2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
September 24–28, 2017, Vancouver, BC, Canada

Algorithms for Shaping a Particle Swarm with a Shared Input by Exploiting Non-Slip Wall Contacts*

Shiva Shahrokhi, Arun Mahadev, and Aaron T. Becker

Abstract— **There are driving applications for large populations of tiny robots in robotics, biology, and chemistry. These robots often lack onboard computation, actuation, and communication. Instead, these "robots" are particles carrying some payload and the particle swarm is controlled by a shared control input such as a uniform magnetic gradient or electric field. In previous works, we showed that the 2D position of each particle in such a swarm is controllable if the workspace contains a single obstacle the size of one particle.**

Requiring a small, rigid obstacle suspended in the middle of the workspace is a strong constraint, especially in 3D. This paper relaxes that constraint, and provides position control algorithms that only require non-slip wall contact in 2D. Both in vivo and artificial environments often have such boundaries. We assume that particles in contact with the boundaries have zero velocity if the shared control input pushes the particle into the wall. This paper provides a shortest-path algorithm for positioning a two-particle swarm, and a generalization to positioning an n-particle swarm. Results are validated with simulations and a hardware demonstration.

I. Introduction

Particle swarms propelled by a uniform field, where each particle receives the same control input, are common in applied mathematics, biology, and computer graphics. As a current example, micro- and nano-robots can be manufactured in large numbers, see [1]–[7]. Someday large swarms of robots will be remotely guided to assemble structures in parallel and through the human body to cure disease, heal tissue, and prevent infection. For each task, large numbers of micro robots are required to deliver sufficient payloads, but the small size of these robots makes it difficult to perform onboard computation. Instead, these robots are often controlled by a broadcast signal. The tiny robots themselves are often just rigid bodies, and it may be more accurate to define the *system*, consisting of particles, a uniform control field, and sensing, as the robot. Such systems are severely underactuated, having 2 degrees of freedom in the shared control input, but $2n$ degrees of freedom for the particle swarm. Techniques are needed that can handle this underactuation. In previous work, we showed that the 2D position of each particle in such a swarm is controllable if the workspace contains a single obstacle the size of one particle.

Positioning is a foundational capability for a robotic system, e.g. placement of brachytherapy seeds. However, requiring a single, small, rigid obstacle suspended in the middle of the workspace is often an unreasonable constraint, especially in 3D. This paper relaxes that constraint, and provides position control algorithms that only require non-slip wall contacts. We assume that particles in contact with the boundaries have zero velocity if the uniform control input pushes the particle into the wall.

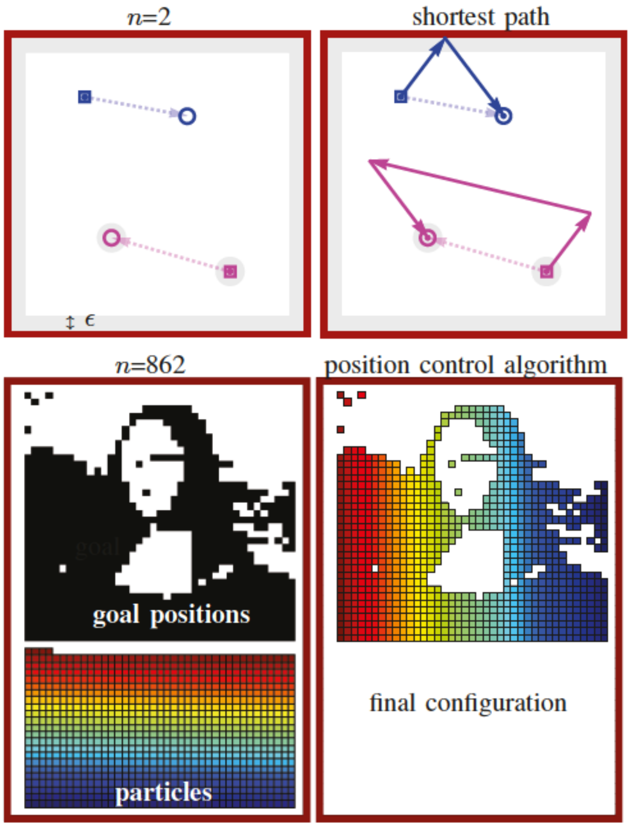


Fig. 1. Positioning particles that receive the same control inputs, but cannot move while a control input pushes them into a boundary. Top row shows the output of a best-first-search algorithm that finds the shortest path for two particles. Top left shows the initial position and goal position of the particles. The shortest path consists of moving at angle 50° until the blue robot contacts the top wall, then moving the magenta robot at angle 165° until the particles reach the desired relative spacing, then moving −60° to the goal positions. The two bottom pictures show n-particle positioning using shared control inputs and boundary interaction.

The paper is arranged as follows. After a review of recent related work in Sec. II, Sec. III introduces a model for boundary interaction. We provide a shortest-path algorithm to arbitrarily position two robots in Sec. IV, and Sec. V extends this to prove a rectangular workspace with non-slip boundaries can position a swarm of n robots arbitrarily within a subset of the workspace. Sec. VI describes im-

*This work was supported by the National Science Foundation under Grant No. [IIS-1553063] and [IIS-1619278].

Authors are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204 USA {sshahrokhi2, aviswanathanmahadev, atbecker}@uh.edu

978-1-5386-2682-5/17/$31.00 ©2017 IEEE

e en a ion o he a ori h in i ua ion and Sec. de cri e hard are e eri en , a ho n in i . . e end i h direc ion or u ure re earch in Sec. .

. AT

on ro in he *shape*, or re a ive o i ion , o a ar o ro o i a ke a i i or a ran e o a ica ion . orre ondin , i ha een udied ro a con ro heore ic er ec ive in o h cen ra i ed and decen ra i ed a roache . or e a e o each, ee he cen ra i ed vir ua ead er in , and he radien a ed decen ra i ed con ro er u in con ro a unov unc ion in . o ever, he e a roache a u e a eve o in e i ence and au ono in individua ro o ha e ceed he ca a i i ie o an e , inc udin curren icro and nano ro o . urren icro and nano ro o , uch a ho e in , , ack on oard co u a ion.

n ead, hi a er ocu e on cen ra i ed echni ue ha a he a e con ro in u o each e er o he ar . reci ion con ro re uire reakin he e r cau ed he uni or in u . S e r can e roken u in a en ha re ond di eren o he uni or con ro i na , ei her hrou h a en a en reac ion , ee ork ode in io o ica ar , or en ineered inho o enei , , . Thi ork a u e a uni or con ro i h ho o enou a en , a in . The echni ue in hi a er are in ired ar i cia orce e d .

Artificial Force-fields: Much re earch ha ocu ed on en era in non uni or ar i cia orce e d ha can e u ed o rearran e a ive co onen . A ica ion have inc uded echni ue o de i n hear orce or en or e ani u a ion o a in e o ec . de on ra ed a co ec ion o orce e d enera ed i de ree o reedo vi ra ion in u o a ri id a e. The e orce e d , inc udin hear orce , cou d e u ed a a e o ri i ive or o ion con ro o eer he or a ion o u i e o ec . o ever un ike he uni or con ro ode in hi a er, heir con ro a u i oda and o i ion de enden .

. T

Using Boundaries: Friction and Boundary Layers

n he a ence o o ac e uni or in u ove a ar iden ica . Sha e con ro re uire reakin hi e r . The o o in ec ion e a ine u in non i oundar con ac o reak he e r cau ed uni or in u .

he i^{h} ar ic e ha o i ion $\mathbf{x}_i\ t$ and ve oci $\mathbf{x}_i\ t$, e a u e he o o in e ode

$$\mathbf{x}_i\ t \quad \mathbf{u}\ t \quad F\ \mathbf{x}_i\ t\ ,\mathbf{u}\ t\ \ ,\ i\in\ \ ,n\ .$$

$$F\ \mathbf{x}_i\ t\ ,\mathbf{u}\ t \quad \begin{cases} -\mathbf{u}\ t & \mathbf{x}_i\ t\ \in\ \text{oundar}\ \textbf{and} \\ & \mathbf{N}\ \ \text{oundar}\ \ \mathbf{x}_i\ t\ \ \cdot\mathbf{u}\ t\ \le \\ & \text{e e} \end{cases}$$

ere $\mathbf{N}$ oundar $\mathbf{x}_i\ t$ i he nor a o he oundar a o i ion $\mathbf{x}_i\ t$, and $F\ \mathbf{x}_i\ t\ ,\mathbf{u}\ t$ i he ric iona orce rovided he oundar .

The e e d na ic re re en ar ic e ar in o e no d nu er environ en , here vi co i do ina e iner ia orce and o ve oci i ro or iona o in u orce . n hi re i e, he in u orce co and $\mathbf{u}\ t$ con ro he ve oci o he ro o . The a e ode can e enera i ed o ar ic e oved uid o here he vec or direc ion o uid o $\mathbf{u}\ t$ con ro he ve oci o ar ic e , or or a ar o ro o ha ove a a con an eed in a direc ion eci ed a uni or in u $\mathbf{u}\ t$. A in our ode , uid o in in a i e ha ero ve oci a on he oundar . Si i ar echanica e e i a ar er ca e , e. . a u er o a co ina ion ook ove uni or une o ruc ed an o ac e. ur con ro ro e i o de i n he con ro in u $\mathbf{u}\ t$ o ake a n ar ic e achieve a a k.

. S T T T B TS S B A T A T

A . u e non i con ac i h a o ar i rari o i ion o ro o in a rec an u ar ork ace. i . ho a Ma he a ica i e en a ion o he a ori h , and i u e u a a vi ua re erence or he o o in de cri ion.

A u e o ro o are ini ia i ed a r_1 and r_2 i h corre ondin oa de ina ion g_1 and g_2. The o u ion i a e r earch a ori h ha ain ain a i o o i e a h *moves* , or ed accordin o an ad i i e heuri ic on a h en h. The a ori h ork e andin he a h i h he hor e e i a ed en h. andin a a h ean ei her ovin direc o he oa , or u hin one ro o o a a and ad u in he re a ive o i ion o he o her ro o . A oon a he oa i reached, he a ori h re urn hi hor e a h.

eno e he curren o i ion o he ro o r_1 and r_2. a ue $.x$ and $.y$ deno e he x and y coordina e , i.e., $r_1.x$ and $r_1.y$ deno e he x and y oca ion o r_1. The a ori h a i n a uni or con ro in u a ever in ance. The oa i o ero he error in o h coordina e , $e \quad e.x,\ \ e.y$ $g - \ r \quad g_2 - g_1 \ - \ r_2 - r_1$ u in a hared con ro in u . The a e ca e occur hen e , . n hi a e ca e, he hor e a h i a rai h ine e een he curren o i ion o each ro o o i oa o i ion, a ho n in i . a. $e\,/$, , e co u e he o ove reacha e e enera ed r i o i i in one ar ic e con ac in he a and hen re o i ionin he econd ar ic e. There are our reacha e e , a ho n in i . . The hori on a and ver ica reacha e e are e uiva en in he con ura ion ace, o e can an in hi ace and choo e o i o i i e he ar ic e c o e o a a . ach non i ove chan e he con ura ion ace, a ho n in i . .

he oa con ura ion can e reached in hree ove , hen m_1 ake one ar ic e hi a a , m_2 ad u he re a ive acin error e o ero, and m_3 ake he ar ic e o heir na o i ion , a ho n in i . . m_2 canno e hor ened, o o i i a ion de end on choo in he oca ion here he ro o hi he a . Since he hor e di ance e een o oin i a rai h ine, re ec in he oa o i ion acro he oundar a and o in a rai h ine ive he o i a

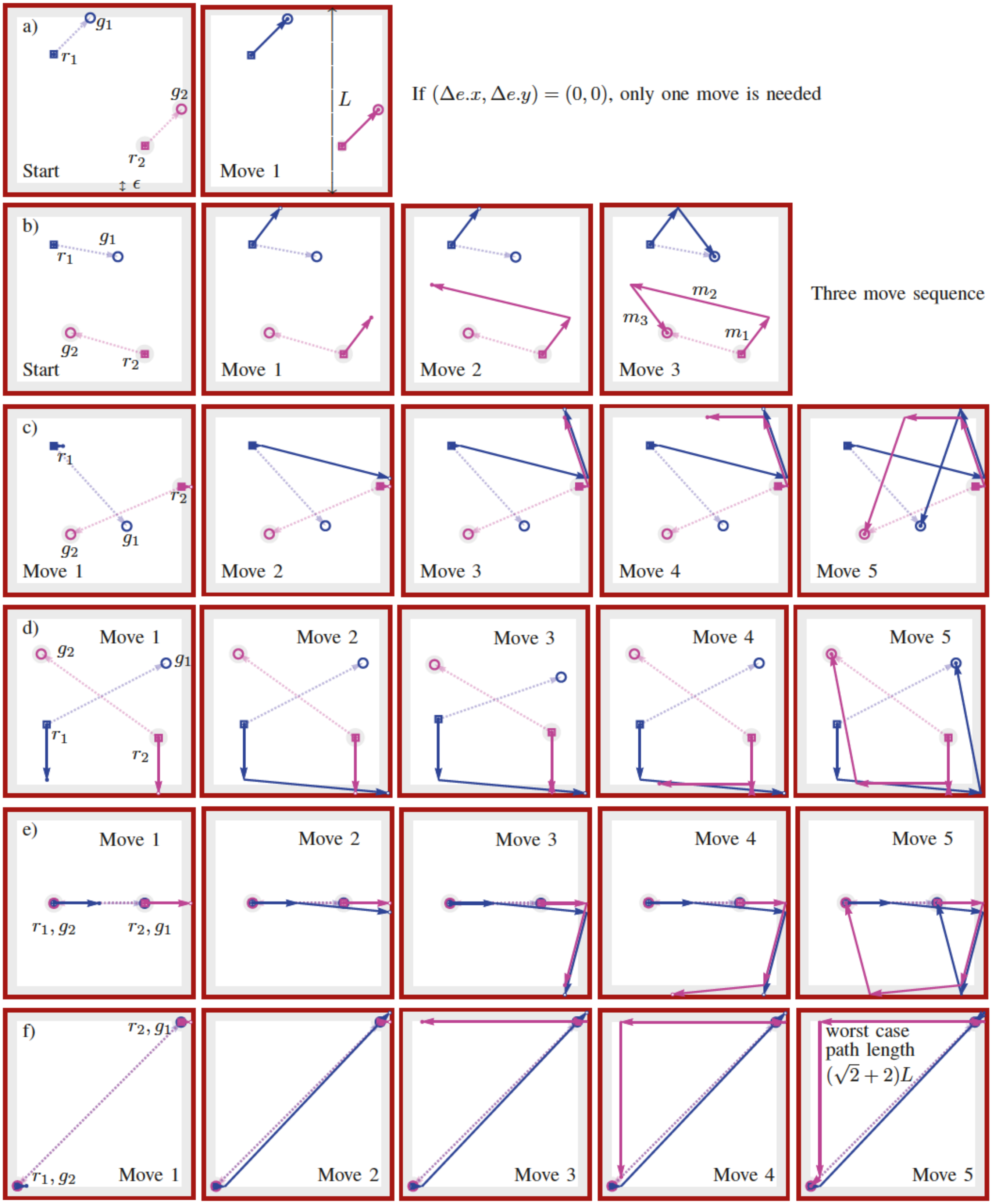


Fig. 2. Frames from an implementation of Alg. 1: two robot positioning using walls with non-slip contacts. Robot start positions are shown by a square, and goal positions by a circle. Dashed lines show the shortest route if robots could be controlled independently. Solid arrows show path given by Alg. 1. Online demonstration and source code at [20].

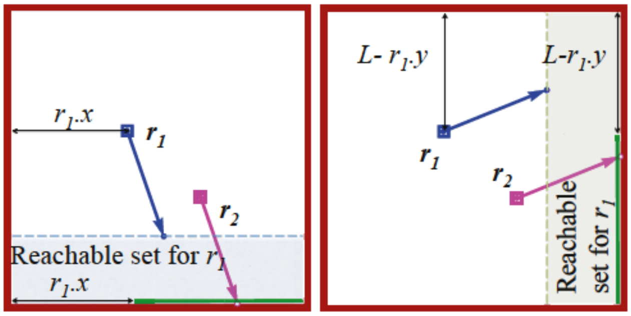


Fig. 3. Boundary interaction is used to change the relative positions of the robots. Each robot gets the same control input. (left) If robot 2 hits the bottom wall before robot 1 reaches a wall, robot 2 can reach anywhere along the green line, and robot 1 can move to anywhere in the shaded area. (right) Similarly, if robot 2 hits the right wall before robot 1 reaches a wall, robot 2 can reach anywhere along the green line, and robot 1 can move to anywhere in the shaded area.

hit location, as shown in Fig. 5. That point is selected when possible, but if this point would cause m_2 to push the moving robot out of the workspace, the hit point is translated until the moving robot will not leave the workspace. If m_2 causes the two particles to overlap, we add or subtract ϵ to $m_2.x$ to avoid collisions. This is shown in Fig. 6 with three different ϵ values.

If Δg is not in the reachable set, we choose the nearest reachable Δx and Δy to Δg.

Alg. 1 uses an admissible heuristic that adds the current path length to the greatest distance from each robot to their goal. This heuristic directs exploration by expanding favorable routes first.

$$h(moves, r_1, r_2, g_1, g_2) = \sum_{i=1}^{|moves|} \|moves_i\| \\ + \max(\|g_1 - r_1\|, \|g_2 - r_2\|) \qquad (2)$$

We exploit symmetry in the solution by labeling the leftmost (or, if they have the same x coordinate, the topmost) robot r_1. If r_1 is not also the topmost robot, we mirror the coordinate frame about the right boundary. As an example, consider the two starting positions, $r_1 = (0.2, 0.2)$ and $r_2 = (0.8, 0.8)$. Because the leftmost robot is not the topmost robot, we mirror the coordinate frame about the right boundary giving $r_1 = (0.2, 0.8)$ and $r_2 = (0.8, 0.2)$. After the path is found, we undo the mirroring to the output path. Similarly, we exploit rotational symmetry and assume the command pushes a robot to hit the top wall. If a different wall is selected, we rotate the coordinate frame by 90°, 180°, or 270° counterclockwise and then push the robot to hit the top wall. After the path is found, we undo the rotation. This symmetry allows us to use a single function, Alg. 2, for collisions with all four walls.

V. Position Control of n Robots Using Boundary Interaction

The ideas from Alg. 1 can be extended to control the position of n particles using walls with non-slip contact. The solution is complete, but not optimal, and requires the starting and final configurations of particles to be disjoint. The solution described here is an iterative procedure with n loops. The k^{th} loop moves the k^{th} robot from a *staging zone* to the desired position in a *build zone*. All robots move according to the uniform input, but due to non-slip wall contacts, at the end of the k^{th} loop, robots 1 through k are in their desired final configuration in the build zone, and robots $k+1$ to n are in the staging zone. See Fig. 8 for a schematic of the build and staging zones.

Assume an open workspace with four axis-aligned walls with non-slip contact. The axis-aligned build zone of dimension (w_b, h_b) containing the final configuration of n robots must be disjoint from the axis-aligned staging zone of dimension (w_s, h_s) containing the starting configuration of n robots. Without loss of generality, assume the build zone is above the staging zone. Let d be the diameter of the particles. Furthermore, there must be at least ϵ space above

Algorithm 1 2-ParticlePathFinder(r_1, r_2, g_1, g_2, L)

```
Require: knowledge of current $(r_1, r_2)$ and goal $(g_1, g_2)$
    positions of two robots. $(0,0)$ is bottom corner, $L$ is
    length of the walls. $PathList$ contains all the paths sorted
    by their path length plus an admissible heuristic.
$PathList \leftarrow \{\}$
$P \leftarrow \{h(\{\}, r_1, r_2, g_1, g_2), \{\}, r_1, r_2\}$   ▷
    $P$ contains the admissible heuristic, the move sequence,
    and the current robot positions
while $P.r_1 \neq g_1$ and $P.r_2 \neq g_2$ do
    for $\theta \in \{0°, 90°, 180°, 270°\}$ do
        $(r_1, r_2, g_1, g_2) \leftarrow$ ROTATE($P.r_1, P.r_2, g_1, g_2, \theta$)
        $\{d, moves, r_1, r_2\} \leftarrow$
            PLANMOVEUP($r_1, r_2, g_1, g_2, L, P.moves$)
        $(moves, r_1, r_2) \leftarrow$ ROTATE($moves, r_1, r_2, -\theta$)
        PUSH $\{d, moves, r_1, r_2\}$ onto $PathList$
    end for
    SORT($PathList$)        ▷ sort by admissible heuristic
    $P \leftarrow$ POP first element of $PathList$
end while
return $moves$
```

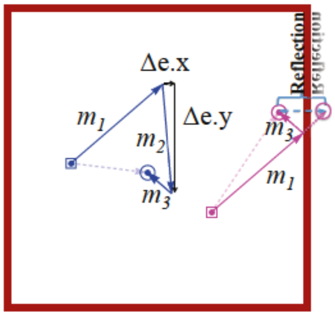


Fig. 5. If the goal configuration can be reached in three moves, the first move makes one particle hit a wall, the second move adjusts the relative spacing error Δe to zero, and the third move takes the particles to their final positions. The second move cannot be shortened, so optimization depends on choosing the location where the robot hits the wall. Since the shortest distance between two points is a straight line, reflecting the goal position across the boundary wall and plotting a straight line gives the optimal hit location.

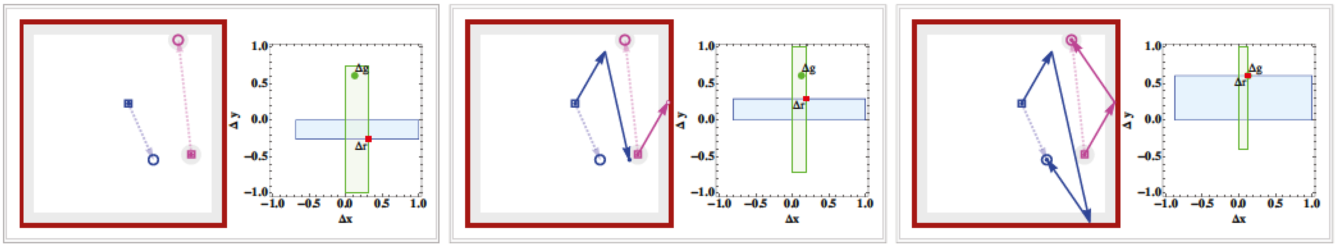


Fig. 4. Workspace and Δ configuration space for three sets of robot configurations with the same final goal. The red square represents the starting Δx and Δy and the green circle represents the goal Δx and Δy. The green rectangle illustrates reachable Δx and Δy when one particle is in contact with a horizontal wall and the blue rectangle illustrates the reachable region when in contact with a vertical wall.

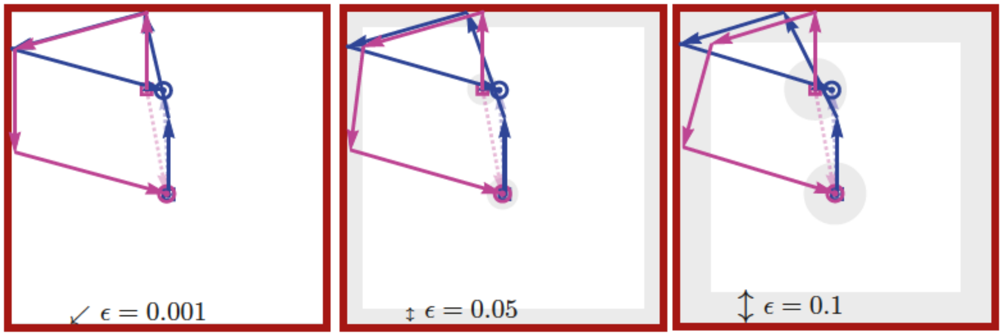


Fig. 6. Changing the minimum spacing ϵ changes the path. ϵ is the minimum spacing between two robots and the minimum separation from the boundaries.

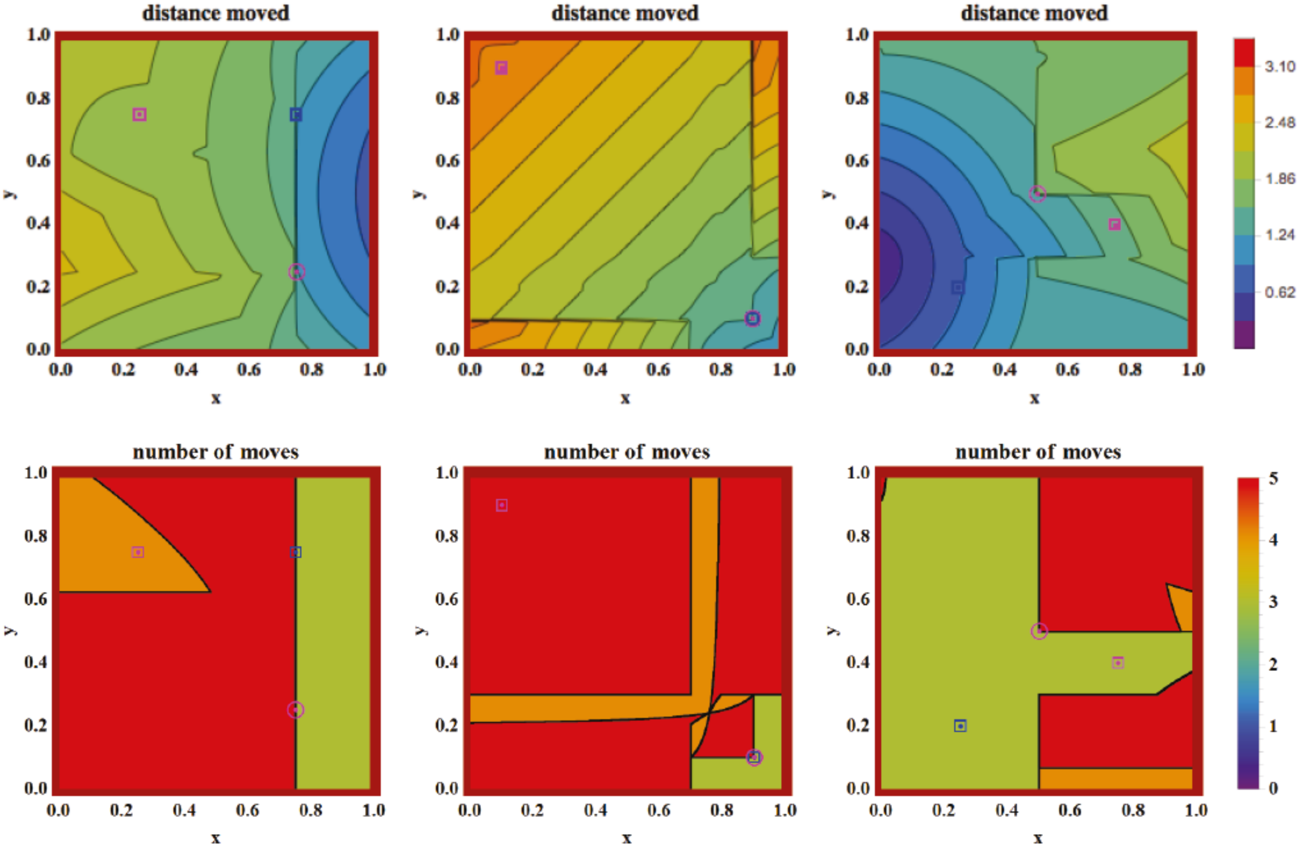


Fig. 7. Starting positions of robots 1 and 2 and goal position of robot 2 are fixed, and $\epsilon = 0.001$. The top row of contour plots show the distance if robot 1's goal position is varied in x and y. The bottom row shows the number of moves required for the same configurations.

Algorithm 2 PLANMOVEUP(r_1, r_2, g_1, g_2, L,*moves*)

```
Require: knowledge of current $(r_1, r_2)$ and goal $(g_1, g_2)$ positions of two robots. $(0,0)$ is bottom corner, $L$ is length of the walls. The array moves is the current sequence of moves up to the current position. Assume $r_1.x < r_2.x$ and $r_1.y \geq r_2.y$. If not, mirror the coordinate frame and swap the robots, then undo the mirroring before returning. $\epsilon$ is a small, nonzero, user-specified value.
Ensure: $(g_1, g_2), (r_1, r_2)$ all at least $\epsilon$ distance from walls the goals and starting points have at least $\epsilon$ distance from each other. $m_1$ is the first move toward the wall or goal. $m_2$ is the second move adjusting $\Delta e$.
$\Delta e \leftarrow (g_2 - g_1) - (r_2 - r_1)$
if $\Delta e = (0,0)$ then                                ▷ base case
    $m_1 \leftarrow g_2 - r_2$
    moves← $\{moves, m_1\}$
    $(r_1, r_2) \leftarrow$APPLYMOVE$(m_1, r_1, r_2)$
    return $\{h(moves, r_1, r_2, g_1, g_2), moves, r_1, r_2\}$
end if
if $r_2.x - r_1.x - 1 + 2\epsilon \leq \Delta g.x \leq 1$ and $r_2.y - r_1.y \leq \Delta g.y \leq 0$ then      ▷ $\Delta g \in$ reachable region
    $m_1 \leftarrow \left(\frac{1-r_1.y}{2-g_1.y-r_1.y}(g_1.x - r_1.x), 1 - r_1.y\right)$
    if $r_2.x + m_1.x > L$ then
        $m_1.x \leftarrow 1 - r_2.x$
    else if $r_2.x + m_1.x < 0$ then
        $m_1.x \leftarrow -r_2.x$
    end if
else
    $m_1 = (0, 1 - r_1.y)$
    $\Delta g \leftarrow$ closest reachable $(\Delta x, \Delta y)$.
end if
moves← $\{moves, m_1\}$
$(r_1, r_2) \leftarrow$APPLYMOVE$(m_1, r_1, r_2)$
$m_2 \leftarrow \Delta g - (r_2 - r_1)$
if robots on each other or on the wall then
    Add $\pm\epsilon$ to $m_2.x$ to avoid collision
end if
moves $\leftarrow \{moves, m_2\}$
$(r_1, r_2) \leftarrow$APPLYMOVE$(m_2, r_1, r_2)$
return $\{h(moves, r_1, r_2, g_1, g_2), moves, r_1, r_2\}$
```

Fig. 8. Illustration of Alg. 3, n robot position control using walls with non-slip contact.

Fig. 9. A DRIFTMOVE$(\alpha, \beta, \epsilon, 0°)$ repeats a triangular movement sequence $\{(\beta/2, -\epsilon), (\beta/2, \epsilon), (-\alpha, 0)\}$. At the sequence end, robot A has moved β units right, and robot B has moved $\beta - \alpha$ units right.

the build zone, ϵ below the staging zone, and $\epsilon + d$ to the left of the build and staging zone. The minimum workspace is then $(\epsilon + d + \max(w_b, w_s), 2\epsilon + h_s, h_b)$.

The n robots position control algorithm relies on a DRIFTMOVE$(\alpha, \beta, \epsilon, \theta)$ control input, described in Alg. 4 and shown in Fig. 9. For $\theta = 0°$, a drift move consists of repeating a triangular movement sequence $\{(\beta/2, -\epsilon), (\beta/2, \epsilon), (-\alpha, 0)\}$. Any particle touching a top wall moves right β units, while every particle not touching the top moves right $\beta - \alpha$.

Let $(0,0)$ be the lower left corner of the workspace, p_k the x, y position of the k^{th} robot, and f_k the final x, y position of the k^{th} robot. Label the robots in the staging zone from left-to-right and bottom-to-top, and the f_k configurations top-to-bottom and right-to-left as shown in Fig. 10.

Alg. 3 proceeds as follows: First, the robots are moved left away from the right wall, and down so all robots in k^{th} row touch the bottom wall. Second, a set of DriftMoves are executed that move all robots in k^{th} row left until k touches the left wall, with no net movement of the other robots. Third, a set of DriftMoves are executed that move only robot k to its target height and return the other robots to their initial heights. Fourth, all robots except robot k are pushed left until robot k is in the correct relative x position compared to robots 1 to $k-1$. Finally, all robots are moved right until robot k is in the desired target position. Running time is $O(n(w+h))$.

The hardware platform depicted in Fig. 10 is an assembled practical setup that assumes that $\epsilon = 1$ cm. The workspace is a 7×7 cm grid space. All particles are 3D-printed plastic whose top is a 1cm diameter cylinder with a narrower base

Algorithm 3 PositionControl$n$Robots(k)

```
Move( $-\epsilon, d/2 - p_{ky}$)
while $p_{kx} > d/2$ do
    DRIFTMOVE$(\epsilon, \min(p_{kx} - d/2, \epsilon), \epsilon, 180°)$
end while
$m \leftarrow$ ceil$(\frac{f_{ky} - d/2}{\epsilon})$
$\beta \leftarrow \frac{f_{ky} - d/2}{m}$
$\alpha \leftarrow \beta - \frac{d/2 - p_{ky} - \epsilon}{m}$
for $m$ iterations do
    DRIFTMOVE$(\alpha, \beta, \epsilon, 90°)$
end for
Move $(d/2 + \epsilon - f_{kx}, 0)$
Move $(f_{kx} - d/2, 0)$
```

Algorithm 4 DRIFTMOVE$(\alpha, \beta, \epsilon, \theta)$

particles touching the wall move β units, while particles not touching the wall move $\beta - \alpha$ units.

```
$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$
MOVE$(R \cdot [\beta/2, -\epsilon]^\top)$
MOVE$(R \cdot [\beta/2, \epsilon]^\top)$
MOVE$(R \cdot [-\alpha, 0]^\top)$
```

that encapsulates a steel bearing ball. Non-slip wall contact is generated by a toothed wall design to keep particles from moving out of place while implementing the drift move. The workspace boundary is mounted on top of a white sheet of cardboard. Underneath the cardboard, a grid of 3mm diameter magnets glued with 1 cm spacing to a thin board generates the uniform control input. A video attachment shows the algorithm at work. This discretized setup requires several modifications to Alg. 3. In this demonstration, all moves are 1 cm in length. All drift moves are a counterclockwise *square* move of size 1 cm×1 cm. Once the k^{th} particle gets to its designated location in each loop, a correction step is implemented. This correction step increases by two the total number of moves required per particle. Fig. 8 shows there are only 6 stages per particle involved in Alg. 3. The fixed step algorithm requires 8 stages per particle as shown in Fig. 10.

A significant difference between Alg. 3 and the fixed move implementation of it is that Alg. 3 enables placing particles at arbitrary, non-overlapping locations, while the fixed move implementation requires goal locations at the center of grid cells.

VI. Simulation

Two simulations were implemented using non-slip contact walls for position control. The first controls the position of two robots, the second controls the position of n robots.

A. Position Control of Two Robots

Algorithm 1 was implemented in Mathematica using point robots (radius = 0). Fig. 2 shows an implementation of this algorithm with robot initial positions represented by hollow squares and final positions by circles. Dashed lines show the shortest route if robots could be controlled independently, while solid lines show the optimal shortest path using uniform inputs.

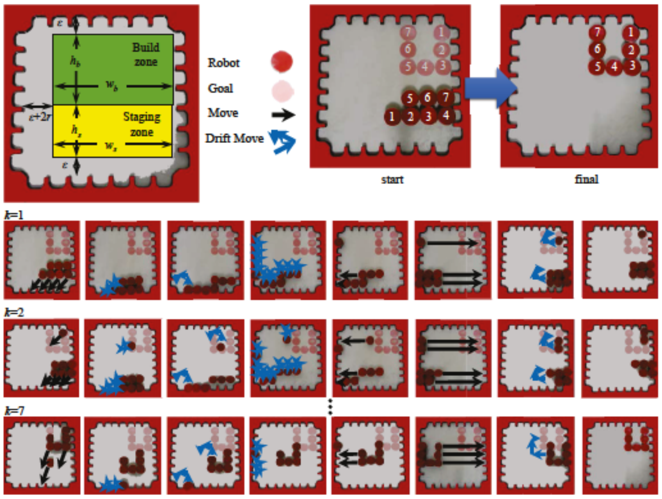


Fig. 10. Illustration of Alg. 3, discretized n robot position control using walls that enforce non-slip contact.

The contour plots in Fig. 7 top row show the length of the shortest path for given s_1, s_2, g_1 with g_2 ranging over all the workspace. This plot clearly shows the nonlinear nature of the path planning, with multiple isolated islands showing regions that are difficult to reach. If the length of each side of the square workspace is L, the worst case path length is $(\sqrt{2} + 2)L$.

The contour plots in Fig. 7 bottom row show the same configurations, but plot the required number of moves. There is never more than one g_2 position reachable in one move. If g_2 results in a contraction of $(\delta x, \delta y)$, there are many three move sequences. Four and five moves are sometimes required.

B. Position Control of n Robots

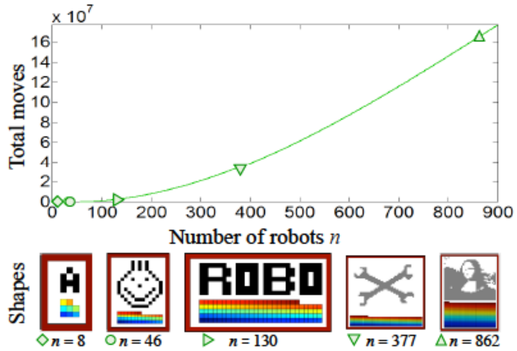
Alg. 3 was simulated in MATLAB using square block robots with unity width. Simulation results are shown in Fig. 11 for arrangements with an increasing number of robots, n= [8, 46, 130, 390, 862]. The distance moved grows quadratically with the number of robots n. A best-fit line $210n^2 + 1200n - 10,000$ is overlaid by the data.

In Fig. 11, the amount of clearance is $\epsilon = 1$. Control performance is sensitive to the desired clearance. As ϵ increases, the total distance decreases asymptotically, as shown in Fig. 12, because the robots have more room to maneuver and fewer DriftMoves are required.

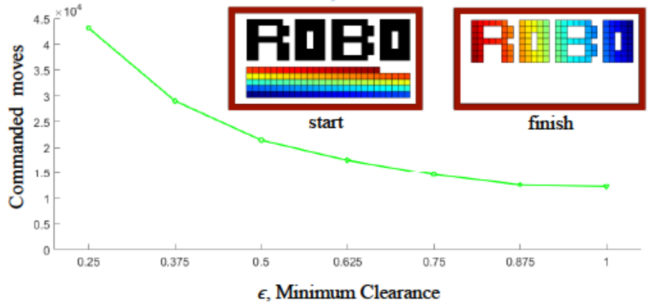
VII. Demonstration

Hardware Demonstration: Position Control of n Robots

A hardware setup with a bounded platform, magnetic sliders, and a magnetic guide board was used to implement Alg. 3. The pink boundary is toothed with a white free space, as shown in Fig 10. Only discrete, 1 cm moves in the x and y directions are used. The goal configuration highlighted in the top right corner represents a ‘U’ made of seven particles. The



i . . The re uired nu er o ove under A . u in non i con ac a o rearran e n uare ha ed ro o ro uadra ica i h n. See code a .



i . . on ro er or ance i en i ive o he de ired c earance ϵ. A ϵ increa e , he o a di ance decrea e a o ica .

dark red con ura ion i he curren o i ion o he ar ic e . ue o he di cre i ed ove en a o ed he oundar , dri ove o o a c uare. ree ar ic e re urn o heir ar o i ion u ar ic e on he oundar o ove a era , enera in a ne idin o ion.

i . o o he o ion o he ar ic e hrou h i er a ion k , , and . A ar ic e receive he a e con ro in u , u oundar in erac ion reak he con ro e r . ar ic e reach heir oa o i ion in a r in, r ou arran e en e innin i h he o o e ar ic e ro he a in one occu in he o ri h o i ion o he ui d one.

. S A T

Thi a er re en ed echni ue or con ro in he o i ion o a ar o ro o u in uni or in u and in erac ion i h oundar ric ion orce . The a er rovided a ori h or reci e o i ion con ro , a e a ro u and e cien covariance con ro . endin a ori h and o i rai h or ard u increa e he co e i . Addi ion a , hi a er a u ed ric ion a u cien o co e e o ar ic e in con ac i h he oundar . The a ori h re uire re oo in o hand e a ric ion coe cien . The a ori h a u ed a rec an u ar ork ace. Thi i a rea ona e a u ion or ar i cia environ en , u in vivo environ en are curved. A e r earch ro ra cou d i ork, u i canno ake advan a e o he o d ro a iona e r a in a rec an u ar environ en . u ure e or hou d e direc ed o ard i rovin he echno o and ai orin i o eci c ro o a ica ion .

S

S. ho dhur , . in , and . . a e eri, on ro in u i e icroro o recen ro re and u ure cha en e , *Journal of Micro-Bio Robotics*, vo . , no. , . . .

S. Mar e , S. Taherkhani, M. Ta ri ian, M. Moha adi, . de anau e, and . e ou , o u er d con ro ed ac eria ran or and a re a ion o icro ia adhered nano co onen , *Journal of Micro-Bio Robotics*, vo . , no. , . , .

. S. S. i , A. Becker, . u, A. A. u iu , and M. . i , ar in a ne ic di o e he ero enei o in erna i ed iron o ide nano ar ic e or icroor ani ar con ro , *Journal of Nanoparticle Research*, vo . , no. , . , .

B. . ona d, . . eve , . a ro n , and . u , annin and con ro or icroa e o ruc ure co o ed o re en ineered e icroro o , *The International Journal of Robotics Research*, vo . , no. , . , .

A. ho h and . i cher, on ro ed ro u ion o ar i cia a ne ic nano ruc ured ro e er , *Nano Letters*, vo . , no. , . , .

. u, . . i , . . i , M. . i , and A. A. u iu , Mo ion con ro o a ne i ed e rah ena ri or i ce a ne ic e d i h ode redic ive con ro , *Int. J. Rob. Res.*, vo . , no. , . , an.

. iu and B. . e on, Ma ne ic he ica icro and nanoro o To ard heir io edica a ica ion , *Engineering*, vo . , no. , . , .

M. er ed and . u, or a ion con rained u i a en con ro , *IEEE Trans. Robotics Automat.*, vo . , . , .

M. A. ieh, . u ar, and . hai o ic , ecen ra i ed con ro er or ha e enera ion i h ro o ic ar , *Robotica*, vo . , no. , . , .

S. Mar e , Ma ne o ac ic ac eria or he ani u a ion and ran or o icro and nano e er i ed o ec , *Micro-and Nanomanipulation Tools*, . , .

. an, . hou, . u, T. u, . en , T. Tan , . en , . Bian, . han , A. erreira, and . han , Ma ne i e nano ruc ured orou ho o he ica icro i er or ar e ed de iver , *Advanced Functional Materials*, vo . , no. , . , .

A. . Ber o i, T. o oko nikov, . Sun, . in k , and . on Brech , in a ern and heir i urca ion in a non oca ode o io o ica ar , *Communications in Mathematical Sciences*, vo . , no. , . , .

T. Bre , on ro o an a en u in e in ruc ion , in *Proceedings of Robotics: Science and Systems*, A an a, A, SA, une , .

A. Becker, . n uk e , T. Bre , and . Mc urkin, on ro in an di eren ia drive ro o i h uni or con ro in u , *Int. J. Robot. Res.*, vo . , no. , . , .

A. Becker, . a i i, . er e , M. u en ein, and . Mc urkin, Ma ive uni or ani u a ion on ro in ar e o u a ion o i e ro o i h a co on in u i na , in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, ov. , . .

. a irau and . . avraki, o i ionin o e ric and non e ric ar u in radia and con an e d o u a ion o a e ui i riu con ura ion , *International Journal of Robotics Research*, vo . , no. , . , .

T. . o e, . anho ar, and . M. nch, S idin ani u a ion o ri id odie on a con ro ed do a e, *The International Journal of Robotics Research*, vo . , no. , . , .

. M. urce , i e a o e no d nu er, *American Journal of Physics*, vo . , no. , . , . n ine . Avai a e h d .doi.or . .

M. u en ein, . Ah er, and . a a, i o o A o co ca a e ro o e or co ec ive ehavior , in *IEEE Int. Conf. Rob. Aut.*, Ma .

S. Shahrokhi and A. T. Becker, Movin T o ar ic e i h Shared on ro n u in a ric ion, o ra e on ra ion ro ec , ov. . n ine . Avai a e h de on ra ion . o ra .co Movin T o ar ic e i hShared on ro n u in a ric ion

A. . Mahadev and A. T. Becker, Arran in a ro o ar i h o a in u and a ric ion di cre e . a a cen ra e e chan e, e . . n ine . Avai a e h . a h ork . co a a cen ra ee chan e