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

Shiva Shahrokhi, *Member, IEEE,* Jingang Shi, Benedict Isichei, and Aaron T. Becker, *Senior Member, IEEE*

***Abstract*—Steered particles offer a method for targeted therapy, interventions, and drug delivery in regions inaccessible by large robots. For example, magnetic actuation of particles 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 position control of particles using uniform forces (the same force is applied everywhere in the workspace). Given a controllable field that can generate bidirectional forces in three orthogonal directions, 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 work, we showed that the individual 2D position of two particles is controllable using global inputs in a square workspace with non-slip wall contact [?]. Because *in vivo* environments are usually not square, this paper extends the previous work to all convex workspaces, and shows how this could be extended to 3D positioning of neutrally buoyant particles. We investigate analytically an idealized variant of this problem with non-slip boundaries and control inputs that are applied uniformly to all particles in the workspace. This paper also implements the algorithms in 2D using a hardware setup inspired by the gastrointestinal tract.**

***Index Terms*—Underactuated Robots, Motion Control, Path Planning for Multiple Robot Systems, Configuration Space.**

I. Introduction

PARTICLE swarms propelled by an external 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 robot as the *system* that consists of particles, a uniform control field, and sensing. Consider a system of point-particles in a 2D planar workspace. 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.

Positioning is a foundational capability for a robotic system, e.g. placement of brachytherapy seeds. 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 [?]. 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.

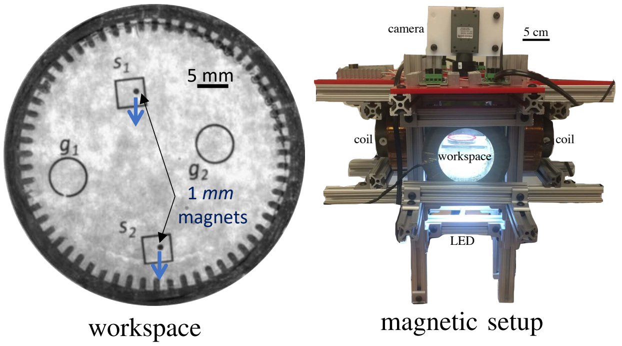


Fig. 1. Workspace and magnetic setup for an experiment to move one particle from s_1 to g_1 and a second particle from s_2 to g_2 when all particles receive the same control inputs, but cannot move while a control input pushes them into a boundary.

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

This paper is an elaboration of preliminary work in a conference paper [?] which considered only square workspaces. This work extends the analysis to convex workspaces and 3D positioning. This paper also implements the algorithms in 2D 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 [?], and the gradient-based decentralized controllers using control-Lyapunov functions in [?]. However, these approaches assume

*This work was supported by the National Science Foundation under Grant No. [IIS-1553063], [IIS-1619278], and [CNS-1646566].

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

Manuscript received May 30, 2018; revised October 13, 2018.

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 [**?**], [**?**], [**?**] lack onboard computation.

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 could be broken using particles that respond differently to the uniform control signal, either through agent-agent reactions [**?**], [**?**], or engineered inhomogeneity [**?**], [**?**], [**?**], [**?**]. The magnetic gradients of MRI scanners are *uniform*, meaning the same force is applied everywhere in the workspace [**?**]. This work, however, assumes a uniform control with homogenous particles, as in [**?**], and breaks the control symmetry using obstacles in the workspace.

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 [**?**]. [**?**] 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 control using magnetic fields has focused on exploiting inhomogeneities in the magnetic field to control multiple micro particles using gradient-based pulling [**?**], [**?**], [**?**], [**?**]. 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 [**?**], [**?**], [**?**]. In contrast, this paper requires only a controllable constant gradient in orthogonal directions to position the particles.

If a control input causes the particles to collide with obstacles at different times, inverting the control input does not undo the action, as in [**?**]. Due to this lack of time-reversibility, techniques that require a bidirectional graph, e.g. PRM [**?**] and RRT* [**?**] are not suitable. Instead, this paper employs a greedy search algorithm. For some configurations, we can obtain the optimal solution. Section V provides shortest-path results for two representative workspaces, squares and disks. While common search strategies such as RRT [**?**] could be used to generate solutions, our algorithm efficiently plans a path that, every two moves, decreases the relative position error between two particles.

III. 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. Our algorithms rely on holding one particle stationary by pushing it into the boundary while moving the other particle. These system dynamics can represent particle swarms in low-Reynolds number environments, where viscosity dominates inertial forces and so velocity is proportional to input force [**?**]. In this regime, the input force command $\mathbf{u}(t)$ controls the velocity of the particles. If the i^{th} particle has position $p_i(t)$ and velocity $\dot{p}_i(t)$, we assume the following system model:

$$\dot{p}_i(t) = \begin{cases} 0 & \begin{array}{c} p_i(t) \in \text{boundary } \textbf{and} \\ \mathbf{N}(\text{boundary}_{p_i(t)}) \cdot \mathbf{u}(t) \leq 0 \end{array} \\ \mathbf{u}(t) & \text{else} \end{cases} . \qquad (1)$$

Here $\mathbf{N}(\text{boundary}_{p_i(t)})$ is the normal to the boundary at position $p_i(t)$ and the frictional force provided by the boundary cancels any control force $\mathbf{u}(t)$ that pushes into the boundary.

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 $\mathbf{u}(t)$ [**?**]. 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 deliver two particles to goal positions.

We implemented a solution to this problem for square workspaces in our previous work, [**?**]. Fig. 2 shows solutions from a *Mathematica* implementation in a square workspace for six representative configurations.

IV. Position Control of Two Particles Using Boundary Interaction

This section presents an algorithm, Alg. 1, that uses non-slip contacts with walls to arbitrarily position two particles in a convex workspace. 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. 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. As an improvement over [**?**], 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.

A. Δ Configuration Space

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. The Δ configuration space is a set of all possible Δp values, defined as the difference in position of the particles: $\Delta p = p_2 - p_1$. We use the Δ configuration space to plan move sequences that achieve the desired relative spacing. Once the particles have the correct relative spacing, they can be delivered to the goal configuration in one move.

The Δ configuration space for an n-sided convex polygon P can be constructed in a method analogous to computing configuration space obstacles for polygons [**?**]. Translate n copies of P so that each copy moves a different vertex of P to $(0, 0)$. Because P is convex, the convex-hull of all these translated vertices is the boundary of the Δ configuration space.

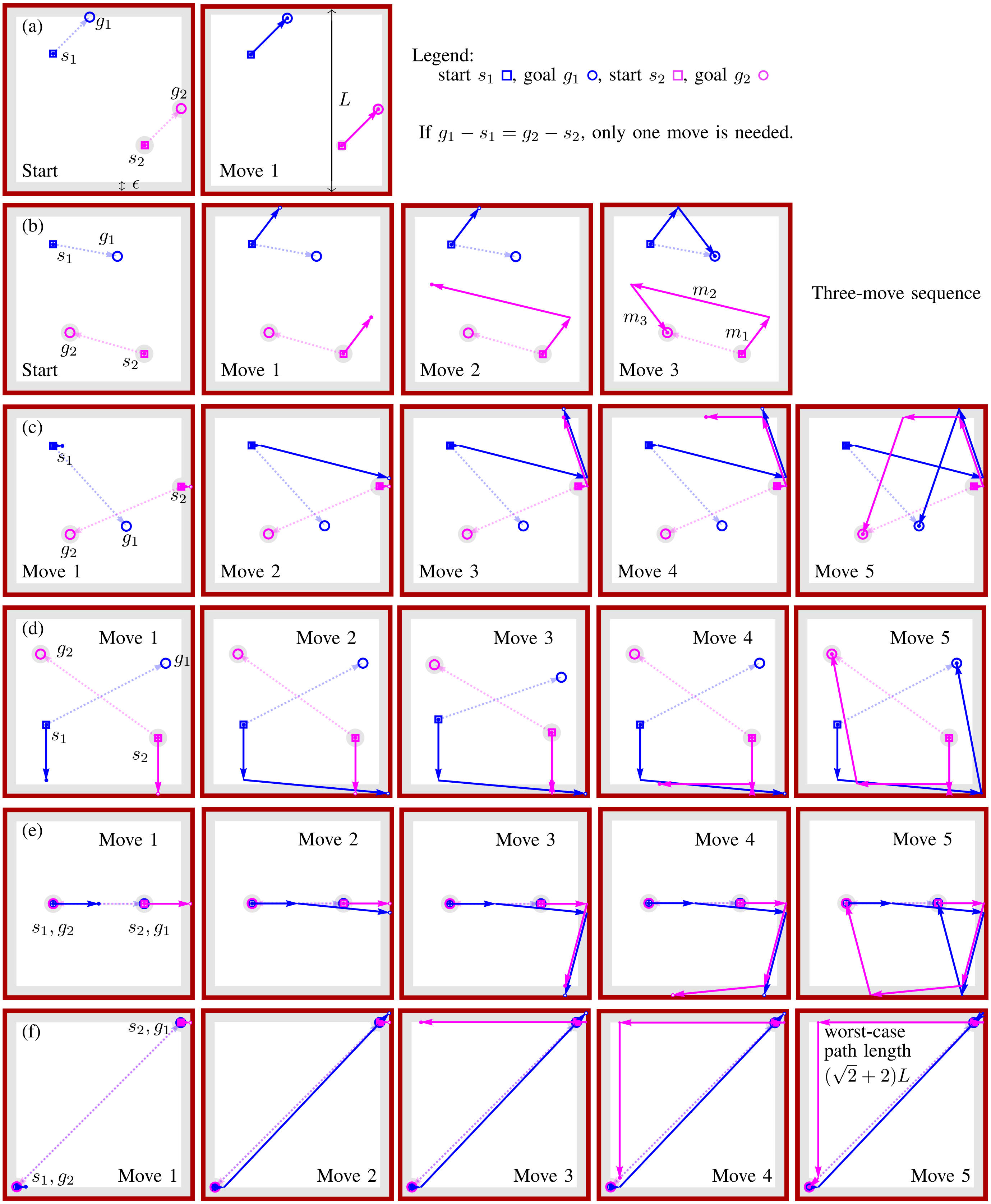


Fig. 2. Frames from an implementation of Alg. 1: two particle positioning using walls with non-slip contacts. Particles move from start positions (□, □), to goal positions (○, ○). Dashed lines show the shortest route if particles could be controlled independently. Solid arrows show path given by Alg. 1. Gray areas denote regions unaccessible by our motion planner. The particle start positions must be distinct ($\|s_2 - s_1\| \geq \epsilon$), and at least one goal position must be farther than ϵ from the boundary, where ϵ is a small but nonzero user-specified constant. The required number of moves increases from (a) to (f).

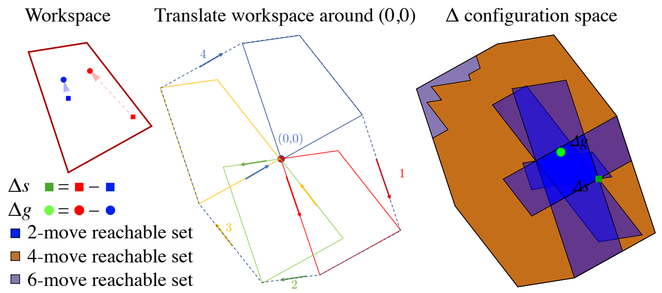


Fig. 3. Workspace and Δ configuration space is shown for an arbitrary convex polygon with $n = 4$ sides.

For an n-sided convex polygon, the Δ configuration space is a $2n$-sided convex polygon. Even-sided regular polygons are a special case in which half the sides align and the Δ configuration space is n-sided. An example Δ configuration space construction is shown in Fig. 3: a four-sided workspace is on the left, the four translated copies with dashed lines outlining the convex hull is in the middle, and the resulting Δ configuration space is on the right.

B. *Two Particle Path-Planning*

```
Algorithm 1  2-PARTICLEPATHPLAN($s_1, s_2, g_1, g_2, P, \epsilon$)
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. $\epsilon$ is an error bound ($\epsilon > 0$).
$(p_1, p_2) \leftarrow (s_1, s_2)$                ▷ $p_1, p_2$ 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 $R_{\text{SET}}$ to $\Delta g$
    $m \leftarrow$move-to-wall corresponding to $\Delta g_c$
    moves $\leftarrow$ Append $m$ to moves
    $(p_1, p_2) \leftarrow$ ApplyMove $m$ to $(p_1, p_2)$
    $\Delta p \leftarrow p_2 - p_1$
moves $\leftarrow$ Append $g_2 - p_2$ to moves       ▷ translate to goal
return moves
```

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. For the given Δs (starting configuration), the rightmost image of Fig. 3 draws the 2-move reachable sets in transparent blue. Figure 4 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. The $k+2$-move reachable set is constructed by the union of 2-move reachable sets starting from the boundary of the k-move reachable set.

The goal is to use a shared control input to move the particles within δ of the goal positions, where δ is an arbitrary small number. We do this by first moving them within δ 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 assigns a uniform control input at every instance. It 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, which is Δg_c.

Achieving a Δg_c configuration requires two moves, the first to move until one particle touches a boundary, and the second to adjust the relative spacing by moving only the particle not touching a boundary. 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.

C. *Convex Polygonal Workspaces: 2-Move Reachable Set*

Figure 4 shows six workspaces, their Δ configuration spaces, and the k-move reachable sets that correspond to representative initial conditions. Figure 5 highlights the construction of the 2-move reachable sets for a square workspace. There are four 2-move reachable sets, but the horizontal (and vertical) reachable sets are equivalent in the Δ configuration space so we can plan in this space and choose between the options to minimize the total distance. Algorithm 2 computes the 2-move reachable set for any convex workspace. The set is constructed by considering each edge of the workspace. We name each vertex as p_i where $1 \leq i \leq n$. If one particle contacts edge $\overline{p_i p_{i+1}}$ before the other (one particle will always contact before the other unless the particles are parallel to the wall), the corresponding 2-move reachable set is 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. Figure 6 illustrates the procedure to construct the 2-move reachable set generated by collisions with the $\overline{p_i p_{i+1}}$ edge.

D. *Convex Workspaces: Accessible Region*

This algorithm allows two particles to be steered to arbitrary positions as long as the initial particle positions are separated by at least ϵ, and at least one goal position is ϵ distance from a wall, where ϵ is a small, positive, user-defined number. For a square workspace where the length of each side is L, the worst case path length is $(\sqrt{2} + 2)L$, and requires at most five moves (see Fig. 2f). As the corner angles increase, the number of moves required to access the entire configuration space increases. As shown in Fig. 4, four moves are sufficient for 3-sided regular polygons, and six moves accesses the entire Δ configuration space for up to 8-sided regular polygons. Eight moves is sufficient for polygons with less than 14-sides, but we have not checked polygons with more sides. For a circular workspace, with corner angles of 180°, the worst-case configuration can only be approached asymptotically, as explained in Section VI. If the polygon is irregular, more moves may be required. The entire configuration space is reachable in four moves for an acute triangle, but obtuse triangles require six moves.

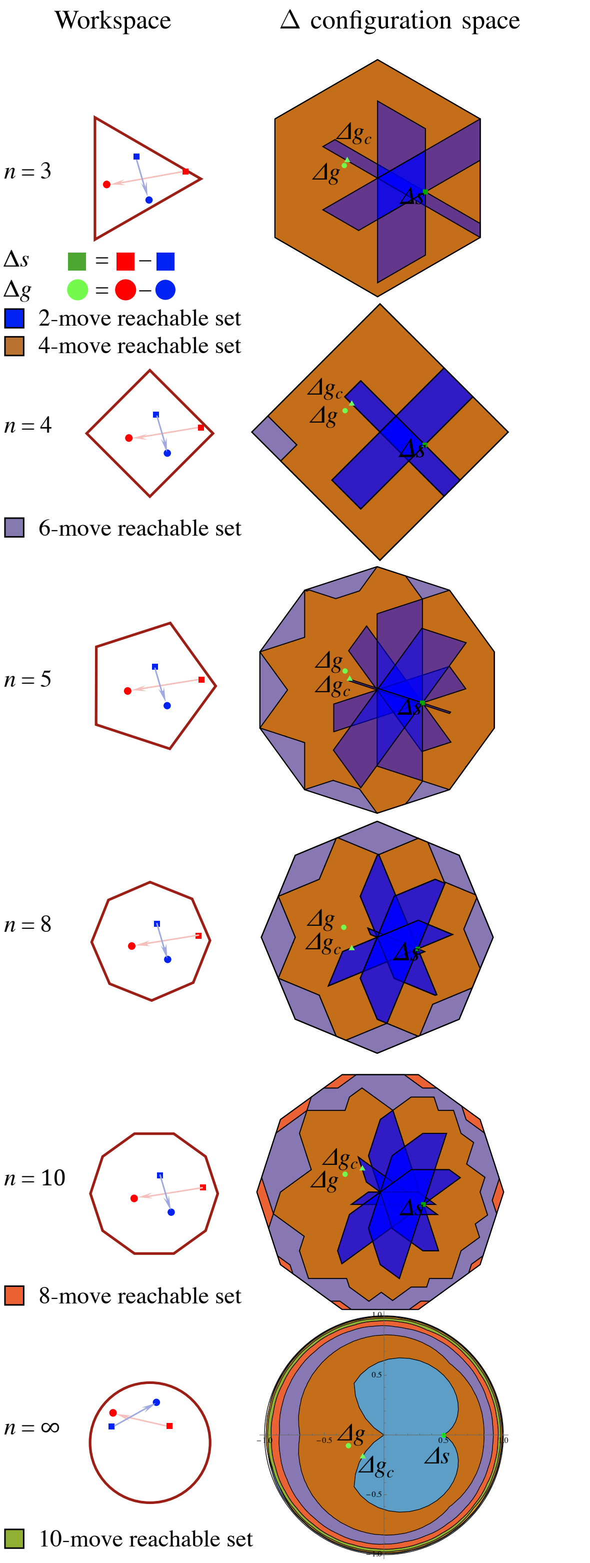


Fig. 4. The Δ configuration space is all possible configurations of $p_2 - p_1$. The sets reachable in two moves, called *2-move reachable sets*, are drawn with transparent blue polygons. A polygon with n sides has n 2-move reachable sets, but if n is even and the polygon is regular, half the reachable 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 (the closest point on the 2-move reachable set to Δg) and apply two moves to achieve Δg_c. We repeat this process until the relative goal position is achieved. Higher-number reachable sets are drawn in distinct colors.

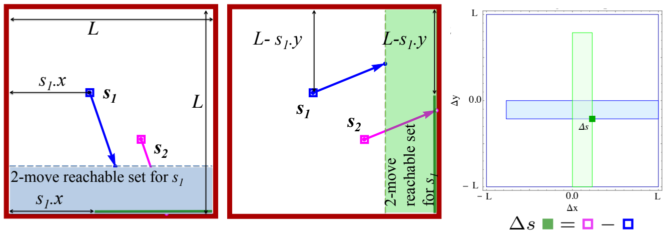


Fig. 5. Boundary interaction is used to change the relative positions of the particles. Each particle gets the same control input. (left) If particle 2 contacts the bottom wall before particle 1 reaches a wall, particle 2 can reach anywhere along the green line, and particle 1 can move to anywhere in the shaded area. (middle) Similarly, if particle 2 contacts 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) All 2-move reachable sets in the Δ configuration space.

Algorithm 2 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.
$R_{\text{SET}} \leftarrow \{\}$
for $p_i$ in $P$ do
    $p'_i \leftarrow s_1 + s_2 - p_i$
    $p'_{i+1} \leftarrow s_1 + s_2 - p_{i+1}$
    $L \leftarrow \overline{p'_i p'_{i+1}}$                ▷ line $(p'_i, p'_{i+1})$
    $l_i, l_{i+1} \leftarrow$ intersections of $L$ and polygon $P$
    if $p'_i$ not inside polygon $P$ then
        $p'_i \leftarrow l_i$
    if $p'_{i+1}$ not inside polygon $P$ then
        $p'_{i+1} \leftarrow l_{i+1}$
    $D \leftarrow s_2 - s_1 - ([l_i, v_{\min}, ..., p_i] - p'_i,$
        $[p_{i+1}, p_{i+2}, ..., v_{\max}, l_{i+1}] - p'_{i+1})$
    $R_{\text{SET}} \leftarrow$ Union of polygon $D$ and $R_{\text{SET}}$
Return $R_{\text{SET}}$
```

E. Circular Workspaces: 2-Move Reachable Set

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

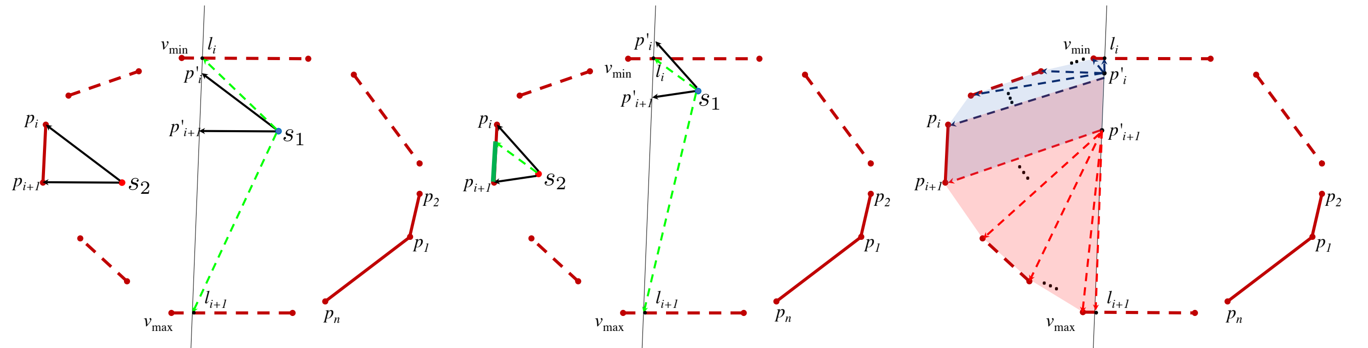
```
Require: knowledge of starting $(s_1, s_2)$ and goal $(g_1, g_2)$ positions of two particles.
Calculate $\psi_{\min}$ and $\psi_{\max}$          ▷ use (2)
Calculate $\gamma(\psi)$                             ▷ use (3)
Calculate $l_1, l_2, l_3, l_4$                       ▷ use (6)
Return the union of $(l_1, l_2, l_3, l_4)$
```

To compute the 2-move reachable set for a circular workspace, first consider all possible first contact locations. The set of boundary points that a particle can touch before the other particle touches are two arcs from $\psi_{\min}$ to $\psi_{\max}$ and from $\pi + \psi_{\min}$ to $\pi + \psi_{\max}$:

$$\psi \in [\psi_{\min}, \psi_{\max}] = \theta + \left[\sin^{-1}\left(\frac{d_{12}}{2r}\right) - \frac{\pi}{2}, \frac{\pi}{2} - \sin^{-1}\left(\frac{d_{12}}{2r}\right)\right], \quad (2)$$

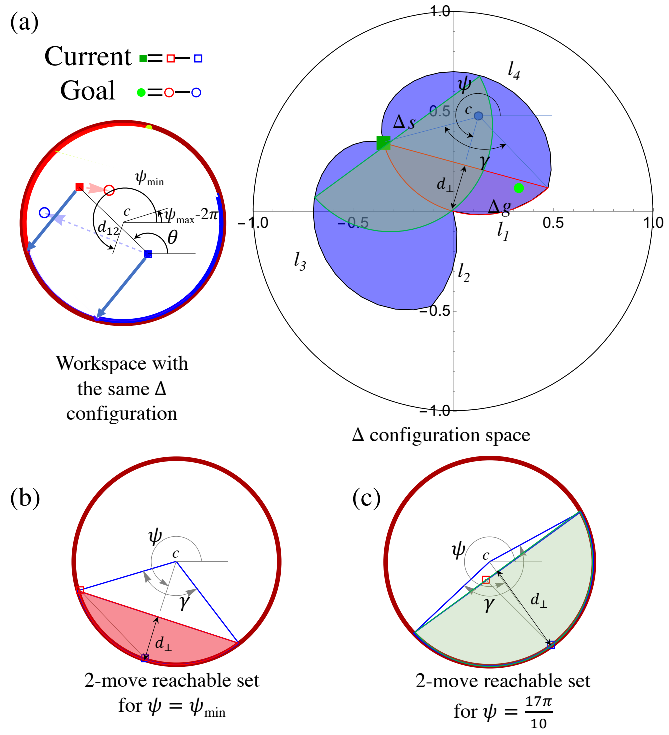
where $d_{12} = \|s_1 - s_2\|_2$, r is the radius of the workspace, and the angle between two particles is $\theta = \arctan(\frac{p_1.x - p_2.x}{p_1.y - p_2.y})$.

A circle has an infinite number of sides, thus infinite reachable sets. However, the 2-move reachable set can be



Case 1: s_1 does not contact boundary   Case 2: s_1 contacts boundary   Polygon generated by first contacts along $\overline{p_i p_{i+1}}$

Fig. 6. Steps to generate the 2-move reachable set when one particle collides with edge $\overline{p_i p_{i+1}}$ of a convex polygonal workspace.



Workspace with the same Δ configuration

Δ configuration space

2-move reachable set for $\psi = \psi_{\text{min}}$

2-move reachable set for $\psi = \frac{17\pi}{10}$

Fig. 7. (a) The possible first contact points for the blue and red particles are shown with blue and red arcs. (b) if the blue particle touches the wall at ψ_{min} (blue square) the other particle (red square) can move anywhere in the pink region. Right bottom: if the blue particle touches the wall at $\psi = \frac{17\pi}{10}$ (blue square) the other particle (red square) can move anywhere in the light green region. (c) The Δ configuration space for the corresponding 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. If the blue particle contacts the boundary at ψ_{min}, the reachable Δ configuration is the red set, or the green set if $\psi = \frac{17\pi}{10}$.

parameterized by the angle of first contact location ψ, as shown in Fig. 7.

Each ψ value generates a 2-move reachable set that is a chord of the disk, with interior angle γ parameterized by ψ:

$$\gamma(\psi) = \cos^{-1}\left(1 - \frac{d_\perp(\psi)}{r}\right), \text{ where:} \tag{3}$$

$$d_\perp(\psi) = 2\|s_1.p_\psi(\psi) - s_2.p_\psi(\psi)\|_2, \tag{4}$$

$$p_\psi(\psi) = r[\cos(\psi), \sin(\psi)]. \tag{5}$$

The 2-move reachable sets with π difference in ψ value are equivalent in the Δ configuration space. 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 $c = r(\cos(\psi - \pi), \sin(\psi - \pi))$. Two such chords are drawn in red and green in Fig. 7.

The equations for the four lines outlining the union of two-move reachable sets are as follows:

$$\begin{aligned} l_1 = r\Big(&\cos\psi_{\text{min}} - \cos(\gamma + \psi_{\text{min}}) \\ &+ \sin\psi_{\text{min}} - \sin(\gamma + \psi_{\text{min}})\Big) \qquad 0 < \gamma < \gamma(\psi_{\text{min}}), \\ l_2 = r\Big(&\cos\psi_{\text{max}} - \cos(\gamma + \psi_{\text{max}}) \\ &+ \sin\psi_{\text{max}} - \sin(\gamma + \psi_{\text{max}})\Big) \qquad \gamma(\psi_{\text{max}}) < \gamma < 0, \\ l_3 = r\Big(&\cos\psi - \cos(\psi + \gamma(\psi)) \\ &+ \sin\psi - \sin(\psi + \gamma(\psi))\Big) \qquad \psi_{\text{min}} < \psi < \psi_{\text{max}}, \\ l_4 = r\Big(&\cos\psi - \cos(\psi - \gamma(\psi)) \\ &+ \sin\psi - \sin(\psi - \gamma(\psi))\Big) \qquad \psi_{\text{min}} < \psi < \psi_{\text{max}}. \end{aligned} \tag{6}$$

We combine these boundaries to compute the 2-move reachable set summarized in Alg. 3. The motion-planner finds 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 ψ_{min} or ψ_{max} using the following two tests:

$$\begin{aligned} &(\Delta g_c.x - c.x)^2 + (\Delta g_c.y - c.y)^2 > r^2 \textbf{ and} \\ &(c.x - \Delta g_c.x)\cos\psi + (c.y - \Delta g_c.y)\sin\psi > r\cos\gamma. \end{aligned} \tag{7}$$

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:

$$\psi = \tan^{-1}\left(\frac{\Delta p.x - \Delta g_c.x}{\Delta p.y - \Delta g_c.y}\right). \tag{8}$$

The particles achieve Δg_c in two moves. The first move causes one particle to touch the wall at p_ψ, (5). The second move achieves the required relative position.

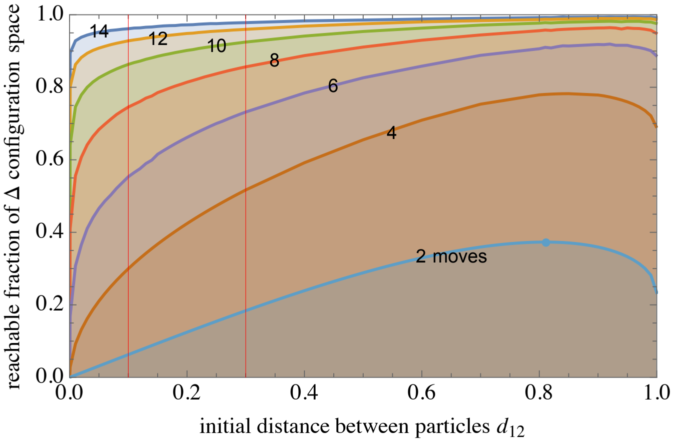


Fig. 8. Reachable fraction of Δ configuration space as a function of initial distance d_{12} for different numbers of total moves. Red vertical lines correspond to the reachable sets for $d_{12} = 0.1$ and 0.3 in Fig. 9.

F. Accessible Workspace

The Δ configuration enables an iterative method to compute the accessible workspace. Due to symmetry of the workspace, the fraction of the Δ configuration space reachable in $2k$ moves is a function of only the initial separation distance d_{12}. The angle θ between the initial particle positions simply rotates the reachable Δ configuration space. As long as the initial configurations are distinct ($s_1 \neq s_2$), the reachable set grows quickly. This relationship is shown in Fig. 8. Only antipodal locations are unreachable ($\|g_2 - g_1\| = 1$), but can be asymptotically approached. Indeed, even with a tiny initial separation of $d_{12} = 0.001$, after 14 moves 90% of the Δ configuration space is reachable. In two moves, the maximum reachable fraction of 0.373 is achieved with $d_{12} \approx 0.81$.

Two example sets of the reachable Δ configuration space for $d_{12} = 0.1$ and $d_{12} = 0.3$ are shown in Fig. 9. After two moves, $d_{12} = 0.1$ reaches only 6.3% of the Δ configuration space, but 30% in four moves, 55% in six moves, 75% in eight, 86% in ten, 93% in twelve, and 96% in fourteen moves. Though these images show reachable sets with initial particle-to-particle angle $\theta = 0$, all sets for other θ values can be formed by rotating these solutions by θ.

G. 3D workspaces: Cylinders and Prisms

This section presents a method for extending Alg. 1 to 3D in workspaces that are right cylinders or right prisms. For ease of analysis, we assume neutrally buoyant particles, and choose a coordinate frame so that the cylinder or prism has end caps in the xy plane. The following method assumes the two particles do not initially have the same x and y positions. The method does not use any contacts with the end caps, and so may be suitable for tubular lumens. 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 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. 10 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. Workspaces that are tortuous or with many obstacles are better handled by other path planners, such as RRT [**?**], or [**?**], which used collisions with protrusions of the workspace

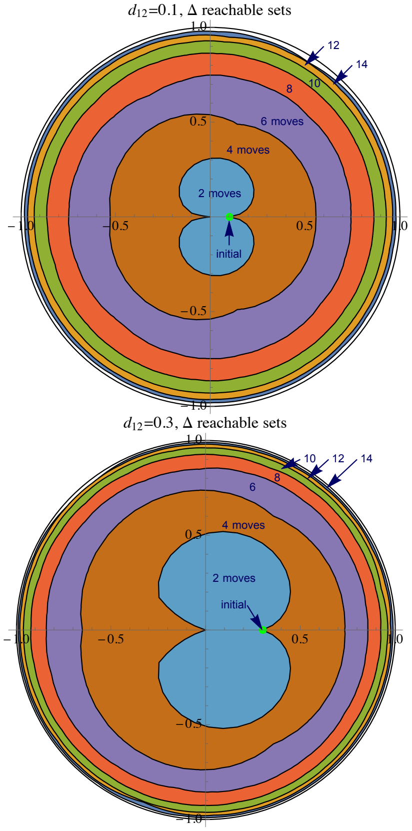


Fig. 9. Plots showing the 2, 4, 6, 8, 10, 12, and 14-move reachable sets in the Δ configuration space for $d_{12} = 0.1$ and 0.3. The numeric method used for plotting strictly underestimates the reachable set.

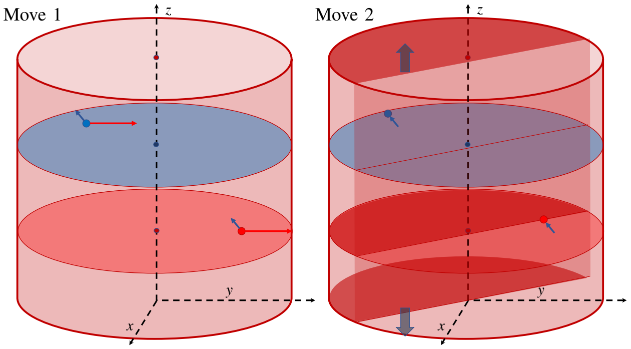


Fig. 10. Illustration on how boundary contacts on the sides of a tube enable 3D positioning. Once one particle contacts a boundary, the other particle's 2-move reachable set is a prism formed by extending the 2D 2-move reachable set in the $\pm z$ direction.

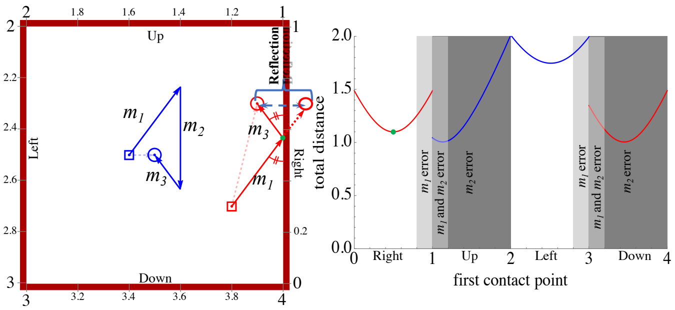


Fig. 11. In a square workspace, the shortest three-move path that reconfigures two particles from starting positions (□, □) to goal positions (○, ○) has the property that the incident angle equals the reflected angle, as shown at left. (right) The first contact is colored red if the red particle is the first to touch a boundary, and colored blue if the blue particle is the first to touch.

to rearrange particles.

V. Two Optimal Results

Algorithm 1 provided a technique to bring two particles to goal positions using global inputs, but did not optimize path length. Changing the relative positions of particles in any workspace requires making one particle contact the boundary. In this section we present two results that can be incorporated into Algorithms 2 and 3 to generate shorter motion paths.

A. 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 contact 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 contacts 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 contact location, as shown in Fig. 11. There are four walls, and four candidate solutions, but some candidate solutions may be invalid because a different boundary is hit before the desired first contact position in move m_1 (light grey regions) or invalid because m_2 cannot generate the goal relative spacing (dark grey regions).

B. Shortest Path in Unit Disk that Intersects Circumference

The shortest path between two points in the unit disk that intersects the circumference is composed of two straight line segments and has an optimal contact point, as shown in Fig. 12. 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 point E by a radius e and angle β: $E = e(\cos\beta, \sin\beta)$. Then the length of the two line segments is

$$\sqrt{(s-\cos\theta)^2 + \sin^2\theta} + \sqrt{(e\cos\beta - \cos\theta)^2 + (e\sin\beta - \sin\theta)^2}, \quad (9)$$

which is minimized by choosing an appropriate θ value.

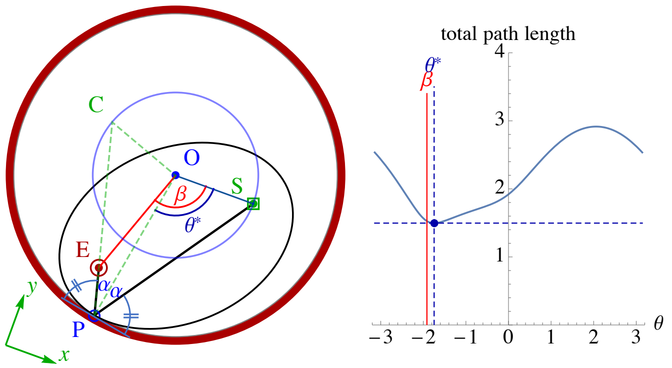


Fig. 12. The shortest path between two points S to E in the unit disk that intersects the circumference. The path length as a function of intersection point, $P = (\cos\theta, \sin\theta)$ is shown at right. See [**?**].

The length of the two line segments as a function of θ is drawn in the right plot of Fig. 12. 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 $-\pi \leq \beta \leq \pi$. We evaluate each and select the solution that results in the shortest length path. For an interactive Mathematica demonstration of this shortest path, see [**?**]. Because the closed form solution is long, it is included in the paper attachments.

VI. Simulation

Algorithm 1 was implemented in Mathematica using particles with zero radius. Figure 13 shows frames of the algorithm in two representative workspaces, square and disk, with two arbitrary starting and goal configurations.

The contour plots in Fig. 14 left show the length of the path for two different settings. The top row considers $\{s_1, s_2, g_1\}$ = $\{(0.2, 0.2), (-0.1, -0.1), (0, 0)\}$ and the bottom row considers $\{s_1, s_2, g_1\} = \{(0.2, 0.2), (-0.1, -0.1), (-0.2, 0)\}$, each in a workspace with $r = 0.5$, and g_2 ranging over all the workspace. Fig. 14 left shows the number of moves and right shows the total distance of the path. This plot shows the nonlinear nature of the path planning. When the goal is in

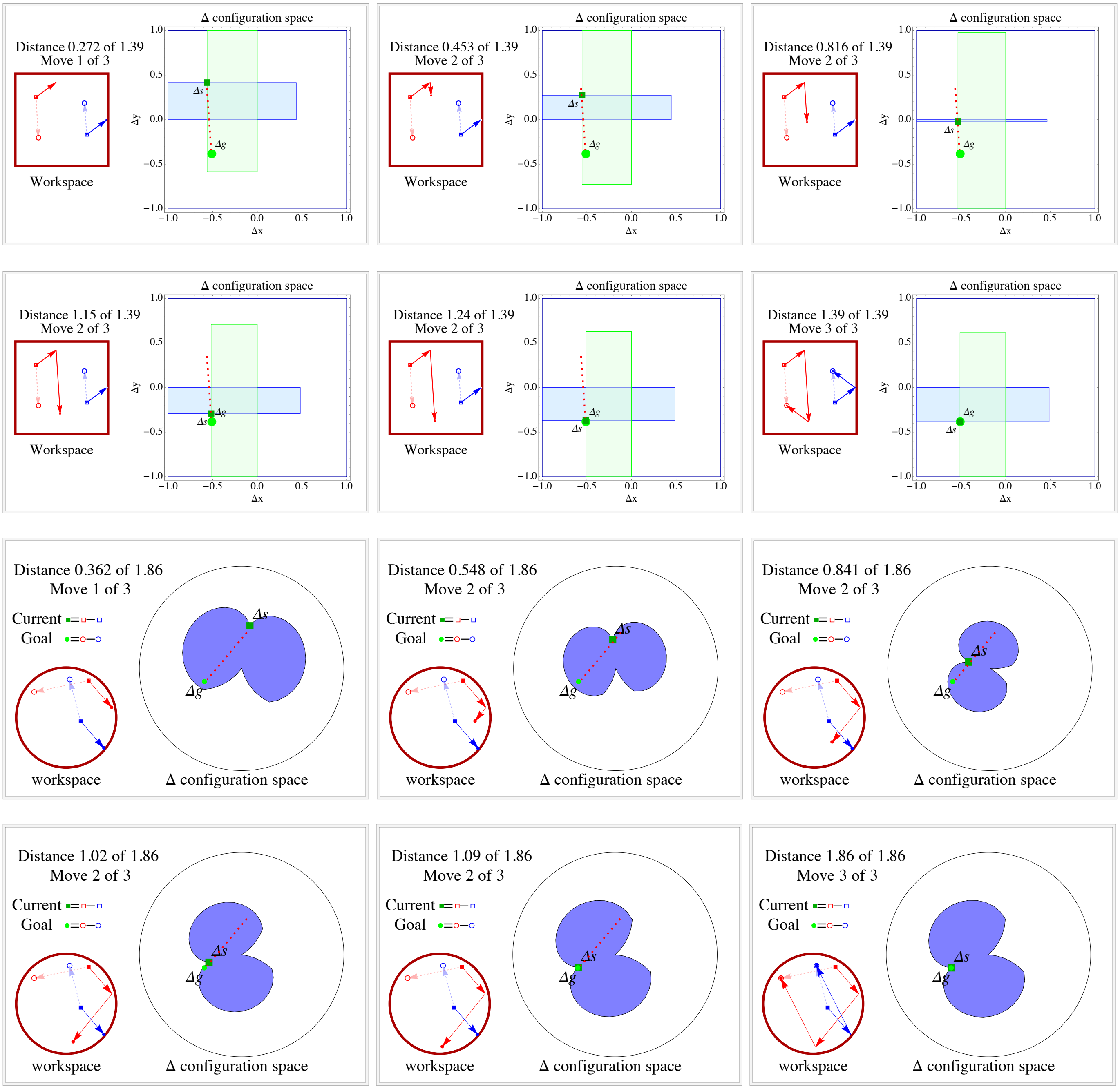


Fig. 13. Frames from reconfiguring two particles. Top six images show a polygonal workspace and the corresponding Δ configuration space with its 2-move reachable sets. Bottom six images show a disk-shaped workspace and the corresponding Δ configuration space with its 2-move reachable sets. For each, moves 1 and 3 are simple translations of both particles and so the reachable sets do not change. The reachable set morphs during move 2 because one particle is held stationary by the boundary. See multimedia attachment for animations of each.

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 breaks the symmetry of the path length in the workspace.

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. 15, which plots the smallest achievable δ radius about each goal position as a function of path length. Figure 16 shows the same concepts in a square workspace. Figure 16 top and middle row considers the particles for three arbitrary starting and goal positions for the particles.

Thus far, this paper has considered the particles to be unique. If particles are interchangeable, the path lengths often decrease, which can be computed by running Alg. 1 twice, but swap the goal positions for the second run and select the shortest path. The bottom row of Fig. 16 considers interchangeable particles with the same configuration as the middle row with unique particles. The worst-case path lengths decrease by 33%, 60%, and 30% for the three test cases shown.

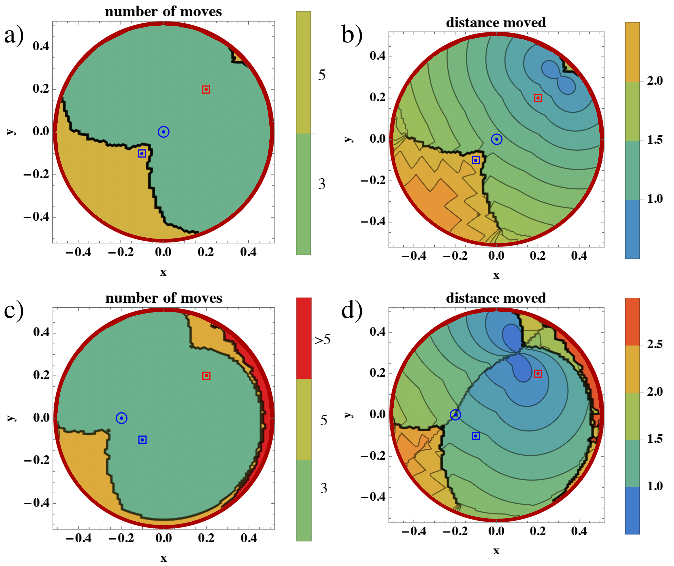


Fig. 14. Circular workspace: contour plots showing the number of moves and distance commanded if red particle's goal position is varied in x and y. Starting positions of red and blue particles (□, □) and goal position of blue particle ○ are fixed. The top row has the blue particle's goal position at the origin, generating symmetric contour plots. Moving the blue particles' goal position to $(-0.2, 0)$, generates non-symmetric contour plots.

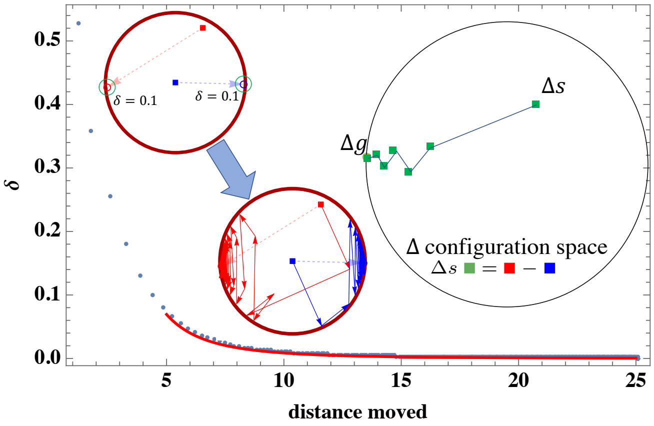


Fig. 15. Circular workspace: the worst-case path length occurs when particles must swap antipodes. This can never be achieved but can be asymptotically approached. Plot shows decreasing error (radius δ around goal positions) as the number of moves grows. Red fit line is $8.66/(\text{distance}^3)$, which has an R-squared value of 0.77.

VII. 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 × 494 pixel camera was attached to the top of the system, focusing on the workspace which was backlit by a 15 W LED light strip.

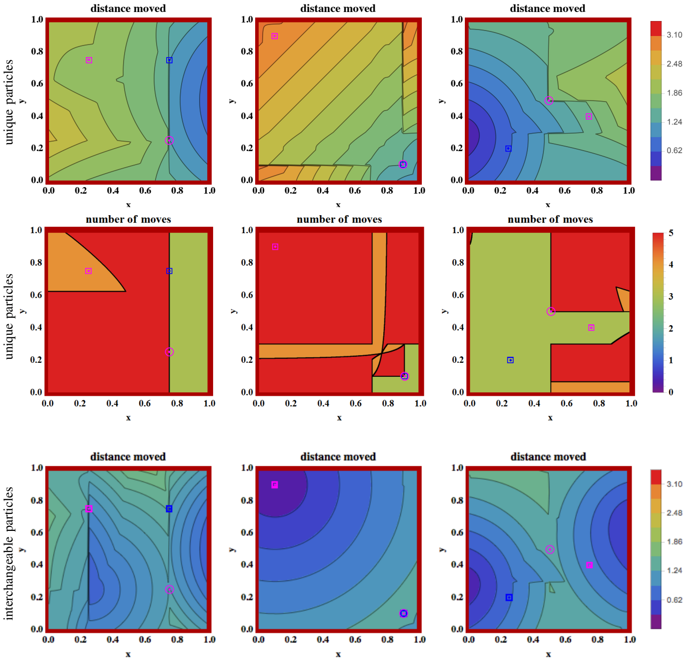


Fig. 16. Square workspace: starting positions of particles 1 and 2 (□, □) and goal position of particle 2 (○) are fixed, and $\epsilon = 0.001$. The top row of contour plots show the distance if particle 1's goal position is varied in x and y. The middle row shows the number of moves required for the same configurations. The bottom row shows the same configuration but when the particles are interchangeable.

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 mm magnets (supermagnetman #SP0100-50) were used as our particles. Our experimental setup did not perfectly implement the system dynamics in (1). In particular, the magnetic field in this setup is only approximately uniform. The magnetic force is increasingly nonuniform as distance from the center increases in both magnitude and orientation. As shown in the video attachment, this non-uniformity causes the particle closer to the coil to move faster than the other particle. This phenomenon makes it easier to increase particle separation than to decrease separation, but this can be compensated because boundary collisions easily decrease the separation. Also, magnetic forces are not exactly parallel, but point toward the center of the activated coil. Algorithm 1 still works despite 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. To model the geometry of intestinal villi, the model consists of a circular ring laser cut from 6 mm thick acrylic. The ring has an outer diameter of 50 mm, an inner diameter of 46 mm, and sixty

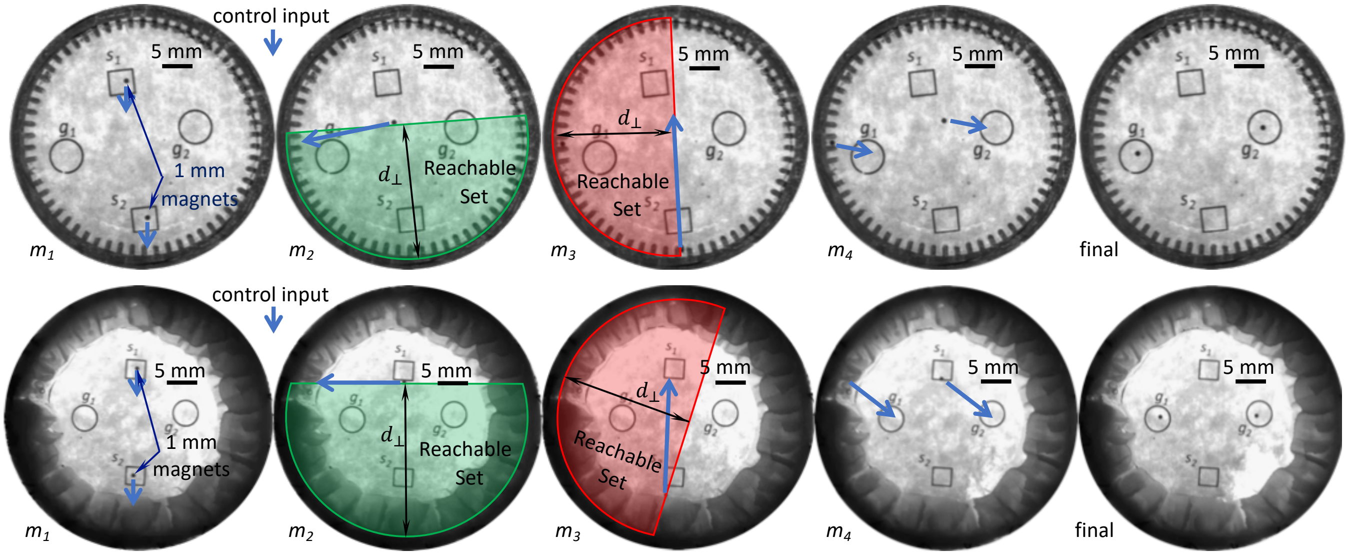


Fig. 17. Frames showing particle positions before and after control inputs. Top row: small intestine phantom. Bottom row: cow stomach tissue.

2 mm long protrusions on its inner surface. The top row of Fig. 17 shows one experiment. Starting and ending positions were printed beneath the workspace on transparency film. Our algorithm successfully delivered the particles to goal positions in 10 out of 10 trials. A video showing one trial of this experiment is available in the supplementary materials.

C. Bovine Stomach Cross-section

Strips of cow stomach approximately 5 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. 17 bottom row. Our algorithm successfully delivered the particles to goal positions in 5 out of 5 trials. A video showing one trial of this experiment is available in the supplementary materials.

VIII. Conclusion and Future Work

This paper presented techniques for controlling the positions of two particles using uniform inputs and non-slip boundary contacts. 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 beyond those mentioned previously. 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 [**?**] and [**?**] could be applied to extend the analysis to more than two particles.

Appendix

We thank our anonymous reviewers for their advice which improved the presentation and content of this paper, especially the analysis of the reachable sets.

The supplemental material includes four videos:

- Video *01Model.mov* animates the concepts of uniform control inputs and non-slip boundary contacts.
- Video *02DeltaConfigurationSpace.mp4* demonstrates how the Δ configuration space is constructed for a variety of workspaces.
- Video *03SimulationWorkSpaces.mp4* shows demonstrations of motion planning in square and disc-shaped workspaces.
- Video *04Hardware Experiments.mp4* shows an experiment trial moving two particles to goal positions in a small intestine phantom, then an experiment trial using cow stomach tissue.

We also include four Mathematica Notebooks (.nb files) containing simulation code.

- *SquareWorkSpace.nb* generates paths in a square workspace for two particles.
- *CircularWorkSpace.nb* generates paths in a circular workspace for two particles.
- *DeltaConfigurationSpacePolygon.nb* generates the Δ configuration space for convex polygons
- *ShortestPathForADisk.nb* analytical solution for shortest path that touches a boundary from one position to another position in a circular workspace.

Shiva Shahrokhi (S'14) received M.S. and PhD degrees in the Electrical and Computer Engineering Department at the University of Houston in 2018. She received the B.S. degree in Computer Engineering from Iran University of Science and Technology, Tehran, Iran. She is involved in studies of swarm manipulation and control for robotics applications. These include steering massively under-actuated mobile robotic ensembles, swarm shape control, and machine learning.

Jingang Shi Jingang Shi received the B.Eng. degree in the School of Materials Science and Engineering at Huazhong University of Science and Technology in 2018. He studied robot control during a four-month internship at the University of Houston in 2017. He is studying for the M.E. degree in the School of Mechanical Science and Engineering at Huazhong University of Science and Technology.

Benedict Isichei received an M.S. degree in the Electrical and Computer Engineering Department at the University of Houston in 2018. His thesis was titled “Robotic Manipulators: Cryogenic Magnetic Manipulation System, Two-body Magnetic Manipulation, and Low-cost Robotic Manipulators”, and features low cost robotics. See his designs at https://www.thingiverse.com/Bisichei.

Aaron T. Becker (S‘06-M‘12-SM‘17) received M.S. and PhD degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 2008 and 2012. He was a post doctoral research scholar at the Multi-Robot Systems Lab at Rice University and at Boston Children’s Hospital & Harvard Medical School, before joining the Electrical and Computer Engineering Department at the University of Houston as an Assistant Professor in 2014. He won the best paper at IROS 2013 and received the NSF CAREER award in 2016.