contents

1	Introduction	3
2	Related Work	6
2.1	Global control of microrobots	6
2.2	Human-swarm interaction	7
2.3	Block pushing and compliant manipulation	8
3	Online experiment	8
3.1	Methods	8
3.1.1	Human subjects	9
3.1.2	Instrumentation	10
3.2	Human-swarm interaction results	10
4	Theory	14
4.1	Independent control of many robots is impossible	15
4.2	Controlling mean position	16
4.3	Controlling the variance of many robots	17
4.4	Controlling both mean and variance of many robots	18
5	Simulation of control laws	20
5.1	Controlling the mean position	20
5.2	Controlling the variance	20
5.3	Hybrid control of mean and variance	20
6	Object manipulation results	22
6.1	Human-controlled object manipulation	22
6.2	Automated object manipulation (simulation)	22
7	Object manipulation with hardware robots	26
7.1	Environmental setup	26
7.2	Automated object manipulation (hardware experiment)	27

Steering a Swarm Using Global Inputs And Swarm Statistics

Shiva Shahrokhi, Chris Ertel*, Mable Wan, Lillian Lin, Aaron T. Becker

August 4, 2016

Abstract

Micro- and nanorobotics have the potential to revolutionize many applications including targeted material delivery, assembly, and surgery. The same properties that promise breakthrough solutions—small size and large populations—present unique challenges to generating controlled motion. We want to use large swarms of robots to perform manipulation tasks; unfortunately, human-swarm interaction studies as conducted today are limited in sample size, are difficult to reproduce, and are prone to hardware failures. We present an alternative.

This paper first examines the perils, pitfalls, and possibilities we discovered by launching SwarmControl.net, an online game where players steer swarms of up to 500 robots to complete manipulation challenges. We record statistics from thousands of players, and use the game to explore aspects of large-population robot control. We present the game framework as a new, open-source tool for large-scale user experiments. One surprising result was that humans completed an object manipulation task *faster* when provided with only the mean and variance of the robot swarm than with full-state feedback. Inspired by human operators, this paper next investigates controllers that use only the mean and variance of a robot swarm. We prove that the mean position is controllable, then provide conditions under which variance is controllable. We next derive automatic controllers for these and a hybrid, hysteresis-based switching control to regulate the first two moments of the robot distribution. Finally, we employ these controllers as primitives for an object manipulation task and implement all the automatic controllers on 100 kilobots controlled by the direction of a global light source.

*S. Shahrokhi and A. Becker are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, 77004 USA
sshahrokhi2@uh.edu, atbecker@uh.edu

1 Introduction

Large populations of micro- and nanorobots are being produced in laboratories around the world, with diverse potential applications in drug delivery and construction, see Peyer et al. (2013); Shirai et al. (2005); Chiang et al. (2011). These activities require robots that behave intelligently. Limited computation and communication rules out autonomous operation or direct control over individual units; instead we must rely on global control signals broadcast to the entire robot population. It is not always practical to gather pose information on individual robots for feedback control; the robots might be difficult or impossible to sense individually due to their size and location. However, it is often possible to sense global properties of the group, such as mean position and density. Finally, many promising applications will require direct human control, but user interfaces to thousands—or millions—of robots is a daunting human-swarm interaction (HSI) challenge.

Our previous work with over a hundred hardware robots and thousands of simulated robots Becker et al. (2013) demonstrated that direct human control of large swarms is possible. Unfortunately, the logistical challenges of repeated experiments with over one hundred robots prevented large-scale tests. This paper presents a tool for investigating HSI methods through statistically significant numbers of experiments. There is currently no comprehensive understanding of user interfaces for controlling multi-robot systems with massive populations.

Our goal was to test several scenarios involving large-scale human-swarm interaction (HSI), and to do so with a statistically-significant sample size. Towards this end, we created SwarmControl.net, an open-source online testing platform suitable for inexpensive deployment and data collection on a scale not yet seen in swarm robotics research. Screenshots from this platform are shown in Fig. 1. All code and experimental results are posted online, Ertel and Becker (2013). TODO: link to 2016 code

Our experiments show that numerous simple robots responding to global control inputs are directly controllable by a human operator without special training, that the visual feedback of the swarm state should be simple to increase task performance, and that humans perform swarm-object manipulation faster using attractive control schemes than repulsive control schemes.

Often robots are difficult or impossible to sense individually due to their size and location. For example, microrobots are smaller than the minimum resolution of a clinical MRI-scanner, see Martel et al. (2014), however it is often possible to sense global properties of the group, such as mean position and variance. To make progress in automatic control with global

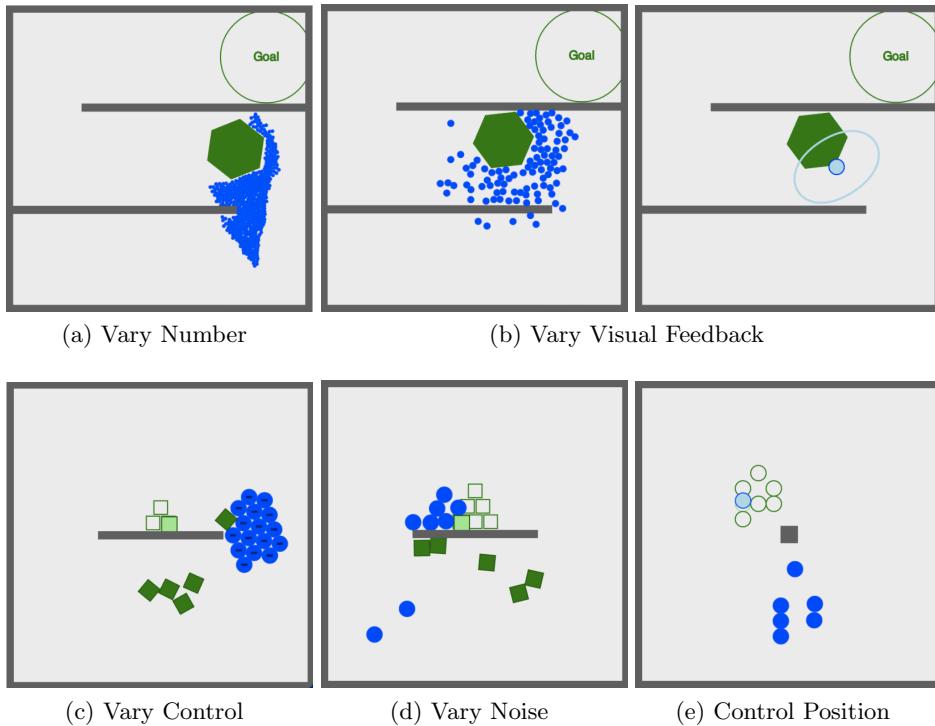


Figure 1: Screenshots from our five online experiments controlling multi-robot systems with limited, global control. **(a)** Varying the number of robots from 1-500 **(b)** Comparing 4 levels of visual feedback **(c)** Comparing 3 control architectures **(d)** Varying noise from 0 to 200% of control authority **(e)** Controlling the position of 1 to 10 robots. See video overview at Shahrokhi and Becker (2015d)

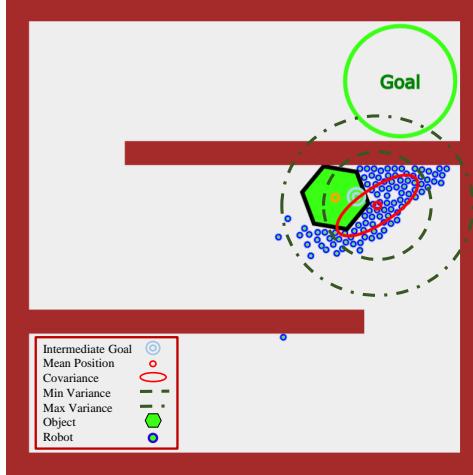


Figure 2: A swarm of robots, all controlled by a uniform force field, can be effectively controlled by a hybrid controller that knows only the first and second moments of the robot distribution. Here a swarm of simple robots (blue discs) pushes a green hexagon toward the goal. See video attachment Shahrokh and Becker (2015d).

inputs, this paper presents swarm manipulation controllers, inspired by our online experiments, that require only mean and variance measurements of the robot’s positions. These controllers are used as primitives to perform the object manipulation task illustrated in Fig. 2.

Our paper is organized as follows. After a discussion of related work in Section 2, we describe our experimental methods for an online human-user experiment in Section 3. We report the results of our online experiments in Section 3.2. Inspired by these results, we prove that the mean and variance of a robot swarm are controllable in Section 4, and present automatic controllers in Section 5. We conclude with implementations of these controllers on hardware robots and use them to complete an object manipulation task with 100+ kilobots in Section 7.

This paper is the synthesis of two preliminary conference papers, the first, Becker et al. (2014a), covering first few months of SwarmControl.net experiments, and the second, Shahrokh and Becker (2015c), with simulations of object manipulation. This paper presents the final results from SwarmControl.net. All hardware validation experiments are new.

2 Related Work

This section describes global control challenges and reviews highlights of human-swarm interaction, block pushing, and compliant manipulation.

2.1 Global control of microrobots

Small robots have been constructed with physical heterogeneity so that they respond differently to a global, broadcast control signal. Examples include *scratch-drive microrobots*, actuated and controlled by a DC voltage signal from a substrate by Donald et al. (2006, 2008); magnetic structures with different cross-sections that could be independently steered by Floyd et al. (2011); Diller et al. (2013); *MagMite* microrobots with different resonant frequencies and a global magnetic field by Frutiger et al. (2008); and magnetically controlled nanoscale helical screws constructed to stop movement at different cutoff frequencies of a global magnetic field by Tottori et al. (2012) and Peyer et al. (2013).

Similarly, our previous work, Becker and Bretl (2012); Becker et al. (2012), focused on exploiting inhomogeneity between robots. These control algorithms theoretically apply to any number of robots—even robotic continuums—but in practice process noise cancels the differentiating effects of inhomogeneity for more than tens of robots. We desire control algorithms that extend to many thousands of robots.

While it is now possible to create many microrobots, there remain challenges in control, sensing, and computation:

Control—global inputs: Many micro- and nanorobotic systems, see Tottori et al. (2012); Shirai et al. (2005); Chiang et al. (2011); Donald et al. (2006, 2008); Takahashi et al. (2006); Floyd et al. (2011); Diller et al. (2013); Frutiger et al. (2008); Peyer et al. (2013) rely on global inputs, where each robot receives an exact copy of the control signal. Our experiments follow this global model.

Sensing—large populations: Parallel control of n differential-drive robots in a 2D workspace requires $3n$ state variables. Even holonomic robots require $2n$ state variables. Numerous methods exist for measuring this state in microrobotics. These solutions use computer vision systems to sense position and heading angle, with corresponding challenges of handling missed detections and image registration between detections and robots. These

challenges are increased at the nanoscale where sensing competes with control for communication bandwidth. We examine control when the operator has access to partial feedback, including only the first and/or second moments of a population’s position, or only the convex-hull containing the robots.

Computation—calculating the control law: In our previous work the controllers required at best a summation over all the robot states, see Becker et al. (2012) and at worst a matrix inversion, see Becker and Bretl (2012). These operations become intractable for large populations of robots. By focusing on *human* control of large robot populations, we accentuate computational difficulties because the controllers are implemented by the unaided human operator.

2.2 Human-swarm interaction

Olsen Jr and Wood (2004) studied human *fanout*, the number of robots a single human user could control. They postulated that the optimal number of robots was approximately the autonomous time divided by the interaction time required by each robot. Their sample problem involved a multi-robot search task, where users could assign goals to robots. Their user interaction studies with simulated planar robots indicated a *fanout plateau* of about 8 robots, after which there were diminishing returns. They hypothesize that the location of this plateau is highly dependent on the underlying task. Indeed, our paper indicates there are some tasks without plateaus. Their research investigated robots with 3 levels of autonomy. We use robots without autonomy, corresponding with their first-level robots.

Squire et al. (2006) designed experiments showing that user-interface design had a high impact on the task effectiveness and the number of robots that could be controlled simultaneously in a multi-robot task.

A number of user studies compare methods for controlling large swarms of simulated robots, for example Bashyal and Venayagamoorthy (2008); Kolling et al. (2012); de la Croix and Egerstedt (2012). These studies provide insights but are limited by cost to small user studies; have a closed-source code base; and focus on controlling intelligent, programmable agents. For instance, the studies Bashyal and Venayagamoorthy (2008), de la Croix and Egerstedt (2012), and Kolling et al. (2012) were limited to a pool of 5, 18, and 32 participants. Using an online testing environment, we conduct similar studies but with sample sizes three orders of magnitude larger.

2.3 Block pushing and compliant manipulation

Unlike *caging* manipulation, where robots form a rigid arrangement around an object, as in Sudsang et al. (2002); Fink et al. (2007), our swarm of robots is unable to grasp the blocks they push, and so our manipulation strategies are similar to *nonprehensile manipulation* techniques, e.g. Lynch (1999), where forces must be applied along the center of mass of the moveable object. A key difference is that our robots are compliant and tend to flow around the object, making this similar to fluidic trapping as in Armani et al. (2006) and Becker et al. (2009).

Our n -robot system with 2 control inputs and $4n$ states is inherently under-actuated, and superficially bears resemblance to compliant, under-actuated manipulators, e.g. Odhner et al. (2014); Deimel and Brock (2014). Like these manipulators, our swarms conform to the object to be manipulated. However our swarms lack the restoring force provided by flexures in Odhner et al. (2014) and the silicone in Deimel and Brock (2014). Our swarms tend to disperse, so we require artificial forces, such as the variance control primitives in Section 4.3, to regroup the swarms.

3 Online experiment

We have developed a flexible testing framework for online human-swarm interaction studies. There are two halves to our framework: the server backend and the client-side (in-browser) frontend. The server backend is responsible for tabulating results, serving webpages containing the frontend code, and for issuing unique identifiers to each experiment participant. The in-browser frontend is responsible for running an experiment—that is to say, accepting user input, updating the state of the robot swarm, and ultimately evaluating task completion.

3.1 Methods

A participant visits the site, initiating a communication between their browser and our server. The web server generates a unique identifier for the participant and sends it along with the landing page to the participant—this identifier is stored as a browser cookie and will be sent along with all results the participant generates. The participant’s browser prompts for confirmation of the terms-of-service and offers a menu of experiments.

Once the participant selects an experiment, their browser makes a new request to the server to load the experiment’s webpage. The server sends

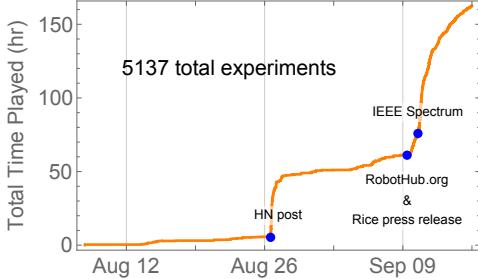


Figure 3: Cumulative time played for completed tests.

scaffold HTML describing the layout of the page and a script block containing the experiment. The script runs the experiment and, upon a successful completion, posts the experiment data to the server. The participant is then given the option of playing again or trying a different experiment.

A participant may view all of the experimental data we have gathered; this information is available as either a webpage, a JSON file, or a comma-separated value file.

Frontend During the **instantiation** phase, an experiment sets up the web page elements with help text and other information, and creates the obstacles, robots, and workpieces that will be present during the experiment. It will also randomly select which mode to run in, if applicable.

The **simulation** phase is the time at which all of the robots are moved according to user input and given a chance to interact with each other and the environment. The simulation phase then draws the current state of the experiment to the canvas of the webpage.

The **evaluation** phase is when the experiment’s completion criteria are applied to the current experiment state: are the robots in the goal zone, are the workpieces in the correct place, and so forth. If the criteria are not met, the experiment loops back into the simulation phase; if they are met, then the experiment proceeds to result submission.

The **submission** phase is when the results of the experiment are combined with other user data, such as the browser user agent string, and submitted to the server for collection.

3.1.1 Human subjects

Because our study involved recording data from human subjects, it required IRB approval before we could legally save user data (IRB #14357-01).

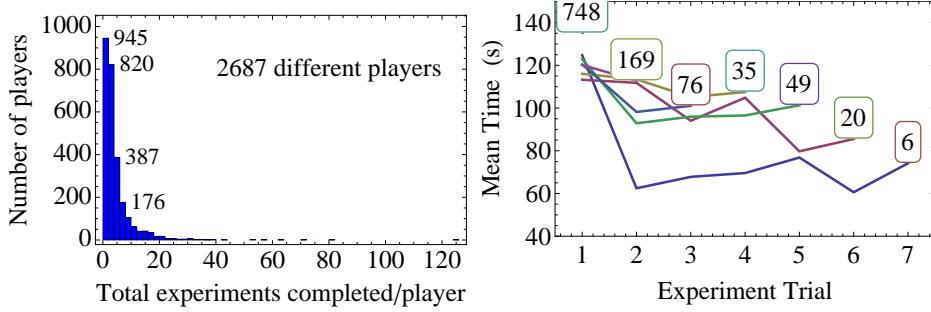


Figure 4: (Left) The total number of games played per player drops off exponentially. (Right) We are able to show that players skill improves as they retry tests using data from *Varying Number*

3.1.2 Instrumentation

When conducting an online experiment, it is helpful to gather data about both the experiment infrastructure and the participants. For the backend, we use a service called Airbrake to monitor the Rails server, getting emails in the events of any errors occurring or suspicious activity. We also use another service called New Relic to provide monitoring and analytics on the server traffic, giving coarse statistics about site visitation, page load time, and other indicators of how our backend is performing.

For the frontend, we use Google Analytics to track user behavior. This tool allows us to see country of origin for users, time spent on the site, relative percent of people who look past the landing page (bounce rate), and user agent information (type of browser, type of device, etc.).

3.2 Human-swarm interaction results

We designed five experiments to investigate human control of large robotic swarms for manipulation tasks. Screenshots of each experiment are shown in Fig. 1. Each experiment examined the effects of varying a single parameter: population of robots for manipulation, four levels of visual feedback, three control architectures, different levels of Brownian noise, and position control with 1 to 10 robots. The users could choose which experiment to try, and then our architecture randomly assigned a particular parameter value for each trial. We recorded the completion time and the participant ID for each successful trial. As Fig. 4 shows, one-third of all participants played only a single game. Still, many played multiple games, and their decreasing completion times demonstrates their skills improved.

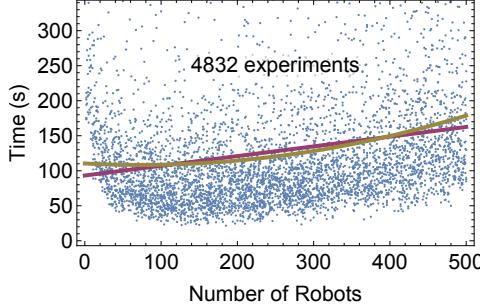


Figure 5: Data from *Varying Number* using robots to push an object through a maze to a goal location. The data indicates that this task has an optimal number of robots, perhaps due to the relative sizes of the robots, obstacles, and object. Best-fit linear and quadratic lines are overlaid for comparison.

Varying number Transport of goods and materials between points is at the heart of all engineering and construction in real-world systems. This experiment varied from 1 to 500 the population of robots used to transport an object. We kept the total area, maximum robot speed, and total net force the swarm could produce constant. The robots pushed a large hexagonal object through an S-shaped maze. Our hypothesis was that participants would complete the task faster with more robots. The results, shown in Fig. 5, do not support our hypothesis, indicating rather that there is a local minima around 130 robots.

Varying control Ultimately, we want to use swarms of robots to build things. This experiment compared different control architectures modeled after real-world devices.

We compared attractive and repulsive control with the global control used for the other experiments. The attractive and repulsive controllers were loosely modeled after scanning tunneling microscopes (STM), but also apply to magnetic manipulation, e.g. Khalil et al. (2013) and biological models, e.g. Goodrich et al. (2012). STMs can be used to arrange atoms and make small assemblies Avouris (1995). An STM tip is charged with electrical potential, and used to repel like-charged or to attract differently-charged molecules. In contrast, the global controller uses a uniform field (perhaps formed by parallel lines of differently-charged conductors) to pull molecules in the same direction. The experiment challenged players to assemble a three-block pyramid with a swarm of 16 robots.

The results were conclusive, as shown in Fig. ??: attractive control was the fastest, followed by global control, with repulsive control a distant last.

The median time using repulsive control was four times longer than with attractive control.

Varying visualization Sensing is expensive, especially on the nanoscale. To see nanocars Chiang et al. (2011), scientists fasten molecules that fluoresce light when activated by a strong light source. Unfortunately, multiple exposures can destroy these molecules, a process called *photobleaching*. Photobleaching can be minimized by lowering the excitation light intensity, but Cazes (2005) showed this increases the probability of missed detections. This experiment explores manipulation with varying amounts of sensing information: **full-state** sensing provides the most information by showing the position of all robots; **convex-hull** draws a convex hull around the outermost robots; **mean** provides the average position of the population; and **mean + variance** adds a confidence ellipse. Fig. 7 shows screenshots of the same robot swarm with each type of visual feedback. Full-state requires $2n$ data points for n robots. Convex-hull requires at worst $2n$, but usually a smaller number. Mean requires two, and variance three, data points. Mean and mean + variance are convenient even with millions of robots. Our hypothesis predicted a steady decay in performance as the amount of visual feedback decreased.

To our surprise, our experiment indicates the opposite: players with just the mean completed the task faster than those with full-state feedback. As Fig. ?? shows, the levels of feedback arranged by increasing completion time are [mean + variance, mean, full-state, convex-hull]. Anecdotal evidence from beta-testers who played the game suggests that tracking 100 robots is

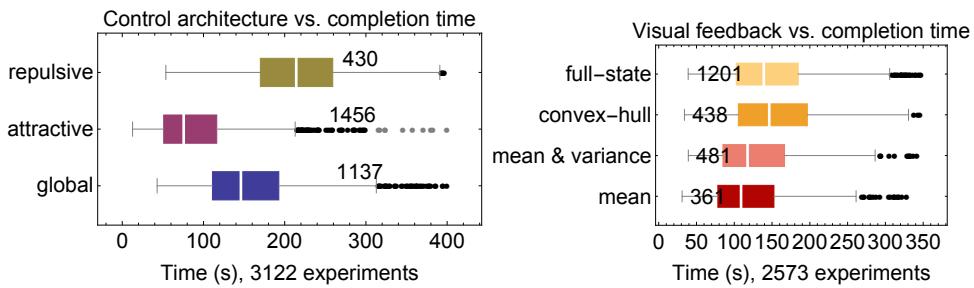


Figure 6: Left: Attractive control resulted in the shortest completion time and repulsive the longest for building a three-block pyramid. Right: Completion-time results for the four levels of visual feedback shown in Fig. 7. Surprisingly, players perform better with limited feedback—subjects with only the mean + variance outperformed all others.

overwhelming—similar to schooling phenomena that confuse predators—while working with just the mean + variance is like using a “spongy” manipulator. Our beta-testers found convex-hull feedback confusing and irritating. A single robot left behind an obstacle will stretch the entire hull, obscuring the majority of the swarm.

Varying noise Microrobots are affected by turbulence caused by random collisions with molecules. The effect of these collisions is called Brownian motion.

This experiment varied the strength of these disturbances to study how noise affects human control of large swarms. Noise was applied at every timestep as follows:

$$\begin{aligned}\dot{x}_i &= u_x + m_i \cos(\psi_i) \\ \dot{y}_i &= u_y + m_i \sin(\psi_i).\end{aligned}$$

Here m_i, ψ_i are uniformly IID, with $m_i \in [0, M]$ and $\psi_i \in [0, 2\pi]$, where M is a constant for each trial ranging from 0 to 200% of the maximum control power (u_{max}).

We hypothesized 200% noise was the largest a human could be expected to control—at 200% noise, the robots move erratically. Disproving our hypothesis, the results in Fig. ?? show only a 40% increase in completion time for the maximum noise.

Position control This online experiment examined how completion time scales with the number of robots n . Using a single square obstacle, users arranged $n \in [1, 10]$ robots into a specified goal pattern. The goal pattern formed a block ‘A’ with 10 robots, and lesser numbers of robots used a subset of these goal positions. Our hypothesis was that completion time

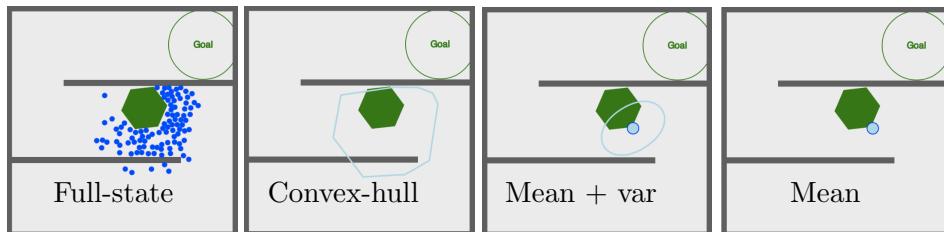


Figure 7: Screenshots from task *Vary Visualization*. This experiment challenges players to quickly steer 100 robots (blue discs) to push an object (green hexagon) into a goal region. We record the completion time and other statistics.

would increase linearly with the number of robots, as with our position control algorithm in Becker et al. (2013). Our results roughly corroborate this, as shown in Fig. ???. Though the number of robots presented to game players is uniformly distributed, larger n are more difficult, and the number of successful experiments drops steadily as n increases.

Note there is a bifurcation between $n=4$ and $n=5$ robots. For $n \in [1, 4]$ the goal patterns are not hollow, but starting at $n=5$ they are. A better experiment design would randomly place the goal positions. Initially we tried this, but our beta-testers strongly disliked trying to arrange robots in random patterns.

4 Theory

Consider holonomic robots that move in the 2D plane. We want to control position and velocity of the robots. First, assume a noiseless system containing one robot with mass m . Our inputs are global forces $[u_x, u_y]$. We define our state vector $\mathbf{x}(t)$ as the x position, x velocity, y position and y velocity. The state-space representation in standard form is:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \quad (1)$$

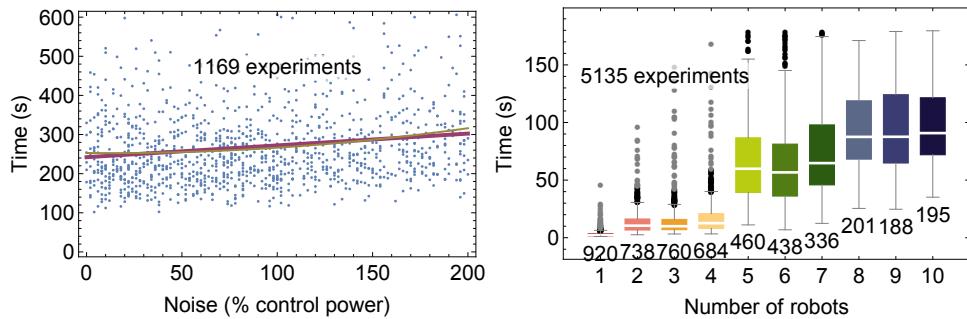


Figure 8: Left: Varying the noise from 0 to 200% of the maximum control input resulted in only a small increase in completion time. Right: Increasing the number of robots resulted in longer completion times. For more than 4 robots the goal pattern contained a void, which may have caused the longer completion times.

and our state space representation as:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & 0 \\ 0 & \frac{1}{m} \end{bmatrix} [u_x, u_y] \quad (2)$$

We want to find the number of states that we can control, which is given by the rank of the *controllability matrix*

$$\mathcal{C} = [B, AB, A^2B, \dots, A^{n-1}B]. \quad (3)$$

$$\text{Here } \mathcal{C} = \left[\begin{array}{cc|cc|cc|cc} 0 & 0 & \frac{1}{m} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{m} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right], \quad (4)$$

and thus all four states are controllable. This Section starts by proving independent position control of many robots is not possible, but the mean position can be controlled. We then provide conditions under which the variance of many robots is also controllable.

4.1 Independent control of many robots is impossible

A single robot is fully controllable, but what happens with n robots? For holonomic robots, movement in the x and y coordinates are independent, so for notational convenience without loss of generality we will focus only on movement in the x axis. Given n robots to be controlled in the x axis, there are $2n$ states: n positions and n velocities. Our state-space representation is:

$$\begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \vdots \\ \dot{x}_n \\ \ddot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ \vdots \\ x_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u_x \quad (5)$$

However, just as with one robot, we can only control two states because \mathcal{C} has rank two:

$$\mathcal{C} = \left[\begin{array}{c|c|c|c} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{array} \right], \dots \quad (6)$$

4.2 Controlling mean position

This means *any* number of robots controlled by a global command have just two controllable states in each axis. We cannot arbitrarily control the position and velocity of two or more robots, but what states are controllable? One option is to control the position and velocity of the j^{th} robot. To find a potentially more useful option, we create the following reduced order system that represents the average x position and x velocity of the n robots:

$$\begin{bmatrix} \dot{\bar{x}} \\ \ddot{\bar{x}} \end{bmatrix} = \frac{1}{n} \begin{bmatrix} 0 & 1 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ \vdots \\ x_n \\ \dot{x}_n \end{bmatrix} + \frac{1}{n} \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u_x$$

Thus:

$$\begin{bmatrix} \dot{\bar{x}} \\ \ddot{\bar{x}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \dot{\bar{x}} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_x \quad (7)$$

We again analyze \mathcal{C} :

$$\mathcal{C} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (8)$$

This matrix has rank two, and thus the average position and average velocity are controllable.

Due to symmetry of the control input, only the mean position and mean velocity are controllable. However, there are several techniques for breaking symmetry, for example by using obstacles as in Becker et al. (2013), or by allowing independent noise sources as in Becker et al. (2014b).

We control mean position with a PD controller that uses the mean position and mean velocity. Our control input is the global force applied to each robot:

$$\begin{aligned} u_x &= K_p(x_{goal} - \bar{x}) + K_d(0 - \bar{v}_x) \\ u_y &= K_p(y_{goal} - \bar{y}) + K_d(0 - \bar{v}_y) \end{aligned} \quad (9)$$

here K_p is the proportional gain, and K_d is the derivative gain.

4.3 Controlling the variance of many robots

The variance, σ_x^2, σ_y^2 , of n robots' position is computed as:

$$\begin{aligned}\bar{x}(\mathbf{x}) &= \frac{1}{n} \sum_{i=1}^n x_i, & \sigma_x^2(\mathbf{x}) &= \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2, \\ \bar{y}(\mathbf{x}) &= \frac{1}{n} \sum_{i=1}^n y_i, & \sigma_y^2(\mathbf{x}) &= \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2.\end{aligned}\quad (10)$$

Controlling the variance requires being able to increase and decrease the variance. We will list a sufficient condition for each. Microscale systems are affected by unmodelled dynamics. These unmodelled dynamics are dominated by Brownian noise, described by Einstein (1956). To model this (1) must be modified as follows:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) + W\varepsilon(t) \quad (11)$$

where $W\varepsilon(t)$ is a random perturbation produced by Brownian noise. Given a large obstacle-free workspace, a *Brownian noise* process increases the variance linearly with time.

$$\dot{\sigma}_x^2(\mathbf{x}(t), \mathbf{u}(t) \Leftarrow 0) = W\varepsilon \quad (12)$$

If faster dispersion is needed, the swarm can be pushed through obstacles such as a diffraction grating or Pachinko board Becker et al. (2013).

If robots with radius r are in a bounded environment with sides of length $[\ell_x, \ell_y]$, the unforced variance asymptotically grows to the variance of a uniform distribution,

$$[\sigma_x^2, \sigma_y^2] = \frac{1}{12}[(\ell_x - 2r)^2, (\ell_y - 2r)^2]. \quad (13)$$

A flat obstacle can be used to decrease variance. Pushing a group of dispersed robots against a flat obstacle will decrease their variance until the minimum-variance (maximum density) packing is reached. For large n , Graham and Sloane (1990) showed that the minimum-variance packing for n circles with radius r is

$$\sigma_{optimal}^2(n, r) \approx \frac{\sqrt{3}}{\pi} nr^2 \approx 0.55nr^2. \quad (14)$$

We will prove the origin is globally asymptotically stabilizable by using a control-Lyapunov function, as in Lyapunov (1992). A suitable Lyapunov function is the squared variance error:

$$\begin{aligned} V(t, \mathbf{x}) &= \frac{1}{2}(\sigma^2(\mathbf{x}) - \sigma_{goal}^2)^2 \\ \dot{V}(t, \mathbf{x}) &= (\sigma^2(\mathbf{x}) - \sigma_{goal}^2)\dot{\sigma}^2(\mathbf{x}) \end{aligned} \quad (15)$$

We note here that $V(t, \mathbf{x})$ is positive definite and radially unbounded, and $V(t, \mathbf{x}) \equiv 0$ only at $\sigma^2(\mathbf{x}) = \sigma_{goal}^2$. To make $\dot{V}(t, \mathbf{x})$ negative semi-definite, we choose

$$u(t) = \begin{cases} \text{move to wall} & \text{if } \sigma^2(\mathbf{x}) > \sigma_{goal}^2 \\ \text{move from wall} & \text{if } \sigma^2(\mathbf{x}) \leq \sigma_{goal}^2. \end{cases} \quad (16)$$

For such a $u(t)$,

$$\dot{\sigma}^2(\mathbf{x}) = \begin{cases} \text{negative} & \text{if } \sigma^2(\mathbf{x}) > \max(\sigma_{goal}^2, \sigma_{optimal}^2(n, r)) \\ W\epsilon & \text{if } \sigma^2(\mathbf{x}) \leq \sigma_{goal}^2, \end{cases} \quad (17)$$

and thus $\dot{V}(t, \mathbf{x})$ is negative definite and the variance is globally asymptotically stabilizable.

A controller to regulate the variance to σ_{ref}^2 is:

$$\begin{aligned} u_x &= K_p(x_{goal}(\sigma_{ref}^2) - \bar{x}) - K_d\bar{v}_x + K_i(\sigma_{ref}^2 - \sigma_x^2) \\ u_y &= K_p(y_{goal}(\sigma_{ref}^2) - \bar{y}) - K_d\bar{v}_y + K_i(\sigma_{ref}^2 - \sigma_y^2). \end{aligned} \quad (18)$$

In a slight abuse of notation we call the gain scaling the variance error K_i because the variance, if unregulated, integrates over time. Eq. (18) assumes the nearest wall is to the left of the robot at $x = 0$, and chooses a reference goal position that in steady-state would have the correct variance according to (13):

$$x_{goal}(\sigma_{ref}^2) = r + \sqrt{3\sigma_{ref}^2} \quad (19)$$

If another wall is closer, the signs of $[K_p, K_i]$ are inverted, and the location x_{goal} is translated.

4.4 Controlling both mean and variance of many robots

The mean and variance of the swarm cannot be controlled simultaneously, however if the dispersion due to Brownian motion is less than the maximum controlled speed, we can adopt the hybrid, hysteresis-based controller

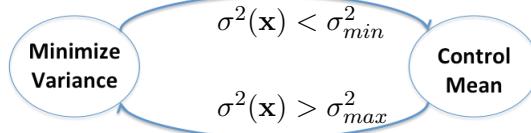


Figure 9: Two states to control mean and variance of a robot swarm.

shown in Alg. 1 to regulate the mean and variance. Such a controller normally controls the mean position, but switches to minimizing variance if the variance exceeds some σ_{max}^2 . Variance is reduced until less than σ_{min}^2 , then control again regulates the mean position. This technique satisfies control objectives that evolve at different rates as in Kloetzer and Belta (2007), and the hysteresis avoids rapid switching between control modes. The process is illustrated in Fig. 9.

Algorithm 1 Hybrid mean and variance control

Require: Knowledge of swarm mean $[\bar{x}, \bar{y}]$, variance $[\sigma_x^2, \sigma_y^2]$, the locations of the rectangular boundary $\{x_{min}, x_{max}, y_{min}, y_{max}\}$, and the target mean position $[x_{target}, y_{target}]$.

```

1: flagx  $\leftarrow$  false, flagy  $\leftarrow$  false
2: xgoal  $\leftarrow$  xtarget, ygoal  $\leftarrow$  ytarget
3: loop
4:   if  $\sigma_x^2 > \sigma_{max}^2$  then
5:     xgoal  $\leftarrow$  xmin
6:     flagx  $\leftarrow$  true
7:   else if flagx and  $\sigma_x^2 < \sigma_{min}^2$ 
8:     xgoal  $\leftarrow$  xtarget
9:     flagx  $\leftarrow$  false
10:  end if
11:  if  $\sigma_y^2 > \sigma_{max}^2$  then
12:    ygoal  $\leftarrow$  ymin
13:    flagy  $\leftarrow$  true
14:  else if flagy and  $\sigma_y^2 < \sigma_{min}^2$ 
15:    ygoal  $\leftarrow$  ytarget
16:    flagy  $\leftarrow$  false
17:  end if
18:  Apply (9) to move toward  $[x_{goal}, y_{goal}]$ 
19: end loop

```

A key challenge is to select proper values for σ_{min}^2 and σ_{max}^2 . The opti-

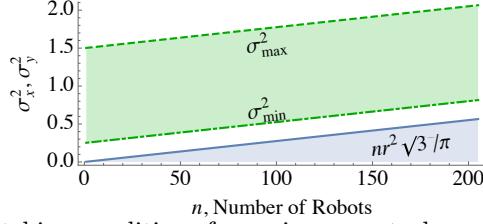


Figure 10: The switching conditions for variance control are set as a function of n , and designed to be larger than the optimal packing density. The above plot uses robot radius $r = 1/10$.

mal packing variance was given in (14). The random packings generated by pushing our robots into corners are suboptimal, so we choose the conservative values shown in Fig. 10:

$$\begin{aligned}\sigma_{min}^2 &= 2.5r + \sigma_{optimal}^2(n, r) \\ \sigma_{max}^2 &= 15r + \sigma_{optimal}^2(n, r).\end{aligned}\quad (20)$$

5 Simulation of control laws

Our simulations use a Javascript port of Box2D, a popular 2D physics engine with support for rigid-body dynamics and fixed-time step simulation Catto (2010). All experiments ran on a Chrome web browser on a 2.6 GHz Macbook. All code is available at Shahrokhi and Becker (2015a).

5.1 Controlling the mean position

We performed a parameter sweep using the PD controller (9) to identify the best control gains . Representative experiments are shown in Fig. 11. 100 robots were used and the maximum speed was 3 meters per second. As shown in Fig. 11, we can achieve an overshoot of 1% and a rise time of 1.52 s with $K_p = 4$, and $K_d = 1$.

5.2 Controlling the variance

For variance control we use the control law discussed in Section 4.3. Results are shown in Fig. 12, with $K_{p,i,d} = [4, 1, 1]$.

5.3 Hybrid control of mean and variance

Fig. 13 shows a simulation run of the hybrid controller in Alg. 1 with 100 robots in a square workspace containing no internal obstacles.

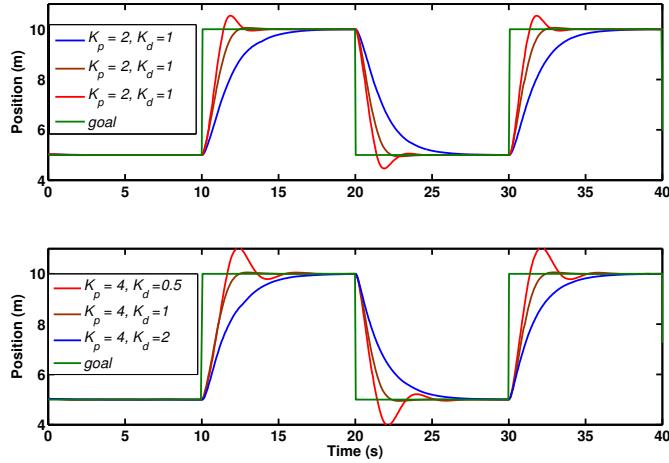


Figure 11: Tuning proportional (K_p , top) and derivative (K_d , bottom) gain values in (9) improves performance with $n = 100$ robots.

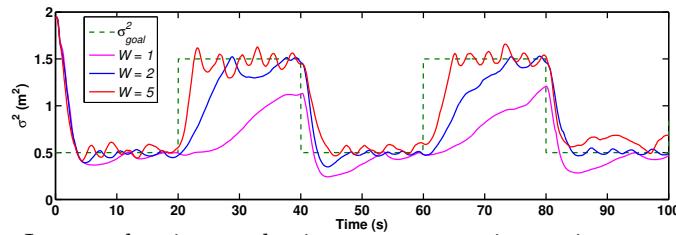


Figure 12: Increased noise results in more responsive variance control because stronger Brownian noise causes a faster increase of variance.

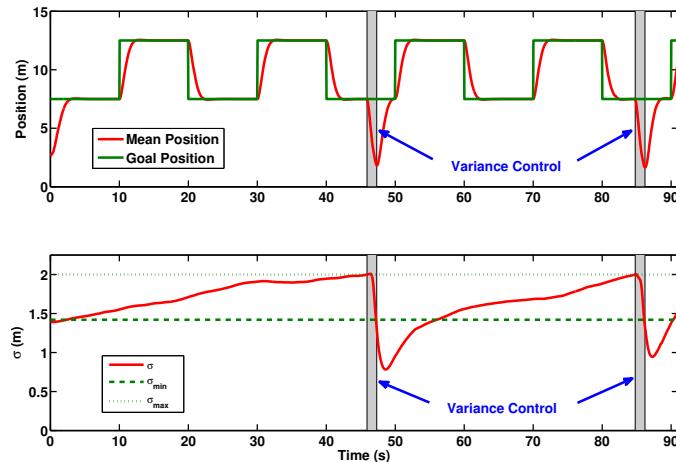


Figure 13: Simulation result with 100 robots under hybrid control Alg. 1, which controls both the mean position (top) and variance (bottom). For ease of analysis, only x position and variance are shown.

6 Object manipulation results

This section analyzes a *object manipulation* task attempted by both our hybrid, hysteresis-based controller and by human users.

6.1 Human-controlled object manipulation

As we saw in previous section, players with just the mean completed the task faster than those with full-state feedback. As Fig. ?? shows, the levels of feedback arranged by increasing completion time are [mean, mean + variance, full-state, convex-hull]. Interviews with beta-testers suggests that tracking 100 robots was overwhelming—similar to schooling phenomenons that confuse predators—while working with just the mean + variance was like using a “spongy” manipulator. Convex-hull feedback was confusing and irritating because a single robot left behind an obstacle would distort the entire hull, obscuring the information about the majority of the swarm.

6.2 Automated object manipulation (simulation)

Fig. 14 shows snapshots during an execution of this algorithm. To solve this object manipulation task, we discretized the environment. On this discretized grid we used breadth-first search to determine \mathbf{M} , the shortest distance from any grid cell to the goal, and generated a gradient map $\nabla\mathbf{M}$ toward the goal as shown in Fig. 15. The object’s center of mass is at \mathbf{b} and has radius r_b . Three constants are needed, where $k_1 > k_2 > 1$ and $1 > k_3 > 0$. All experiments used $[k_1, k_2, k_3] = [2.5, 1.5, 0.1]$. The robots were directed to assemble behind the object at $\mathbf{b} - k_2 r_b \nabla\mathbf{M}(\mathbf{b})$, then move to $\mathbf{b} - k_3 r_b \nabla\mathbf{M}(\mathbf{b})$ to push the object toward the goal location. We use the hybrid hysteresis-based controller in Alg. 1 to track the desired position, while maintaining sufficient robot density to move the object by switching to minimize variance whenever variance exceeds a set limit. The minimize variance control law (18) is slightly modified to choose the nearest corner further from the goal than \mathbf{b} with an obstacle-free straight-line path to \mathbf{b} . The control algorithm for object manipulation is listed in Alg. 2. Experimental results are summarized in Fig. ???. Although larger populations of robots can apply more force, minimizing the variance requires more time with larger populations and dominates task completion time.

Algorithm 2 is an imperfect solution and has a failure mode if the robot swarm becomes multi-modal with modes separated by an obstacle, as shown in Fig. ???. In this case, moving toward a corner will never reduce the

Algorithm 2 Object-manipulation controller for a robotic swarm.

Require: Knowledge of swarm mean $[\bar{x}, \bar{y}]$, variance $[\sigma_x^2, \sigma_y^2]$, moveable object's center of mass \mathbf{b} , map of the environment, and the locations of all convex corners \mathbf{C}

Require: Robot distribution is unimodal

Require: Obstacle-free, straight-line path from swarm to moveable object

- 1: Compute \mathbf{M} , the distance to goal, with breadth-first search
- 2: Compute the gradient, $\nabla\mathbf{M}$
- 3: $\mathbf{C} \leftarrow \text{sort}(\mathbf{C})$ according to $-\mathbf{M}$
- 4: **while** \mathbf{b} is not in goal region **do**
- 5: $\sigma^2 \leftarrow \max(\sigma_x, \sigma_y)$
- 6: **if** $\sigma^2 > \sigma_{\max}^2$ **then**
- 7: **while** $\sigma^2 > \sigma_{\min}^2$ **do**
- 8: $\mathbf{c}_i \leftarrow$ the nearest corner in \mathbf{C} to $[\bar{x}, \bar{y}]$
- 9: $[x_{goal}, y_{goal}] \leftarrow \mathbf{c}_i$
- 10: **if** $\mathbf{M}(\mathbf{b}) > \mathbf{M}(\mathbf{c}_i)$ **then**
- 11: $[x_{goal}, y_{goal}] \leftarrow \mathbf{c}_{i-1}$
- 12: Apply (9) to move toward $[x_{goal}, y_{goal}]$
- 13: **end if**
- 14: **end while**
- 15: **else**
- 16: **if** $\text{distance}(\mathbf{b}, [x_{goal}, y_{goal}]) > k_1 r_b$ **then**
- 17: $r_p \leftarrow k_2 r_b$ ▷ guarded move
- 18: **else**
- 19: $r_p \leftarrow k_3 r_b$ ▷ pushing move
- 20: **end if**
- 21: $[x_{goal}, y_{goal}] \leftarrow \mathbf{b} - r_p \nabla \mathbf{M}(\mathbf{b})$
- 22: **end if**
- 23: Apply (9) to move toward $[x_{goal}, y_{goal}]$
- 24: **end while**

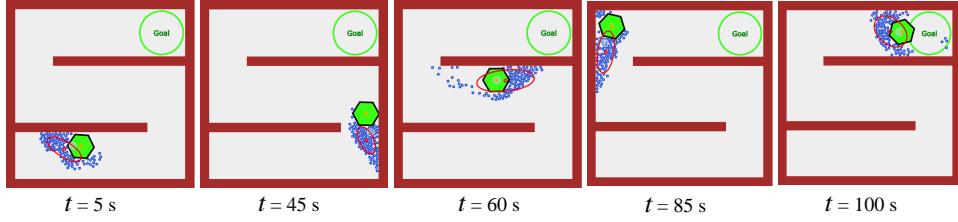


Figure 14: Snapshots showing an object manipulation simulation with 100 robots under automatic control. See the video attachment for an animation Shahrokhi and Becker (2015d).

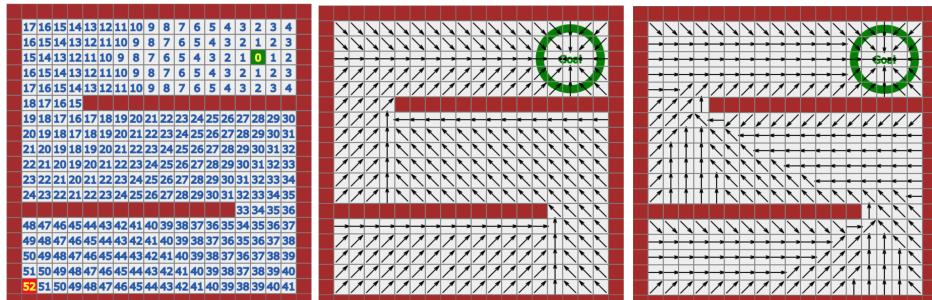


Figure 15: The BFS algorithm finds the shortest path for the moveable object (left), which is used to compute gradient vectors (middle). Using policy iterations enables encoding penalties for being near obstacles (right).

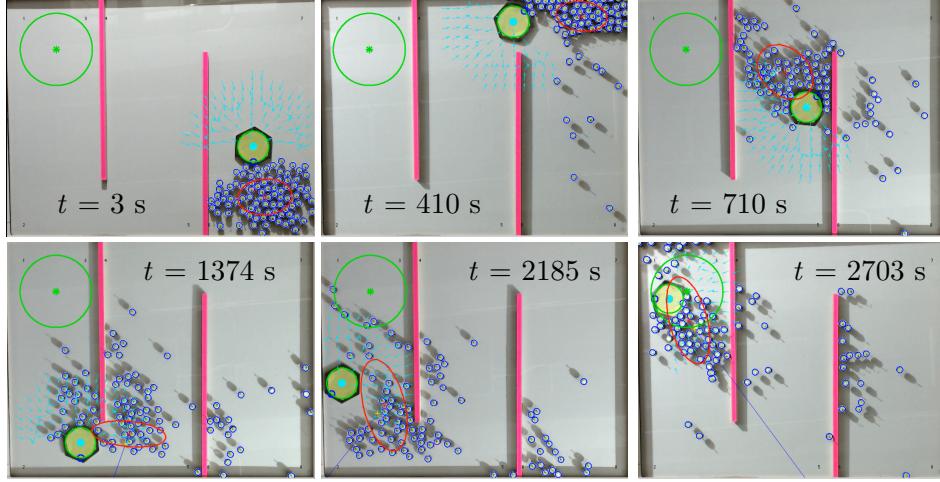


Figure 16: Snapshots showing the object manipulation experiment with 100 kilo-
bots under automatic control. The automatic controller will see the pink objects
as obstacles and finds its path to the goal. See the video attachment for an anima-
tion Shahrokh and Becker (2015d)

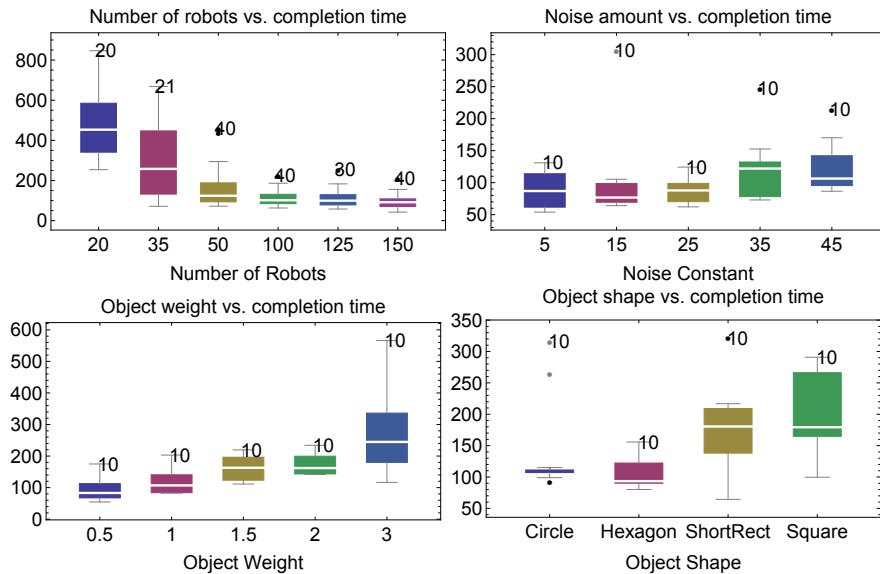


Figure 17: Parameter sweep for a) number of robots, b) different noise values, c)
object weight, and d) object shape. Each bar is labelled with the number of trials.

variance below σ_{min}^2 .

The first challenge is to identify when the distribution has become multi-modal. Measuring just the mean and variance is insufficient to determine if a distribution is no longer unimodal, but if the swarm is being directed to a corner, and the variance does not reduce below σ_{min}^2 , the swarm has become separated. In this case, we must either manipulate with a partial swarm, or run a gathering algorithm. For the S-shaped workspace in this study, an open-loop input that commands the swarm to move in succession {WEST, NORTH, EAST, SOUTH} will move the swarm to the bottom right corner. This is not true for all obstacle fields. In a T-shaped workspace, it is not possible to find an open-loop input that will move the entire swarm to the bottom of the T.

Using only the mean and variance may be overly restrictive. Many heuristics using high-order moments have been developed to test if a distribution is multimodal Haldane (1951). Often the sensor data itself, though it may not resolve individual robots, will indicate multi-modality. For instance CCD images reveal clusters of bacteria, and MRI scans show agglomerations of particles Stuber et al. (2007). This data can be fitted with k -means or expectation maximization algorithms, and manipulation could be performed with the nearest swarm of sufficient size.

7 Object manipulation with hardware robots

7.1 Environmental setup

Our experiments use centimeter-scale hardware systems called *kilobots*. While those are far larger than the micro scale devices we model, using kilobots allows us to emulate a variety of dynamics, while enabling a high degree of control over robot function, the environment, and data collection. The kilobot Rubenstein et al. (2012, 2014) is a low-cost robot designed for testing collective algorithms with large numbers of robots. It is available commercially or as an open source platform K-Team (2015). Each robot is approximately 3 cm in diameter, 3 cm tall, and uses two vibration motors to move on a flat surface at speeds up to 1 cm/s. Each robot has one ambient light sensor that is used to implement *phototaxis*, moving towards a light source. In these experiments as shown in Fig. 18, we used $n=100$ kilobots, a 1.5 m \times 1.2 m whiteboard as the workspace, and lights: four 50W LED floodlights at the corners and four 30W LED floodlights on the sides of a 6 m square centered on the workspace and 1.5 m above the table. The lights were controlled using an Arduino Uno board connected to an 8 relay shield

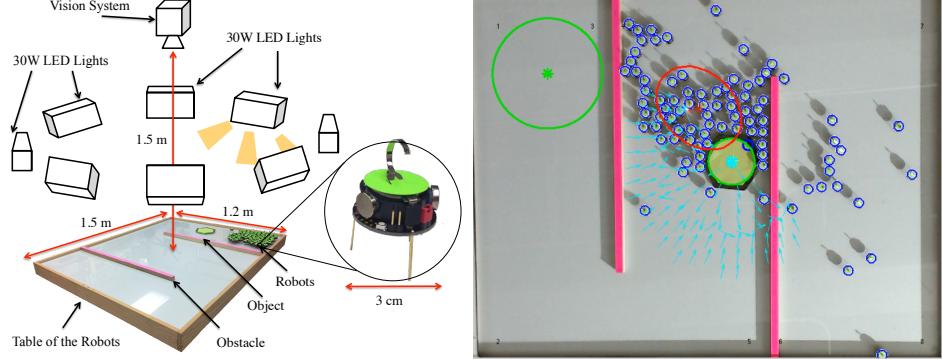


Figure 18: Hardware platform: table with 1.5×1.2 m workspace, surrounded by eight remotely triggered 30W LED floodlights, with an overhead machine vision system (Left). A swarm of robots, all controlled by a uniform force field, can be effectively controlled by a hybrid controller that knows only the first and second moments of the robot distribution. Here is a swarm of hardware robots (kilobots) that pushes a green hexagon toward the goal (Right). See video attachment Shahrokh and Becker (2015d)

board. Above the table, an overhead machine vision system tracks the position of the swarm. Laser-cut patterns for the neon green fiducial markers and MATLAB tracking code are available at our github repository Shahrokh and Becker (2015b).

7.2 Automated object manipulation (hardware experiment)

Heuristics to handle outliers and improve performance The variance controller in Alg. 1 is a greedy algorithm that is susceptible to outliers. The controller in Shahrokh and Becker (2015a) failed in 14% trials, often because workspace obstacles made some robots unable to reach the object. This failure rate increases if object weight increases or ground-robot friction increases. The mean and covariance calculations (10) included all robots in the workspace. Robots that cannot reach the object due to obstacles skew these calculations. The state machine in Fig. 20.a solves this problem by creating two states for the maze: either main or transfer. Each state has a set of regions representing a discretized visibility polygon. Whenever the object crosses a region boundary the state toggles. The main regions are generated by extending obstacles until they meet another obstacle shown in Fig. 20.b. The transfer regions are perpendicular to obstacle boundaries,

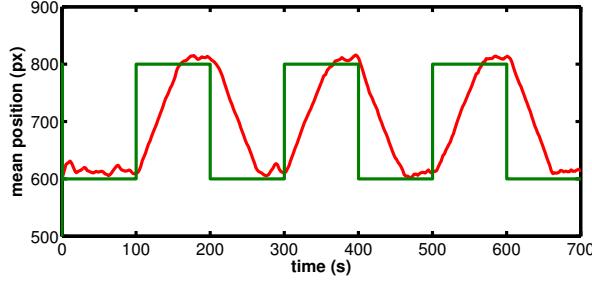


Figure 19: Mean Control plot with kilobots.

and act as a buffer between two main regions shown in Fig. 20.c. This filtering increases experimental success because the mean calculation only includes nearby robots that can directly interact with the object. In the example, we want the robots to push the object to the right. Without filtering the robots, the orange star is the mean and the algorithm would instruct the robots to push the object southeast. The filtered mean is at the yellow star and the algorithm instructs the robots to push the object directly east. This heuristic improves performance by 50% or less regarding the object heaviness.

Potential fields for swarm management Unfortunately, when the swarm is in front of the object, control law (9) pushes the object backwards. To fix this, we implement a potential field approach Spong and Vidyasagar (2008) that attracts the swarm to the intermediate goal, but repulses the swarm from in front of the object. The repulsive potential field is centered at object's COM and is active for a radius ρ_0 , but is implemented only when the swarm mean is within θ of the desired direction of motion as it is shown in Fig. 21.

$$F_{att} = -\zeta \Delta \rho / \rho \quad (21)$$

$$F_{rep} = \begin{cases} \eta(1/\rho - 1/\rho_0) \frac{1}{\rho^2} \Delta \rho & \rho \leq \rho_0 \text{ \& } \theta > \pi/2 \\ 0 & \text{otherwise} \end{cases}$$

In simulations, $\theta = \pi/2$, $\eta = 75$, $\zeta = 2$ and $\rho_0 = 3$. Because the kilobot hardware experiments have a slower time constant, they use $\theta = \pi/2$, $\eta = 50$, $\zeta = 1$ and $\rho_0 = 7.5$.

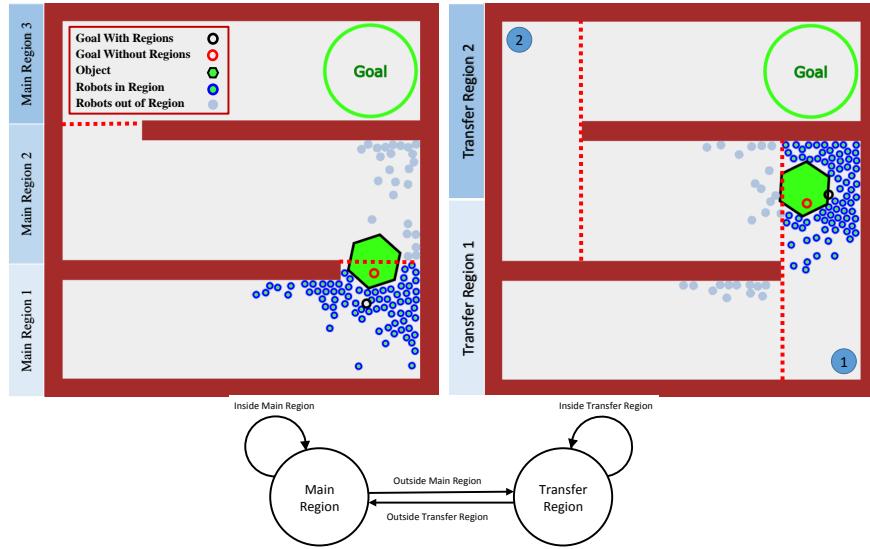


Figure 20: The state machine and regions.

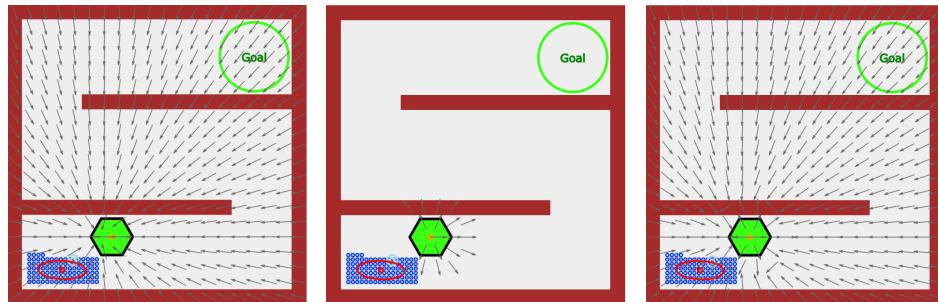


Figure 21: The attractive field source is centered on (left). Repulsive field source is centered at object's COM(middle). We use a combined forces to use this method to avoid pushing the object backwards(right).

References

- Michael D. Armani, Satej V. Chaudhary, Roland Probst, and Benjamin Shapiro. Using feedback control of microflows to independently steer multiple particles. *Journal of Microelectromechanical systems*, 15(4), August 2006.
- Phaedon Avouris. Manipulation of matter at the atomic and molecular levels. *Accounts of chemical research*, 28(3):95–102, 1995.
- Shishir Bashyal and Ganesh Kumar Venayagamoorthy. Human swarm interaction for radiation source search and localization. In *Swarm Intelligence Symposium (SIS)*, pages 1–8. IEEE, 2008.
- A. Becker, R. Sandheinrich, and T. Bretl. Automated manipulation of spherical objects in three dimensions using a gimbaled air jet. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 781 –786, oct. 2009. doi: 10.1109/IROS.2009.5354427. URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5354427>.
- Aaron Becker and Timothy Bretl. Approximate steering of a unicycle under bounded model perturbation using ensemble control. *IEEE Trans. Robot.*, 28(3):580–591, June 2012.
- Aaron Becker, Cem Onyuksel, and Timothy Bretl. Feedback control of many differential-drive robots with uniform control inputs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2012.
- Aaron Becker, Golnaz Habibi, Justin Werfel, Michael Rubenstein, and J. McLurkin. Massive uniform manipulation: Controlling large populations of simple robots with a common input signal. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 520–527, November 2013.
- Aaron Becker, Chris Ertel, and James McLurkin. Crowdsourcing swarm manipulation experiments: A massive online user study with large swarms of simple robots. In *IEEE International Conference on Robotics and Automation*, 2014a.
- Aaron Becker, Cem Onyuksel, Timothy Bretl, and James McLurkin. Controlling many differential-drive robots with uniform control inputs. *The international journal of Robotics Research*, 33(13):1626–1644, 2014b.

- Erin Catto. User manual, Box2D: A 2D physics engine for games, <http://www.box2d.org>, 2010. URL <http://www.box2d.org>.
- Jack Cazes. *Encyclopedia of Chromatography*, volume 2. Taylor & Francis Group, New York, second edition, 2005. ISBN 0824727878.
- Pinn-Tsong Chiang, Johannes Mielke, Jazmin Godoy, Jason M. Guerrero, Lawrence B. Alemany, Carlos J. Villagómez, Alex Saywell, Leonhard Grill, and James M. Tour. Toward a light-driven motorized nanocar: Synthesis and initial imaging of single molecules. *ACS Nano*, 6(1):592–597, February 2011. doi: 10.1021/nn203969b.
- Jean-Pierre de la Croix and Magnus Egerstedt. Controllability characterizations of leader-based swarm interactions. In *2012 AAAI Fall Symposium Series*, 2012.
- Raphael Deimel and Oliver Brock. A novel type of compliant, underactuated robotic hand for dexterous grasping. *Robotics: Science and Systems, Berkeley, CA*, pages 1687–1692, 2014.
- Eric Diller, Joshua Giltinan, and Metin Sitti. Independent control of multiple magnetic microrobots in three dimensions. *The International Journal of Robotics Research*, 32(5):614–631, 2013. URL <http://ijr.sagepub.com/content/32/5/614.abstract>.
- B.R. Donald, C.G. Levey, C.D. McGray, I. Paprotny, and D. Rus. An untethered, electrostatic, globally controllable MEMS micro-robot. *J. of MEMS*, 15(1):1–15, February 2006. ISSN 1057-7157.
- B.R. Donald, C.G. Levey, and I. Paprotny. Planar microassembly by parallel actuation of MEMS microrobots. *J. of MEMS*, 17(4):789–808, August 2008.
- Albert Einstein. *Investigations on the Theory of the Brownian Movement*. Courier Corporation, 1956.
- Chris Ertel and Aaron Becker. Swarmcontrol git repository, September 2013. URL <https://github.com/crertel/swarmmanipulate.git>.
- J. Fink, N. Michael, and V. Kumar. Composition of vector fields for multi-robot manipulation via caging. In *Robotics Science and Systems*, Atlanta, GA, June 2007.

- Steven Floyd, Eric Diller, Chytra Pawashe, and Metin Sitti. Control methodologies for a heterogeneous group of untethered magnetic micro-robots. *Int. J. Robot. Res.*, 30(13):1553–1565, November 2011.
- Dominic Frutiger, Bradley Kratochvil, Karl Vollmers, and Bradley J. Nelson. Magmites - wireless resonant magnetic microrobots. In *IEEE Int. Conf. Rob. Aut.*, Pasadena, CA, May 2008.
- Michael A Goodrich, Sean Kerman, Brian Pendleton, and PB Sujit. What types of interactions do bio-inspired robot swarms and flocks afford a human? In *Robotics: Science and Systems*, 2012.
- Ronald L. Graham and Neil JA Sloane. Penny-packing and two-dimensional codes. *Discrete & Computational Geometry*, 5(1):1–11, 1990.
- JBS Haldane. Simple tests for bimodality and bitangentiality. *Annals of eugenics*, 16(1):359–364, 1951.
- K-Team. Kilobot, www.k-team.com/mobile-robotics-products/kilobot, 2015.
- Islam S. M. Khalil, Frank van den Brink, Ozlem Sardan Sukas, and Sarthak Misra. Microassembly using a cluster of paramagnetic microparticles. In *IEEE International Conference on Robotics and Automation*, pages 5507–5512, Karlsruhe, Germany, May 2013.
- Marius Kloetzer and Calin Belta. Temporal logic planning and control of robotic swarms by hierarchical abstractions. *Robotics, IEEE Transactions on*, 23(2):320–330, 2007.
- Andreas Kolling, Steven Nunnally, and Michael Lewis. Towards human control of robot swarms. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 89–96. ACM, 2012.
- Aleksandr Mikhailovich Lyapunov. *The General Problem of the Stability of Motion*, translated and edited by A.T. Fuller. Taylor & Francis, London, 1992.
- K.M. Lynch. Locally controllable manipulation by stable pushing. *IEEE Trans. Robot. Autom.*, 15(2):318–327, April 1999.

- Sylvain Martel, Samira Taherkhani, Maryam Tabrizian, Mahmood Mommadi, Dominic de Lanauze, and Ouajdi Felfoul. Computer 3d controlled bacterial transports and aggregations of microbial adhered nanocomponents. *Journal of Micro-Bio Robotics*, 9(1-2):23–28, 2014.
- Lael U Odhner, Leif P Jentoft, Mark R Claffee, Nicholas Corson, Yaroslav Tenzer, Raymond R Ma, Martin Buehler, Robert Kohout, Robert D Howe, and Aaron M Dollar. A compliant, underactuated hand for robust manipulation. *The International Journal of Robotics Research*, 33(5):736–752, 2014.
- Dan R. Olsen Jr and Stephen Bart Wood. Fan-out: Measuring human control of multiple robots. In *SIGCHI Conference on Human Factors in Computing Systems*, pages 231–238, Vienna, Austria, April 2004.
- Kathrin E. Peyer, Li Zhang, and Bradley J. Nelson. Bio-inspired magnetic swimming microrobots for biomedical applications. *Nanoscale*, 2013.
- M. Rubenstein, C. Ahler, and R. Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *IEEE Int. Conf. Rob. Aut.*, pages 3293–3298, May 2012.
- Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.
- Shiva Shahrokhi and Aaron T. Becker. BlockPushingIROS2015, <https://github.com/aabecker/swarmcontrolsandbox/blob/master/exam-plecontrollers/blockpushingiros2015.html>, July 2015a.
- Shiva Shahrokhi and Aaron T. Becker. ”Wall-Friction Swarm Shape Control”, <https://github.com/aabecker/swarmcontrolsandbox/tree/master/papers/icra2016>, August 2015b.
- Shiva Shahrokhi and Aaron T. Becker. Stochastic swarm control with global inputs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page tbd, September 2015c.
- Shiva Shahrokhi and Aaron T. Becker. Stochastic swarm control, <https://youtu.be/tcej-9e6-4o>, October 2015d. URL <https://youtu.be/tcej-9e6-4o>.

Yasuhiro Shirai, Andrew J. Osgood, Yuming Zhao, Kevin F. Kelly, and James M. Tour. Directional control in thermally driven single-molecule nanocars. *Nano Letters*, 5(11):2330–2334, February 2005. doi: 10.1021/nl051915k.

Mark W Spong and Mathukumalli Vidyasagar. *Robot dynamics and control*. John Wiley & Sons, 2008.

Peter Squire, Greg Trafton, and Raja Parasuraman. Human control of multiple unmanned vehicles: effects of interface type on execution and task switching times. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, HRI '06, pages 26–32, New York, NY, USA, 2006. ACM. ISBN 1-59593-294-1. doi: 10.1145/1121241.1121248. URL <http://doi.acm.org/10.1145/1121241.1121248>.

Matthias Stuber, Wesley D Gilson, Michael Schär, Dorota A Kedziorek, Lawrence V Hofmann, Saurabh Shah, Evert-Jan Vonken, Jeff WM Bulte, and Dara L Kraitchman. Positive contrast visualization of iron oxide-labeled stem cells using inversion-recovery with on-resonant water suppression (iron). *Magnetic Resonance in Medicine*, 58(5):1072–1077, 2007.

A. Sudsang, F. Rothganger, and J. Ponce. Motion planning for disc-shaped robots pushing a polygonal object in the plane. *IEEE Trans. Robot. Autom.*, 18(4):550–562, August 2002.

K. Takahashi, K. Hashimoto, N. Ogawa, and H. Oku. Organized motion control of a lot of microorganisms using visual feedback. In *IEEE Int. Conf. Rob. Aut.*, pages 1408–1413, May 2006. doi: 10.1109/ROBOT.2006.1641906. URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1641906&isnumber=34383>.

S. Tottori, L. Zhang, F. Qiu, K.K. Krawczyk, A. Franco-Obregón, and B. J. Nelson. Magnetic helical micromachines: Fabrication, controlled swimming, and cargo transport. *Advanced Materials*, 24(811), 2012.