Question 1.1

To update the position of the individual robots, we utilize the following double integrator formula
in MATLAB.

```
Vx(i,k)=Vx(i,k-1) + Ux(i)*dt;

Vy(i,k)=Vy(i,k-1) + Uy(i)*dt;

X(i,k)=X(i,k-1) + Vx(i,k)*dt;

Y(i,k)=Y(i,k-1) + Vy(i,k)*dt;
```

This just discretizes the system and updates the robot's velocity and position one step at a time. But now we must implement some flocking algorithms to make this useful.

Question 1.2

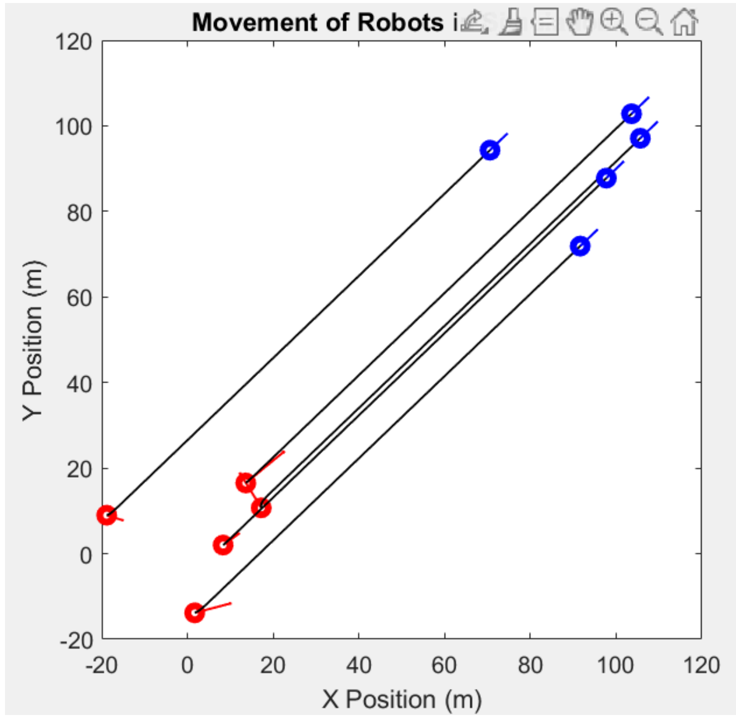
The first flocking algorithm is aligning the velocity of the robots. This is implemented using the following formula:

$$\ddot{x}_i(t) = -\alpha \sum_{j=1}^{n} a_{ij}(t)\left[\dot{x}_i(t) - \dot{x}_j(t)\right]$$

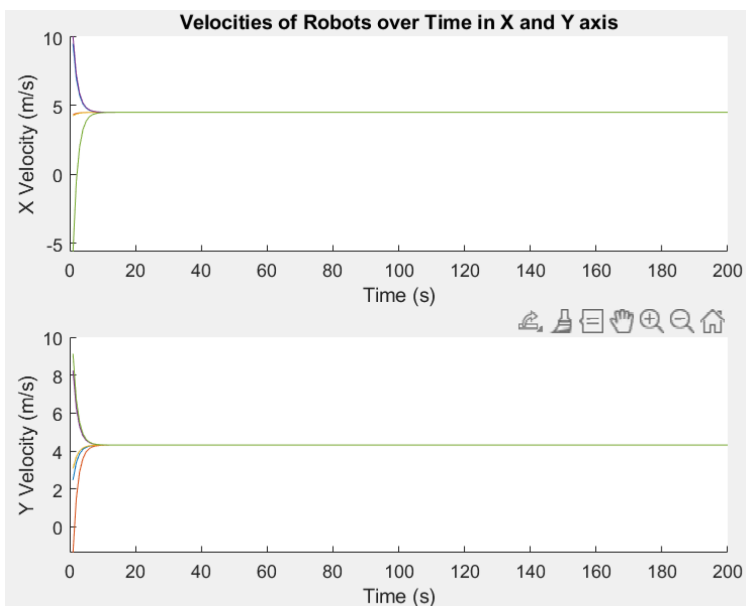
This is the same as the consensus algorithm implemented in the earlier labs. But the only change is that we are now getting consensus velocity rather than consensus position.

Question 1.3

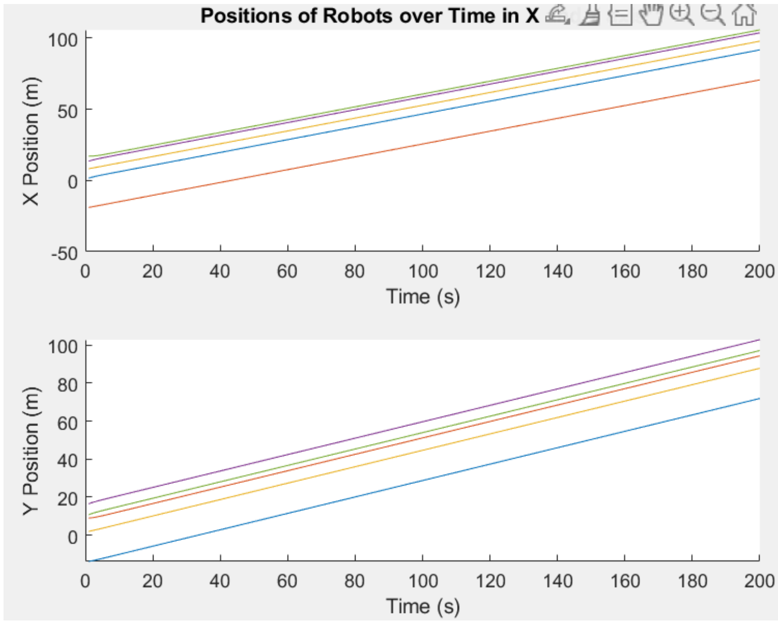
For N = 5:



Figure(1)

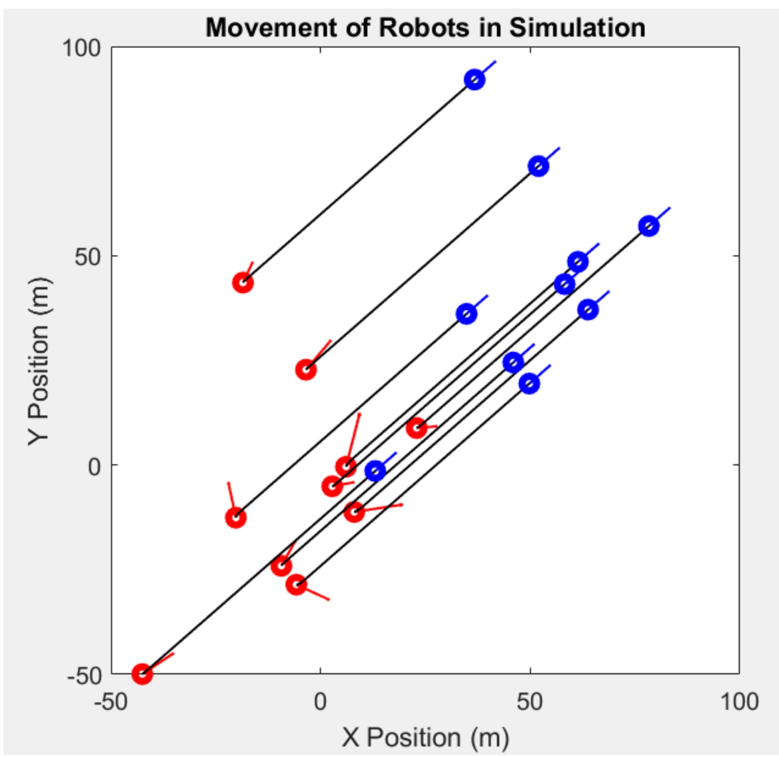


Figure(2)

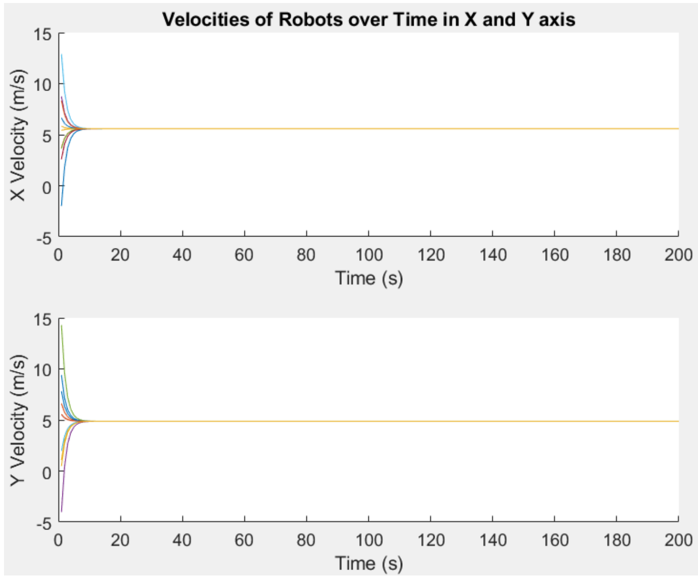


Figure(3)

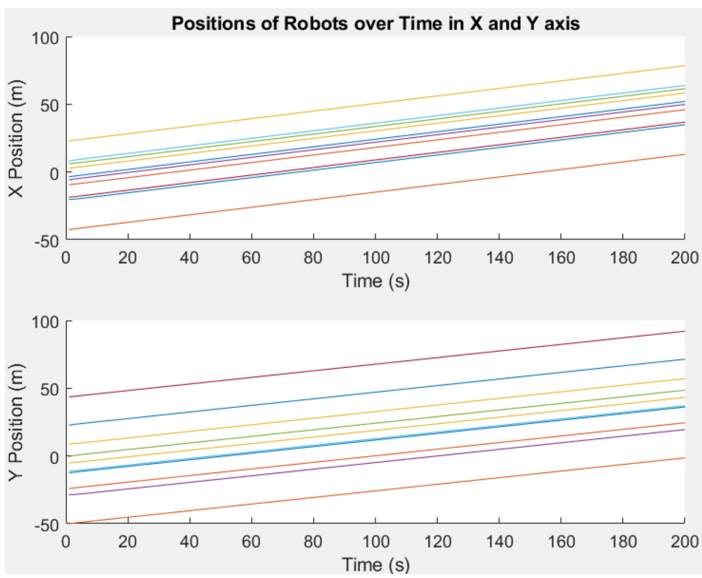
For N = 10:



Figure(4)



Figure(5)



Figure(6)

As shown in figures (1, 2, 3, 4, 5, 6) for different N , with alignment, we set α=1 and β=0 γ=0. This ensures that the velocities of all 5 robots align by minimizing the differences between their velocities.

Question 1.4

We can see that the position is uncontrolled, but the velocities are able to reach consensus. We are able to determine the velocities of the robots if we know the initial conditions. It is just an average in both X and Y direction for the robots. A quick check is completed on Figure 2. The x velocity converges to 5, which if we average the approximate initial Y velocities of robots we also get 4.3.

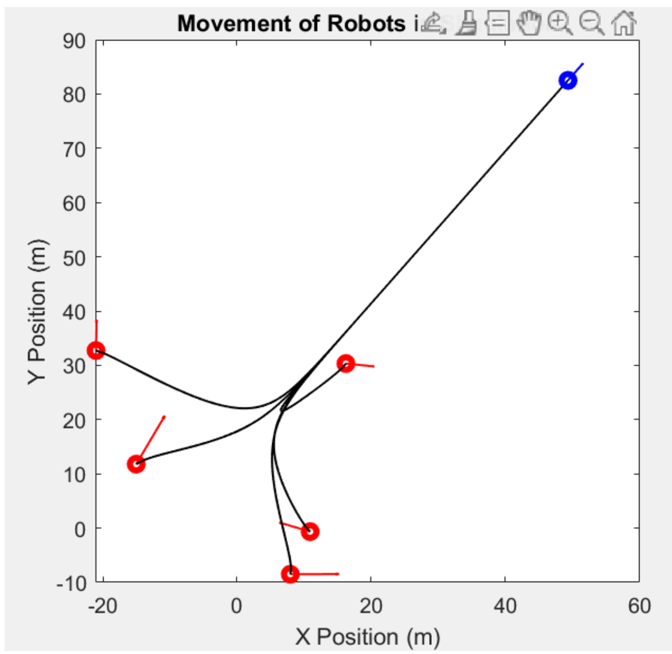
Question 1.5

we gather from the slides that we must add another term. This gives us the equation below:

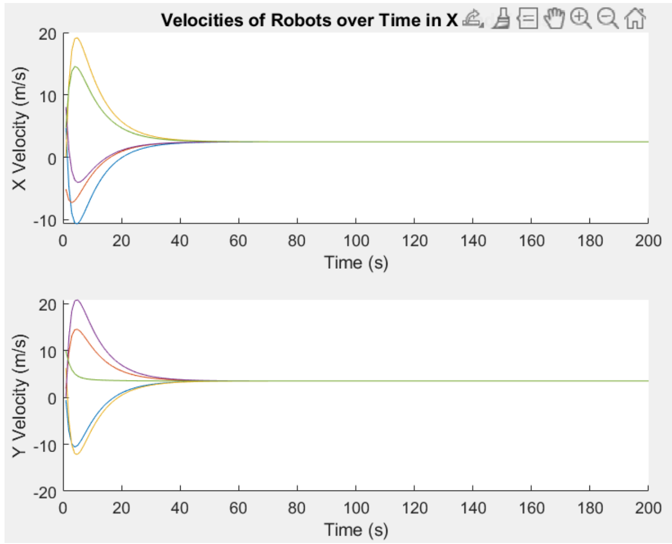
$$\ddot{x}_i(t) = -\alpha \sum_{j=1}^{n} a_{ij}(t)\left[\dot{x}_i(t) - \dot{x}_j(t)\right] - \beta\nabla U_{coh}(x)$$

Question 1.6

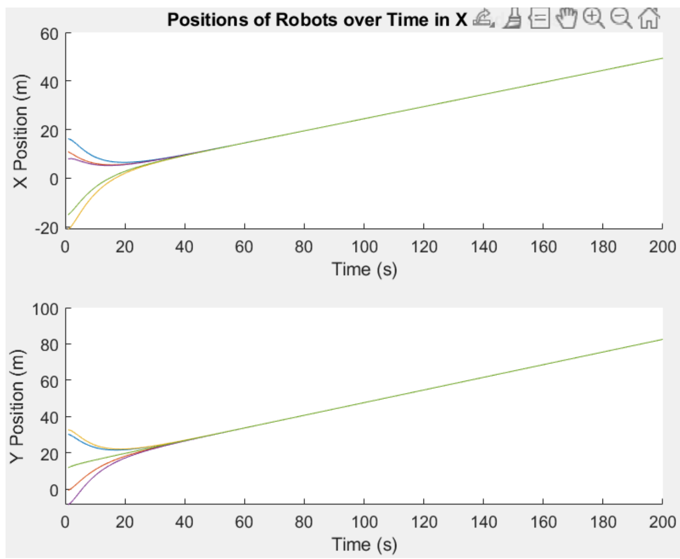
For N = 5:



Figure(7)

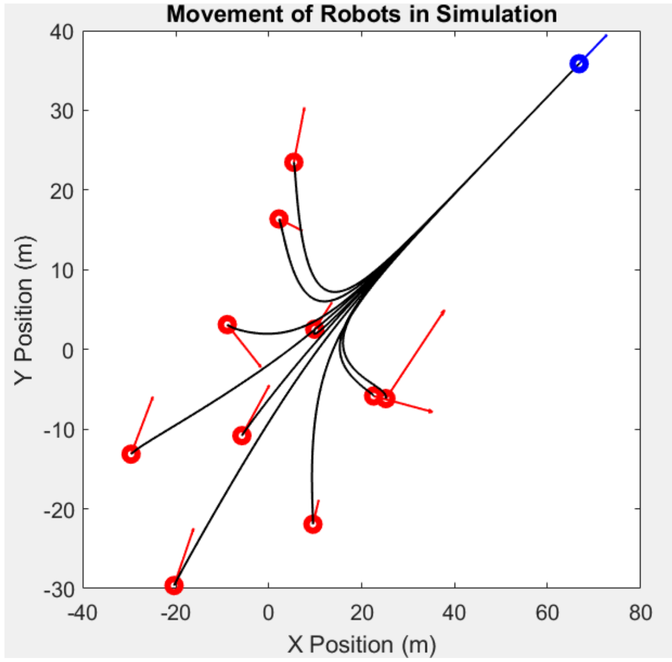


Figure(8)

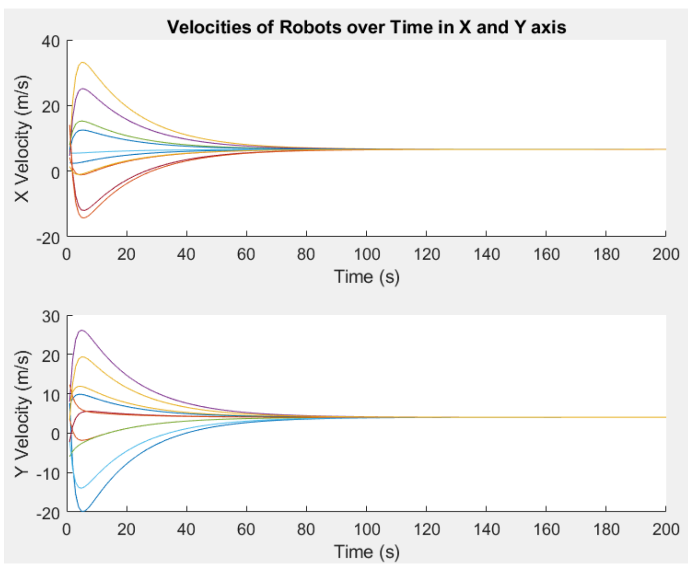


Figure(9)

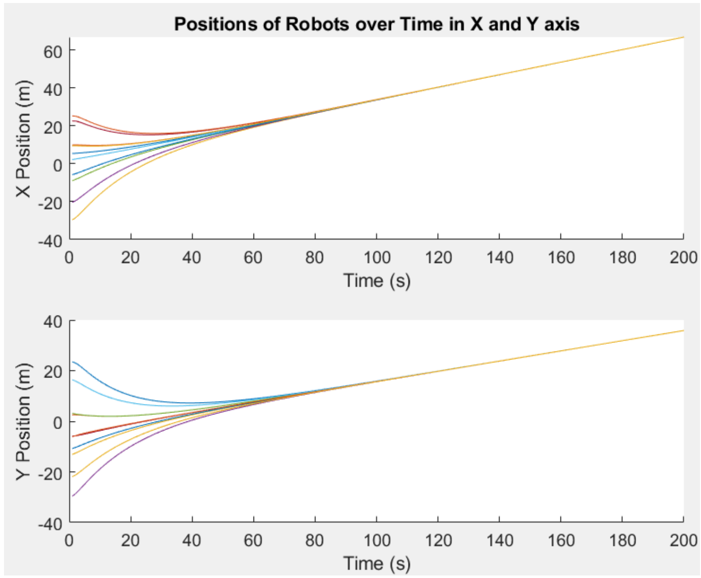
For N = 10:



Figure(10)



Figure(11)



Figure(12)

As shown in figures (7, 8, 9, 10, 11, 12) for different N, the cohesion with alignment, we set α=1, β=1, and γ=0. This ensures the robots not only align their velocities but also move toward the centroid of their neighbors, maintaining group cohesion.

- **Alignment effect (α)**: Velocities align, creating coordinated movement.
- **Cohesion effect (β)**: Robots are pulled toward the group center, forming a tighter cluster.

This results in a flocking behavior where robots stay together and move cohesively.

Question 1.7

$$u_i = \gamma \sum \frac{P_i - P_j}{\|P_i - P_j\|^3}$$

- **Neighbor Identification**:
  - For each robot i, compute the distance to all other robots j≠i.
  - If the distance is below a threshold (typically determined by the communication radius ρ), apply the separation force.
- **Repulsive Force**:

- The force is inversely proportional to the cube of the distance. This ensures:
  - Strong repulsion at very close distances to prevent collisions.
  - Minimal influence when robots are far apart.

- **Dynamic Adjustment**:
  - As robots move closer, the separation force increases, pushing them apart.
  - This balances the cohesion term, which pulls robots together, creating a stable flocking formation.

Question 1.8

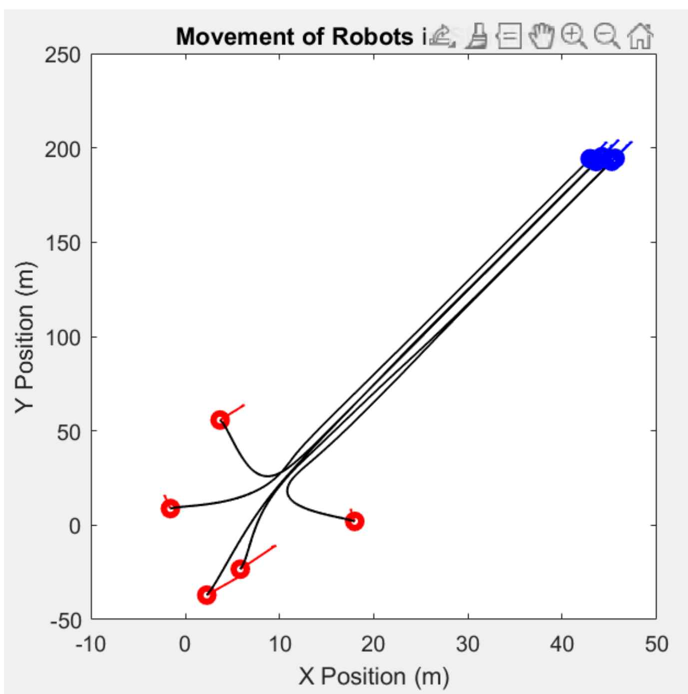
We have now included the separation into the equation which follows as below:

$$\ddot{x}_i(t) = \; - \alpha \sum_{j=1}^{n} a_{ij}(t)\left[\dot{x}_i(t) - \dot{x}_j(t)\right] - \beta \nabla U_{coh}(x) + \gamma \nabla U_{sep}(x)$$

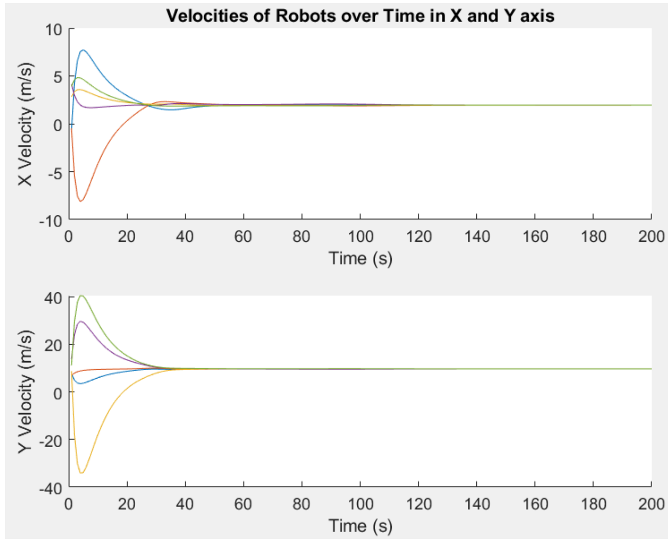
Question 1.9

Implementing in matlab, we receive the following graphs: It is hard to tell, but these robots are able to resist each other a little, but they are flying in a very tight formation.

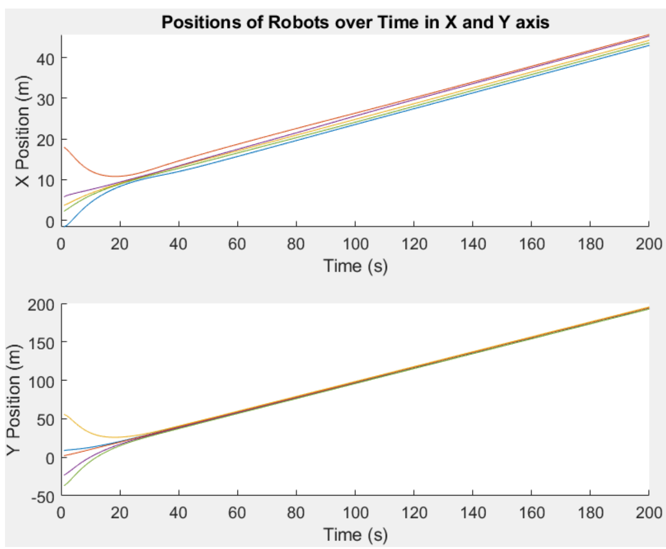
For N = 5 , $\beta = 1, \gamma = 10$:



Figure(13)

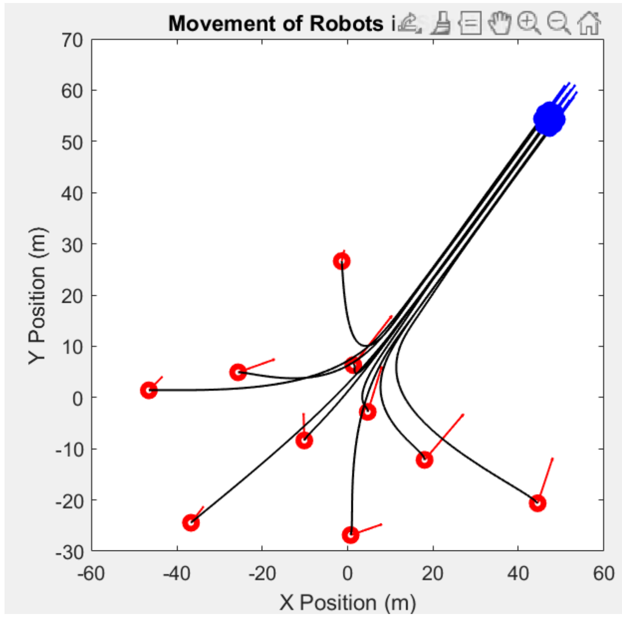


Figure(14)

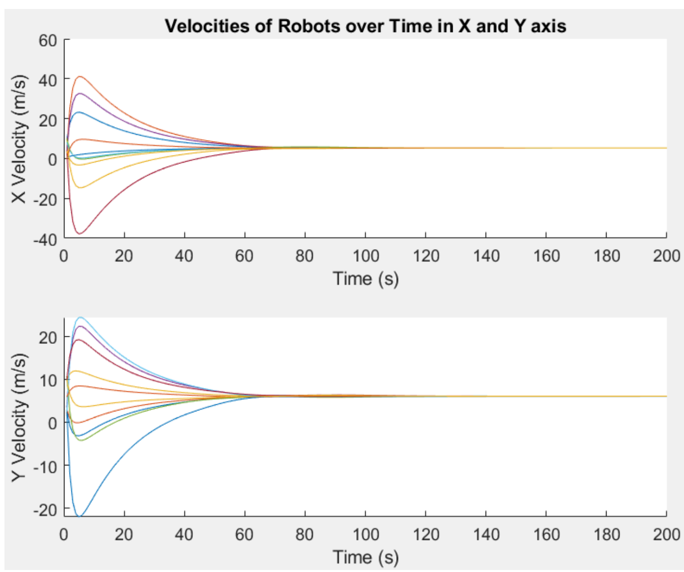


Figure(15)

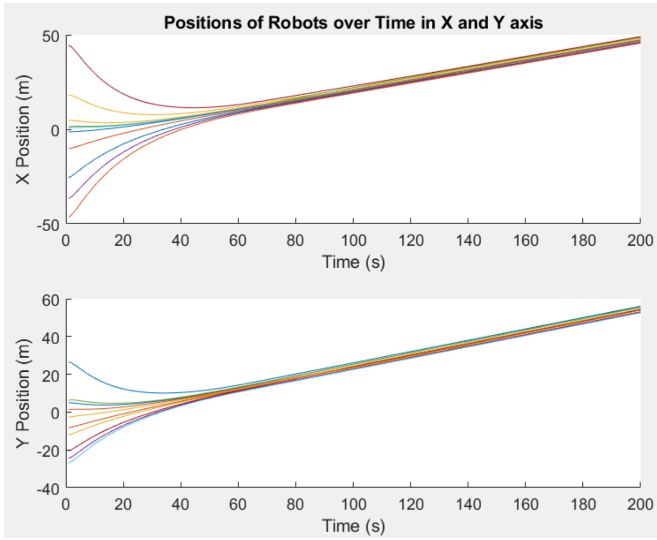
For N = 10 , $\beta = 1, \gamma = 10$:



Figure(16)



Figure(17)



Figure(18)

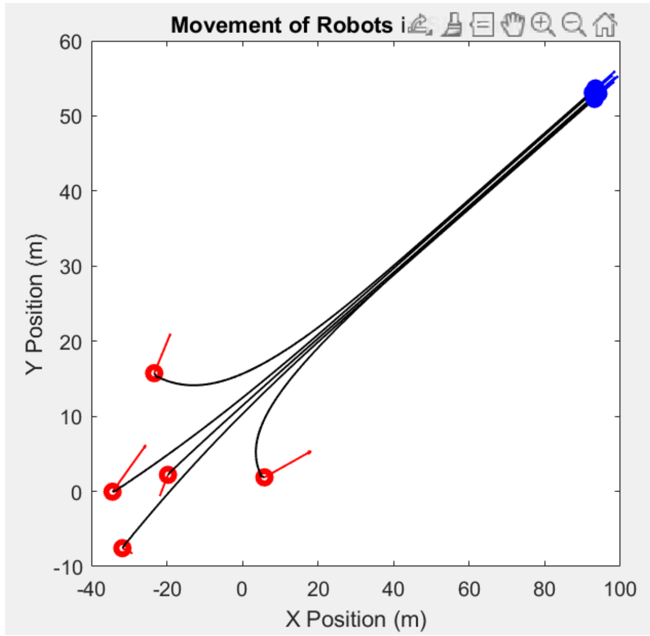
As shown in figures (13, 14, 15, 16, 17, 18) for alignment, cohesion, and separation, we set α=1, β=1, and γ=10 This ensures:

1. **Alignment (α)**: Velocities align for coordinated movement.
2. **Cohesion (β)**: Robots move toward the centroid, maintaining group cohesion.
3. **Separation (γ)**: Robots repel each other when too close, preventing collisions.

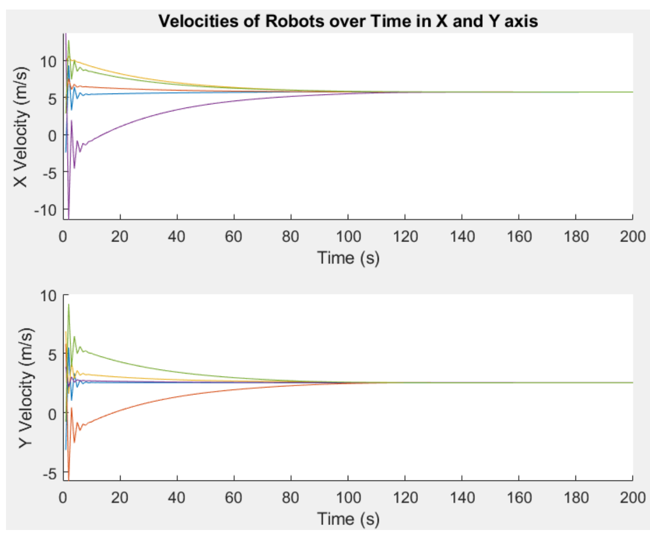
Result: The robots align, cluster together, and maintain safe distances, achieving stable and collision-free flocking.

Question 1.10

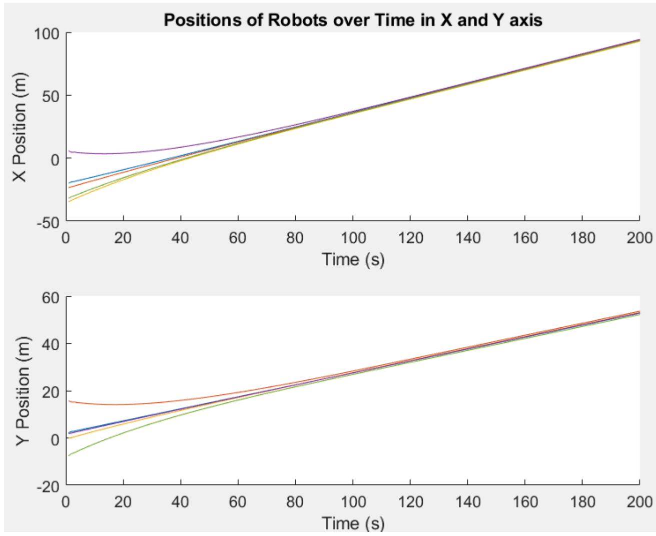
For N = 5, $\alpha = 3, \beta = 1, \gamma = 1$:



Figure(19)



Figure(20)



Figure(21)

For N=5 robots, with $\alpha=3$, $\beta=1$ and $\gamma=1$,

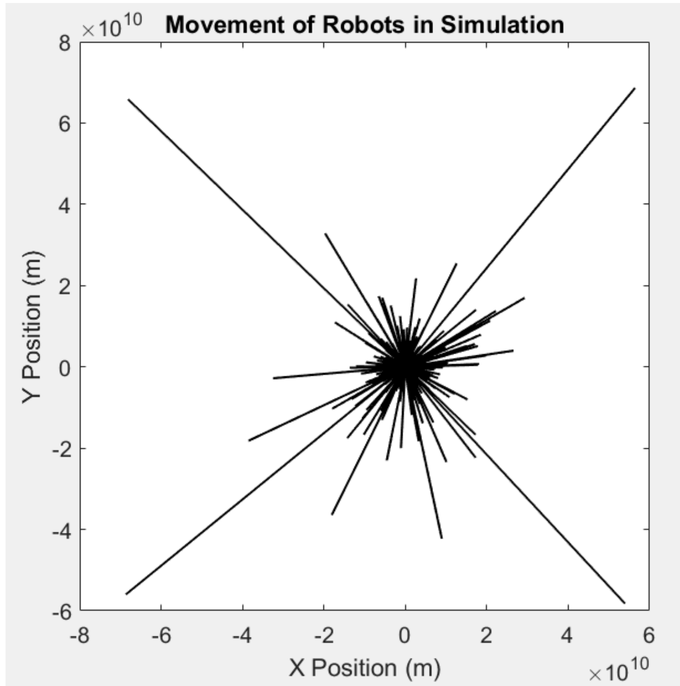
Effects of Parameters:

1. **$\alpha=3$ = 3** (Strong alignment):
   - Robots quickly align their velocities for coordinated movement.
2. **$\beta=1$ = 1** (Moderate cohesion):
   - Robots move toward the centroid, maintaining group cohesion.
3. **$\gamma=1$ = 1** (Moderate separation):
   - Robots maintain safe distances, avoiding collisions without excessive repulsion.

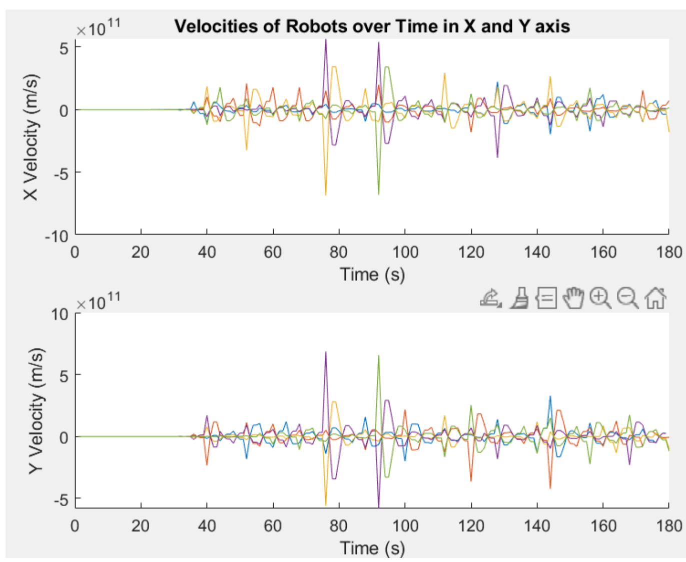
Result:

The robots form a cohesive flock with aligned velocities, balanced by a safe inter-robot distance. This configuration ensures both stability and collision-free movement.

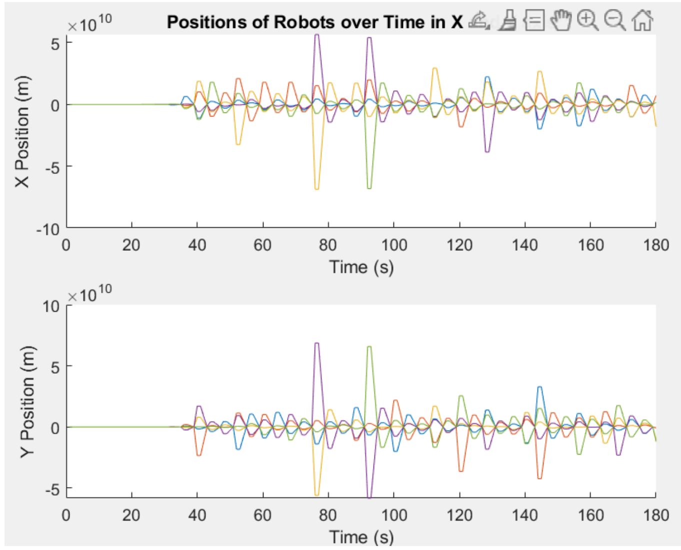
By changing the values of $\alpha = 1, \beta = 10, \gamma = 1$, we got the following results:



Figure(22)

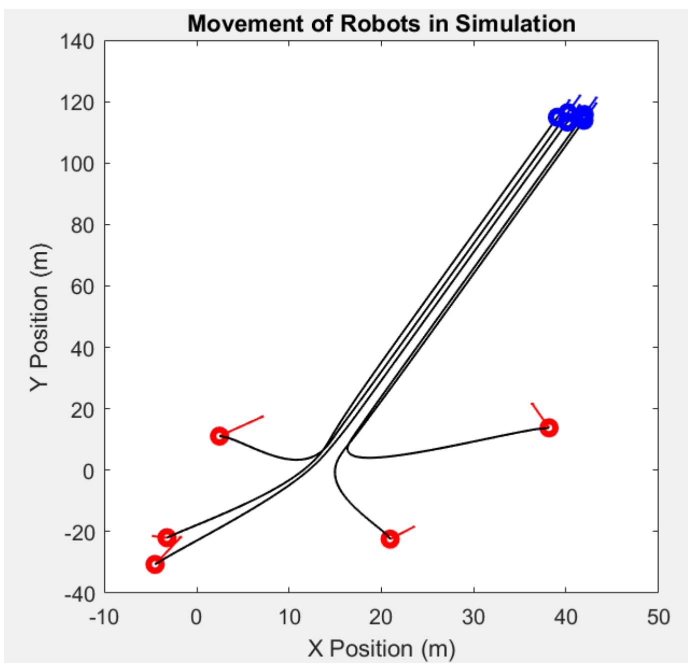


Figure(23)

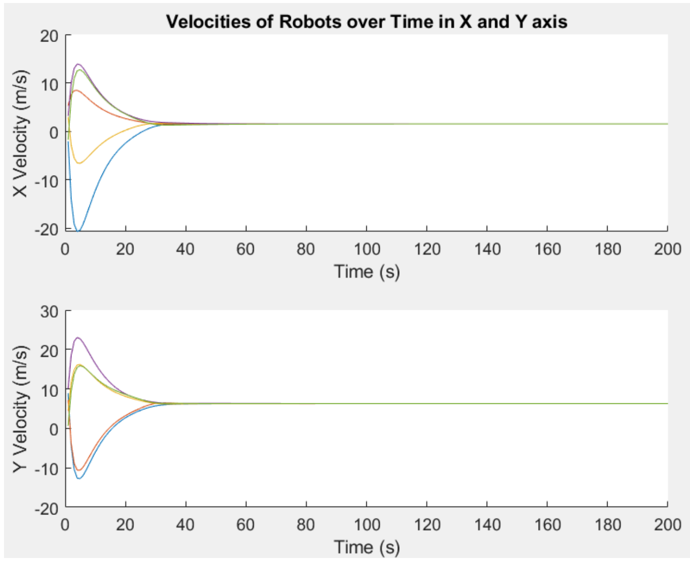


Figure(24)

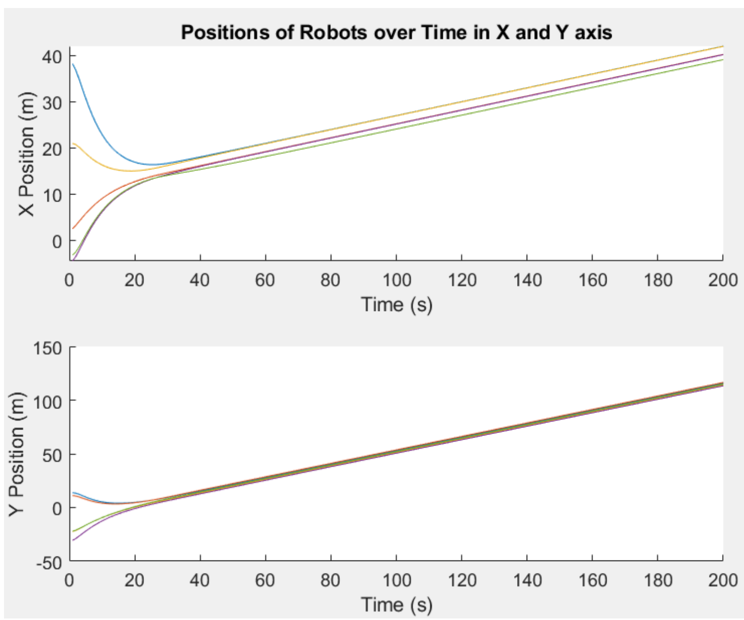
And when changing the values of $\alpha = 1, \beta = 1, \gamma = 15$, we got the following results:



Figure(25)



Figure(26)



Figure(27)

For N=5 robots, with α=1, β=10, and γ=1,

Parameter Effects:

1. **α=1** (Moderate alignment):

    ○ Velocities gradually align, achieving coordinated movement.

2. **β=10**(Strong cohesion):
    - Robots are strongly pulled toward the centroid, ensuring a tight cluster formation.
3. **γ=1** (Moderate separation):
    - Ensures robots maintain safe distances to avoid collisions without disrupting cohesion.

Result:

- The robots will form a tight flock due to the strong cohesion (β=10).
- They will maintain safe spacing because of the separation term (γ=1).
- The velocities of the robots will gradually align, achieving a cohesive and well-aligned group movement.

Question 2:

Question 2.1

We just need to update our Laplacian matrix to take into account if the robots are close enough.

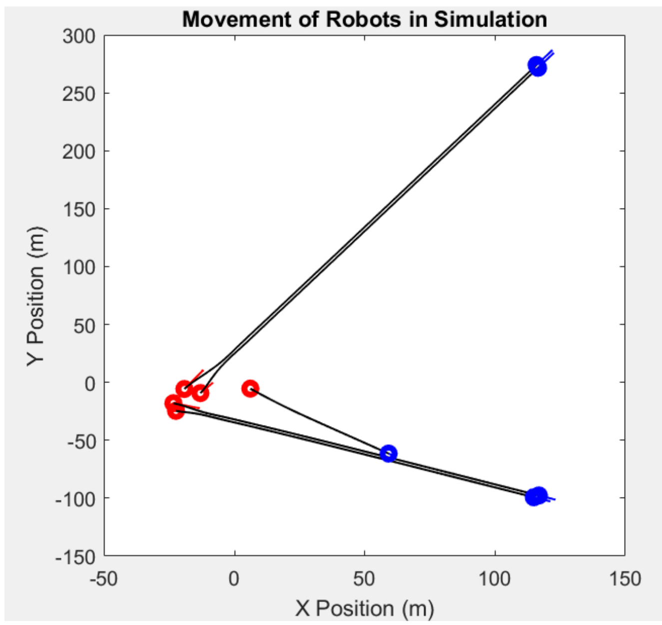
The following code was used to create this simulated communication network.

```
% communication radius
    rho=100;
    L = zeros(N,N);
    for i=1:N
        for j=1:N
            if i~=j
                vector = [X(i,k-1)-X(j,k-1) Y(i,k-1)-Y(j,k-1)];
                if norm(vector) < rho
                    L(i,j) = -1;
                else
                    L(i,j) = 0;
                end
            end
```

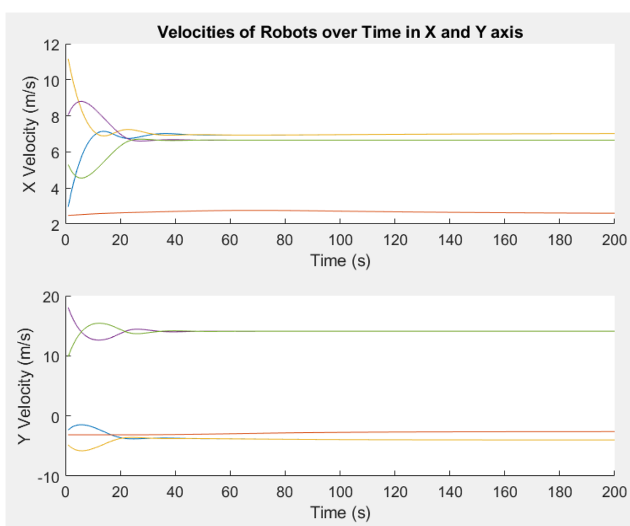
```
        end
    end
    for i=1:N
        L(i,i) = -sum(L(i,:));
    end
```

Question 2.2

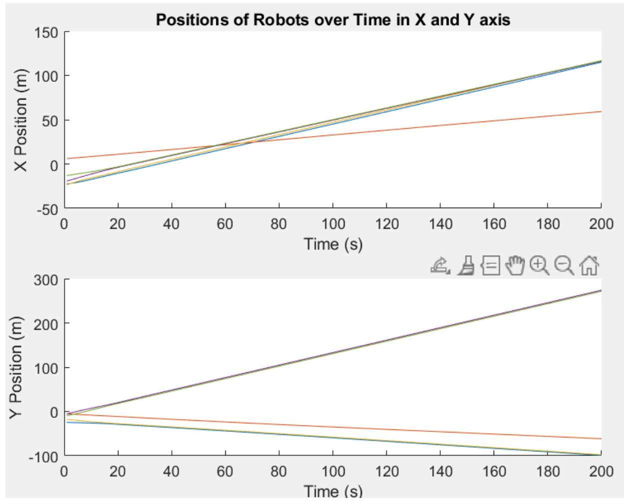
For $\gamma = 15$, rho=10:



Figure(28)



Figure(29)

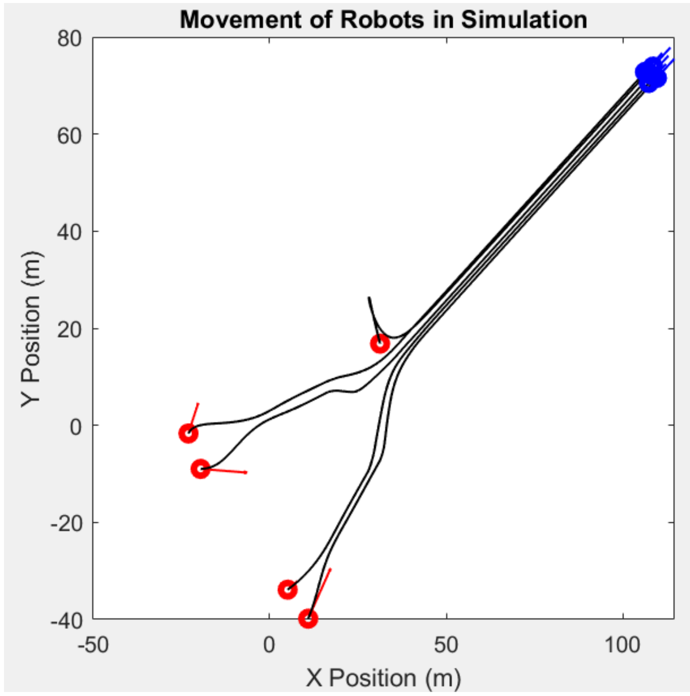


Figure(30)

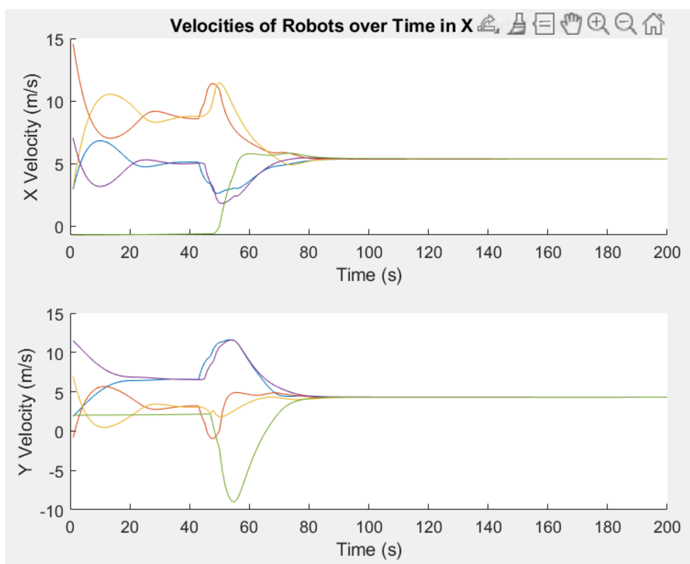
as shown in the figures above the parameter Effects:

1. **α=1(Alignment)**:
   - Ensures robots' velocities gradually align, promoting coordinated movement.
2. **β=1 (Cohesion)**:
   - Pulls robots toward the group centroid, maintaining cohesion.
3. **γ=15 (Strong Separation)**:
   - Strongly repels robots that get too close, ensuring a safe distance.
4. **ρ=10 (Limited Communication)**:
   - Robots only interact with neighbors within a distance of 10 units, resulting in **localized flocking**.

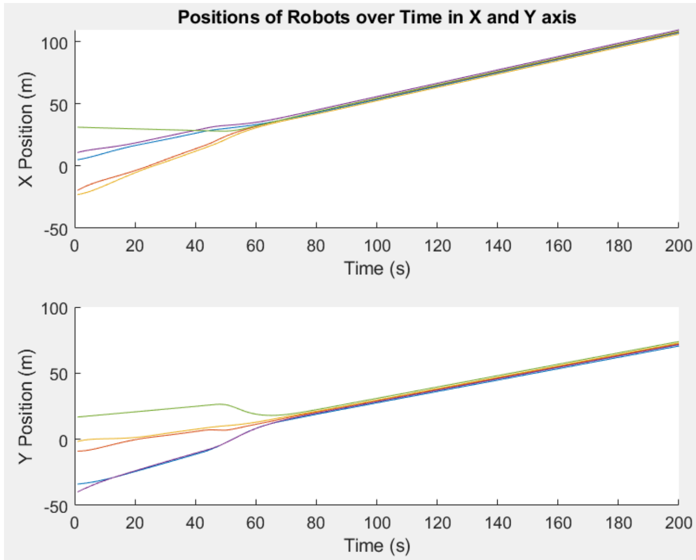
For $\gamma = 15$, rho = 20, we got the following results:



Figure(31)



Figure(32)



Figure(33)

With these parameters, the robots will achieve stable flocking behavior within their local neighborhoods, forming a cohesive group while avoiding collisions. Adjusting ρ allows control over the extent of interaction, balancing local and global behaviors.

Question 4

At the end of this lab and after implementing the three rules of Reynold in order to achieve af locking behavior in a group of double integrator robots, we can conclude what we have learned in the following

1. The Importance of Flocking Rules (R1, R2, R3)

- ***Alignment* (R1)** ensures that robots align their velocities, leading to a cohesive movement in a common direction. It promotes velocity synchronization across the group.
- **Cohesion (R2)** pulls robots toward the centroid of their local neighbors, facilitating clustering and group formation.
- ***Separation* (R3)** prevents robots from getting too close, ensuring collision avoidance and maintaining a safe distance between agents.

2. Impact of Communication Radius (ρ)

- **Large ρ**: Allows all-to-all communication, resulting in global cohesion. The flock converges to a single centroid, and all robots move in unison.
- **Small ρ**: Limits communication to local neighbors, leading to localized interactions. T

3. Role of Control Gains (α,β,γ)

- **Alignment Gain (α)**:
  - Higher αaccelerates velocity synchronization but may cause instability if too large.
- **Cohesion Gain (β)**:
  - Higher β results in faster clustering, but overly high values may cause overshooting or oscillatory motion around the group centroid.
- **Separation Gain (γ)**:
  - Higher γ ensures better collision avoidance by creating strong repulsive forces. However, excessive γ may lead to erratic behavior as robots repel too aggressively.

4. Initial Conditions Matter

- **Scattered Initial Positions**:
  - Robots take longer to converge, especially with small ρ, and may form sub-flocks.
- **Clustered Initial Positions**:
  - Faster convergence and more cohesive behavior, even with moderate or small ρ.