Massive Uniform Manipulation:
Controlling Large Populations of Simple Robots with a Common Input Signal

Aaron Becker, Golnaz Habibi, Justin Werfel, Michael Rubenstein, and James McLurkin

***Abstract*— Roboticists, biologists, and chemists are now producing large populations of simple robots, but controlling large populations of robots with limited capabilities is difficult, due to communication and onboard-computation constraints. Direct human control of large populations seems even more challenging.**

In this paper we investigate control of mobile robots that move in a 2D workspace using three different system models. We focus on a model that uses broadcast control inputs specified in the global reference frame.

In an obstacle-free workspace this system model is uncontrollable because it has only two controllable degrees of freedom—all robots receive the same inputs and move uniformly. We prove that adding a single obstacle can make the system controllable, for any number of robots. We provide a position control algorithm, and demonstrate through extensive testing with human subjects that many manipulation tasks can be reliably completed, even by novice users, under this system model, with performance benefits compared to the alternate models.

We compare the sensing, computation, communication, time, and bandwidth costs for all three system models. Results are validated with extensive simulations and hardware experiments using over 100 robots.

I. Introduction

It is now possible to make and field very large (10^3–10^{14}) populations of simple robots. Potential applications for these robots include targeted therapy, sensing, and actuation. With large populations come two fundamental challenges: (1) how to perform state estimation for these robots, and (2) how to control the robots.

Traditional approaches often assume independent control signals for each robot [1], but each additional independent signal requires engineering and communications bandwidth. This becomes more challenging as the robot size decreases. At the molecular scale [2], there is a bounded number of individual-specific modifications that can be made.

More recently, 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 [3], [4]; magnetic structures with different cross-sections that could be independently steered [5]; *MagMite* microrobots with different resonant frequencies and a global magnetic field [6]; and magnetically controlled nanoscale helical screws constructed to stop movement at different cutoff frequencies of a global magnetic field [7]. In our previous work with robots that can be modeled as nonholonomic unicycles, we showed that an inhomogeneity in turning speed is enough to make even an infinite number of robots controllable with regard to position. All these approaches show promise, but they also require both excellent state estimation and perfect heterogeneity (no duplication). In addition, the controllers required at best a summation over all the robot states [8] and at worst a matrix inversion [9]. These approaches become impractical for large robot populations.

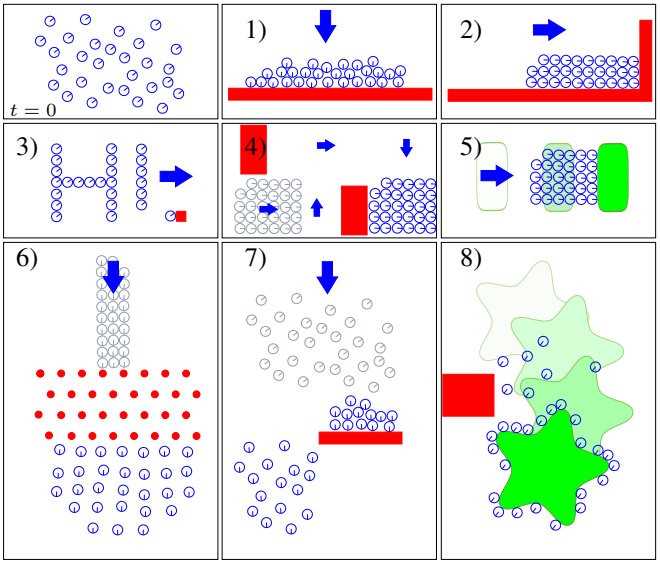

Fig. 1. Examples of manipulating many simple robots using uniform global inputs. Shown are 1) 1D compression, 2) 2D compression, 3) position control, 4) obstacle avoidance, 5) pushing an object, 6) dispersion, 7) splitting, and 8) guiding an object along a trajectory. Many of these can be realized with an economy of sensing and communication bandwidth. See the video attachment for simulation and hardware demonstrations, http://www.youtube.com/watch?v=px5RdSvGD2Q.

In this paper we take a very different approach, illustrated in Fig. 1. We assume a population of approximately identical planar robots and one global control signal consisting of a vector all robots should move along. This system is not controllable because the robots move uniformly, and applying a control signal transposes the entire group identically along the vector. However, we show that a single square obstacle is sufficient to control the final position of every robot under mild workspace constraints. Moreover, we catalog primitive operations, such as techniques for gathering or dispersing robots in 2D. These manipulation primitives can be accomplished with a constant number of commands that,

A. Becker, G. Habibi, and J. McLurkin are with the Computer Science Department, Rice University, Houston, TX 77005 USA aabecker@gmail.com, {gh4, jmclurkin}@rice.edu, J. Werfel and M. Rubenstein are with Harvard University, Cambridge, MA 02138 USA {mrubenst@seas, justin.werfel@wyss}.harvard.edu

unlike techniques relying on inhomogeneity, do not increase with population size.

Our paper is organized as follows. After a discussion of related work in Section II, we describe our problem and algorithmic results in Section III. We discuss the results of simulations and hardware experiments in Section IV, and end with concluding remarks in Section V.

II. Related Work

Our previous work [8], [9] 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. This paper seeks to establish control algorithms that extend to many thousands of agents. These techniques derive inspiration from the field of nonprehensile manipulation [10].

A. Three challenges for massive manipulation

While it is now possible to create many micro- and nanorobots, there remain challenges in control, sensing, and computation.

1) Control—global inputs: The systems [2]–[7], [11]–[13] all rely on global inputs, where each robot receives an exact copy of the control signal. Two reasonable questions are "What tasks are possible with many robots, all under uniform control inputs?" and "What tasks are impossible with many robots, all under uniform control inputs?"

2) Sensing—large populations: Parallel control of n differential-drive robots in a plane requires $3n$ state variables. Even holonomic robots require $2n$ state variables. Numerous methods exist for measuring this state in micro- and nanorobotics. These solutions use computer vision systems to sense position and heading angle, with corresponding challenges of handling missed detections and registration of detections with corresponding robots. These challenges are increased at the nanoscale where sensing competes with control for communication bandwidth. In this paper we outline control techniques that require only the first and second moments of a population's position, or the bounding box containing all robots of interest.

3) Computation—calculating the control law: In our previous work the controllers required at best a summation over all the robot states [8] and at worst a matrix inversion [9]. These operations become intractable for large populations of robots. This paper, by focusing on direct *human* control of large robot populations, accentuates computational difficulties because the controllers are implemented by the unaided human operator. We present an approach that, for many tasks, bypasses these large-population problems by allowing the user to command the entire population as a single unit. For position control–bringing each robot to a desired final position–we cannot bypass this problem, but we provide an algorithm that scales linearly in the number of robots.

B. Nonprehensile manipulation

In *nonprehensile manipulation*, a robot affects its environment without grasping [10], [14], [15]. In some ways, our problem formulation is the inverse of nonprehensile manipulation. Rather than just use a robot to restructure the environment, we use the environment to restructure a population of robots.

We can also use a large population of robots for traditional nonprehensile tasks, such as transporting objects using the flow of the robots [16], and manipulating an object too heavy for a single robot. Our control formulation enables efficient control of this kind of transport.

III. System Model

A. Architectures

We compare three n-robot system architectures with the following motion models:

$$\begin{bmatrix}\dot{x}_i\\ \dot{y}_i\\ \dot{\theta}_i\end{bmatrix} = \underbrace{\delta_{ai}\begin{bmatrix}v\cos(\theta_i)\\ v\sin(\theta_i)\\ \omega\end{bmatrix}}_{\text{Addressable}}, \underbrace{\begin{bmatrix}v\cos(\theta_i)\\ v\sin(\theta_i)\\ \epsilon_i\omega\end{bmatrix}}_{\text{Local}}, \underbrace{\begin{bmatrix}v\cos(\theta_i)\\ v\sin(\theta_i)\\ k\sin(\psi-\theta_i)\end{bmatrix}}_{\text{Global}} \quad (1)$$

The state of the ith robot is $[x_i, y_i, \theta_i] \in \mathbb{R}^3$ and the state of the system is $\in \mathbb{R}^{3n}$. This paper primarily focuses on the Global control architecture.

*1) Addressable control with independent inputs (*Addressable*):* This architecture has the finest level of control, but requires each robot to be addressable. There are three scalar inputs, the forward speed $v \in \mathbb{R}$, the angular turning rate $\omega \in \mathbb{R}$, and the *address* $a \in \{1, \ldots, n, \texttt{all}\}$. The ith robot only moves if the address is i or `all`. This system is fully controllable because each robot can be steered independently to a goal position and heading angle.

*2) Control with uniform inputs in the robot's local coordinate frame (*Local*):* The system has two scalar control inputs, the forward speed $v \in \mathbb{R}$ and the angular turning rate $\omega \in \mathbb{R}$. We proved in [8] that the position (but not the heading angle) of all the robots is controllable if each robot has a unique parameter ϵ_i that scales the turning rate.

*3) Control with uniform inputs in the global coordinate frame (*Global*):* The system has two scalar control inputs, the forward speed $v \in \mathbb{R}$ and the desired heading $\psi \in [0, 2\pi)$. The parameter k scales the turning command. In an obstacle-free workspace, this system is not controllable. The robots translate and turn at the same rate and so their final positions are the result of the same homogenous transformation being applied to the starting pose of each robot. However, as shown in Section III-D, adding a single obstacle is sufficient to break symmetry and control the final position of each robot.

B. BlockWorld Abstraction

We illustrate our points with a simplified BlockWorld abstraction. The workspace is a rectangular $m_1 \times m_2$ grid in which each square is marked either *free*, *fixed*, or *robot*. All *robots* are controlled by a shared input command from the set $\{\uparrow, \rightarrow, \leftarrow, \downarrow, \emptyset\}$, and can move horizontally and vertically

in the grid, as long as there are no *fixed* squares stopping the robot. The boundary of the grid is composed of *fixed* squares.

The general case of motion-planning in a world composed of even a single robot and both *fixed* and *moveable* squares is in the complexity class PSPACE-complete [17]. Adding an additional robot does not decrease this complexity: given any single-robot problem, we can place a second robot in the boundary of the world and surround it with *fixed* squares without changing the original problem's complexity. Still, there are many tractable subproblems. We will use this abstraction to present two representative algorithms, the first for removing all the robots from a specified region and the second for position control of every robot. Both algorithms require only a single fixed obstacle and we can explicitly state the space and number of commands required.

C. Clearing a rectangular region

In this section we provide an algorithm to remove all the robots from within an axis-aligned rectangular region A using a single, rectangular obstacle composed of *fixed* squares. The procedure consists of moving robots initially in A to the obstacle, sweeping the robots in a back-and-forth pattern past the obstacle, and returning this newly cleared area to A.

Given: an area to clear with bottom left corner at $[A_x, A_y]$ of width A_w and height A_h, and an obstacle with bottom left corner at $[O_x, O_y]$ of width O_w and height O_h.

Algorithm 1 CLEARREGION(A, O)

1: move $\leftarrow O_x - A_x$ ▷ move obstacle to bottom left
2: move $\downarrow\ O_y - A_y$
3: $c = 0$
4: **while** $c \cdot O_w < A_w$ **do**
5: move $\downarrow\ A_h$ ▷ clear column down
6: move $\leftarrow O_w$
7: $c = c + 1$
8: **if** $c \cdot O_w < A_w$ **then**
9: move $\uparrow\ A_h$ ▷ clear column up
10: move $\leftarrow O_w$
11: $c = c + 1$
12: **end if**
13: **end while**
14: move $\rightarrow A_x - O_x - c \cdot O_w$ ▷ return cleared area to A
15: move $\uparrow\ A_y - O_y$

Algorithm 1 requires space proportional to the area of A and O:

$$(|A_x - O_x| + A_w) \times (|(A_y + A_h) - (O_y + O_h)| + A_h)\,.$$

The total distance moved is linear in the area of A:

$$\lceil \frac{A_w}{O_w} \rceil (A_h - O_x) + A_x + 2|O_x - A_x| + 2|O_y - A_y|.$$

D. Position control

This section presents an algorithm to control the position of n robots using a single obstacle. We employ the BLOCK-WORLD abstraction, where the robots and the obstacle are unit squares. Each call to Algorithm 2 moves one robot from its starting position to its goal position.

a) Notation: The starting position of the kth robot in world coordinates is $k^W(0)$, its desired final position is k^W_{goal}, and its position at time t is $k^W(t)$. We define fixed-size, axis-aligned bounding boxes S and F such that $k^W(0) \in S^W(0)$ and $k^W_{\text{goal}} \in F^W(0)\ \forall k \in [1, n]$. The bottom left corners of S and F are $[S^W_x(t), S^W_y(t)]$ and $[F^W_x(t), F^W_y(t)]$, and are of width S_w, F_w and height S_h, F_h. Because all robots are identical, without loss of generality the robot indices are arranged in raster-scan order left-to-right, top-to-bottom in S and top-to-bottom, left-to-right in F. We note that the position of the kth robot may be specified in local reference frame: $k^W(t) = F^W(t) + k^F(t)$. The unmoving obstacle is located at $[O^W_x, O^W_y]$. We assume the obstacle position $O_{x,y}$, the starting positions $S_{x,y}$, and the final positions $F_{x,y}$ are disjoint. Without loss of generality, we will assume that S is to the lower right of the obstacle and F is to the upper left of the obstacle, as illustrated in Fig. 2.

b) Procedure: At the beginning of the kth call, the time is t, the bounding boxes S and F have been returned to their initial positions on opposite corners of O, the first $k-1$ robots have been moved to their proper positions in F, the remaining robots are in their original columns in S, and O is between S and F. The kth robot starts in position $[k^W_x(t), k^W_y(t)]$ and should be moved to $[k^W_{\text{goal},x}, k^W_{\text{goal},y}]$.

The algorithm consists in "popping" the kth robot out of the $S(t)$ bounding box (steps 1–3), pushing the kth robot to the correct x coordinate relative to $F_x(t)$ (steps 4–7), pushing the kth robot to the correct y coordinate relative to $F_y(t)$ (steps 8–10), and returning the S and F bounding boxes to their original positions on either side of O (steps 11–12).

The commanded distance to move the kth robot from $k^W(0)$ to the final destination k^W_{goal} is bounded by:

$$\text{Commanded distance}(k) \leq 2(2S_h + S_w + F_h + F_w + 2)$$

The total distance commanded for position control of n robots is the sum:

$$\begin{aligned}\text{Commanded distance} &= \sum_{k=1}^{n} \text{Commanded distance}(k) \\ &\leq 2n(2S_h + S_w + F_h + F_w + 2).\end{aligned}$$

c) Analysis: Algorithm 2 always requires $12n$ control switches. The worst-case running time for Algorithm 2 occurs when S and F are sparse and/or have large aspect ratios, and the algorithm runs in $O(n \cdot \max\{S_w, S_h, F_w, F_h\})$ time. For more reasonable arrays, when S and F are dense with aspect ratios near 1, the running time approaches $O(n\sqrt{n})$.

Algorithm 2 requires at least $S_w + F_w + 1$ free space to the left, $S_w + F_w$ to the right, $S_h + F_h + 1$ above, and $S_h + F_h$ below the obstacle:

$$(2S_h + 2F_h + 1) \times (2S_w + 2F_w + 1)\,.$$

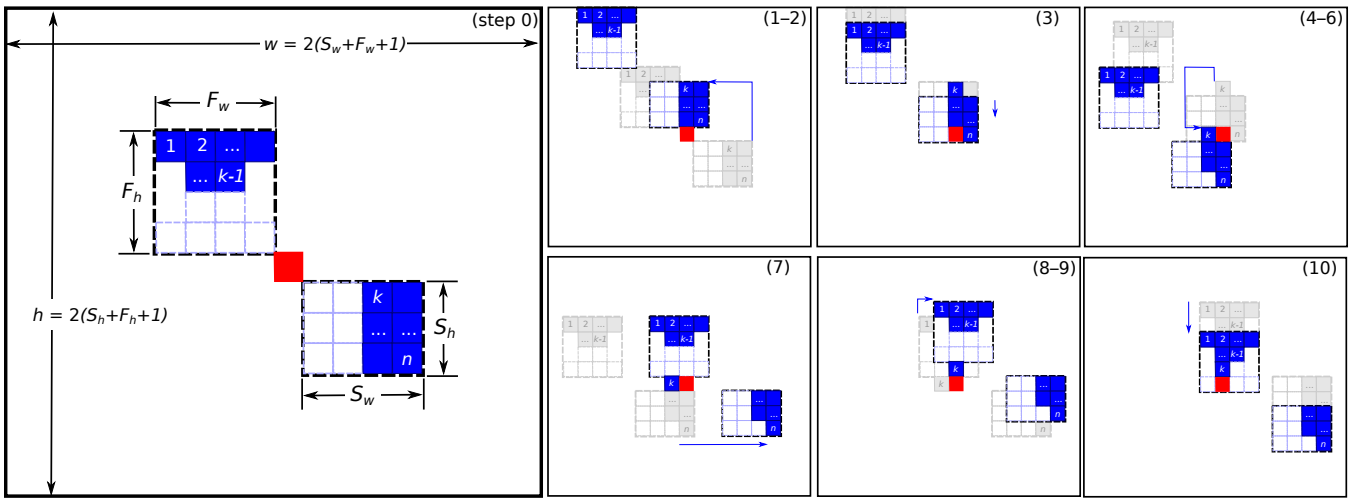

Fig. 2. A single rectangular obstacle is sufficient to enable position control of n robots. We provide an $O(n^2)$ algorithm to accomplish this. Shown above are frames from moving the kth robot into position. The robots are initially within the box at $S(t)$, which is of width S_w and height S_h. We want to move these robots to their final positions within a box at $F(t)$, which is of width F_w and height F_h and disjoint from $S(t)$. Given a simple square obstacle O, the algorithm requires at least $S_w + F_w + 1$ space on the left, $S_w + F_w$ on the right, $S_h + F_h + 1$ above, and $S_h + F_h$ below the obstacle.

Algorithm 2 POSITIONCONTROL(S, F, O, k)

1: move $\uparrow$ until $S_y^W(t) > O_y^W$
2: move $\leftarrow$ until $k_x^W(t) = O_x^W$
3: move $\downarrow$ until $k_y^W(t) > S_y^W(t) + S_h$
4: move $\uparrow$ until $S_y^W(t) > O_y^W$
5: move $\leftarrow$ until $S_x^W(t) < O_y^W - S_w$
6: move $\downarrow$ until $k_y^W(t) = O_y^W$
7: move $\rightarrow$ until $k_x^W(t) = F_{\text{goal},x} + k_{\text{goal},x}^F$
8: move $\uparrow$ 1
9: move $\rightarrow$ 1
10: move $\downarrow$ until $k_y^W(t) = F_{\text{goal},y}^W + k_{\text{goal},y}^F$
11: move $\uparrow$ until $F_y^W(t) > O_y^W$
12: move $\leftarrow$ until $F_x^W(t) < O_x^W - F_w$

d) Simulation: Simulation results are shown in Fig. 3 for five arrangements with an increasing number of robots. We compare the total distance moved and commanded with the *LAP distance*—the shortest distance according to the Linear Assignment Problem using Manhattan distance. Because all robots are interchangeable, the LAP distance reduces to

$$\text{LAP} = \sum_{k=1}^{n} \left|k_x^W(0) - k_{\text{goal},x}^W\right| + \left|k_y^W(0) - k_{\text{goal},y}^W\right|.$$

IV. EXPERIMENTAL VALIDATION

To demonstrate the feasibility of human control of many simple robots, we performed experiments on three platforms, the r-one (n=8), the kilobot (n=101), and a simulated environment (n=2000). We used the r-one robot [18] to contrast three different control architectures, and the Kilobot robot platform [19] to demonstrate manipulation tasks with large populations of robots. Finally, in simulations we applied the same control techniques to control thousands of robots.

A. User Study: Multi-Robot Manipulation Experiment

1) Objective: compare the three system architectures of Eqn. (1) (Section III-A) with hardware experiments using n robots to manipulate objects to goal positions and orientations.

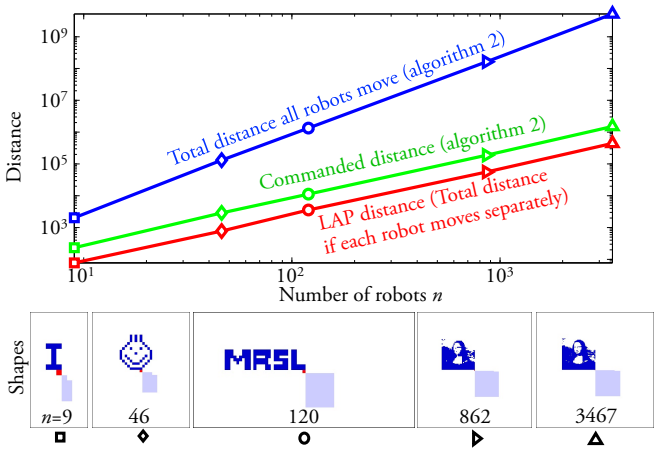

Fig. 3. The required number of moves using Algorithm 2 employing a single square obstacle to rearrange n square-shaped robots. The log-log plot compares *Total distance*—the sum of the moves made by every robot, with *Commanded distance*—the distance broadcast to all robots, and with *LAP distance*—the shortest distance according to the Linear Assignment Problem using Manhattan distance. In each metric moving $\{\uparrow, \rightarrow, \leftarrow, \downarrow\}$ one unit counts as one move. These were calculated for the five patterns shown above. Dark blue is the target position, red is the obstacle, and light blue is the initial configuration. The outline shows the minimum required free space for the algorithm. See hardware implementation and simulation at http://youtu.be/5p_XIad5-Cw. Code available at http://www.mathworks.com/matlabcentral/fileexchange/42889.

2) Hypothesis: the GLOBAL technique will result in lower completion times than ADDRESSABLE and LOCAL. Moreover, as the number of robots grows, the completion times for each technique scale differently as the number of robots is increased.

3) Equipment: This experiment used 1 to 8 r-one robots [18]. The r-one is a low-cost, open-hardware, differential-drive robot with a 10 cm diameter circular profile shown in Fig. 4.

For each technique, the inputs are velocity and turning rate (by tilting the controller forward/backwards to control velocity and tilting it side to side to control turning rate.)

(a) r-one robot (b) Manipulation experiment with Velcro-skirt grippers

Fig. 4. **(a)** (foreground) The r-one differential-drive mobile robot used for manipulation experiments (background) many r-one robots. **(b)** Manipulation experiments. Each robot is wearing a circular skirt covered in Velcro fasteners. The object has the alternate type of fastener, so the robots remain attached to the object after they collide with it. These experiments demonstrate the feasibility of multi-robot manipulation with simple robots and uniform inputs.

The inputs are low-pass filtered to provide a slow control.

There are two differences from Eqn. (1) due to hardware constraints: 1.) Under LOCAL control, each robot receives the same inputs. Over time, due to robot and environment differences, the robots will not be aligned. This significant process noise obviates the need to provide unique ϵ_i values to each robot. 2.) The r-one robots do not have global heading information. To emulate a global input, we send the same velocity and turning commands to all the robots, and augment this control with a distributed algorithm where each robot turns to align itself with the average of its neighbor's orientation [20]. This algorithm is simple to implement, and allows us to easily change the number of robots used from experiment to experiment.

4) Task: Using each controller technique, a human user will steer the robots from a starting position in the bottom left side of the environment, to move an object to the goal region (a position and an orientation). The experimenter records the position/orientation of the object using a tracking system and the time required for each task.

5) Results: The completion times are shown in Fig. 5, and representative paths from one user in Fig. 6. Each shape was tested with at least 5 subjects. GLOBAL had the shortest mean completion times, while LOCAL had the longest mean completion times for experiments with 5 and 8 robots.

Observing test subjects strive to control the robots led to interesting conclusions. With ADDRESSABLE, the subjects would align all the robots in a vector straight to the goal, then enable them all to drive the robot to the target position, then turn them to align tangent to the shape to correct the orientation. The time to switch between robots was the largest limitation to this approach and became increasingly costly as the number of robots increased. With the GLOBAL control, the subjects could very quickly bring the object to the desired position—much faster than with ADDRESSABLE—but controlling the orientation was difficult because our arena was a rectangular shape with no convex obstacles to pivot the object. Instead our test subjects relied on dragging the object. The sum of the forces often created a moment about the object, allowing the object to slowly spin. The LOCAL control took the greatest amount of time with more robots, requiring almost 40 minutes to push a 1 m object a 2 m distance in two trials of the 8-robot case. The users were eventually successful, by iterating between rotating the robots in place with alternating forwards and backwards velocity inputs. Occasionally the forward and backward inputs opportunistically result in the object moving in the general direction of the goal, and our test subjects would continue this input until the object stopped progress.

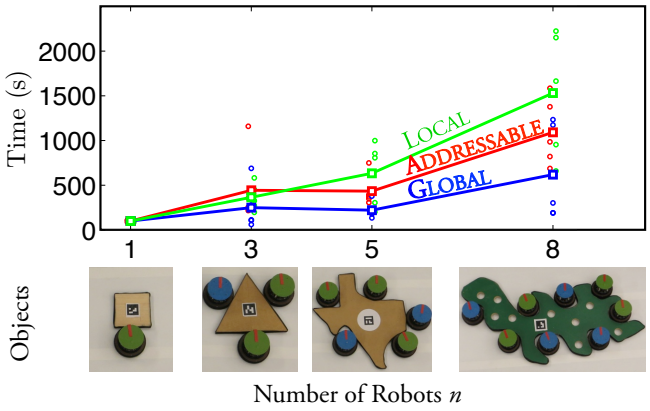

Fig. 5. We measured the completion time for human users using the three control techniques in Eqn. (1). Users controlled $n = 1$ robots to manipulate the box shape, 3 for the triangle, 5 for Texas, and 8 for the amoeba shape. GLOBAL had the shortest mean completion times overall and LOCAL the longest for experiments with 5 and 8 robots. Also shown are the means and the final times for each trial (the same subject tested all three control techniques, and each shape was tested with at least 5 subjects).

B. Hardware Demonstrations: Large Robot Populations

The Kilobot [19] is a low-cost robot that allows one to easily test collective algorithms with large numbers of robots. It is available as an open source platform and is also commercially available. Each robot is approximately 3 cm in diameter and 3 cm tall, and uses two vibration motors to move on a flat surface at speeds up to 1 cm/s. Each robot has a single ambient light sensor which is used to implement phototaxis (i.e., moving towards a light source).

Phototaxis is implemented by turning on one of the two vibration motors, which causes the robot to slowly move forward while turning. Since the ambient light sensor is located on the back of the robot, this rotation will cause the sensed light value to decrease until the robot is pointed in the direction of the light source, at which point any further rotation will cause the sensed light value to increase. Once the light value begins to increase, the robot switches which motor is active, causing the robot to rotate in the opposite direction. The process of rotating until the sensed light value increases and then switching directions is run continually, which causes the robot to move towards the light source.

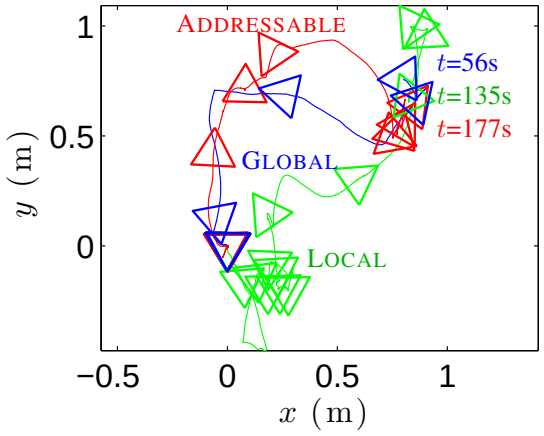

Fig. 6. Representative paths of the triangle object, which is manipulated by $n = 3$ robots to a goal position and orientation at $(0.9, 0.6)$ m. The same human subject repeated the task with three controllers: ADDRESSABLE in red, LOCAL in green, and GLOBAL in blue. Outlines of the object every 10s reveal that the LOCAL controller had the longest and least smooth path. GLOBAL control makes steady progress to the goal, while ADDRESSABLE stops forward movement periodically to rotate each robot towards the goal.

The following tests in Figs. 7 and 9 were conducted with n=101 kilobots, a 1.2×1.2 m whiteboard as our workspace, using either a set of five lights arranged at the $\{W, SW, S, SW, E\}$ vertices of a 4 m square centered on the workspace, or (for the *Assembly* task) moving a light along the circumference of a 3 m radius circle centered on the workspace.

1) 1D and 2D compression: Compression tasks use the environmental obstacles to reduce the position variance. By driving the robots into a vertical boundary, we can reduce the horizontal variance. Driving into a horizontal boundary reduces the vertical variance. Iterating between the two reduces both variances.

2) Dispersion: Dispersion separates the robots to raise the position variance. This requires a more complex workspace. We use a *Galton array*, an arrangement of circular obstacles in interleaved rows with spacing between obstacles sufficient so that only one robot can pass through at a time. Robots enter from the top, and bounce stochastically left or right as they collide with obstacles. The probability distribution for the resulting spread of robots can be computed analytically—Pascal's triangle provides the number of paths to each subsequent row. Experimental results are shown in Fig. 8.

3) Object manipulation with obstacles: In this test a single object was pushed to a goal region at the end of an S-shaped maze. The object was weighted to require multiple robots pushing to break static friction. Control consisted of alternating between a) actively pushing the object and b) regrouping sufficient robots behind the object to return to active pushing.

4) Object manipulation with orientation control: To rotate an object, the robots must exert a force with a line of action offset from the object's center of mass. In this test a single object was steered to a desired position and orientation.

5) Assembly: The robots were initialized uniformly-randomly in the workspace. Two components of a compound

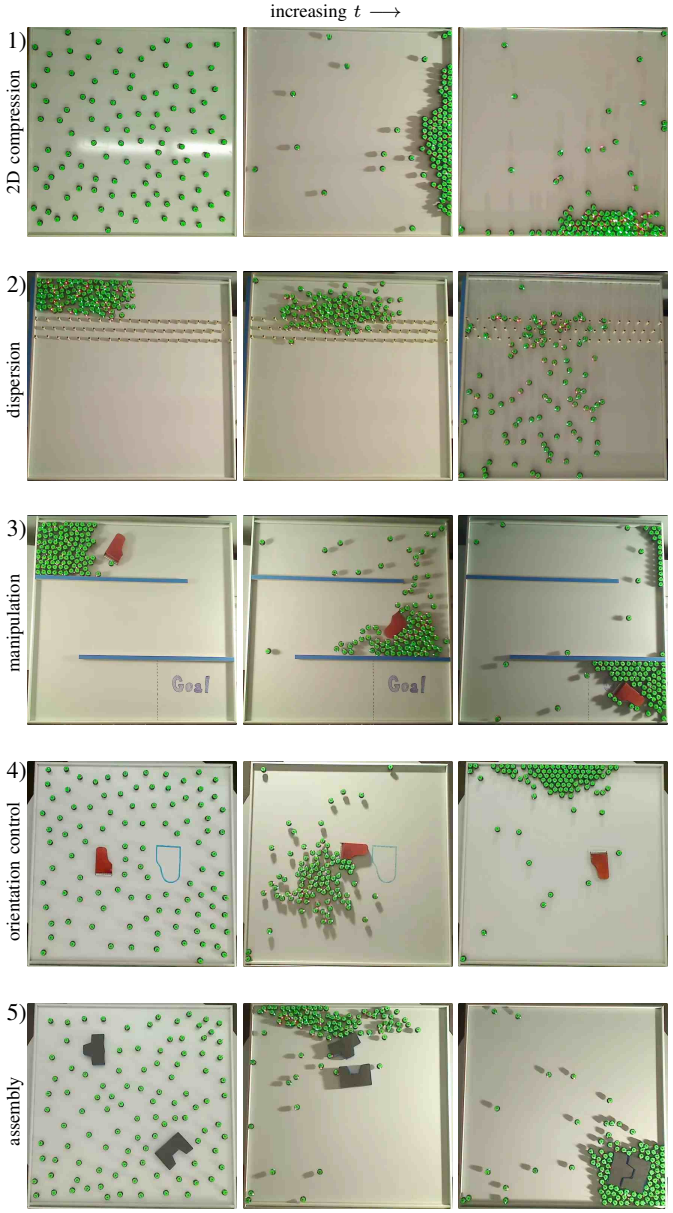

Fig. 7. Five tasks with 101 Kilobots: 2D compression, dispersion, manipulating an object through a maze, manipulation with orientation control, and assembly. In each task the light source's position was the only input to the robot population. This input is robust to individual robot failures—during each test several robots failed due to glitches, poor calibration, or low batteries—but each demonstration was ultimately successful.

object, each requiring several robots to manipulate, both required reorientation before they could be assembled. Control consisted of preparing both parts for mating, pushing the objects together, and finally delivering the assembly to the goal location.

6) Maze navigation: In this experiment we steered varying numbers of robots through an S-shaped maze and recorded the number of robots at the goal position as a function of time. Fig. 9 summarizes the results. This experiment indicates that there is an additional cost in time to steer additional robots through a maze, but the growth appears to be much less than linear. In single trials, $n = 10$ required 2.6× as much time as $n = 1$, and $n = 100$ required only 3.7× as much time as $n = 1$.

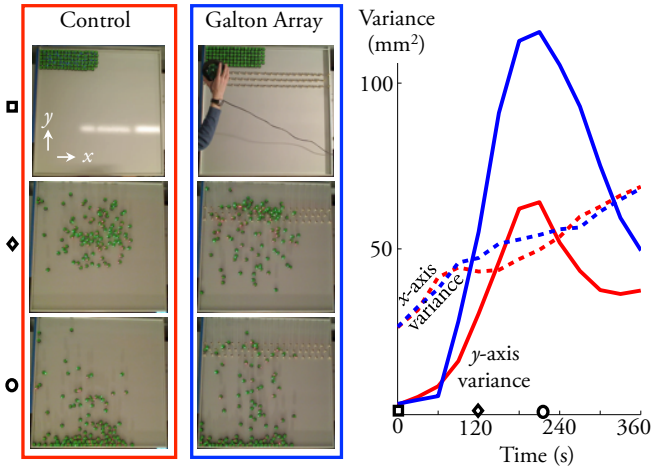

Fig. 8. Comparison of variance as a function of time for 101 Kilobot robots with and without circular obstacles in interleaved rows (a Galton array). With the obstacles the maximum y-axis variance of the group is twice the maximum y-axis variance without obstacles.

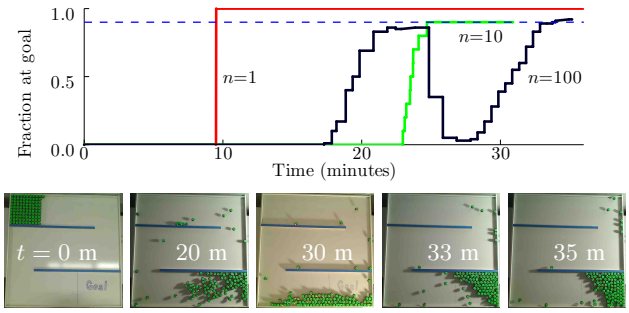

Fig. 9. Navigating a simple maze with $n = \{1, 10, 100\}$ Kilobot robots. The input selects one of five lights arranged at the $\{W, SW, S, SW, E\}$ vertices of a 4 m square centered on 1.2×1.2 m workspace. Robots started in the NW corner and the trial was ended when 90% of the robots reached the goal region in the SE corner. $n = 1$ required 9m31s, $n = 10$ 24m52s, and $n = 100$ 35m05s. Top: fraction at goal as a function of time, bottom: frames from the 100-robot test at different times. For n=100, 88 robots reached the goal within 22 minutes, but the remaining robots were stuck behind an obstacle. To achieve 90%, the group had to first be retracted.

C. Feasibility Demonstrations in Simulation

In this section we use simulations to demonstrate that, unlike the ADDRESSIBLE or LOCAL control techniques, GLOBAL control scales in a fashion manageable by a human user. We implemented GLOBAL control of large numbers of robots using the 2D physics engine Box2d (`http://box2d.org`) to simulate phototaxis behavior on each robot. Fig. 10 shows frames from a test using 2,000 robots initially uniformly randomly distributed in the workspace. In a short sequence of commands, the human subject demonstrated a 2D compression task using a workspace corner to cluster all the robots into a rectangular clump. The subject then treated this clump as a single component and steered the robots collectively above an array of interleaved obstacles. The subject used these workspace obstacles to disperse the robots through the workspace.

In Fig. 11, a human user steers 200 disk-shaped robots to assemble a structure composed of 5 blocks. Robots conform to the manipulated object, aiding fine control (see video).

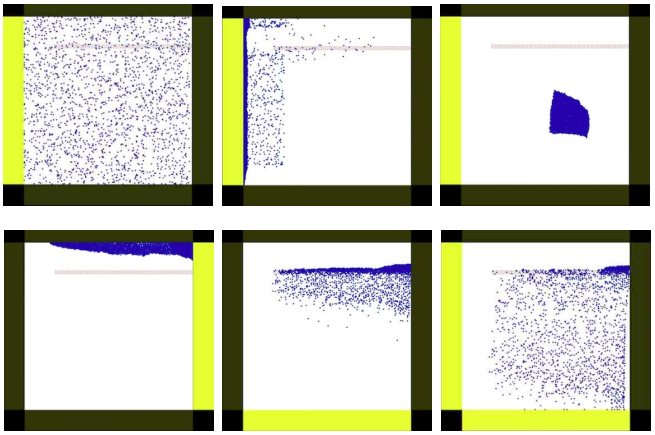

Fig. 10. Human-controlled simulation using 2,000 disk-shaped robots (blue) under the GLOBAL control law. Yellow bars represent lights that attract the robots {north, south, east, west}. The top row demonstrates a compression task using a workspace corner to cluster all the robots into a rectangular clump. The user can then treat this clump as a single component. In the bottom row, the user disperses the robots with an array of interleaved obstacles.

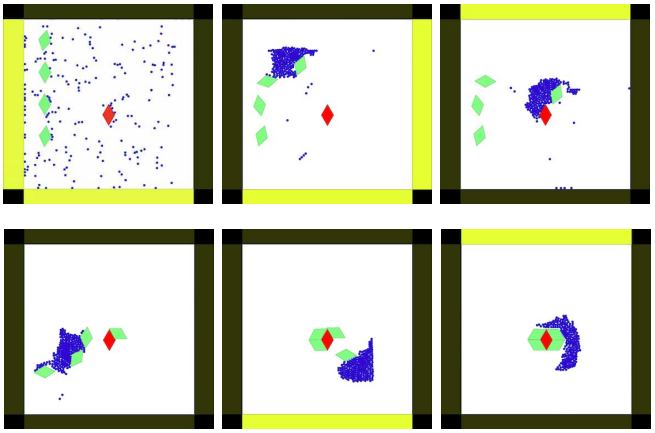

Fig. 11. Human-controlled simulation using 200 disk-shaped robots (blue) under the GLOBAL control law to assemble a specified structure from four movable blocks (red) and one fixed block (red). The robots conform to the object shape. If the robot density was insufficient to apply the desired force on an object, the user re-grouped the robots using one or more of the walls.

V. CONCLUSION

In this paper we investigated control of mobile robots that move in a 2D workspace using three different system models. Table I compares theoretical performance of these models for several tasks in $O(\cdot)$ notation. A key feature is that compression, unison movement, and dispersion can be accomplished under GLOBAL control by commanding a vector direction into the appropriate obstacle, and require sensing only the bounding box (4 scalars), while the demands of other models almost uniformly scale with the number of robots.

We focused on the GLOBAL model using broadcast control inputs specified in the global reference frame. This system model, in an obstacle-free workspace, is uncontrollable because it has only two controllable degrees of freedom. We proved that adding a single obstacle makes the system controllable for any number of robots. Additionally, we showed

TABLE I

TASK COMPLEXITY FOR n ROBOTS AND THE THREE CONTROL ARCHITECTURES OF EQN. (1). MANIPULATION IS COMPARED FOR A PATH WITH m STRAIGHT-LINE SEGMENTS.

	Commands $O(\cdot)$			Sensing		
Task	ADDRESSABLE	LOCAL	GLOBAL	ADDRESSABLE	LOCAL	GLOBAL
position control	n	n	n	$3n$	$3n$	$2n$
compression	n	n	1	$3n$	$3n$	4
move in unison	n	n	1	0	$3n$	0
dispersion	n	n	1	$3n$	$3n$	4
manipulation	nm	nm	m	$2n$	$3n$	4
form closure	n	n	1	$3n$	$3n$	$2n$
force closure	n	n	N/A	$3n$	$3n$	N/A

that motion-planning in BLOCKWORLD with multiple robots and *fixed* and *moveable* squares is PSPACE-complete. An open problem is determining the complexity of motion planning with multiple robots and only *fixed* squares. Our algorithm for position control uses a single obstacle. This algorithm concentrates the task complexity into the (large) sequence of moves. An alternative approach is to design complex environments that can enable position control with a small number of moves. We will explore this design space of environmental vs. movement-sequence complexity in future work.

We demonstrated through extensive testing with human subjects that many manipulation tasks can be reliably completed, even by novice users, under this system model. Results were validated through hardware experiments using over 100 robots, and simulations with large populations of robots.

There are a number of things we cannot do with the GLOBAL model of Eqn. (1). If two robots are in identical environments, and start in identical states, they cannot be differentiated. The same control sequence applied to each will produce the same results. We note that this only applies to deterministic systems. Stochastic systems, such as the interleaved array of obstacles in Fig 1, *can* differentiate identical robots. Force closure on an object is a second impossible task, but we demonstrated that equipping robots to adhere to an object enables them to steer the object along a trajectory. Our hardware experiments illustrated that orientation control in a concave boundary with no interior obstacles is difficult. In a frictionless environment, orientation control would be impossible. Similarly, we cannot simultaneously generate velocities in different directions.

Many of the tasks we demonstrated could be combined for tasks that require large populations such as (1) search and coverage tasks, (2) distributed object manipulation and assembly, (3) parallel procurement and delivery. Future work should expand our experiments through large-scale online simulators, enabling larger populations of robots and refined experimental control. We want to explore the level of state feedback required to complete a task—could a trained user steer a million robots through a maze using only the mean and variance of the position distribution? Would the bounding-box of the population be sufficient?

VI. ACKNOWLEDGEMENTS

The authors thank Alex Cornejo for providing software and expertise to track the Kilobots. This work was supported by the National Science Foundation under CPS-1035716.

REFERENCES

[1] A. Sudsang, F. Rothganger, and J. Ponce, "Motion planning for disc-shaped robots pushing a polygonal object in the plane," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 550–562, Aug. 2002.

[2] P.-T. Chiang, J. Mielke, J. Godoy, J. M. Guerrero, L. B. Alemany, C. J. Villagómez, A. Saywell, L. Grill, and J. M. Tour, "Toward a light-driven motorized nanocar: Synthesis and initial imaging of single molecules," *ACS Nano*, vol. 6, no. 1, pp. 592–597, Feb. 2011.

[3] B. Donald, C. Levey, C. McGray, I. Paprotny, and D. Rus, "An untethered, electrostatic, globally controllable MEMS micro-robot," *J. of MEMS*, vol. 15, no. 1, pp. 1–15, Feb. 2006.

[4] B. Donald, C. Levey, and I. Paprotny, "Planar microassembly by parallel actuation of MEMS microrobots," *J. of MEMS*, vol. 17, no. 4, pp. 789–808, Aug. 2008.

[5] S. Floyd, E. Diller, C. Pawashe, and M. Sitti, "Control methodologies for a heterogeneous group of untethered magnetic micro-robots," *Int. J. Robot. Res.*, vol. 30, no. 13, pp. 1553–1565, Nov. 2011.

[6] D. Frutiger, B. Kratochvil, K. Vollmers, and B. J. Nelson, "Magmites - wireless resonant magnetic microrobots," in *IEEE Int. Conf. Rob. Aut.*, Pasadena, CA, May 2008.

[7] K. E. Peyer, L. Zhang, and B. J. Nelson, "Bio-inspired magnetic swimming microrobots for biomedical applications," *Nanoscale*, 2013.

[8] A. Becker, C. Onyuksel, and T. Bretl, "Feedback control of many differential-drive robots with uniform control inputs," in *IEEE Int. Rob. and Sys.*, Oct. 2012.

[9] A. Becker and T. Bretl, "Approximate steering of a unicycle under bounded model perturbation using ensemble control," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 580–591, Jun. 2012.

[10] K. Lynch, "Locally controllable manipulation by stable pushing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 318–327, Apr. 1999.

[11] S. Tottori, L. Zhang, F. Qiu, K. Krawczyk, A. Franco-Obregón, and B. J. Nelson, "Magnetic helical micromachines: Fabrication, controlled swimming, and cargo transport," *Advanced Materials*, vol. 24, no. 811, 2012.

[12] Y. Shirai, A. J. Osgood, Y. Zhao, K. F. Kelly, and J. M. Tour, "Directional control in thermally driven single-molecule nanocars," *Nano Letters*, vol. 5, no. 11, pp. 2330–2334, Feb. 2005.

[13] 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.*, May 2006, pp. 1408–1413. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1641906&isnumber=34383

[14] O. C. Goemans, K. Goldberg, and A. F. van der Stappen, "Blades: a new class of geometric primitives for feeding 3d parts on vibratory tracks," in *Int. Conf. Rob. Aut.*, May 2006, pp. 1730–1736.

[15] T. Vose, P. Umbanhowar, and K. Lynch, "Friction-induced velocity fields for point parts sliding on a rigid oscillated plate," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 1020–1039, 2009. [Online]. Available: http://ijr.sagepub.com/content/28/8/1020.abstract

[16] K. Sugawara, N. Correll, and D. Reishus, "Object transportation by granular convection using swarm robots," in *Distributed Algorithmic Robotics*, Nov. 2012.

[17] E. D. Demaine and R. A. Hearn, *Games of No Chance 3*. Mathematical Sciences Research Institute Publications, Cambridge University Press, 2009, vol. 56, ch. Playing Games with Algorithms: Algorithmic Combinatorial Game Theory, pp. 3–56. [Online]. Available: http://arXiv.org/abs/cs.CC/0106019

[18] J. McLurkin, A. Lynch, S. Rixner, T. Barr, A. Chou, K. Foster, and S. Bilstein, "A low-cost multi-robot system for research, teaching, and outreach," *Distributed Autonomous Robotic Systems*, pp. 597–609, 2010.

[19] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *IEEE Int. Conf. Rob. Aut.*, May 2012, pp. 3293–3298.

[20] R. Olfati-Saber, J. Fax, and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.