Lab 3: Simulation and Control of Multi-Robot Systems
By: Dr. Randy C. Hoover
Date Assigned: October 11, 2019
Date Due: Friday October 18, 2019
You may team up with another student to work on this particular lab.
Submit a single report with both partners names on it.

1 Introduction

In this lab we will investigate the n-robot rendezvous problem, restricting our attention (for now) to operation on $\mathbb{R}^2$. Recall, in class we discussed the problem of a two robot system planning to "meet up" in their operating space. Our system dynamics were derived as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

where the state vector $\mathbf{x} = [x_1 \;\; x_2]^T$ is the position of each robot restricted to operation on $\mathbb{R}^1$, and we can directly control the velocity. Considering a similar problem, the n-robot problem (depicted graphically in Figure 1) on $\mathbb{R}^2$ we derived the equations of motion as:

$$\begin{aligned} \dot{x}_1 &= u_{1x} \\ \dot{y}_1 &= u_{1y} \\ \dot{x}_2 &= u_{2x} \\ \dot{y}_2 &= u_{2y} \\ \dot{x}_3 &= u_{3x} \\ \dot{y}_3 &= u_{3y} \\ &\vdots \\ \dot{x}_n &= u_{nx} \\ \dot{y}_n &= u_{ny} \end{aligned}$$

where we assume we can measure the position and control the velocity. Furthermore, each robot is free to move in any direction within $\mathbb{R}^2$ without consequence of collision with other agents.

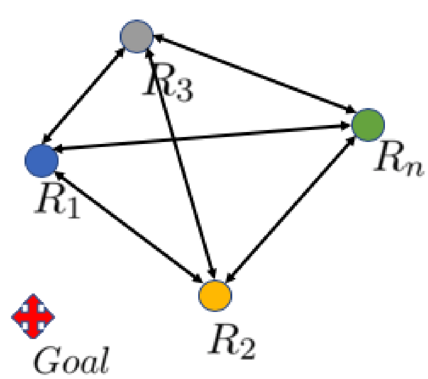

Figure 1: Graphical depiction of the n-robot rendezvous problem.

2 Deliverables

2.1 Modeling

We have developed the mathematical model of the n-robot problem above but you are to design different control based on the final configurations listed below.

2.2 Design Configurations

Using MATLAB, design and simulate your multi-robot rendezvous control for n robots assuming the following:

1. To get started, assume $n = 4$ and assume each robot can actually measure their distance with respect to a desired goal-state. For this simulation, let the goal be located at $(x_d, y_d = (5, 5)$ and let the initial positions of each robot be determined by a random number generator (you can assume the "world" coords are bounded by $(-20, 20)$ in both the x and y directions (i.e., you should generate random values between $(-20, 20)$ for the initial location of each of the n robots.

2. Once you have this working to your satisfaction, now assume the robots have no way of measuring the world around them, but they can measure where other robots are with respect to themselves (within some radius specified by the sensors they carry - in this case, they can measure all robot locations within 5 units at all times). Again using random initial starting configurations for $n = 4$ robots, design a control law for them to all converge to the same *unknown* (x, y) location.

3. In many situations in robotic swarms, we use a leader-follower approach where one robot is considered the "leader" and the others configure into a particular formation behind the leader. Let $n = 5$ robots and configure them to form a delta shape (again from random starting configurations) where the top of the delta is the leader position. Once you have this accomplished, can you transition them from a delta, to a diomond formation and back again (as illustrated in Figure 2?

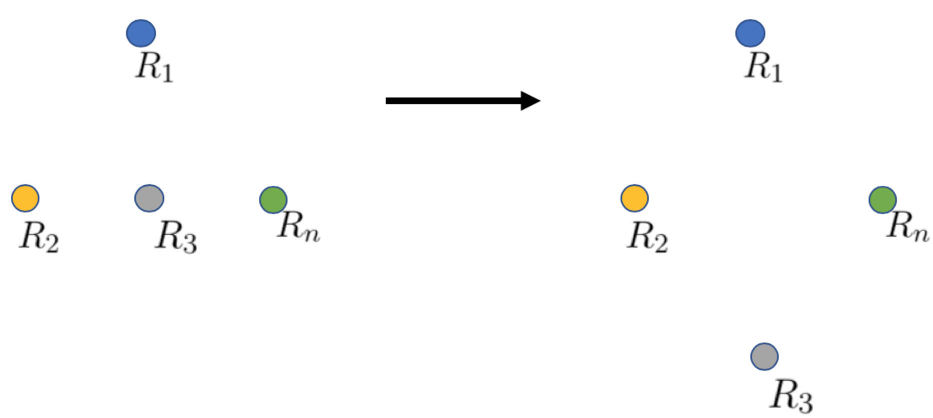

Figure 2: Leader-follower formation control.

2.3 Submission

Submit in a single .pdf your code along with simulation plots illustrating the simulations (both simulation and design) verifying your control does indeed enable the quad to track a desired reference altitude. **You should use the MATLAB publisher for this task $\longrightarrow$ change the settings from html (default) to pdf.**.

In addition, you will need to create a Matlab video illustrating your working product for each deliverable - there are many tutorials online about how to create videos from your simulation plots via MATLAB.