# *Supplement to* Shaping a Swarm With a Shared Control Input Using Boundary Walls and Wall Friction

Paper-ID [add your ID here]

***Abstract*—Includes algorithms and equations too lengthy for main paper, but potentially useful for the community.**

## I. Calculations for Modeling Swarm as Fluid in a Simple Planar Workspace

Two workspaces are used, a square and a circular workspace.

### A. Square Workspace

This section provides formulas for the mean, variance, covariance and correlation of a very large swarm of robots as they move inside a square workplace under the influence of gravity pointing in the direction $\beta$. The swarm is large, but the robots are small in comparison, and together cover an area of constant volume $A$. Under a global input such as gravity, they flow like water, moving to a side of the workplace and forming a polygonal shape. The workspace is

The range of possible angles for the global input angle $\beta$ is [0,2$\pi$). In this range of angles, the swarm assumes eight different polygonal shapes. The shapes alternate between triangles and trapezoids when the area $A$<1/2, and alternate between squares with one corner removed and trapezoids when $A$>1/2.

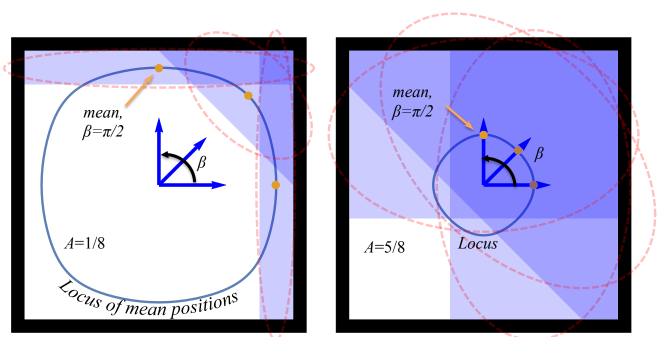


Fig. 1. A swarm in a

### B. Circle Workspace

## II. Algorithm for Generating Desired $y$ Spacing Between Two Robots Using Wall Friction

**Algorithm 1** GenerateDesired$y$-spacing$(s_1, s_2, e_1, e_2, L)$

**Require:** Knowledge of starting $(s_1, s_2)$ and ending $(e_1, e_2)$ positions of two robots. $(0, 0)$ is bottom corner, $s_1$ is rightmost robot, $L$ is length of the walls. Current position of the robots are $(r_1, r_2)$.

**Ensure:** $r_{1x} - r_{2x} \equiv s_{1x} - s_{2x}$

$\Delta s_y \leftarrow s_{1y} - s_{2y}$
$\Delta e_y \leftarrow e_{1y} - e_{2y}$
$r_1 \leftarrow s_1$, $r_2 \leftarrow s_2$
**if** $\Delta e_y < 0$ **then**
  $m \leftarrow (L - \max(r_{1y}, r_{2y}), 0)$ ▷ Move to top wall
**else**
  $m \leftarrow (-\min(r_{1y}, r_{2y}), 0)$ ▷ Move to bottom wall
**end if**
$m \leftarrow m + (0, -\min(r_{1x}, r_{2x}))$ ▷ Move to left
$r_1 \leftarrow r_1 + m$, $r_2 \leftarrow r_2 + m$ ▷ Apply move
**if** $\Delta e_y - (r_{1y} - r_{2y}) > 0$ **then**
  $m \leftarrow (\min(|\Delta e_y - \Delta s_y|, L - r_{1y}), 0)$ ▷ Move top
**else**
  $m \leftarrow (-\min(|\Delta e_y - \Delta s_y|, r_{1y}), 0)$ ▷ Move bottom
**end if**
$m \leftarrow m + (0, \epsilon)$ ▷ Move right
$r_1 \leftarrow r_1 + m$, $r_2 \leftarrow r_2 + m$ ▷ Apply move
$\Delta r_y = r_{1y} - r_{2y}$
**if** $\Delta r_y \equiv \Delta e_y$ **then**
  $m \leftarrow (e_{1x} - r_{1x}, e_{1y} - r_{1y})$
  $r_1 \leftarrow r_1 + m$, $r_2 \leftarrow r_2 + m$ ▷ Apply move
  **return** $(r_1, r_2)$
**else**
  **return** GenerateDesired$y$-spacing$(r_1, r_2, e_1, e_2, L)$
**end if**