MAE 598: Multi Robot Systems Final Project

Multi Robot Object Transport Using Potential Field and Symmetric Formation Control.

Team: Gautham Manoharan(A), Vikramaditya Nanhai (B), Parth Borse (C), Nichol Rodrigues(D)

I. Abstract

Multi robot systems have been extensively studied for their applications in various fields like search and rescue, construction, medical, agriculture and object transport in the past decade. This report focuses on multi robot object transport using the potential field and formation control approach. A brief review of the state of the art can be found in [1] and [2]. In [1] object transport has been classified as pushing, caging and grasping. In this project we have divided the process of object transport in three steps: 1) Approach, 2) Formation Control and 3) Transport. We have used the potential field equations [3] for the approach section which also includes obstacle avoidance. Formation control is applied using graph theory and minimum communication between robots. Final transport mode uses the same attractive potential field which is comparatively weaker than the Approach mode. In section II, the mathematical model is presented along with the parameters and constants for potential fields and formation control. In section III properties of these models are discussed and verified. MATLAB simulation is done in section IV and the results are verified. The formation control and transport to goal position is simulated in the 2-D Robotarium simulator.

II. Mathematical Model

Our analysis includes mathematical models for attractive potential fields for the go to object task. We integrate this model with an obstacle avoidance repulsive potential field model for the approach mode. Along with this we have also applied inter robot collision avoidance. The number of robots required to trap the object is also studied for various shapes. We have defined nodes and edges for formation control and derived the laplacian matrix and distance matrix based on the desired formation. A different attractive potential field equation with lower gain is used for transportation.

Potential Field Model (B and C)

We have used the following equation for attractive potential field:

$$V_{ig}^{a} = \frac{1}{2} k_{ig} r_{ig}^{2} \tag{1}$$

where $k_{ig}$is a constant and $r_{ig} = ||r_i - r_g||$, which is the distance between the i_{th} robot and the final goal point of that robot around the object. This model produces an attractive potential field which guides each robot to a predetermined fixed location around the object. The goal point varies for each robot depending on the position it is supposed to reach.

Repulsive potential is defined for obstacle avoidance using the following equation:

$$V_{io}^{r} = \frac{k_{io}}{r_{io}} \tag{2}$$

In this equation $r_{io}$is the distance between the obstacle and the i_{th} robot. The same equation is used for inter robot collision avoidance where the distance between the i_{th} and j_{th} robot is considered.

The gradient of the potential field is calculated to decide the direction of movement of the robot which is given as:

$$Gr = \left[\frac{\partial V}{\partial x}\ \frac{\partial V}{\partial y}\right]^{T}$$

where V is the potential field. The forces on the robot in x and y directions are given by:

$$Fx = -\frac{\partial V(x,y)}{\partial x}$$

$$Fy = -\frac{\partial V(x,y)}{\partial y}$$

The robot will follow the direction with the least value of the potential field. It continuously searches for the minimum value. We first look at the attractive potential field equation and derive the forces in x and y direction. Here x_g and y_g are the x and y coordinates of the goal or object. Here $r_{ig} = \sqrt{(x - x_g)^2 + (y - y_g)^2}$

$$Fxg = -\frac{\partial V_{ig}^{a}(x,y)}{\partial x} = -\frac{\partial(\frac{1}{2}k_{ig}r_{ig}^{2})}{\partial x} = k_{ig}(x_g - x) \tag{3}$$

$$Fyg = -\frac{\partial V_{ig}^{a}(x,y)}{\partial y} = -\frac{\partial(\frac{1}{2}k_{ig}r_{ig}^{2})}{\partial y} = k_{ig}(y_g - y) \tag{4}$$

To calculate forces on the robot due to the obstacles we follow the repulsive potential field which is shown in equation (2).

$$Fxo = -\frac{\partial V_{io}^{r}(x,y)}{\partial x} = -\frac{\partial(\frac{k_{io}}{r_{io}})}{\partial x} = \frac{k_{io}(x_o - x)}{r_{io}^{3}} \tag{5}$$

$$Fyo = -\frac{\partial V_{io}^{r}(x,y)}{\partial y} = -\frac{\partial(\frac{k_{io}}{r_{io}})}{\partial y} = \frac{k_{io}(y_o - y)}{r_{io}^{3}} \tag{6}$$

The total force in x direction is:

$$F_x = Fxg + Fxo1 + Fxo2 + Fxo3 \tag{7}$$

The total force in y direction is:

$$F_y = Fyg + Fyo1 + Fyo2 + Fyo3 \tag{8}$$

Three obstacles are considered in our case which are numbered as 1, 2 and 3. The repulsive forces for each are represented as (Fxo1, Fyo1), (Fxo2, Fyo2) and (Fxo3, Fyo3).

Controller design (A and B)

We design a Proportional-Integral-Derivative (PID) controller for the differential drive robot with unicycle dynamics in our simulations. This design is robust to input step disturbances (bias) and follows performance specifications we set on our controller which will be described in detail in the theoretical analysis.

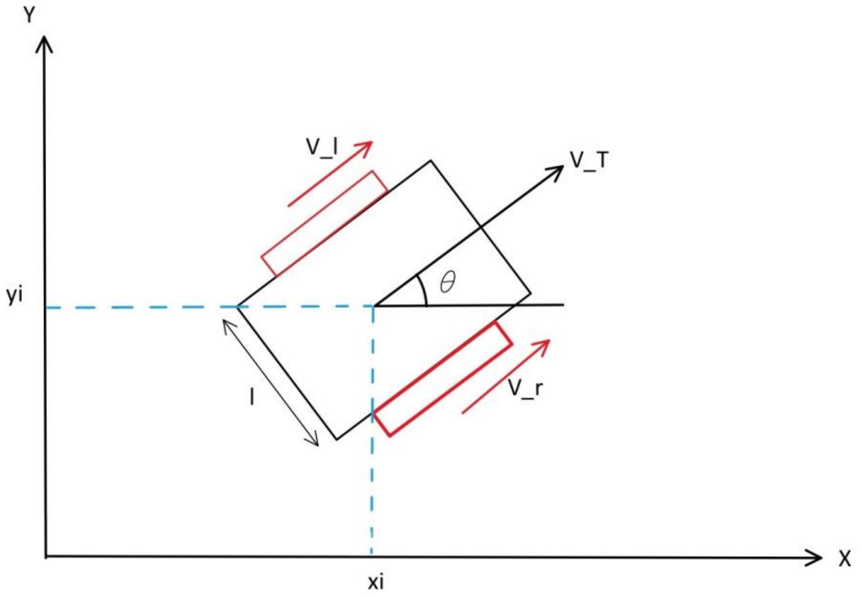


Figure 1: Differential drive robot pose and velocity.

Velocity for the left and right wheel is v_l and v_r , the translation velocity is v_t and the heading angle between the x axis and v_t is θ. For a robot with heading θ, position given by vector p and desired heading angle θ_{des}, the controller is defined as follows with proportional, integral, and derivative terms.

$$u = K_P(\theta_{des} - \theta) + K_I \int_0^t (\theta_{des} - \theta) + K_D\left(\dot{\theta_{des}} - \dot{\theta}\right)$$

$$|\dot{p}| = v_T \Rightarrow \begin{cases} \dot{x} = v_T \cos(\theta) \\ \dot{y} = v_T \sin(\theta) \end{cases}$$

K_P, K_I, K_D are the proportional, integral, and derivative controller coefficients respectively. To determine θ_{des} we use the gradients of the artificial potential field.

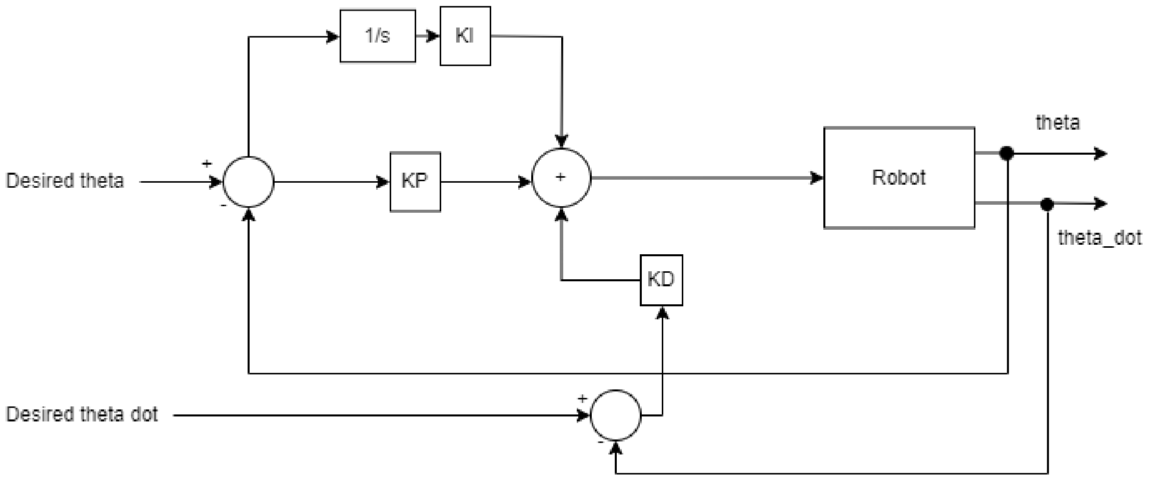


Figure 2: Block diagram of controller design

$$\theta_{des} = tan^{-1}(\frac{F_y}{F_x})$$

The discrete time implementation of the controller in the MATLAB simulations is as follows. An additional term $eI(t)$is computed for the integral term.

$$e(t) = \theta_{des}(t) - \theta(t)$$
$$eI(t) = eI(t - \Delta t) + \Delta t * e(t)$$
$$u = K_P e(t) + K_I eI(t) + K_D \frac{\left(e(t) - e(t - \Delta t)\right)}{\Delta t}$$

Formation control (B and D)

The formation control is inspired by the graph network theory. Each robot is defined as a node. After the approach mode is completed, we switch to the formation controller. The communication between each robot is defined by the edges and the inter-robot distances. The graph is not directed and is derived in such a way that minimum communication is needed to maintain a robust rigid formation.

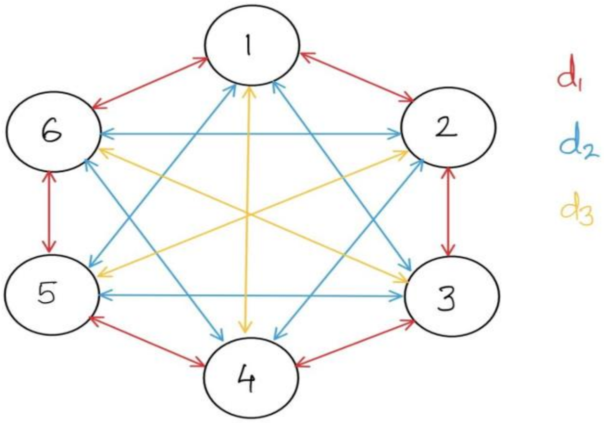


Figure 3: Graph and edges between the robots in desired formation

This graph network is used to entrap the object using 6 robots where each of them acts as a node. Every robot communicates with all other robots to maintain fixed distance between them. The value for the distances is d, d_2 and d_3 as shown in figure 3. These are implemented in the algorithm using the distance matrix W. The velocities in x and y direction for the ith robot are controlled in an algorithm. The Laplacian matrix which represents the formation is given as:

$$L = \begin{bmatrix} 5 & -1 & -1 & -1 & -1 & -1 \\ -1 & 5 & -1 & -1 & -1 & -1 \\ -1 & -1 & 5 & -1 & -1 & -1 \\ -1 & -1 & -1 & 5 & -1 & -1 \\ -1 & -1 & -1 & -1 & 5 & -1 \\ -1 & -1 & -1 & -1 & -1 & 5 \end{bmatrix}$$

The distance matrix is represented as:

$$W = \begin{bmatrix} 0 & d & d_2 & d_3 & d_2 & d \\ d & 0 & d & d_2 & d_3 & d_2 \\ d_2 & d & 0 & d & d_2 & d_3 \\ d_3 & d_2 & d & 0 & d & d_2 \\ d_2 & d_3 & d_2 & d & 0 & d \\ d & d_2 & d_3 & d_2 & d & 0 \end{bmatrix}$$

The following mathematical model is used for the formation control from [4].

$$\begin{bmatrix}\dot{x}_i\\ \dot{y}_i\end{bmatrix} = \sum_{j=1}^{N} G\left(\left|\left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 - W_{ij}\right)\right|\right) \times \begin{bmatrix}x_j - x_i\\ y_j - y_i\end{bmatrix} \dots (9)$$

In equation (9), (x_i,y_i) are the x and y coordinates of the ith robot, and (x_j,y_j) are the coordinates of the jth neighbor of the ith robot. The velocity is initially set to zero and at each time step velocity is added to the initial velocity till the desired formation is reached. Once control of the desired formation has been achieved, another velocity vector is added to equation (9), which is dependent on the position of the centroid of the formation.

$$\begin{bmatrix}\dot{x}_i\\ \dot{y}_i\end{bmatrix} = \sum_{j=1}^{N} G\left(\left|\left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 - W_{ij}\right)\right|\right) \times \begin{bmatrix}x_j - x_i\\ y_j - y_i\end{bmatrix} + \frac{\begin{bmatrix}x_c - x_g\\ y_c - y_g\end{bmatrix}}{t_s}$$

where, (x_c, y_c) are the coordinates of the centroid, (x_g, y_g) are the coordinates of the goal point, and t_s is the time interval. We can find the coordinates of the centroid by averaging the coordinates of the robots:

$$\begin{bmatrix}x_c\\ y_c\end{bmatrix} = \frac{\sum_{i=1}^{N}\begin{bmatrix}x_i\\ y_i\end{bmatrix}}{N}$$

In the case of the current experiment setup, N=6, so the centroid is the center of a regular hexagon.

Given the velocity vector between the centroid and the goal point, we can drive the formation towards the goal point. As seen in the simulations conducted [4], we can verify that the system reliably does this.

III. Theoretical Analysis

Controller Stability (A and B)

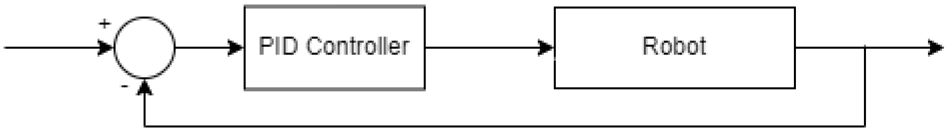


Figure 4: Simplified block diagram

To simplify the design process of the controller, the block diagram in figure 2 is simplified to the diagram in figure 4 through block diagram reduction techniques. The input to the robot is the angular velocity and the output is the angle of heading, hence the Robot in the block diagram can be simplified to an integrator. The PID Controller is represented in the Laplace domain as follows:

$$PID = K_{eff} = K_P + \frac{K_I}{s} + K_D s$$

The closed loop transfer function is given by:

$$\frac{\dfrac{K_P s + K_I + K_D s^2}{1 + K_D}}{s^2 + (\dfrac{K_p}{1 + K_D})s + (\dfrac{K_I}{1 + K_D})}$$

The steady state gain of the closed loop transfer function is 1. This transfer function follows an approximate second order dynamics. This is so because of the presence of zeros in the transfer function. To achieve desired behavior, we have set some desired parameters like the overshoot % < 10 %, settling time should be 2 s. Comparing the closed loop transfer function with standard second order transfer function we find values for $\omega_n = 3.33$.

For the initial design, we assign the value for K_P as 5 and calculate corresponding K_I, and K_D. But since the system is an approximate second order system, we further tune the value for controller coefficients to match our design specifications. Finally, we get the values of K_P= 8, K_I=13.9 and K_D=0.25. After using this in the controller the overshoot was 11% and settling time was less than 2 s as can be seen in figure 5.

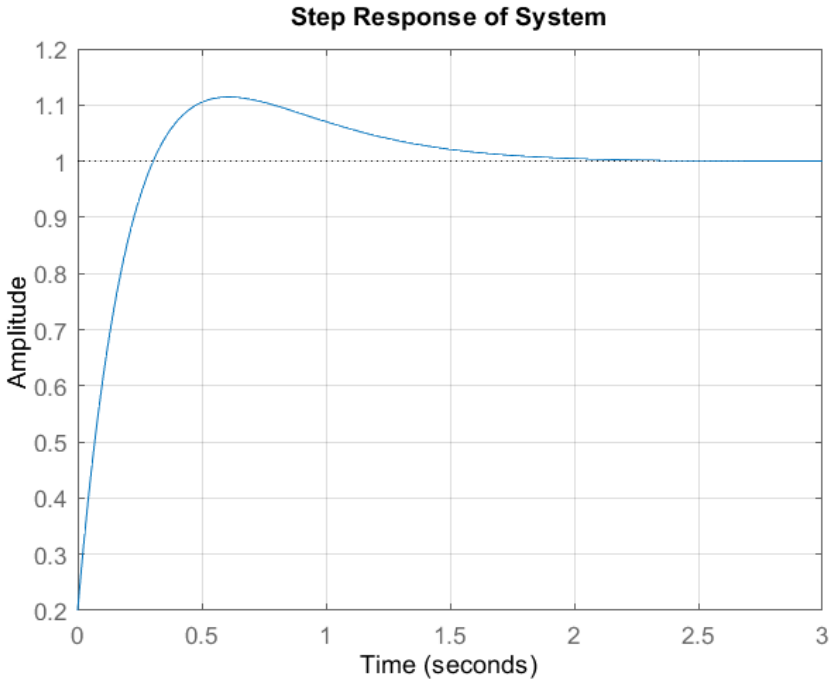


Figure 5: Step response plot

To prove the error rejection capability of the controller, we invoke final value theorem [5]. The transfer function from the input error d_i to the output $\theta$is given as follows.

$$\frac{\Theta(s)}{D_i(s)} = \frac{1/s}{1 + \frac{K_D s^2 + K_P s + K_I}{s^2}} = \frac{s}{(1 + K_D)s^2 + K_P s + K_I}$$

For a step disturbance $D_i(s) = \frac{1}{s} \Rightarrow \Theta(s) = \frac{1}{(1+K_D)s^2+K_P s+K_I}$

$$\lim_{t\to\infty}\theta(t) = \lim_{s\to 0} s\Theta(s) = \lim_{s\to 0}\frac{s}{(1 + K_D)s^2 + K_P s + K_I} = 0$$

This proves that the disturbance is rejected by the system in the steady state.

Local minima in Potential Field (C)

One of the major problems faced in potential field-based approaches is the local minimum. When the repulsive and attractive potential interact at a certain point and cancel out each other the agent can get stuck at the local minimum. There are many ways to approach this problem. One way is to use the navigation functions as shown in [6]. However, for our case to make analysis easier we have made a few assumptions in the environment and potential gain values in the potential field equations as follows:

Assumption 1: The 3 obstacles in the environment are far away from each other such that when the robot is near obstacle 1, the repulsive forces by other obstacles can be ignored.

Assumption 2: The value of attractive potential field gain(k_{io}) is far greater than repulsive field gain(k_{ig}).

The critical points are present in the environment when the gradient value is equal to zero. To prove that local minima does not exist, it is shown that the gradient value is zero only at goal positions. For analysis we are using forces in x direction and show that the gradient is zero when x = x_g.

$$F_x = Fxg + Fxo1$$

$$k_{ig}(x_g - x) + \frac{k_{io}(x_o - x)}{r_{io}^3} = 0$$

After rearranging equation, we get,

$$\left(\frac{x_o - x}{x_g - x}\right) = -\left(\frac{k_{ig}}{k_{io}}\right) r_{io}^3$$

Using assumption 1 we can say that the value of k_{ig} is far greater than k_{io}. Thus the term on the right hand side tends to infinity. This can only happen when the denominator on the left-hand side is equal to zero which implies that $x = x_g$.
A similar approach is used for the forces in y direction, and we get $y = y_g$.
Thus, the gradient of the net potential field will be equal to zero only when the robot is at its goal position.

For collision avoidance it can be said that as the distance between the obstacle and robot tends to zero the forces between them tends to infinity. This ensures that collision does not take place. This can be proved by limiting r_{io} to zero in equation (5). The repulsive force between robot and obstacle tends to infinity.

Formation Velocity Consensus (D and A)

In this section we prove that the graph used for transport mode is well connected and balanced. Further we check if average consensus can be reached using the Laplacian matrix mentioned in section II. It is assumed that if average consensus can be reached any other reference velocity can be reached using the same matrix.

First, we mention that the graph is directed and strongly connected as all the vertices have directed arrows to all other vertices. To show that the graph is well balanced we use the Laplacian matrix and observe that the sum of all elements in any column is equal to zero. After calculating the eigenvalues for Laplacian, it is observed that zero is a simple eigenvalue of the matrix. So, it can be concluded that the graph is strongly connected.

The algebraic connectivity of the graph in the figure is given by the second eigenvalue of the matrix L_s.

$$L_s = \frac{(L + L^T)}{2}$$

We have considered the graph to be bidirectional and unweighted, so we know that the Laplacian matrix is symmetric, i.e. $L = L^T$. Thus, $L_s = L$.

The eigenvalues of this Laplacian matrix are:

$$eig(L) = \begin{bmatrix} 0 \\ 6 \\ 6 \\ 6 \\ 6 \\ 6 \end{bmatrix}$$

Thus, we know that $\lambda_2 = 6$, and from this we get the algebraic connectivity. Thus, we know that the system will converge to a consensus value at a speed at least as fast as λ_2 [7].

To figure out the inter-robot distances, we make use of the potential formula as described in [8]:

$$V_{ij} = \frac{1}{4}\left(\left\|p_i - p_j\right\|^2 - {d_{ij}}^2\right)$$

Where, p_i and p_j are the position vectors of robots i and j, and d_{ij} is the desired distance between those robots. The gradient between this is:

$$\nabla_{p_i} V_{ij} = \left(\left\|p_i - p_j\right\|^2 - d_{ij}\right)\left(p_j - p_i\right)$$

Summing this for all of the neighbors of each robot, we get the velocities that each robot needs to have to maintain the formation. To get this formation to travel somewhere else, we add a velocity vector to this gradient, in the desired direction.

IV. Validation in Simulation

Approach Mode (A and C)

In this section we are designing a model for the robots to approach the object that needs to be transported. We are considering a potential field-based approach for moving the robots to the goal positions. We have added a high potential field on the border of the object being transported instead of the center to avoid the robots from going through it. The distance between the goal and the current position of the robots is calculated. Each robot is given a specific position around the object. There is high attractive potential at individual goal points for each robot. We are considering three obstacles having fixed positions. When the robot gets within the sensing range of the obstacle it gets repulsed by its potential field. A repulsive potential field is also applied between the robots themselves. This prevents any collision between them. An if-else statement is added for these purposes. All these potentials are added for each robot in an iteration within the loop. The ODE45 function in MATLAB is used to find out and simulate these conditions over a time of 30 sec. These methods have shown the results that we expected. The robots are appropriately avoiding obstacles and each other and are successfully reaching the goal point.

The behavior of the system in approach mode is validated in MATLAB. The robot trajectories are plotted in figure 6. The actual and desired heading angles of the robot are plotted in figure 7.

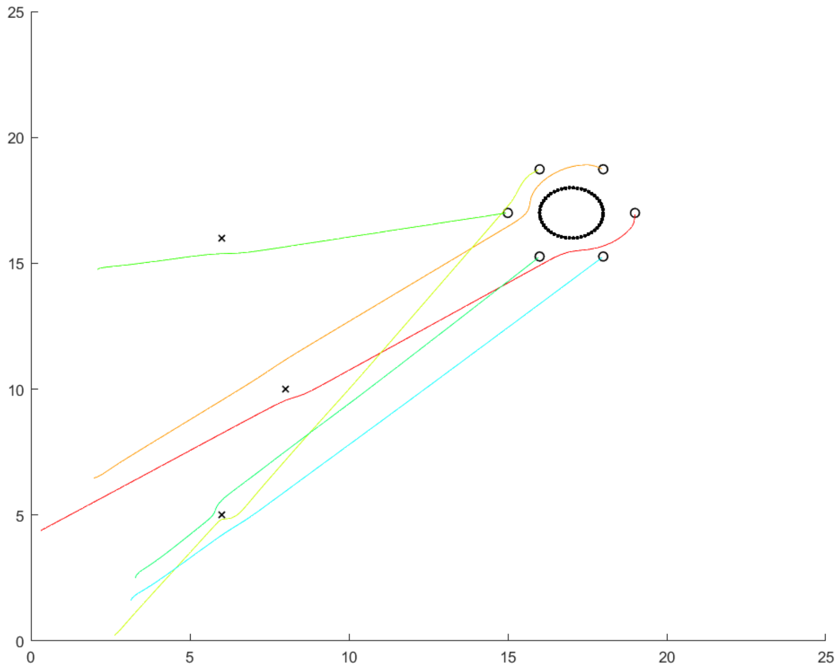


Figure 6: Trajectories of Robots in Approach Mode

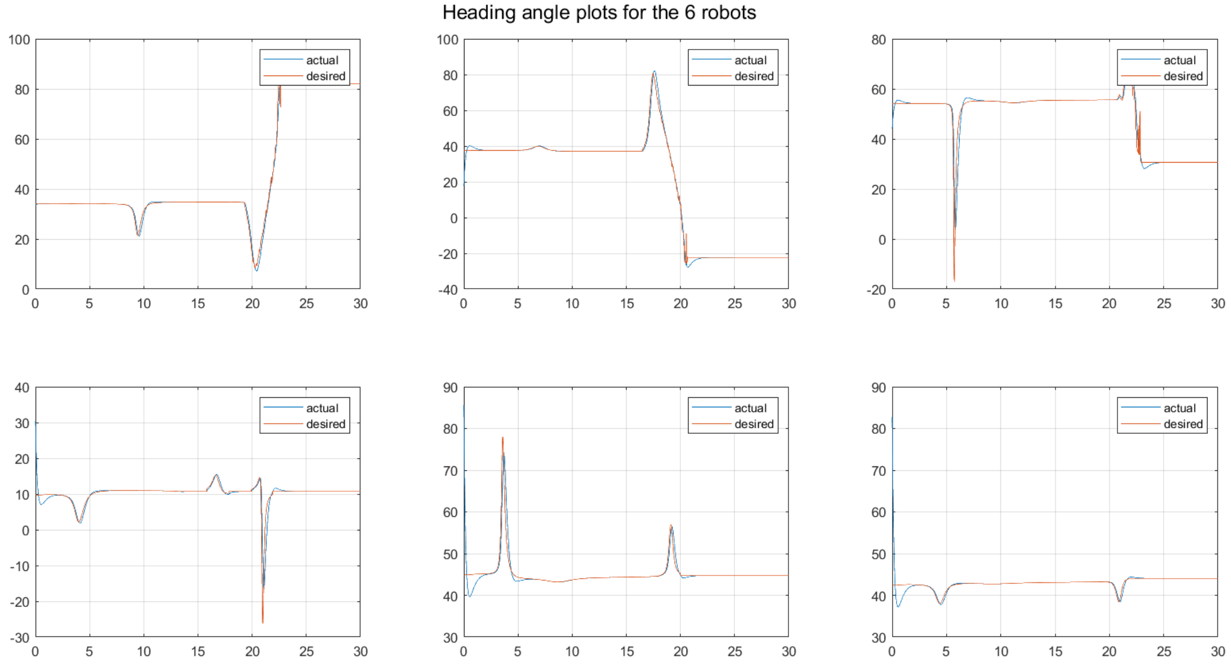


Figure 7: Actual and Desired Heading Angles for individual robots

Controller Validation (A and D)

The performance specifications of the controller are validated in MATLAB and the robustness to a constant disturbance at the robot input is verified. The step response of the system is shown in figure 5. The θ values of the system in the presence of bias added to the control input is shown in figure 8. The bias disturbance is rejected by the controller and all robots arrive at their destinations.

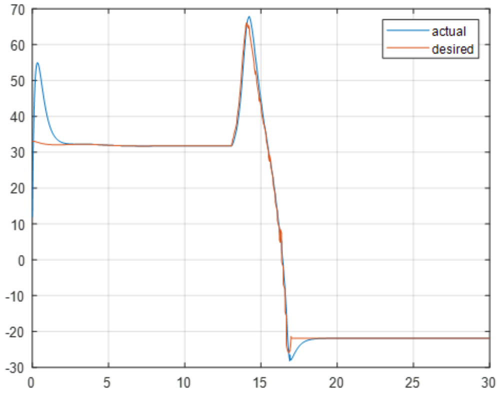


Figure 8: Actual vs Desired heading angle for robot 1 with disturbance

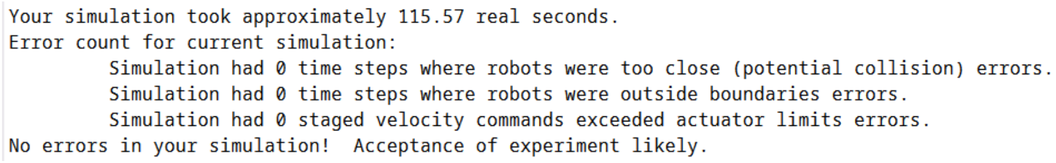
Formation and Transport (A, B, C and D)

The MATLAB simulator of Robotarium was used to simulate the arrangement of the robots. The code is divided into two main parts: the formation control and the transport control. The formation control is run over 2000 iterations, for the formation to be formed, and then the additional centroid velocity vector is added, to get the formation to the goal point. As shown in 10(b), the green square is the goal, set as a set of coordinates, and the robots are initialized with random coordinates in the simulated Robotarium testbed.

At each iteration, the robot velocity is initialized to zero, and the velocity of each robot, determined by the inter-robot potential gradient, is added to the empty velocity vector. After 2000 iterations, once the robots have reached the desired formation, as shown in 10(a), the centroidal velocity vector is added, to give the formation velocity towards the pre-defined goal point.

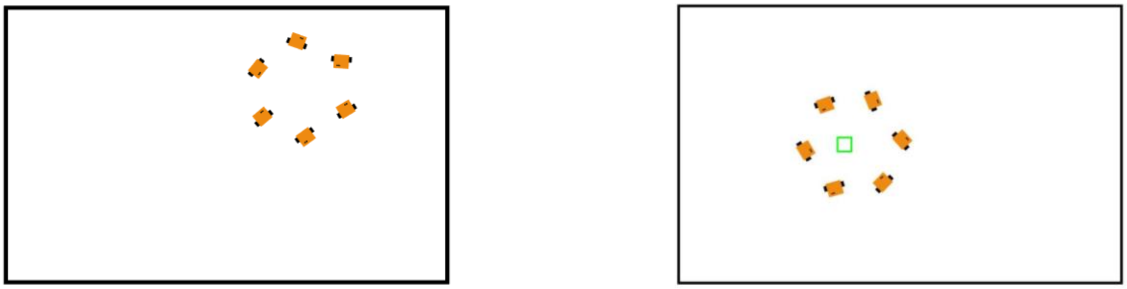
This is performed in single-integrator dynamics and using the Robotarium's native function to convert the single-integrator dynamics to the unicycle dynamic model as used in the simulator. This is done because the robots used in the Robotarium simulator are non-holonomic; in fact, they are differential drive robots, which make use of unicycle dynamics. These are also used in barrier certificates and controllers in the Robotarium code, to prevent the positions and velocities from going out-of-bounds, and also to prevent inter-robot collision.

If run on the Robotarium’s physical testbed at Georgia Tech, the simulation would take about under two minutes to complete. This is found through the debugging function provided by Robotarium.



```
Your simulation took approximately 115.57 real seconds.
Error count for current simulation:
         Simulation had 0 time steps where robots were too close (potential collision) errors.
         Simulation had 0 time steps where robots were outside boundaries errors.
         Simulation had 0 staged velocity commands exceeded actuator limits errors.
No errors in your simulation!  Acceptance of experiment likely.
```

Figure 9: Output of the debugging function in Robotarium for a simulation



(a)

(b)

Figure 10: Snapshots of the Robotarium simulation, (a) while arranging themselves in formation, (b) approaching the goal position

References:

[1] Parker L. (2015) Collective Manipulation and Construction. In: Kacprzyk J., Pedrycz W. (eds) Springer Handbook of Computational Intelligence. Springer Handbooks. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-43505-2_72

[2] Tuci E, Alkilabi MH and Akanyeti O (2018). Cooperative Object Transport in Multi-Robot Systems: A Review of the State-of-the-Art. *Front. Robot. AI* 5:59. doi: 10.3389/frobt.2018.00059

[3] Peng Song and V. Kumar, "A potential field based approach to multi-robot manipulation," Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), 2002, pp. 1217-1222 vol.2, doi: 10.1109/ROBOT.2002.1014709.

[4] Sean Wilson, et al., *"The Robotarium: Globally Impactful Opportunities, Challenges, and Lessons Learned in Remote-Access, Distributed Control of Multirobot Systems,"* in IEEE Control Systems Magazine, vol. 40, no. 1, pp. 26-44, Feb. 2020.

[5] Schiff, J. L. (1999). The Laplace transform: theory and applications. Springer Science & Business Media.

[6] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," in IEEE Transactions on Robotics and Automation, vol. 8, no. 5, pp. 501-518, Oct. 1992, doi: 10.1109/70.163777.

[7] Olfati-Saber, Reza, J. Alex Fax, and Richard M. Murray. "Consensus and cooperation in networked multi-agent systems." Proceedings of the IEEE 95.1 (2007): 215-233.

[8] Brian D.O. Anderson, Zhiyong Sun, Toshiharu Sugie, Shun-ichi Azuma, Kazunori Sakurama, Formation shape control with distance and area constraints, IFAC Journal of Systems and Control, Volume 1, 2017, Pages 2-12, ISSN 2468-6018, https://doi.org/10.1016/j.ifacsc.2017.05.001.