Kurtis Bertauche

February 25, 2024

CS 455 – Mobile Sensor Networks

## Project 1

#### Case 1: Algorithm 1 – Fragmentation

1. The initial deployment of 100 nodes in the area of 50x50:

A blue and purple grid

Description automatically generated

1. 6 Snapshots of the of the MSN to show how it is fragmenting:

A blue and purple lines and dots

Description automatically generatedA diagram with dots and lines

Description automatically generated

A graph with many triangles

Description automatically generated with medium confidenceA graph with purple triangles

Description automatically generated

A graph with purple triangles

Description automatically generatedA graph with purple triangles

Description automatically generated

1. Trajectory of all nodes in xy coordinates.

A diagram of a network

Description automatically generated

1. Velocity of all nodes in the xy coordinates

A graph with a line

Description automatically generatedA graph of a number of different colored lines

Description automatically generated

1. Connectivity of the MSN:

A graph with blue lines

Description automatically generated

As expected, the sensor nodes with algorithm 1 applied fragmented. In addition, the nodes did not reach any consensus velocity, and over time their connectivity decreased.

#### Case 2: Algorithm 2 – Static Target

1. Initial deployment of 100 nodes in area of 50x50:

A blue and purple network

Description automatically generated

1. 6 Snapshots to show how the MSN is flocking and tracking a moving target:

A graph with lines and dots

Description automatically generatedA blue and purple grid

Description automatically generated

A graph with a blue circle

Description automatically generatedA graph with a blue circle

Description automatically generated

A blue and purple grid with dots

Description automatically generatedA blue and red dotted circle

Description automatically generated

As shown in the 6 screenshots, the flock quickly forms as it moves towards the target of (150, 150), shown in red. However, the flock passes the target and overshoots, before slowing down and moving in the opposite direction back towards the target, which continues back and forth. This is expected, as the target tracking term in Algorithm 2 continues to add acceleration towards the target until the node has already passed the target, at which point as small acceleration is added back towards the target, but since the node already has a high velocity in the other direction, it takes some time to slow down the flock and turn it back towards the target, which is why the overshooting occurs.

(report continues on next page)

1. Plot the trajectory of all sensor nodes in the xy coordinate:

A graph with a black line

Description automatically generated

1. Plot the velocity of all sensor nodes in the xy coordinate:

A graph of a graph

Description automatically generatedA graph of a number of lines

Description automatically generated with medium confidence

The hyperbolic nature of velocity graphs is consistent with the overshooting behavior observed in the MSN. At the top peak, the flock has reached the goal, and only after passing the goal does the acceleration in the opposite direction begin to apply slowing the flock down as it overshoots. The valley is when the flock has come to a complete stop and is at the furthest point of its “overshoot”, it then speeds up again as it starts heading back to the target. We can see that the nodes fairly quickly math the velocity of the other nodes.

(report continues on next page)

1. Plot the connectivity of the MSN:

A graph with a line

Description automatically generated

The MSN spawned in a fully connected state and the nodes stayed together and cohesive throughout the entire simulation, producing a constant connectivity of 1.

#### Case 3: Algorithm 2 – Sin Wave

1. The initial deployment of 100 nodes in a 150x150 space:

A graph with red lines and green dots

Description automatically generated

1. 6 Snapshots of the MSN to show how it is flocking:

A graph with a red line and a blue line

Description automatically generatedA graph with a blue and red line

Description automatically generated

A graph with red and blue lines

Description automatically generatedA graph with a blue dot and red line

Description automatically generated

A graph with lines and dots

Description automatically generatedA graph of a diagram

Description automatically generated with medium confidence

The flock quickly forms in the direction of the target and begins following it. The flock tracks the target well, with the shown center of mass tracking a near sin wave, with a slight phase shift from the target, which is expected as the flock will not be able to perfectly follow the target as it constantly changes direction and location. In addition, we can see that the flock mates avoid running into each other over time, while the flock is dense, they are not overlapping and they occupy a space of roughly 100x100 units over time, which is plenty of room for the flock to move without neighbors running into each other. This can also be seen in the trajectory graph on the next page, as the lines run parallel to each other.

(report continues on next page)

1. Trajectory of all sensor nodes:

A graph of a wave

Description automatically generated

1. Velocity of all sensor nodes:

A graph with a line

Description automatically generatedA graph of a number of colors

Description automatically generated with medium confidence

The changes in velocity are due to the fact that the target moves slower and the peaks and valleys of the sin wave. It is good that the flock also exhibits this, as this means that the flock is matching the velocity of the target. Additionally, we can see that the nodes continually get closer and closer to having the same velocity over time.

(report continues on next page)

1. Center of Mass of the MSN and the Trajectory of the Target:

A graph of a graph

Description automatically generated

The center of mass makes a near perfect sin wave as it tracks the target, just with a slight phase shift as it catches up to the target which is constantly changing direction.

1. Connectivity of the MSN:

A graph with lines and text

Description automatically generated

As shown, the flock does not start out fully connected, but very quickly all the flock mates meet as they track the target and stay connected for the duration of the simulation.

(report continue on next page)

#### Case 4: Algorithm 2 – Circle

1. Initial deployment of 100 nodes in 150x150 area:

A diagram of a circle with a red circle and blue dots

Description automatically generated

1. 6 Snapshots of the MSN tracking the target:

A diagram of a graph

Description automatically generated25A graph of a diagram

Description automatically generated

A diagram of a circle with a blue dot and red line

Description automatically generatedA diagram of a circle with a blue and red circle

Description automatically generated

A graph with a red circle and blue circles

Description automatically generatedA diagram of a circle with a blue dot

Description automatically generated

1. Trajectory of all sensor nodes in the xy coordinate:

A black circle with a purple center

Description automatically generated

1. Velocity of all nodes in the xy coordinate:

A graph with a line drawn on it

Description automatically generatedA graph of a number of colors

Description automatically generated

As shown the nodes quickly speed up from the initial state of 0 movement, and then come to a consensus velocity fairly quickly.

(report continue on next page)

1. Plot the Center of Mass of the MSN against the Trajectory of the Target:

A graph with a red circle

Description automatically generated

The flock tracks the target well. After forming the flock, it appears to start following the trajectory, just with a slightly slower response so that it does not make as tight of a circle as the trajectory.

1. Plot the connectivity:

A graph with blue lines

Description automatically generated

The flock quickly becomes connected after starting out disconnected and remains connected for the duration of the simulation.

The code for the simulations is contained within project1.py (Python). It uses the matplotlib, numpy, and some functions from the math library. In order for the code to run, you must also have a “circle”, “sin”, “scatter”, and “static” folder in the working directory of the file. The images shown in this report are generated and saved in those directories as the file runs.

The file can be run using Python, no command line arguments are accepted or necessary.