Exploiting Non-Slip Wall Contacts to Position Two Particles Using The Same Control Input

Shiva Shahrokhi, Jingang Shi, Benedict Isichei, and Aaron T. Becker

***Abstract*— Steered particles offer a method for targeted therapy, interventions, and drug delivery in regions inaccessible by large robots. Magnetic actuation has the benefits of requiring no tethers, being able to operate from a distance, and in some cases allows imaging for feedback (e.g. MRI). This paper investigates particle control with uniform magnetic gradients (the same force is applied everywhere in the workspace). Given three orthogonal magnetic fields, steering one particle in 3D is trivial. Adding additional particles to steer makes the system underactuated because there are more states than control inputs. However, the walls of in vivo and artificial environments often have surface roughness such that the particles do not move unless actuation pulls them away from the wall. In previous works, we showed that the individual 2D position of two particles is controllable in a square workspace with non-slip wall contact [1]. Because in vivo environments are usually not square, this paper extends the previous work to convex workspaces including circles and 3D positioning. This paper also implements the algorithms using a hardware setup inspired by the gastrointestinal tract.**

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.

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 planar control input, but $2n$ degrees of freedom for the n-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 [2].

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.

*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,atbecker}@uh.edu

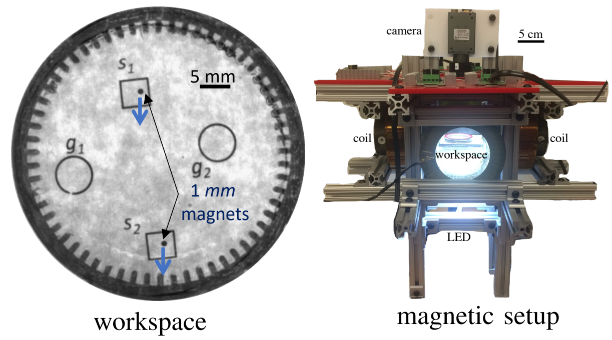


Fig. 1. Workspace and magnetic setup for an experiment of positioning particles that receive the same control inputs, but cannot move while a control input pushes them into a boundary.

The paper is arranged as follows. After a review of recent related work in Sec. II, Sec. III introduces a model for boundary interaction and two shortest path results for representative workspaces. We provide a shortest-path algorithm to arbitrarily position two particles in Sec. IV. Section V describes implementations of the algorithms in simulation and Sec. VI describes hardware experiments, as shown in Fig. 1. We end with directions for future research in Sec. VII.

This paper is an elaboration of preliminary work in a conference paper [1] which considered only square workspaces. This work extends the analysis to convex workspaces and 3D positioning. This paper also implements the algorithms using a hardware setup inspired by the anatomy of the gastrointestinal tract.

II. Related Work

Controlling the *shape*, or relative positions, of a swarm of robots is a key ability for a range of applications. Correspondingly, it has been studied from a control-theoretic perspective in both centralized and decentralized approaches. For examples of each, see the centralized virtual leaders in [3], and the gradient-based decentralized controllers using control-Lyapunov functions in [4]. However, these approaches assume a level of intelligence and autonomy in individual robots that exceeds the capabilities of many systems, including current micro- and nano-robots. Current micro- and nano-robots, such as those in [5]–[7] lack onboard computation.

Instead, this paper focuses on centralized techniques that apply the same control input to both particles. Precision control requires breaking the symmetry caused by the uniform input. Symmetry can be broken using particles that respond differently to the uniform control signal, either through agent-agent reactions [8], or engineered inhomogeneity [9]–[11]. This work assumes a uniform control with homogenous particles, as in [2], and breaks the control symmetry using obstacles in the workspace. The magnetic gradients of MRI scanners are *uniform*, meaning the same force is applied everywhere in the workspace [12].

Alternative techniques rely on non-uniform inputs, such as artificial force-fields. Applications have included techniques to design shear forces for sensorless manipulation of a single object by [13]. [14] demonstrated a collection of 2D force fields generated by six degree-of-freedom vibration inputs to a rigid plate. These force fields, including shear forces, could be used as a set of primitives for motion control to steer the formation of multiple objects.

Similarly, much recent work in magnet control has focused on exploiting inhomogeneities in the magnetic field to control multiple micro particles using gradient-based pulling [15], [16]. Unfortunately, using large-scale external magnetic fields makes it challenging to independently control more than one microrobot unless the distance between the electromagnetic coils is at the same length scales as the robot workspace [15]–[17]. In contrast this paper requires only a controllable constant gradient in orthogonal directions to position the particles.

III. Theory

If a control input causes the particles to collide with obstacles at different times, inverting the control input does not undo the action. Due to this lack of time-reversibility, techniques that require a bidirectional graph, e.g. PRM [18] and RRT* [19] are not suitable. Instead, this paper employs a graph search. This section starts with a boundary interaction model in subsection III-A.

Our algorithms rely on holding one particle stationary by pushing it into the boundary while moving the other particle. In subsections III-B and III-C we provide shortest-path results for two representative workspaces, squares and circles.

A. Boundary Interaction Model

In the absence of obstacles, uniform inputs move a swarm identically. Independent control requires breaking this symmetry. The following sections examine using non-slip boundary contacts to break the symmetry caused by uniform inputs.

The system dynamics represent particle swarms in low-Reynolds number environments, where viscosity dominates inertial forces and so velocity is proportional to input force [20]. In this regime, the input force command $\mathbf{u}(t)$ controls the velocity of the particles. If the i^{th} particle has position $\mathbf{x}_i(t)$ and velocity $\dot{\mathbf{x}}_i(t)$, we assume the following system model:

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}(t) + F\left(\mathbf{x}_i(t), \mathbf{u}(t)\right), \ i \in [1, n]. \tag{1}$$

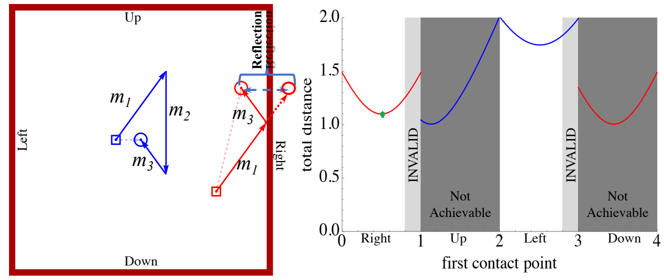


Fig. 2. The shortest three-move path that reconfigures two particles has the property that the incident angle equals the reflected angle.

$$F(\mathbf{x}_i(t), \mathbf{u}(t)) = \begin{cases} -\mathbf{u}(t) & \begin{array}{c}\mathbf{x}_i(t) \in \text{boundary } \textbf{and} \\ \mathbf{N}(\text{boundary}(\mathbf{x}_i(t))) \cdot \mathbf{u}(t) \leq 0\end{array} \\ 0 & \text{else.} \end{cases},$$

Here $F(\mathbf{x}_i(t), \mathbf{u}(t))$ is the frictional force provided by the boundary, and $\mathbf{N}(\text{boundary}(\mathbf{x}_i(t)))$ is the normal to the boundary at position $\mathbf{x}_i(t)$.

The same model can be generalized to particles moved by fluid flow where the vector direction of fluid flow $\mathbf{u}(t)$ controls the velocity of particles, or for a swarm of particles that move at a constant speed in a direction specified by a uniform input $\mathbf{u}(t)$ [21]. As in our model, fluid flowing in a pipe has zero velocity along the boundary. Similar mechanical systems exist at larger scales, e.g. all tumblers of a combination lock move uniformly unless obstructed by an obstacle. Our control problem is to design the control inputs $\mathbf{u}(t)$ to make all n particles achieve a task.

B. Example: Shortest Path in a Square Workspace

If the goal configuration cannot be reached in one move but can be reached in three moves, the shortest path has a simple solution. The first move, m_1, makes one particle hit a wall, m_2 adjusts the relative spacing error to zero, and m_3 takes the particles to their final positions. m_2 cannot be shortened, so optimization depends on choosing the location where the particle 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, as shown in Fig. 2. There are four walls, and four candidate solutions, but some candidate solutions may be *invalid* because they would cause both particles to hit the boundary in move m_1 (light grey regions) or *not achievable* because they would cause both particles to hit the boundary in move m_2 (dark grey regions).

C. Shortest Path in Unit Disk that Intersects Circumference

Changing the relative positions of particles in any workspace requires making one particles contact the boundary. The shortest path between two points in the unit disk that intersects the circumference is composed of two straight line segments, as shown in Fig. 3. The problem can be simplified by choosing the coordinate system carefully. We define the x-axis along the line from the circle center to the starting point: $S = (s, 0)$, and define the point of intersection by the angle θ from the x-axis: $P = (\cos\theta, \sin\theta)$. Define the final

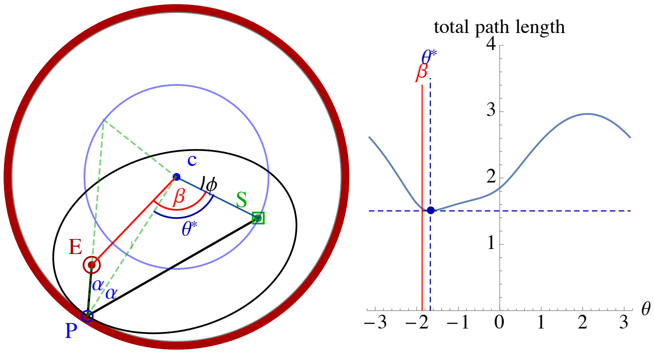


Fig. 3. The shortest path between two points S to E in the unit disk that intersects the circumference. The path length as a function of intersection point, $P = (\cos\theta, \sin\theta)$ is shown at right. See [22].

point E by a radius e and angle β: $E = e(\cos\beta, \sin\beta)$. Then the length of the two line segments is

$$\sqrt{(s-\cos\theta)^2+\sin^2\theta}+\sqrt{(e\cos\beta-\cos\theta)^2+(e\sin\beta-\sin\theta)^2}, \quad (2)$$

which is minimized by choosing an appropriate θ value.

The length of the two line segments as a function of θ is drawn in the right plot of Fig. 3. There are several simple solutions. If s is 1 or e is 0 or β is 0, the optimal angle θ^* is 0. If e is 1 or s is 0, the optimal angle is β. Label the origin O. The optimal path satisfies the law of reflection off the unit circle, with angle of incidence equal to angle of reflection. The angle $\angle OPS$ (from the origin to P to S) is the same as the angle $\angle OPE$ (from the origin to P to E). We name these angles α. This can be proved by drawing an ellipse whose foci are S and E. When the ellipse is tangent to the circle, the point of tangency is P. Since the distance from the origin to P is always 1, we can set up three equalities using the law of sines: From triangle OSP: $\frac{\sin\alpha}{s} = \frac{\sin(\alpha+\theta)}{1} = \frac{\sin\theta}{||SP||}$, and from triangle OEP: $\frac{\sin\alpha}{e} = \frac{\sin(\beta-\theta)}{||EP||}$. If we mirror the point S about line $\overline{OP}$ and label this point C, from triangle CEO: $\frac{\sin(\alpha+\theta)}{e} = \frac{\sin(2\theta-\beta)}{||CE||}$.

Simplifying this system of equations results in $s = e\csc\theta(s\sin(2\theta-\beta)+\sin(\beta-\theta))$. Solving this last equation results in a quartic solution that has a closed-form solution with four roots, each of which can be either a clockwise or a counterclockwise rotation θ, depending on the sign of β, with $-\pi \le \beta \le \pi$. We evaluate each and select the solution that results in the shortest length path. For an interactive Mathematica demonstration of this shortest path, see [22]. Because the closed form solution is long, it is included in the paper attachments.

IV. Position Control of Two Particles Using Boundary Interaction

This section presents algorithms that use non-slip contacts with walls to arbitrarily position two particles in a convex workspace. Our previous work used a square workspace [1]. Fig. 4 shows a Mathematica implementation of a square workspace for different cases. Alg. 1 can now handle any convex workspace, including the special limit case of a circular workspace. In the last subsection we present techniques to control 3D positioning of two particles.

Workspaces are 2D convex polygons with no internal obstacles. Assume two particles are initialized at s_1 and s_2 with corresponding goal destinations g_1 and g_2. Denote the current positions of the particles p_1 and p_2.

A. Two Particle Path Planning

The configuration space for two particles is a four dimensional manifold. Translating both particles the same amount is a trivial operation, but changing the relative positions requires boundary interaction. For this reason, our algorithms use the two dimensional Δ configuration space, defined as the difference in position of the first particle from the second particle: $\Delta p = p_2 - p_1$.

The Δ configuration space is a set of all possible Δp values. Δ configuration spaces for a representative set of workspaces are shown in Fig. 6. The *2-move reachable set* is the locus of points in the Δ configuration space corresponding to any two-move sequence where the first move brings one particle into contact with the boundary, and the second move translates the second particle without moving the first. Fig. 9 shows the starting and ending relative positions as Δs and Δg in the Δ configuration space. The next subsections give procedures to compute the 2-move reachable set.

Values $.x$ and $.y$ denote the x and y coordinates, i.e., $p_1.x$ and $p_1.y$ denote the x and y locations of p_1. Alg. 1 assigns a uniform control input at every instance. The goal is to move the particles within ϵ of the goal positions using a shared control input where ϵ is an arbitrary small number. We do this by first moving them within ϵ of the correct relative position and then translating the particles to the goal. The relative position is $||\Delta g - \Delta p|| = ||(g_2 - g_1) - (p_2 - p_1)||$.

Algorithm 1 first computes the 2-move reachable set. If the goal relative position is in the 2-move reachable set, we move particles to achieve that relative position. If it is not in the 2-move reachable set, we move particles to achieve the closest point on this reachable set from Δg.

Algorithm 1 first computes the 2-move reachable set. If the goal relative position is in the reachable set, we move particles to achieve that relative position. If it is not in the reachable set, we move particles to achieve the closest point on the reachable set from Δg. ¿¿¿¿¿¿¿ Stashed changes *Achieving* a Δ configuration requires two-moves, the first to move until one particle touches a wall, and the second to adjust the relative spacing. Once the correct *relative* position has been achieved, a final translation delivers both particles to their goal destinations. Otherwise, we iterate until we reach the goal.

B. Δ Configuration Space

Given an n-sided convex polygon P, the Δ configuration space is a $2n$-sided convex polygon. A special case is for even-sided regular polygons, in which half the sides align and the Δ configuration space is n-sided. To give an intuition,

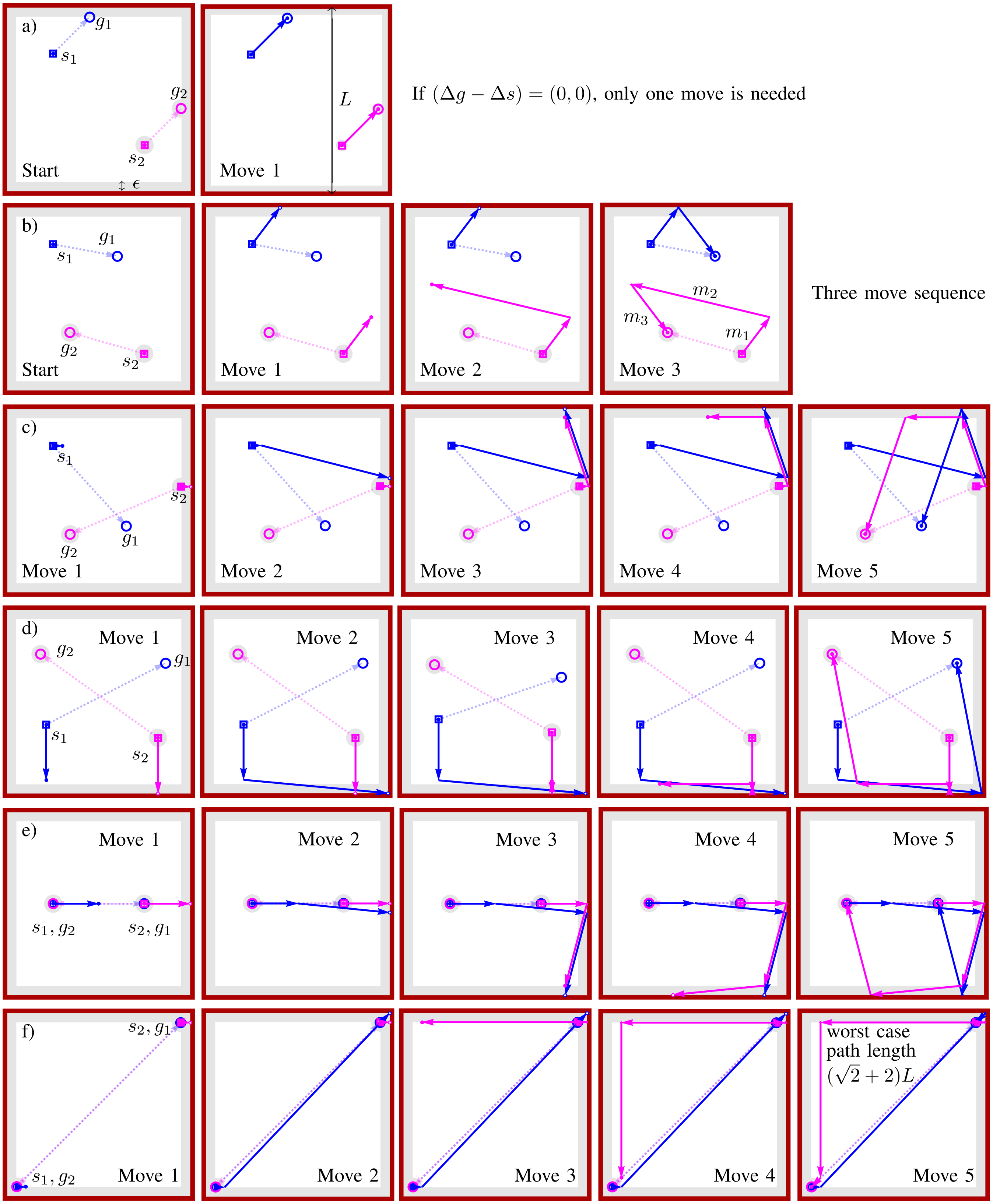


Fig. 4. Frames from an implementation of Alg. 1: two particle positioning using walls with non-slip contacts. Particle start positions are shown by squares, and goal positions by circles. Dashed lines show the shortest route if particles could be controlled independently. Solid arrows show path given by Alg. 1.

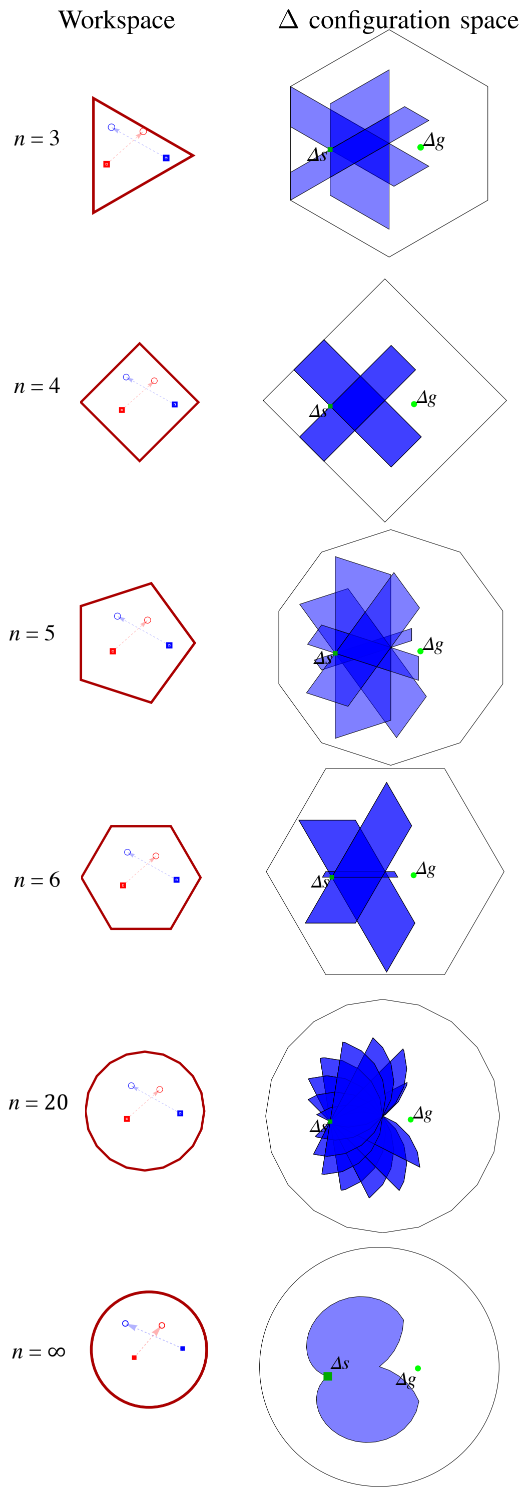


Fig. 5. The Δ configuration space is all possible configurations of $p_2 - p_1$. The sets reachable in two moves, called *reachable sets* are drawn with transparent blue polygons. A polygon with n sides has n reachable sets, but if n is even and the polygon is regular, half the reachable sets overlap. If Δg is in the reachable sets, we can achieve the required relative position in two moves. If Δg is not in the reachable set, we define a temporary goal, Δg_c, which is the closest point on the reachable set to Δg and apply two moves that make the relative position closer to Δg. We do it until we reach the relative goal position.

Algorithm 1 2-PARTICLEPATHPLANNING$(s_1, s_2, g_1, g_2, P, \epsilon)$

Require: knowledge of starting (s_1, s_2) and goal (g_1, g_2) positions of two particles. P is a description of the workspace. ϵ is a positive error bound.

$(p_1, p_2) \leftarrow (s_1, s_2)$ ▷ p_1, p_2 are current positions
moves $\leftarrow \{\}$
$\Delta p \leftarrow p_2 - p_1$
$\Delta g \leftarrow g_2 - g_1$
while $||\Delta p - \Delta g|| > \epsilon$ **do**
  $R_{\text{SET}} \leftarrow$ Compute 2-move reachable set ▷ use Alg. 2 or 3
  $\Delta g_c \leftarrow$nearest point in R_{SET} to Δg
  $m \leftarrow$move-to-wall corresponding to Δg_c
  moves $\leftarrow$ Append m to moves
  $(p_1, p_2) \leftarrow$ ApplyMove m to (p_1, p_2)
  $\Delta p \leftarrow p_2 - p_1$
end while
moves $\leftarrow$ Append $g_2 - p_2$ to moves ▷ translate to goal
return moves

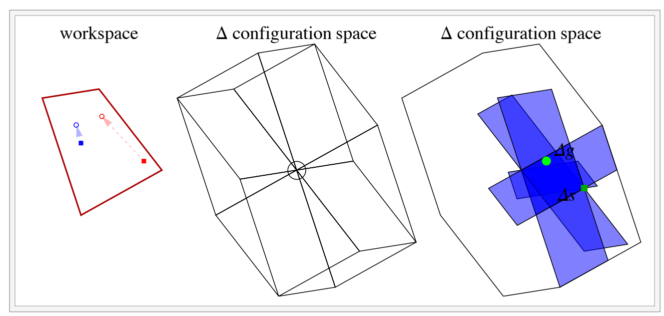


Fig. 6. Shiva, this image is nice. A few suggestions: ?TODO: Label the middle image 'Translate workspace around (0,0)', and draw the workspace outlines with different colors and thicknesses Workspace and Δ configuration space is shown for an arbitrary convex polygon with $n = 4$ sides. The 2-move reachable sets are drawn in transparent blue.

suppose that the first particle is positioned on a vertex of the workspace. The Δ configuration space is mainly determined by moving the second particle along the edges of the workspace and considering its relative distance to the first particle. In particular, to generate the Δ configuration space we iterate through all vertices v_i. At each iteration, we compute $v_i - v_j$ $(1 \leq j \leq n)$ and then make an n-sided polygon according to these values. Since our current workspaces are convex, the convex hull of these n-sided polygons determines the boundary of the Δ configuration space. To give a concrete example, Fig. 5 shows an arbitrary 4-sided workspace on the left. All the 4-sided polygons made by the computation of $(v_i - v_j)$s are shown in the middle and accordingly, Δ configuration space is shown on the right with 2-move reachable sets drawn in transparent blue. We will discuss how to generate 2-step reachable sets in the following subsection.

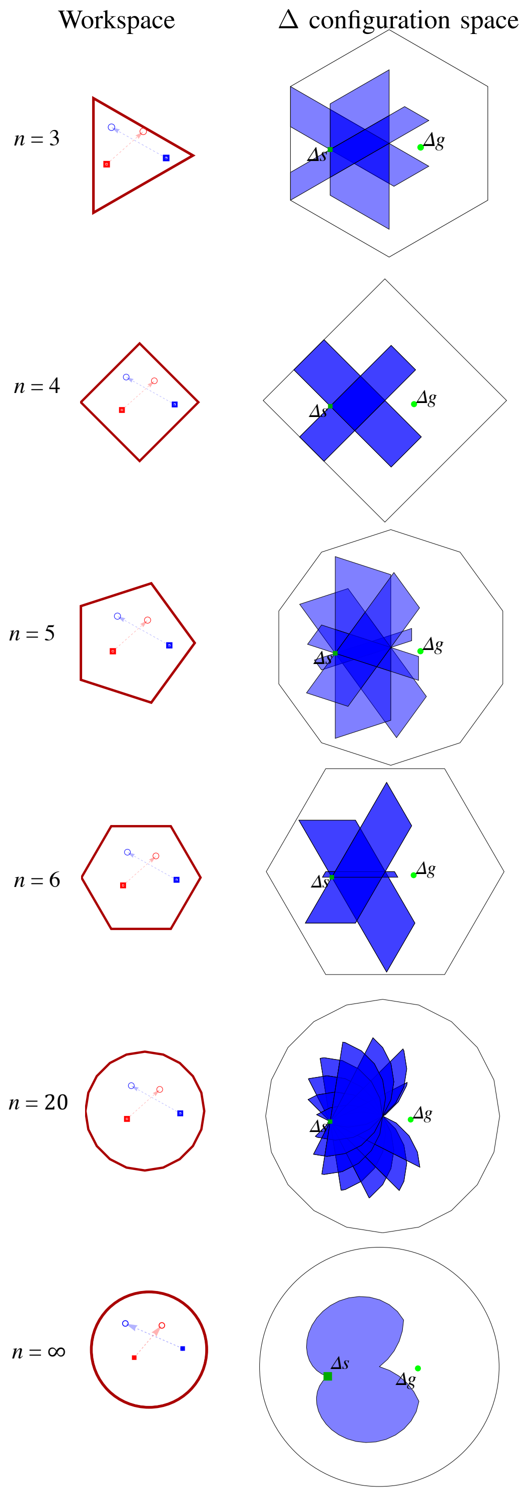


Fig. 7. The Δ configuration space is all possible configurations of $p_2 - p_1$. The sets reachable in two moves, called *2-move reachable sets* are drawn with transparent blue polygons. A polygon with n sides has n 2-move reachable sets, but if n is even and the polygon is regular, half the reachable sets overlap. If Δg is in the 2-move reachable sets, we can achieve the required relative position in two moves. If Δg is not in the 2-move reachable set, we define a temporary goal, Δg_c, which is the closest point on the 2-move reachable set to Δg and apply two moves that make the relative position closer to Δg. We do it until we reach the relative goal position.

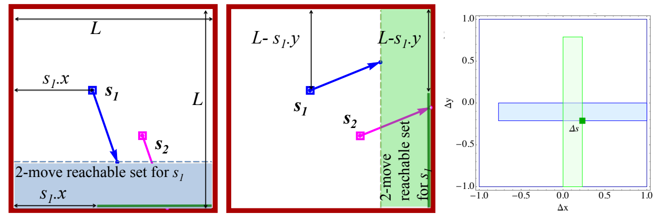


Fig. 8. Boundary interaction is used to change the relative positions of the particles. Each particle gets the same control input. (left) If particle 2 hits the bottom wall before particle 1 reaches a wall, particle 2 can reach anywhere along the green line, and particle 1 can move to anywhere in the shaded area. (middle) Similarly, if particle 2 hits the right wall before particle 1 reaches a wall, particle 2 can reach anywhere along the green line, and then particle 1 can move to anywhere in the shaded area. (right) The 2-move reachable sets in the Δ configuration space is shown.

C. Convex Polygonal Workspaces: 2-Move Reachable Set

Fig. 6 shows different workspaces and their representative Δ configuration spaces.

If a particle is touching a wall, the other particle can move freely in the reachable set as shown in Fig. 7. Alg. 2 computes the 2-move reachable set for any convex workspace. Fig. 8 illustrates the procedure to construct the 2-move reachable set generated by collisions with the i^{th} side. If one particle hits side i before the other (one particle will always hit before the other unless the particles are parallel to the wall), the 2-move reachable set is defined by a polygon, constructed in lines 2-13 of Alg. 2. The union of these polygons for all n sides is the 2-move reachable set of Δ configurations.

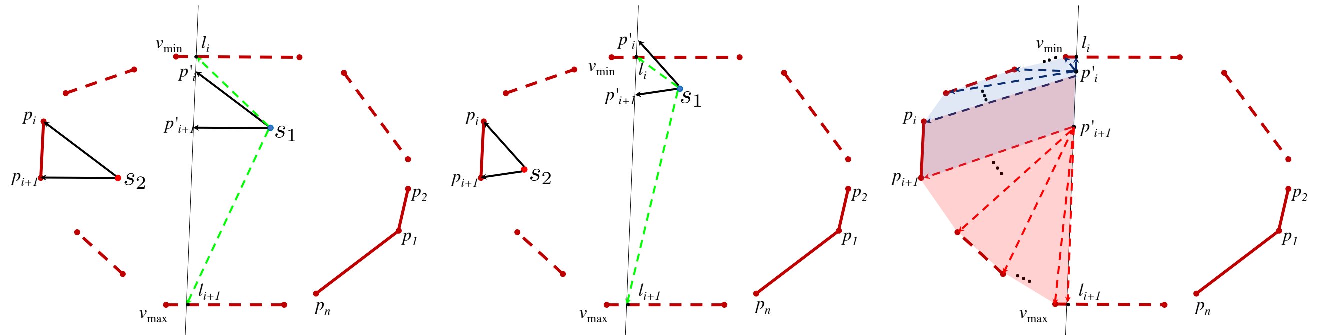
Algorithm 2 $\textsc{ReachableSetPolygon}(s_1, s_2, g_1, g_2, P)$

Require: knowledge of starting (s_1, s_2) and goal (g_1, g_2) positions of two particles. P is a list of the vertices of a convex polygon.

1: $R_{\text{SET}} \leftarrow \{\}$
2: **for** p_i in P **do**
3: $\quad p'_i \leftarrow s_1 + s_2 - p_i$
4: $\quad p'_{i+1} \leftarrow s_1 + s_2 - p_{i+1}$
5: $\quad L \leftarrow \overline{p'_i p'_{i+1}}$ ▷ line (p'_i, p'_{i+1})
6: $\quad l_i, l_{i+1} \leftarrow$ intersections of L and polygon P
7: $\quad$ **if** p'_i not inside polygon P **then**
8: $\quad\quad p'_i \leftarrow l_i$
9: $\quad$ **end if**
10: $\quad$ **if** p'_{i+1} not inside polygon P **then**
11: $\quad\quad p'_{i+1} \leftarrow l_{i+1}$
12: $\quad$ **end if**
13: $\quad D \leftarrow s_2 - s_1 - ([l_i, v_{\min}, \ldots, p_i] - p'_i, [p_{i+1}, p_{i+2}, \ldots, v_{\max}, l_{i+1}] - p'_{i+1})$
14: $\quad R_{\text{SET}} \leftarrow$ Append polygon D to R_{SET}
15: **end for**
16: Return R_{SET}

D. Circular Workspaces: 2-Move Reachable Set

To compute the 2-move reachable set for a circular workspace, first we consider all possible first contact locations. The set of boundary points that a particle can



Case 1: s_1 does not hit boundary Case 2: s_1 hits boundary Generates polygon defined by $\overline{p_i p_{i+1}}$

Fig. 9. Steps to generate the 2-move reachable set when one particle collides with edge $i, i+1$ of a convex polygonal workspace.

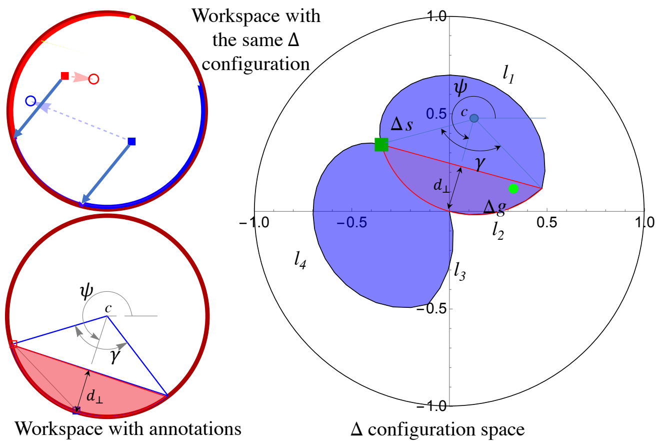


Fig. 10. Left top: The possible first contact points for the blue and red particles are shown with blue and red arcs. Left bottom: Once the blue particle touches the wall (blue square) the other particle (red square) can go anywhere in the reachable set (red region). Right: The Δ configuration space for the corresponded starting positions of the particles is shown. The possible 2-move reachable sets before contact are shown in the Δ configuration as a blue region. Once the blue particle contacts the boundary as shown, the reachable Δ configuration is the red set.

touch before the other particle touches is an arc of angle $2(\pi - \frac{\arcsin d_{12}}{r})$, where $d_{12} = ||s_1 - s_2||$ and r is the radius of the workspace. We define the angle between two particles as $\theta = \arctan(\frac{p_1.x - p_2.x}{p_1.y - p_2.y})$.

A circle has an infinite number of sides, thus infinite reachable sets. However, the 2-move reachable set can be parameterized by the angle of first contact location ψ, as shown in Fig. 9 where

$$\psi \in [\psi_{\min}, \psi_{\max}] = \theta + \left[\frac{\sin^{-1} d}{2r} - \frac{\pi}{2}, \frac{\pi}{2} - \frac{\sin^{-1} d}{2r}\right]. \tag{3}$$

The possible first contact locations are on an arc with interior angle γ parameterized by ψ:

$$\gamma(\psi) = \cos^{-1}\left(1 - \frac{d_\perp(\psi)}{r}\right), \text{ where:} \tag{4}$$

$$d_\perp(\psi) = 2||s_1.p_\psi(\psi) - s_2.p_\psi(\psi)||, \tag{5}$$

$$p_\psi(\psi) = r[\cos(\psi), \sin(\psi)]. \tag{6}$$

2-move eachable sets with π difference in ψ value are equivalent in the Δ configuration space, so we can plan in this space and choose between the two options to immobilize the particle closest to a wall. The reachable Δ configuration set for any first contact point defined by ψ is the area under a chord from angle $\psi - \frac{\gamma(\psi)}{2}$ to $\psi + \frac{\gamma(\psi)}{2}$, for a circle of radius r centered at $c = r(\cos(\psi - \pi), \sin(\psi - \pi))$. One such chord is drawn in red in Fig. 9.

The equations for the four lines outlining the union of reachable Δ configuration sets are as follows:

$$\begin{aligned} l_1 = r\Big(&(\cos\psi_{\min} - \cos(\gamma + \psi_{\min})) \\ &+ (\sin\psi_{\min} - \sin(\gamma + \psi_{\min}))\Big) \quad 0 < \gamma < \gamma(\psi_{\min}), \\ l_2 = r\Big(&(\cos\psi_{\max} - \cos(\gamma + \psi_{\max})) \\ &+ (\sin\psi_{\max} - \sin(\gamma + \psi_{\max}))\Big) \quad \gamma(\psi_{\max}) < \gamma < 0, \\ l_3 = r\Big(&(\cos\psi - \cos(\psi + \gamma(\psi))) \\ &+ (\sin\psi - \sin(\psi + \gamma(\psi)))\Big) \quad \psi_{\min} < \psi < \psi_{\max}, \\ l_4 = r\Big(&(\cos\psi - \cos(\psi - \gamma(\psi))) \\ &+ (\sin\psi - \sin(\psi - \gamma(\psi)))\Big) \quad \psi_{\min} < \psi < \psi_{\max}. \end{aligned} \tag{7}$$

We combine these boundaries to compute the 2-move reachable set summarized in Alg. 3. Next, find a ψ that would enable us to reach Δg_c, the nearest point in the 2-move reachable set to Δg. We first check if Δg_c is in the Δ configuration space chords defined by either $\psi_{\min}$ or $\psi_{\max}$ using:

$$\begin{aligned} &(\Delta g_c.x - c.x)^2 + (\Delta g_c.y - c.y)^2 > r^2 \textbf{ and} \\ &(c.x - \Delta g_c.x)\cos\psi + (c.y - \Delta g_c.y)\sin\psi > r\cos\gamma. \end{aligned} \tag{8}$$

If Δg_c is not in either chord, we draw a line from Δg_c to the current relative position, Δp. This line is a chord of the circle centered at c. The ψ to this chord obeys:

$$\psi = \tan^{-1}\left(\frac{\Delta p.x - \Delta g_c.x}{\Delta p.y - \Delta g_c.y}\right). \tag{9}$$

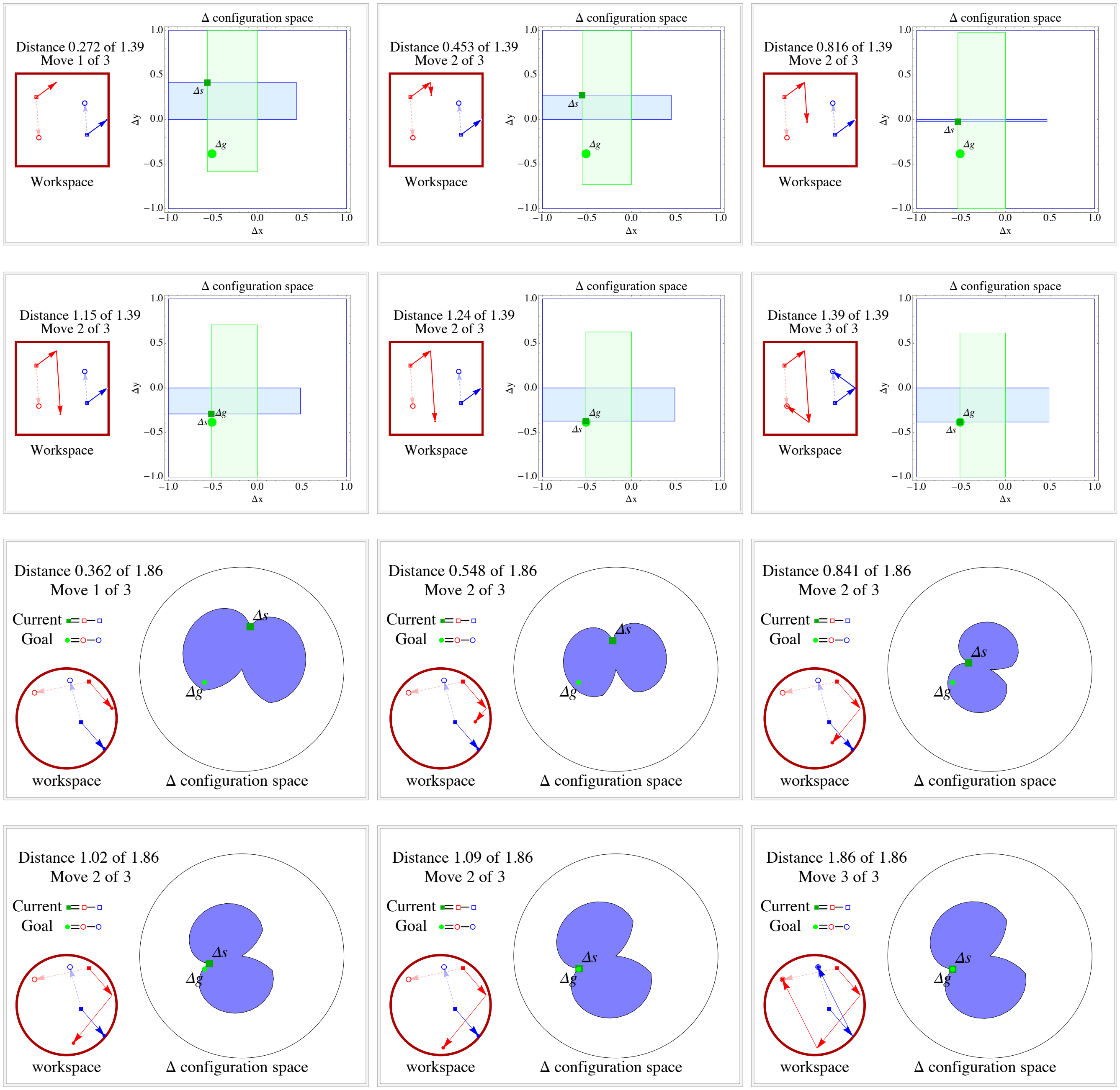


Fig. 11. Top row shows a polygonal workspace with its 2-move reachable sets. Bottom row, left circle shows the workspace. Right shows the Δ configuration space and the 2-move reachable set that is shown in red is representative of the point we need to go to get to the goal relative distance in one move.

The particles achieve Δg_c in two moves. The first move causes one particle to touch the wall at p_ψ, (6). The second move achieves the required relative position.

Algorithm 3 REACHABLESETCIRCLE(s_1, s_2, g_1, g_2)

Require: knowledge of starting (s_1, s_2) and goal (g_1, g_2) positions of two particles.

Calculate p_ψ ▷ use (6)
Calculate γ ▷ use (4)
Calculate l_1, l_2, l_3, l_4 ▷ use (7)
Return the union of (l_1, l_2, l_3, l_4)

E. 3D workspaces: Cylinders and Prisms

Extending path planning to 3D is possible only if the two particles do not initially have the same x and y positions. For ease of analysis, we assume the workspace boundaries extend in the $\pm z$ direction to form either right cylinders or right prisms. If the 3D projection is at a different angle, redefine the 2D workspace as a region perpendicular to the projection. First, we move the closest particle to the boundary, which prevents its z coordinate from changing. We next apply actuation in either the $\pm z$ direction to achieve the desired Δz. Then the particles are actuated away from the

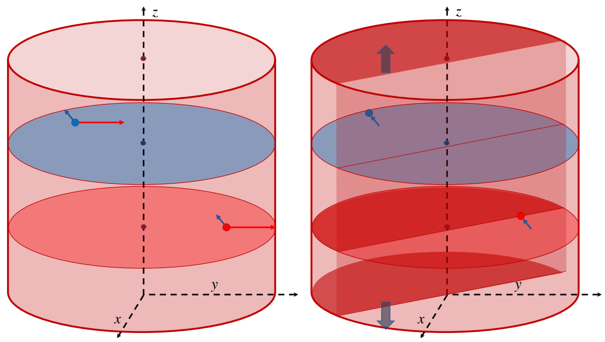


Fig. 12. Illustration on how boundary contacts enable 3D positioning. Once one particle contacts a boundary, the other particle's 2-move reachable set is a prism formed by extending the 2D 2-move reachable set in the $\pm z$-direction.

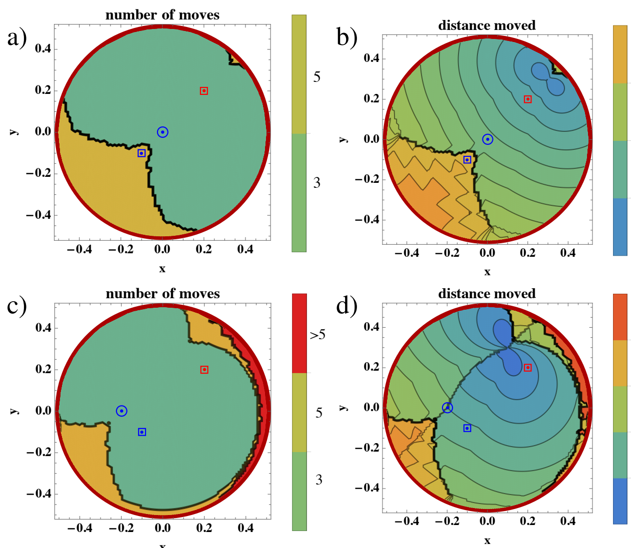


Fig. 13. Plots show performance with one goal on the boundary.

boundary and to the appropriate z positions. Path planning continues using Alg. 1 to position the particles to the desired x and y positions. As an example, consider Fig. 11 which shows a cylindrical workspace. The blue particle starts in the blue disk and the red particle starts in the red disk. The two candidate shortest-length paths that touch the wall are shown with parallel arrows. Each arrow will cause one of the particles to touch the wall, enabling the other particle to move freely in the z-axis to achieve the required relative position. This can be extended to other 3D workspaces if the workspace can be locally approximated as a 3D prism or cylinder. Other workspaces may be better handled by other path planners, such as [2], which used collisions with protrusions of the workspace to rearrange particles.

V. Simulation

Algorithm 1 was implemented in Mathematica using particles with zero radius.

The contour plots in Fig. 12 left show the length of the path for two different settings. Top row considers $\{s_1, s_2, g_1\}$

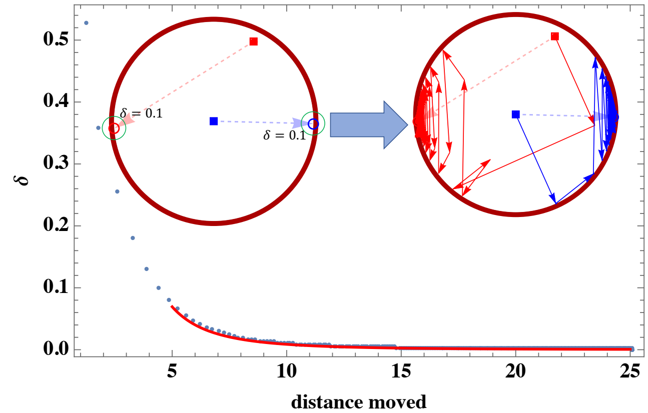


Fig. 14. The worst case path length occurs when particles must swap antipodes. This can never be achieved but can be asymptotically approached. Plot shows decreasing error as the number of moves grows.

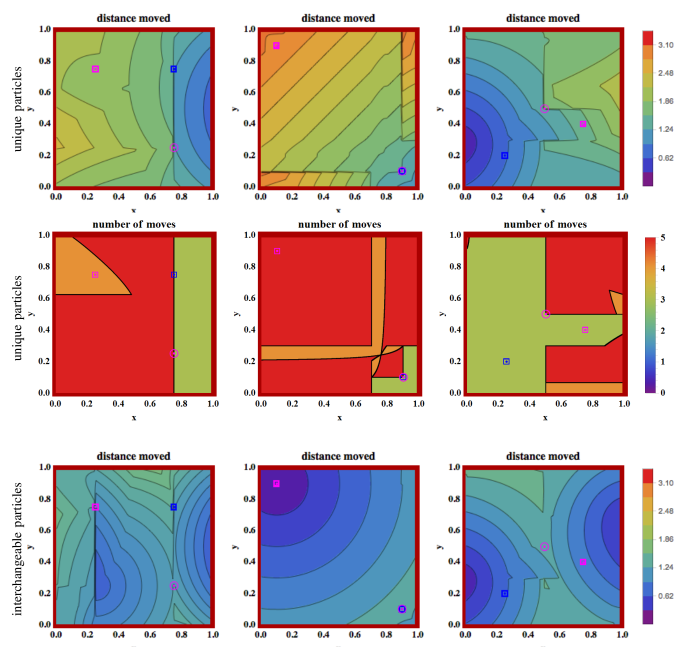


Fig. 15. Starting positions of particles 1 and 2 and goal position of particle 2 are fixed, and $\epsilon = 0.001$. The top row of contour plots show the distance if robot 1's goal position is varied in x and y. The bottom row shows the number of moves required for the same configurations.

= $\{(0.2, 0.2), (-0.1, -0.1), (0, 0)\}$ and bottom row considers $\{s_1, s_2, g_1\} = \{(0.2, 0.2), (-0.1, -0.1), (-0.2, 0)\}$ in a workspace with $r = 0.5$, and g_2 ranging over all the workspace. Fig. 12 right shows the number of moves and left shows the total distance of the path. This plot shows the nonlinear nature of the path planning. When the goal is in the middle of the workspace, a symmetry in the path length is expected as the top row shows. The bottom row shows a shift in the goal position which will break the symmetry of the path length in the workspace.

The path length grows when the goals have π difference and are very close to the boundary. The worst case occurs when the ending points are at antipodes along the boundary (π angular distance). This can never be achieved but can be asymptotically approached as shown in Fig. 13. Fig. 14 shows the same concepts in a square workspace. Fig. 14 top and middle row considers the particles for three arbitrary starting and goal positions for the particles. All of the discussed plots have considered the particles to be unique. If particles are interchangeable and it does not matter to put the specific particle to its goal location, the path length will be significantly smaller. The bottom row of Fig. 14 considers interchangeable particles with the same configuration as the middle row with unique particles.

VI. Experimental Results

To demonstrate Alg. 1 experimentally, we performed several tests. Each used the same magnetic setup shown in Fig. 1. Two different intestine models were employed, the first a 3D-printed cross-section representation of a small intestine, and the second a cross-section of a bovine stomach.

A. Magnetic Manipulation Setup

The magnetic manipulation system has two pairs of electromagnetic coils, each with iron cores at their centers, and arranged orthogonal to each other. The iron core at the center of each coil concentrates the magnetic field towards the workspace. An Arduino and four SyRen regenerative motor drivers were used for control inputs to the coils. Finally, a FOculus F0134SB 659 x 494 pixel camera was attached to the top of the system, focusing on the workspace which was backlit by a $15\,\mathrm{W}$ LED light strip.

To obtain experimental data, the test samples (the phantom intestine model and the bovine cross section) were placed in laser-cut acrylic discs and then immersed in corn syrup. Corn syrup was used to increase the viscosity to 12000 cP for the experiments. Spherical $1\,\mathrm{mm}$ magnets (supermagnetman #SP0100-50) were used as our particles. The magnetic field in this setup is only approximately uniform. The magnetic force varies in both magnitude and orientation. As shown in the video attachment, the particle closer to the coil moves faster than the other particle. Also, magnetic forces are not exactly parallel, but point toward the center of activated coil. Algorithm 1 is robust to these non-uniformities, but sometimes requires additional iterations.

B. Intestine Phantom Model

The intestine phantom model was used first and was made to mimic the geometry of an intestine and its villi. The model consists of a circular ring with an outer diameter of $50\,\mathrm{mm}$, an inner diameter of $46\,\mathrm{mm}$, and 60 $2\,\mathrm{mm}$ long protrusions on its inner surface cut out of $6\,\mathrm{mm}$ thick acrylic to model the geometry of intestinal villi. Fig. 15 top row shows an experiment. Starting and ending positions were printed beneath the workspace on transparency film.

C. Bovine Stomach Cross-section

Strips of cow stomach approximately $5\,\mathrm{mm}$ thick were cut and sewn to acrylic cylinder and then glued to an acrylic substrate using cyanoacrylate (superglue). This assembly was then filled with corn syrup. The experiment is shown in Fig. 15 bottom row.

VII. Conclusion and Future Work

This paper presented techniques for controlling the positions of two particles using uniform inputs and non-slip boundary contact. The paper provided algorithms for precise position control. The algorithms relied on calculating reachable sets in a 2D Δ configuration space. Extending Alg. 1 to 3D was straightforward, but increased the complexity. Hardware experiments illustrated the algorithms in ex vivo and in artificial workspaces that mimic the geometry of biological tissue.

There are several avenues for future work. This paper assumed friction was sufficient to completely stop particles in contact with the boundary. The algorithms would require retooling to handle small friction coefficients. The techniques in [1] and [2] could be applied to extend the analysis to more than two particles.

References

[1] S. Shahrokhi, A. Mahadev, and A. T. Becker, "Algorithms for shaping a particle swarm with a shared input by exploiting non-slip wall contacts," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, 2017.

[2] A. Becker, G. Habibi, J. Werfel, M. Rubenstein, and J. McLurkin, "Massive uniform manipulation," in *IEEE International Conference on Intelligent Robots and Systems*, Nov. 2013.

[3] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Trans. Robotics Automat.*, vol. 17, pp. 947–951, 2001.

[4] M. A. Hsieh, V. Kumar, and L. Chaimowicz, "Decentralized controllers for shape generation with robotic swarms," *Robotica*, vol. 26, no. 05, pp. 691–701, 2008.

[5] S. Chowdhury, W. Jing, and D. J. Cappelleri, "Controlling multiple microrobots: recent progress and future challenges," *Journal of Micro-Bio Robotics*, vol. 10, no. 1-4, pp. 1–11, 2015.

[6] S. Martel, "Magnetotactic bacteria for the manipulation and transport of micro-and nanometer-sized objects," *Micro-and Nanomanipulation Tools*, pp. 308–317, 2015.

[7] X. Yan, Q. Zhou, J. Yu, T. Xu, Y. Deng, T. Tang, Q. Feng, L. Bian, Y. Zhang, A. Ferreira, and L. Zhang, "Magnetite nanostructured porous hollow helical microswimmers for targeted delivery," *Advanced Functional Materials*, vol. 25, no. 33, pp. 5333–5342, 2015.

[8] A. L. Bertozzi, T. Kolokolnikov, H. Sun, D. Uminsky, and J. Von Brecht, "Ring patterns and their bifurcations in a nonlocal model of biological swarms," *Communications in Mathematical Sciences*, vol. 13, no. 4, pp. 955–985, 2015.

[9] B. R. Donald, C. G. Levey, I. Paprotny, and D. Rus, "Planning and control for microassembly of structures composed of stress-engineered mems microrobots," *The International Journal of Robotics Research*, vol. 32, no. 2, pp. 218–246, 2013.

[10] T. Bretl, "Control of many agents using few instructions," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007, pp. 1–8.

[11] A. Becker, C. Onyuksel, T. Bretl, and J. McLurkin, "Controlling many differential-drive robots with uniform control inputs," *Int. J. Robot. Res.*, vol. 33, no. 13, pp. 1626–1644, 2014.

[12] Z. Nosrati, N. Li, F. Michaud, S. Ranamukhaarachchi, S. Karagiozov, G. Soulez, S. Martel, K. Saatchi, and U. O. Hfeli, "Development of a coflowing device for the size-controlled preparation of magnetic-polymeric microspheres as embolization agents in magnetic resonance navigation technology," *ACS Biomaterials Science & Engineering*, vol. 4, no. 3, pp. 1092–1102, 2018.

[13] F. Lamiraux and L. E. Kavraki, "Positioning of symmetric and non-symmetric parts using radial and constant fields: Computation of all equilibrium configurations," *International Journal of Robotics Research*, vol. 20, no. 8, pp. 635–659, 2001.

[14] T. H. Vose, P. Umbanhowar, and K. M. Lynch, "Sliding manipulation of rigid bodies on a controlled 6-dof plate," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 819–838, 2012.

[15] S. Salmanipour and E. Diller, "Eight-degrees-of-freedom remote actuation of small magnetic mechanisms," in *IEEE International Conference on Robotics and Automation*, 2018.

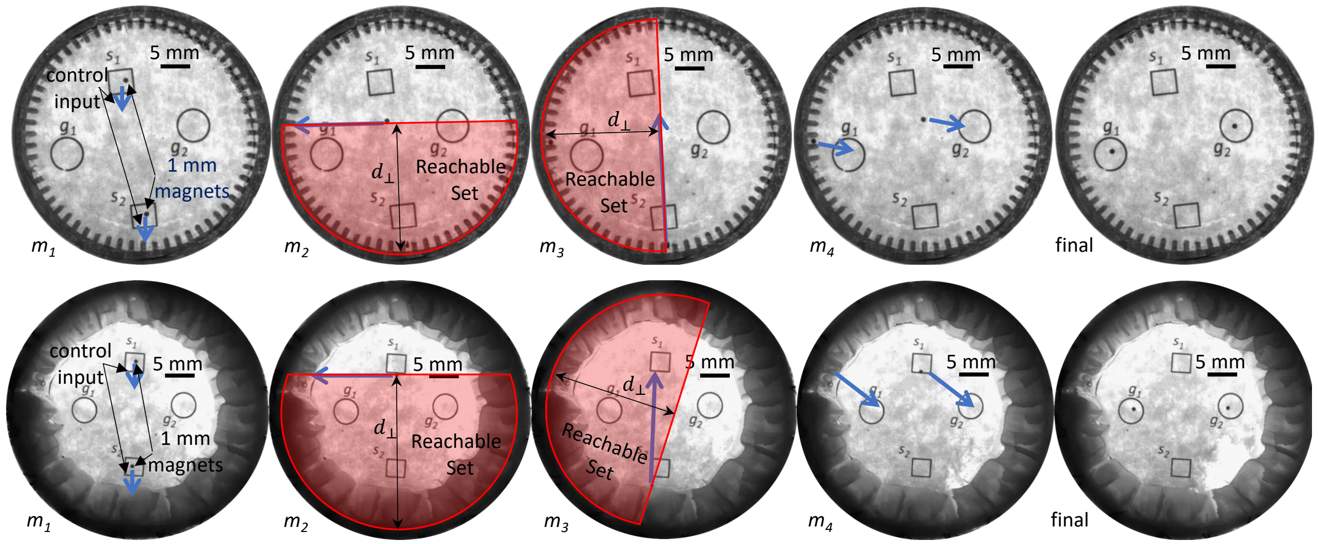


Fig. 16. Frames showing particle positions before and after control inputs. Top row: small intestine phantom. Bottom row: cow stomach tissue.

[16] A. Denasi and S. Misra, "Independent and leader follower control for two magnetic micro-agents," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 218–225, Jan 2018.

[17] E. Diller, J. Giltinan, G. Z. Lum, Z. Ye, and M. Sitti, "Six-degree-of-freedom magnetic actuation for wireless microrobotics," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 114–128, 2016.

[18] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[19] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[20] E. M. Purcell, "Life at low Reynolds number," *American Journal of Physics*, vol. 45, no. 1, pp. 3–11, 1977. [Online]. Available: http://dx.doi.org/10.1119/1.10903

[21] 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.

[22] J. Shi and A. T. Becker, "Shortest path between two points in the unit disk reflecting off the circumference," Sep. 2017. [Online]. Available: http://demonstrations.wolfram.com/ShortestPathBetweenTwoPointsInTheUnitDiskReflectingOffTheCir/