Tania Jaswal

Project 2: MSN Flocking Formation Control and Obstacle Avoidance

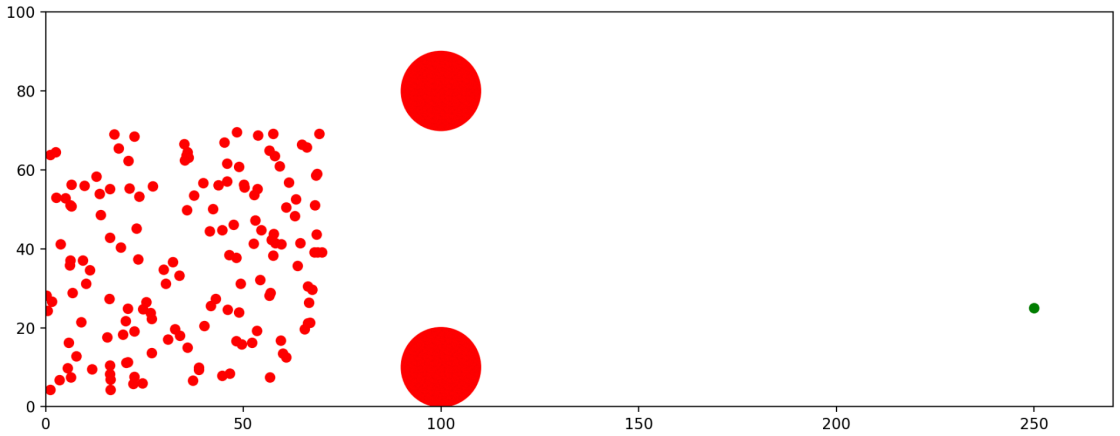
Project Overview:
This project aims to investigate Multi-Sensor Networks (MSN) dynamics in achieving flocking formation control while incorporating obstacle avoidance. Through the implementation of Algorithm 3, we explore how MSN nodes interact to exhibit flocking behaviors and adapt to avoid obstacles in different scenarios.

Project parameters:
Number of sensor nodes: n =150.
Space dimensions: m = 2.
Desired distance among sensor node: d = 15.
Scaling factor: k = 1.2 and interaction range r = k*d.
Epsilon = 0.1 and Delta_t = 0.009

Case 1: Algorithm 3(MSN Quasi-Lattice Formation) with obstacle avoidance

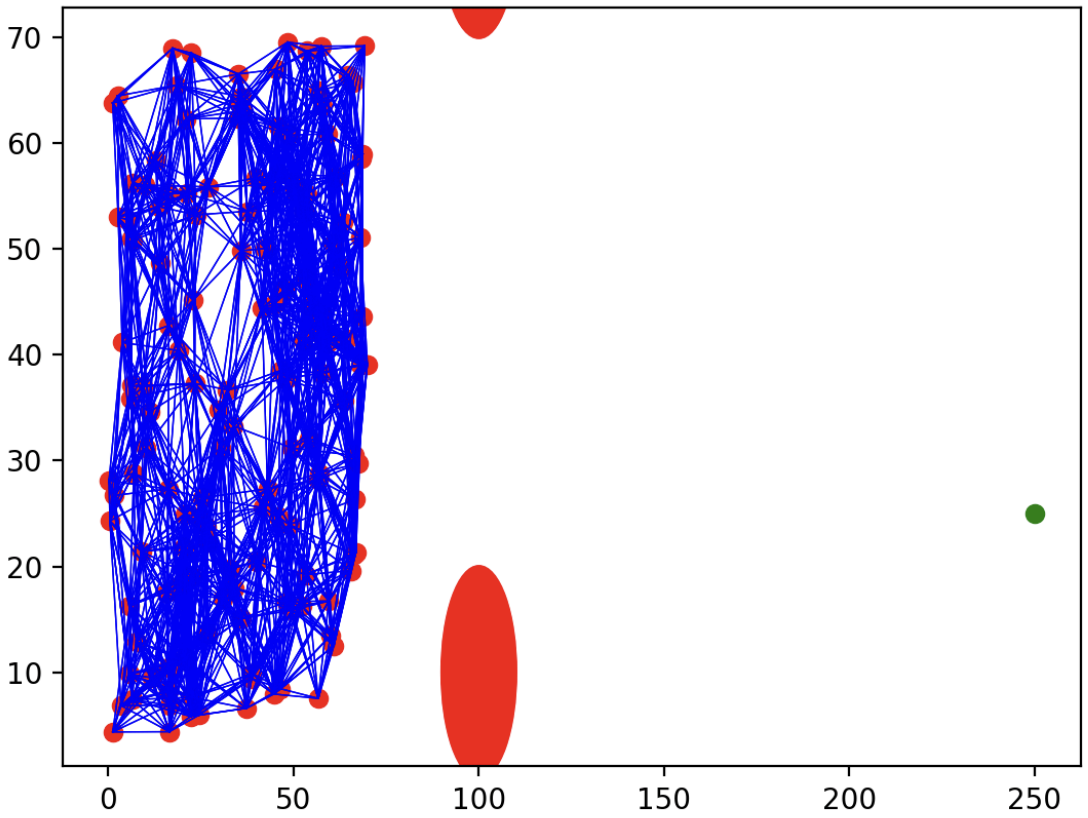
Implementation Details:
Generate a connected network of 150 sensor nodes within a 70x70 area, with a static target positioned at coordinates (250, 25). Employ Algorithm 3 to achieve quasi-lattice formation around the static target while ensuring obstacle avoidance.

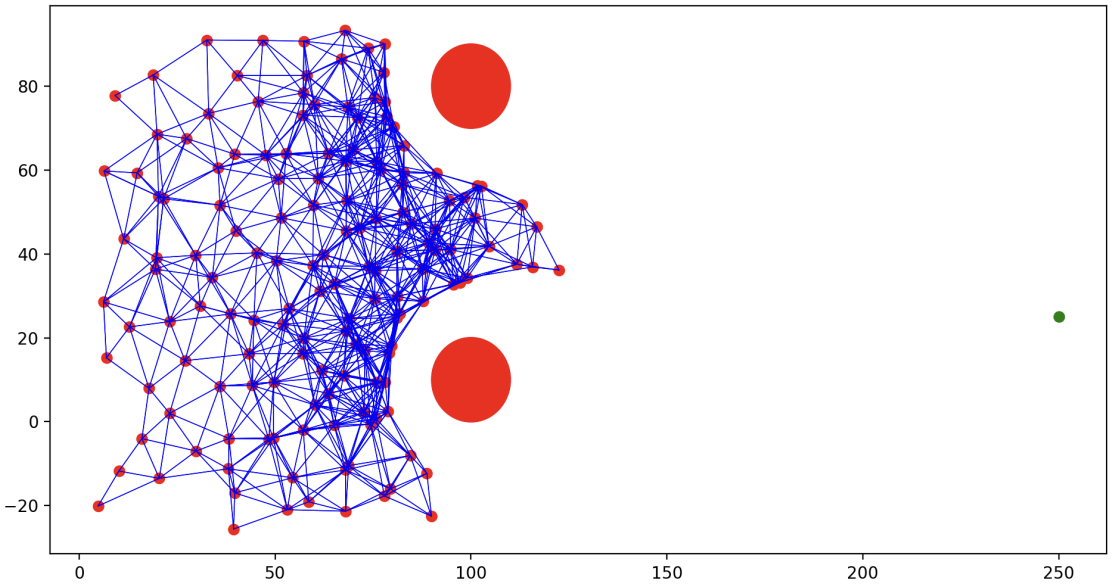
Results:
1. **Initial Deployment:**

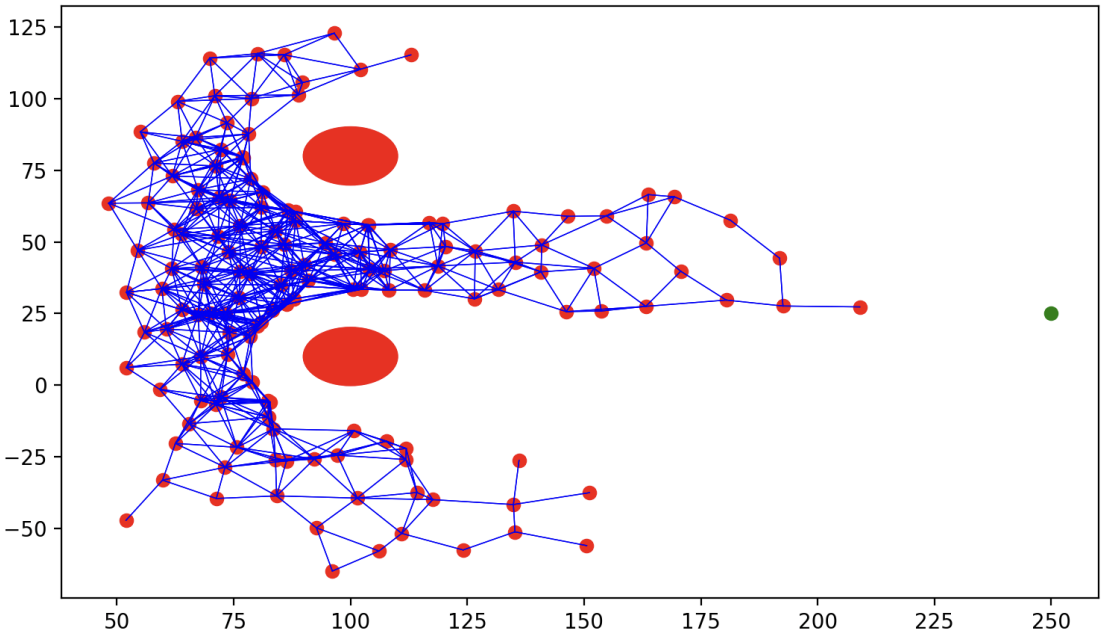


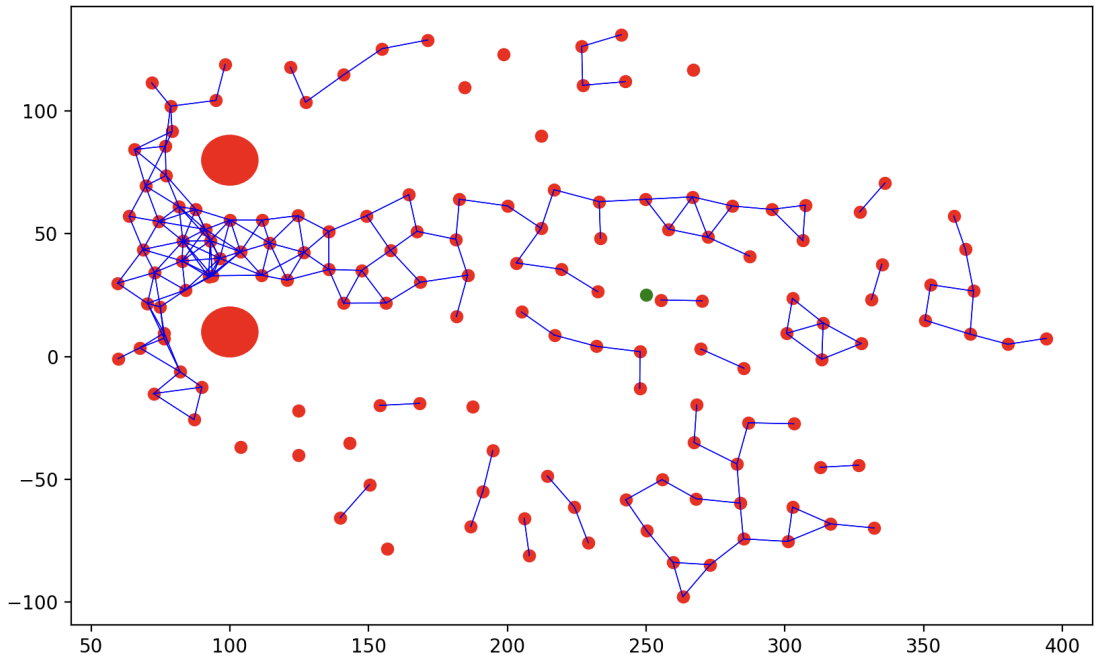
- Visualize the initial deployment of sensor nodes, obstacles and the static target, highlighting the connections between neighboring nodes.

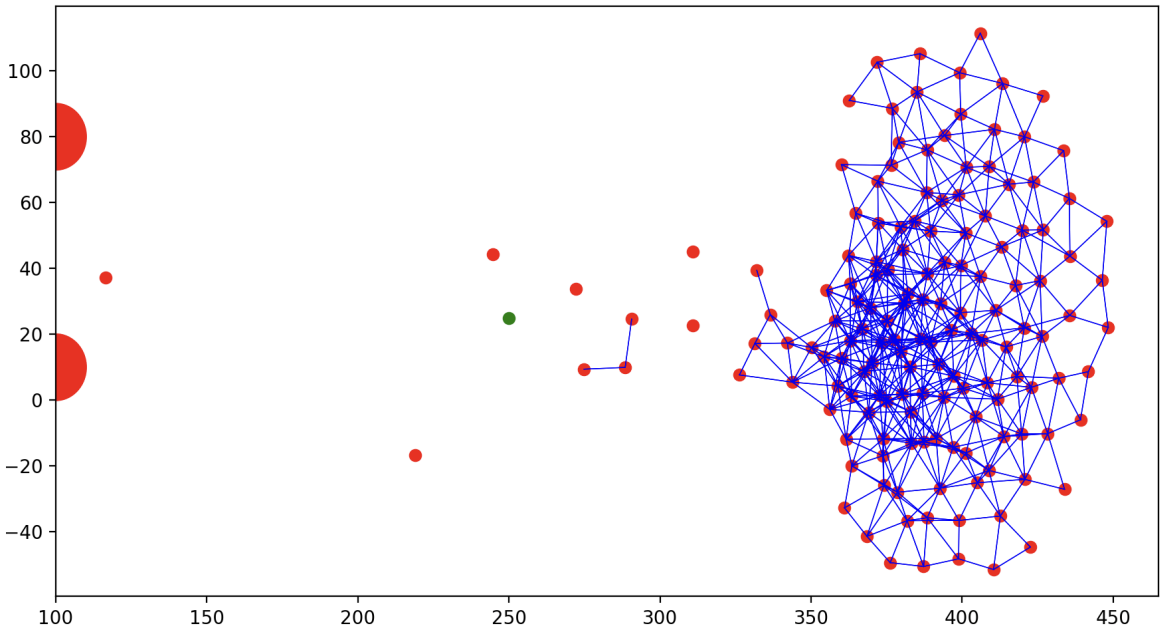
2. **Flocking Snapshots:**

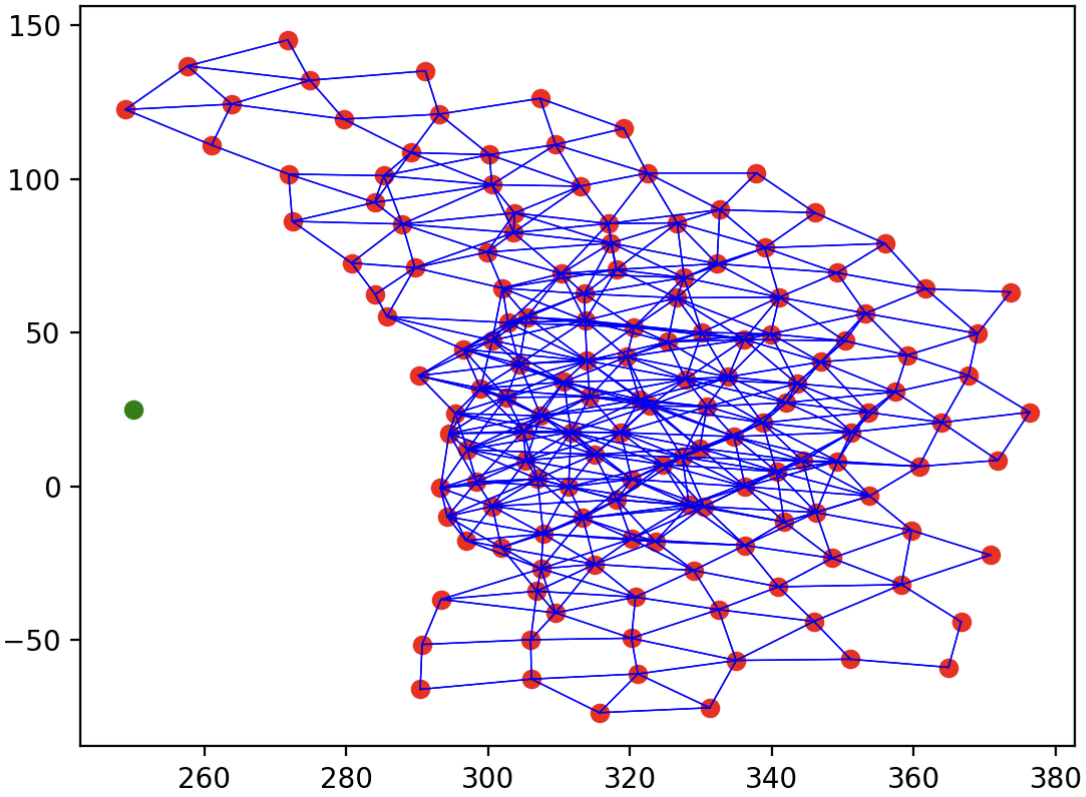



80
60
40
20
0
−20
0
50
100
150
200
250


125
100
75
50
25
0
−25
−50
50
75
100
125
150
175
200
225
250

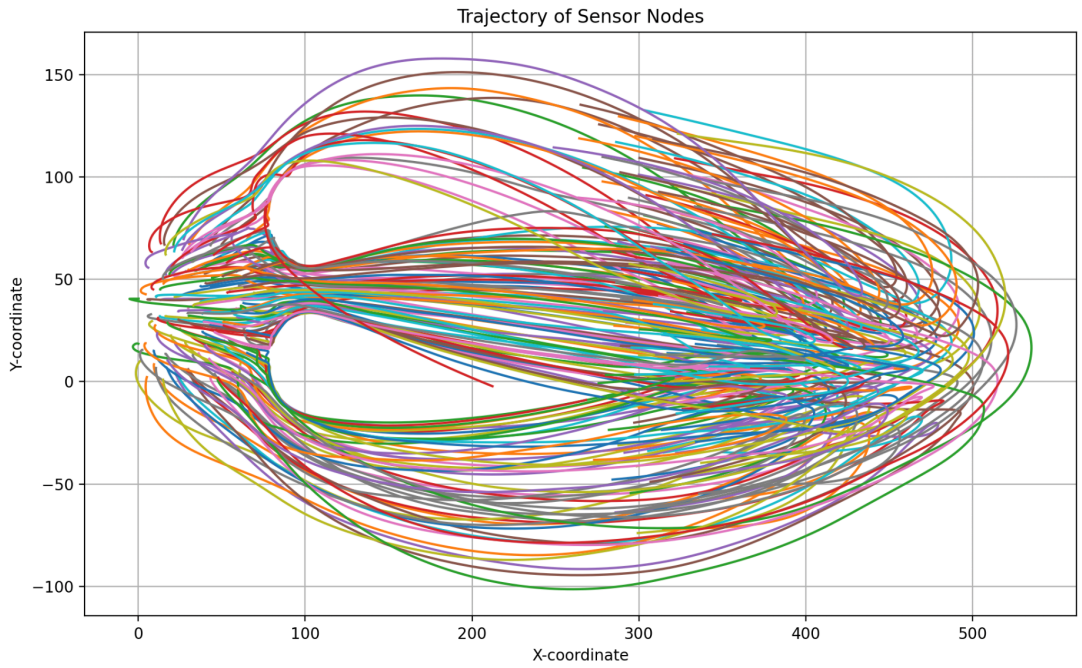

100
50
0
−50
−100
50
100
150
200
250
300
350
400


100
80
60
40
20
0
−20
−40
100
150
200
250
300
350
400
450



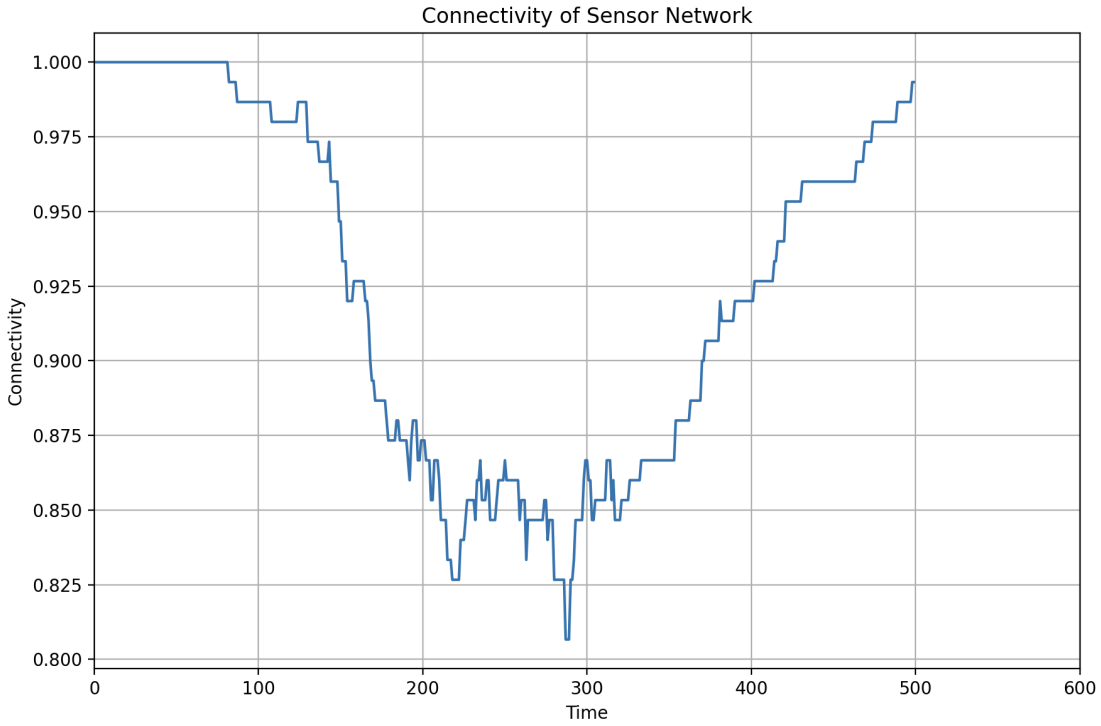
- A series of snapshots are presented to illustrate how sensor nodes flock towards the static target while avoiding obstacle collisions with each other.

3. **Trajectory**



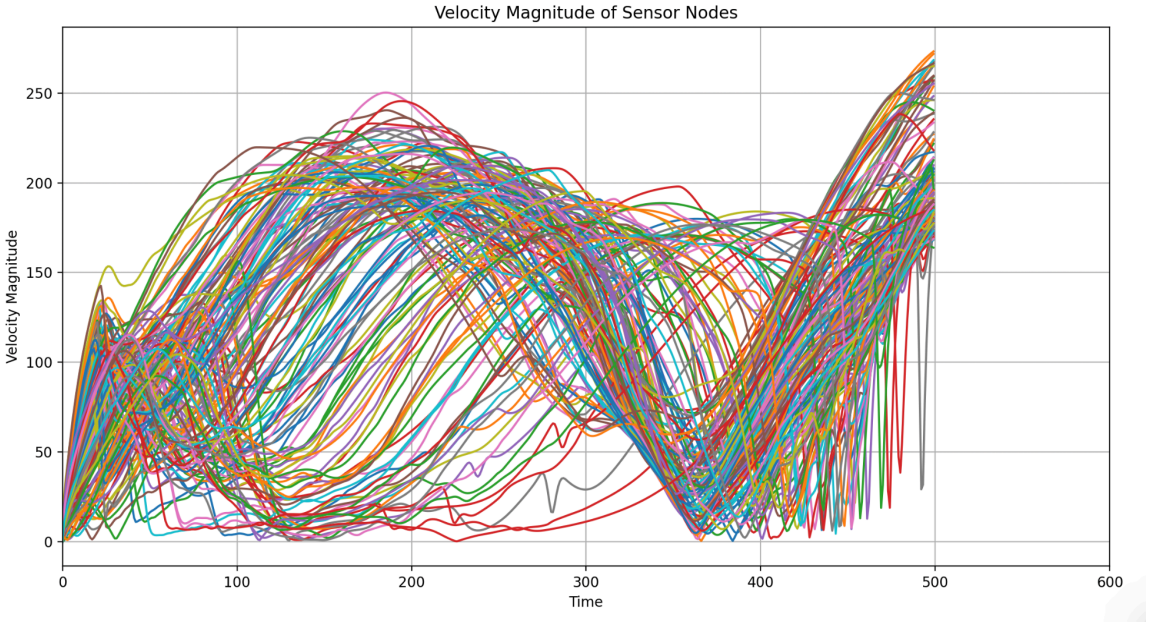
- Trajectory plots provide insights into the movement characteristics of sensor nodes as they navigate towards the target.

4. **Connectivity Check:**



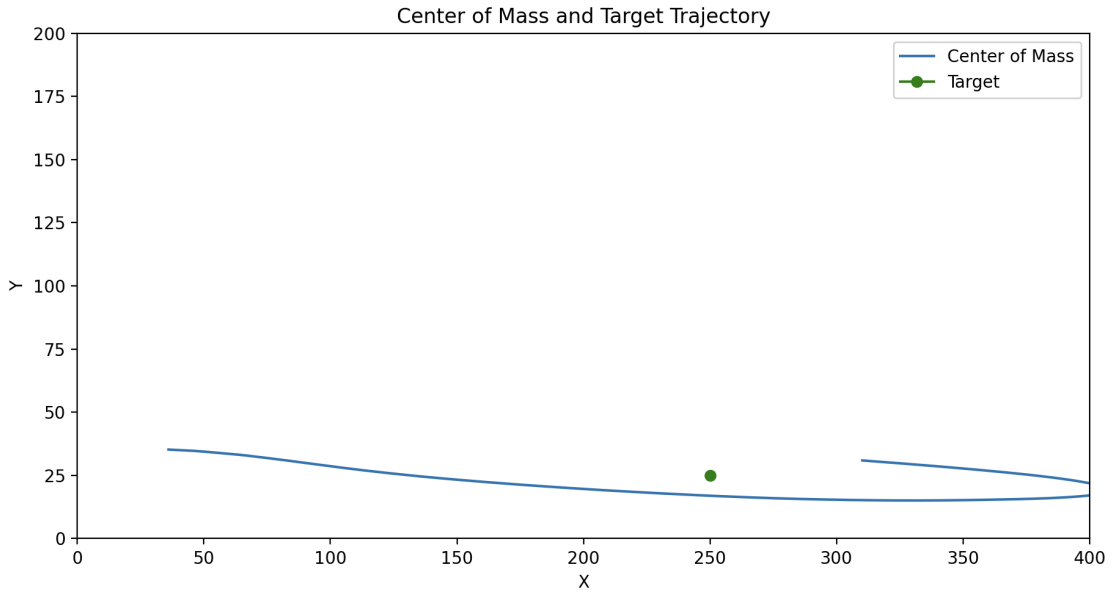
- We assess the connectivity of the network and present a plot to demonstrate the network's connectivity status throughout the simulation.

5. **Velocity**



- Velocity plots provide insights into the movement characteristics of sensor nodes as they navigate towards the target.

6. **Center of Mass (COM) of the MS**



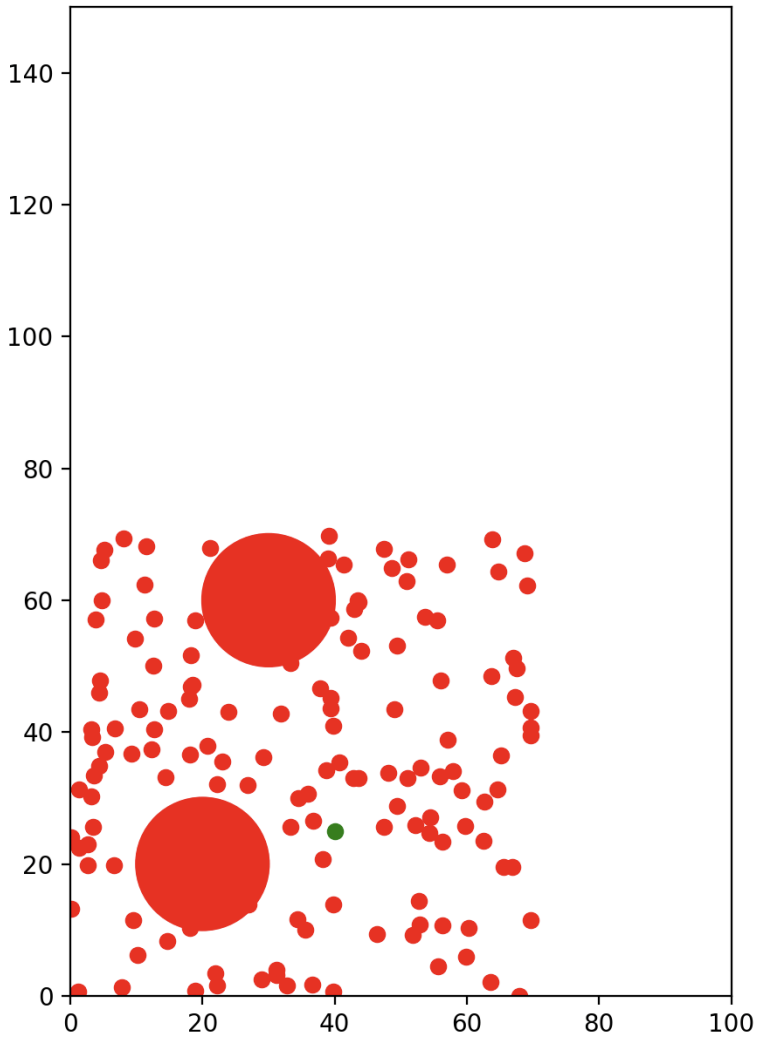
- Track the trajectory of the Center of Mass (COM) of the sensor nodes and overlay it with the trajectory of the static target to showcase the tracking behavior of the MSN.

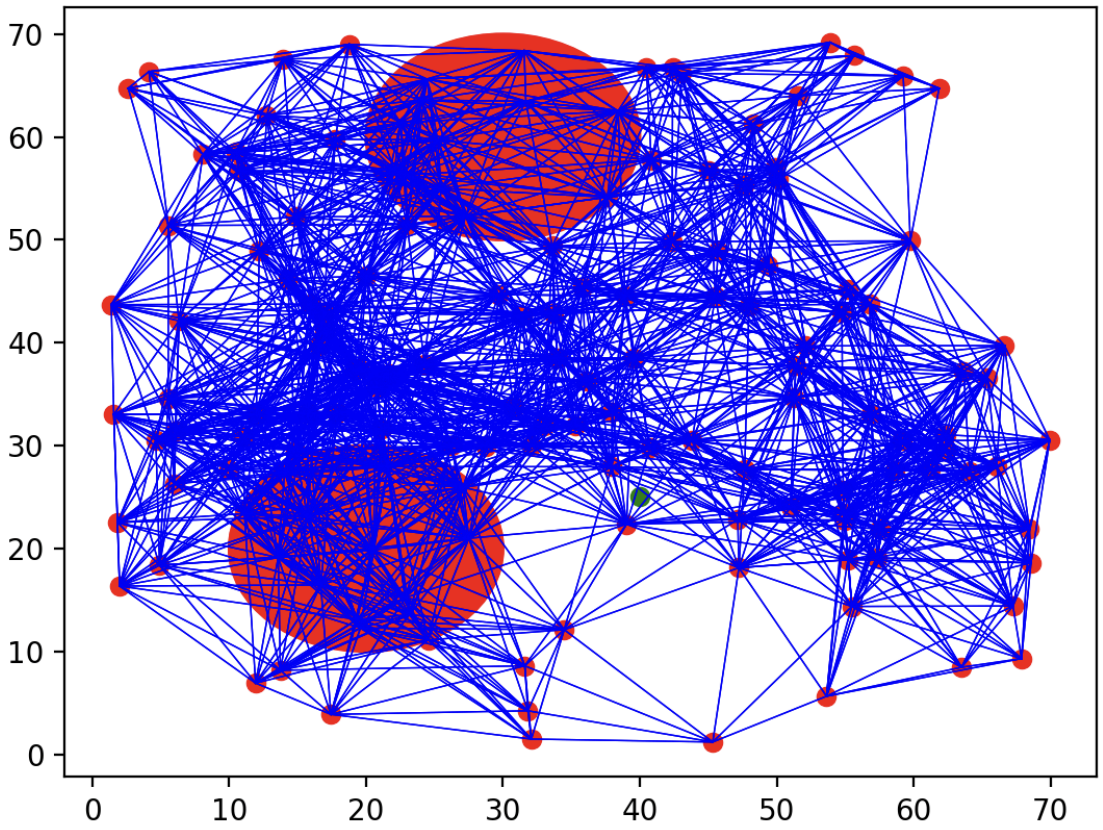
Case 2: MSN Quasi-Lattice Formation with Dynamic Target (Sine Wave) with obstacle avoidance

Implementation Details:
Create a connected network of 150 sensor nodes within a 70x70 area, with a target moving along a sine wave trajectory. Utilize Algorithm 3 to achieve quasi-lattice formation around the moving target while ensuring obstacle avoidance.

Results:

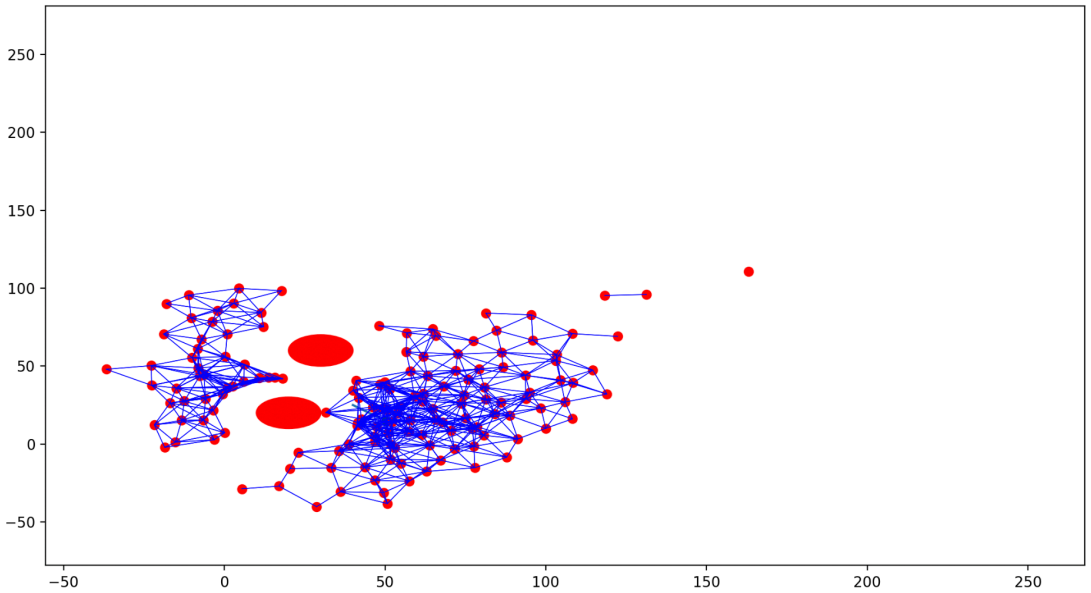
1. **Initial Deployment:**

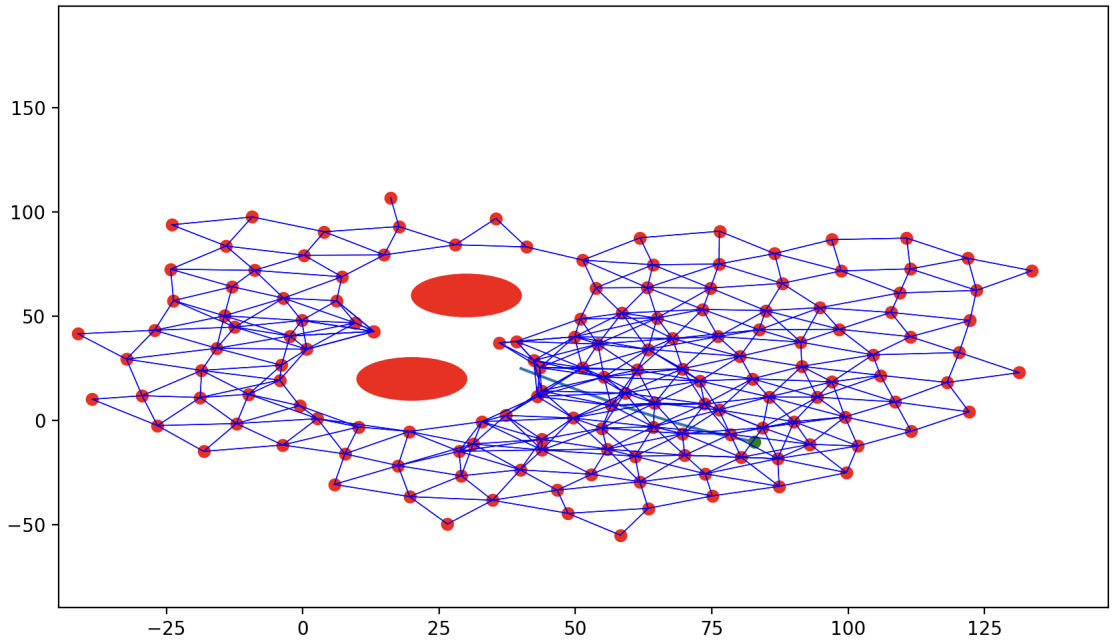


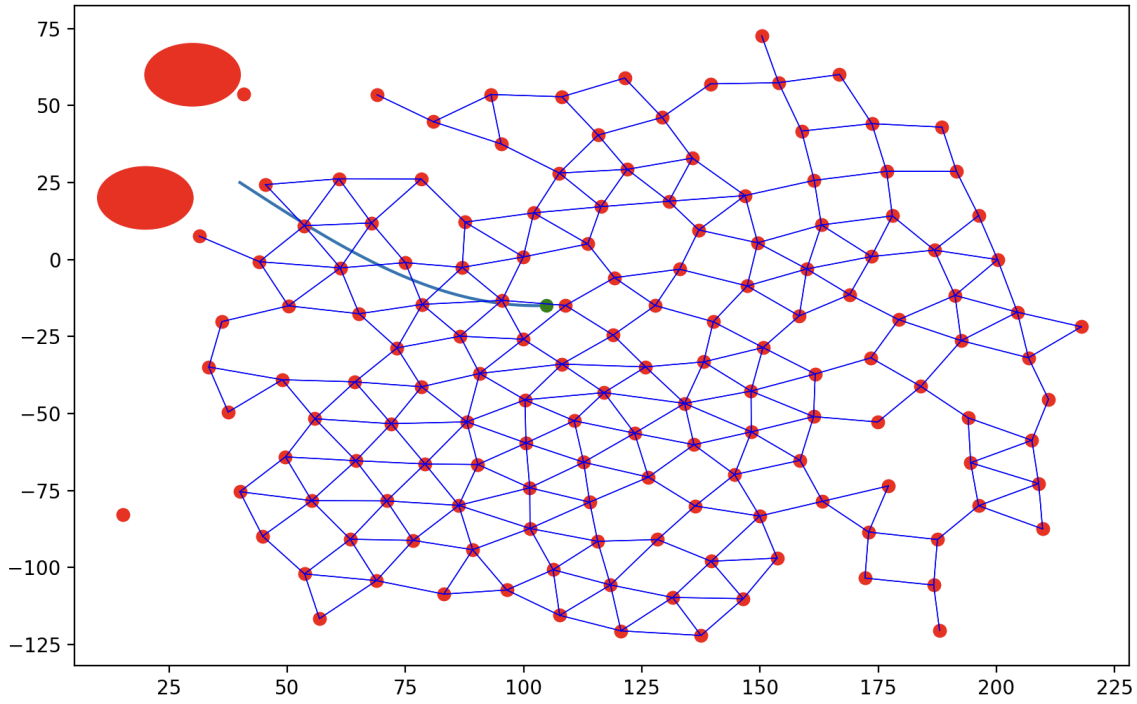


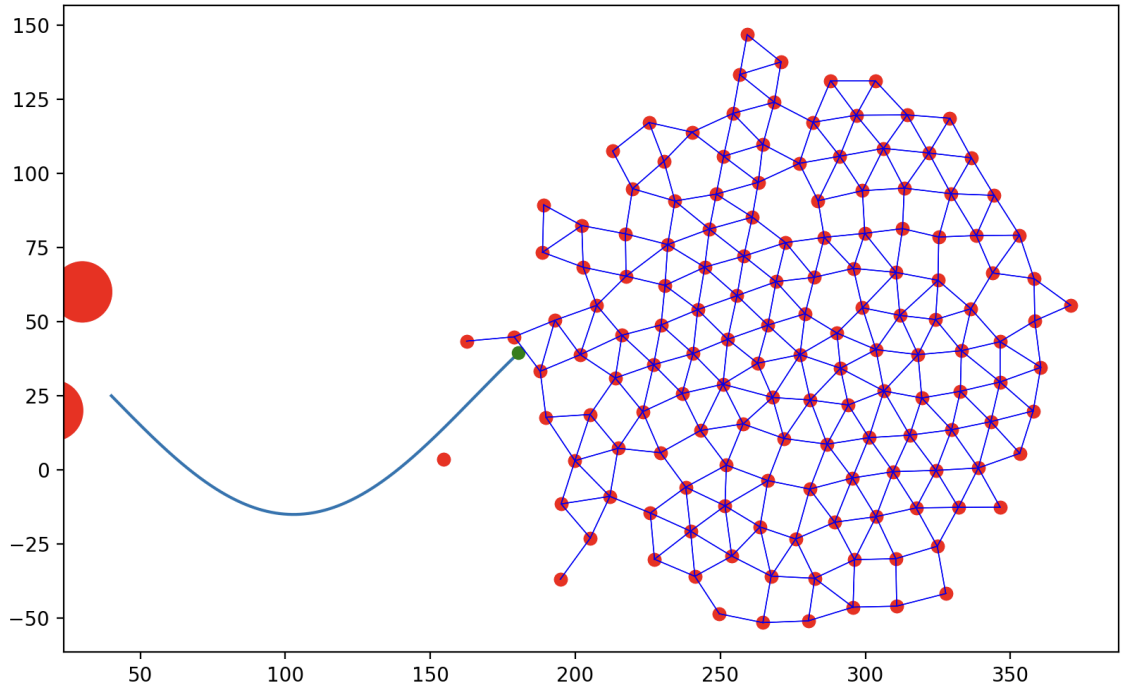
- We visualize the initial deployment of sensor nodes and the moving target, highlighting the connections between neighboring nodes.

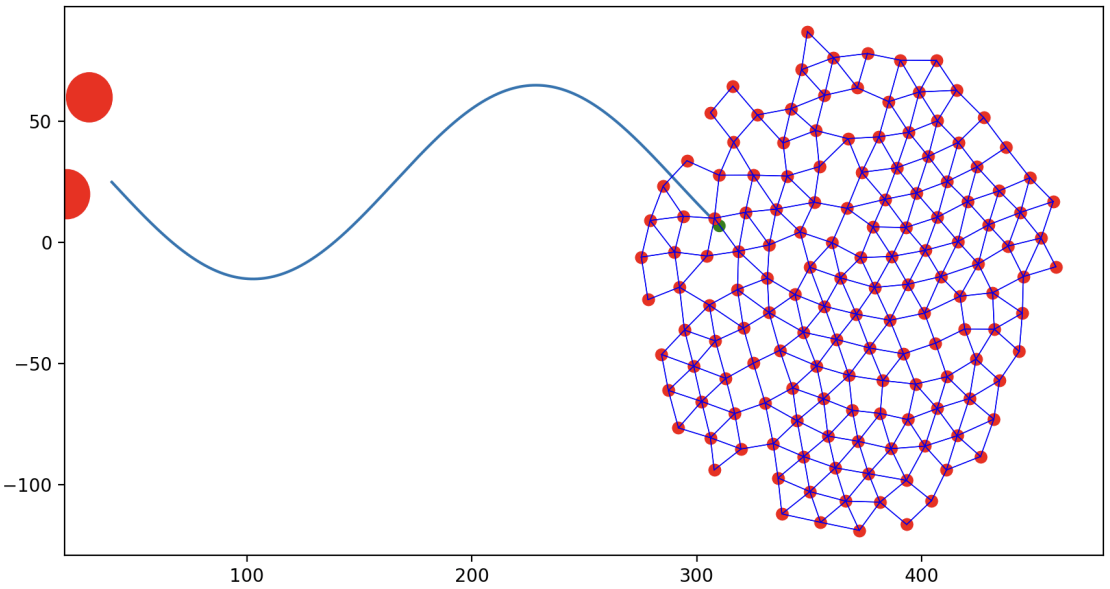
2. **Flocking Snapshots:**





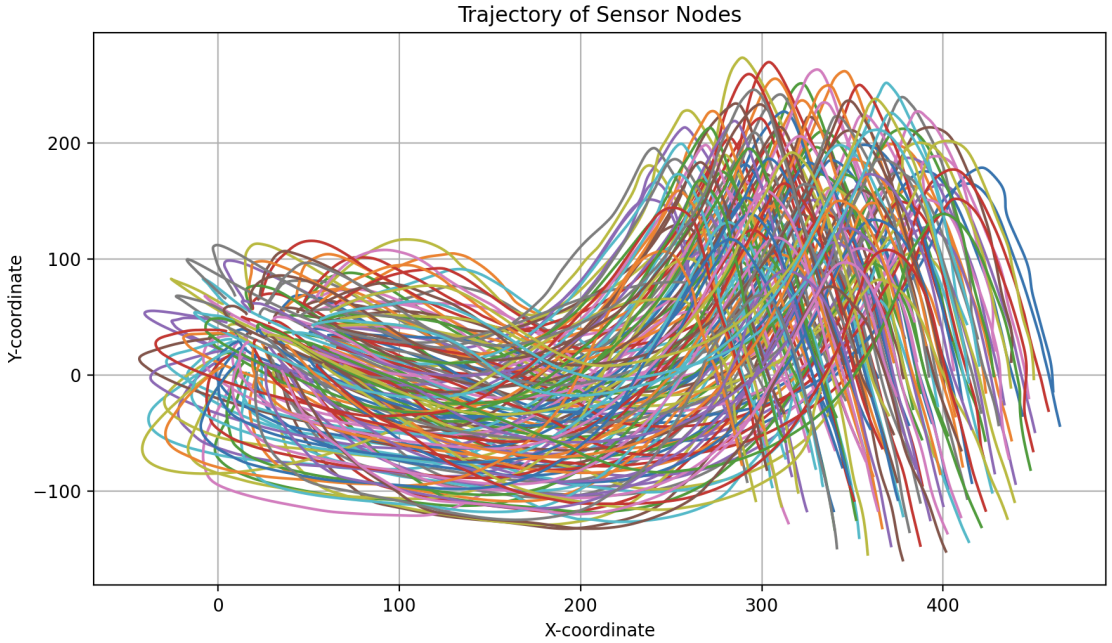

75
50
25
0
−25
−50
−75
−100
−125
25
50
75
100
125
150
175
200
225

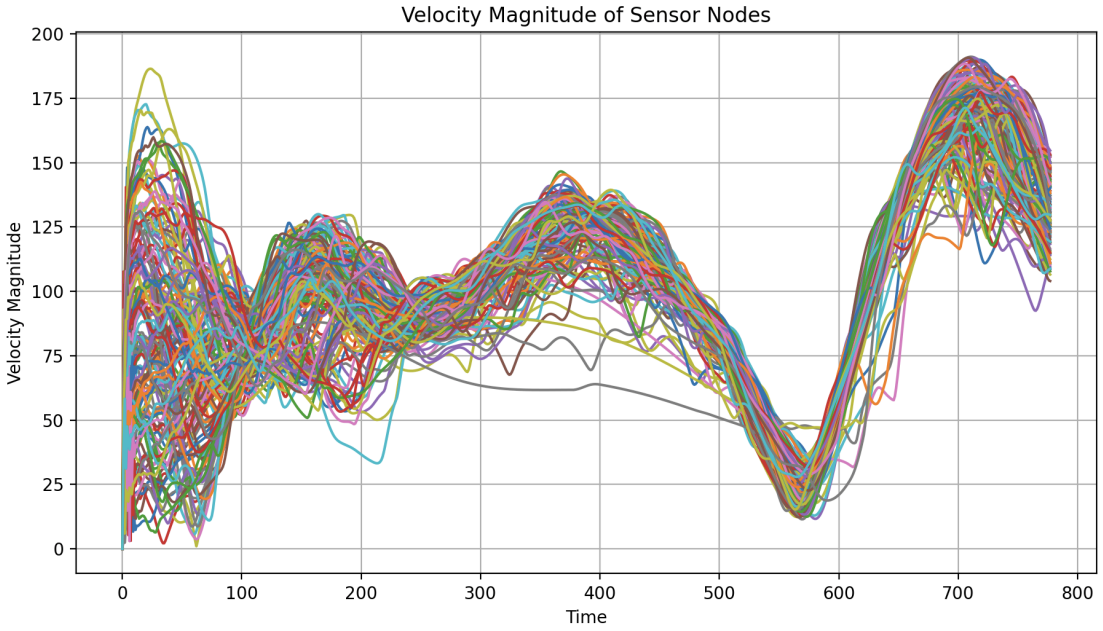

150
125
100
75
50
25
0
−25
−50
50
100
150
200
250
300
350



- A series of snapshots are provided to illustrate how sensor nodes dynamically adjust their positions to track the moving target following a sine wave trajectory.

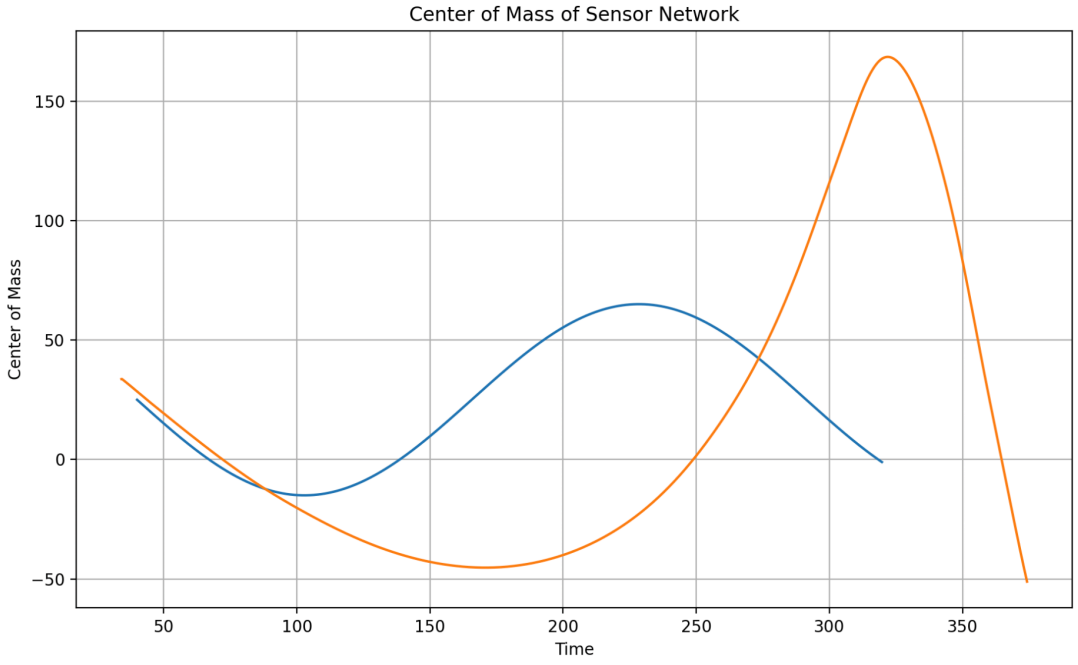
3. **Trajectory and Velocity:**





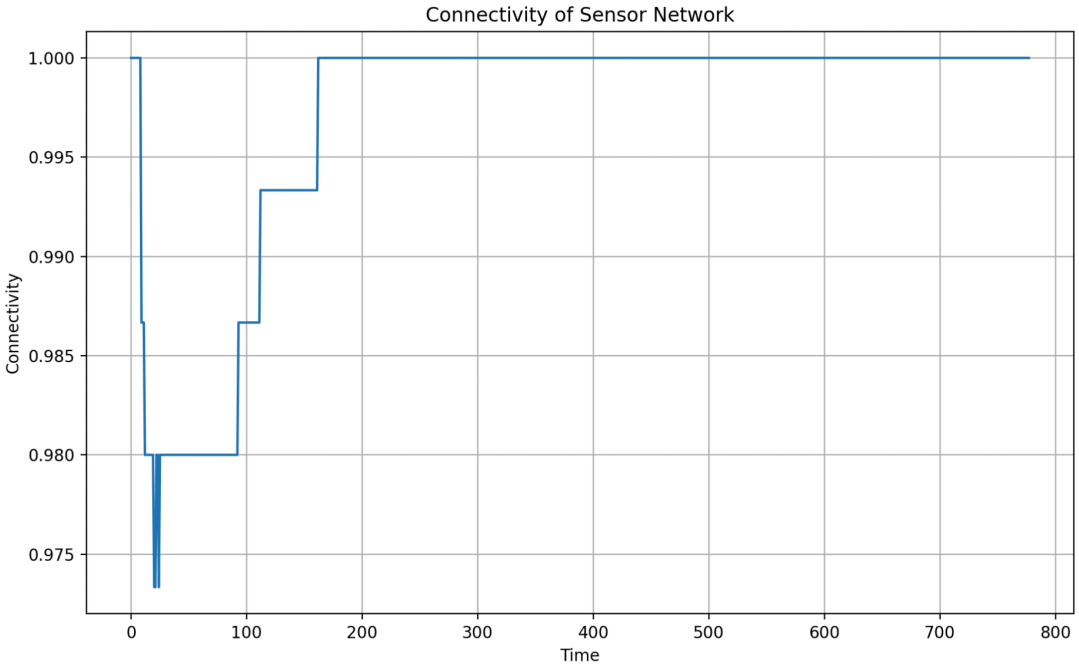
- Trajectory and velocity plots depict the movement patterns of sensor nodes and the target in the xy coordinate space.

4. **Center of Mass Tracking:**



- We track the trajectory of the Center of Mass (COM) of the sensor nodes and overlay it with the trajectory of the moving target to showcase the tracking behavior of the MSN.

5. **Connectivity Check:**



- Connectivity analysis is conducted to evaluate the network's connectivity status over time, with results presented in a plot.

How to Run:

1. Extract the compressed files and open them in an EDI.
2. Install the required libraries mentioned below.

3. Run the script.
4. Run python file using “python filename.py”

Required Libraries:

- numpy: Install using pip install numpy
- matplotlib: Install using pip install matplotlib