ME-GY 7943
Network Robotic Systems, Cooperative Control and Swarming

Exercise series 1

Exercise 1

We want to deploy a team of robots for a recognition and rescue mission. We have three drones and nine wheeled robots. We create three teams, containing each three wheeled robots and one drone. Due to the terrain conditions, the wheeled robots can only communicate with the same members of the team and do not have access to the other robots. The drones can communicate with the other drones but not with the mobile robots of other teams.

1 Draw the communication graph associated to this scenario (label the vertexes with the robot type).

2 Write down the graph in terms of a set of edges and vertexes.

3 What type of graph is this? (explain)

4 Is there a path from a mobile robot of the first team to a mobile robot of the second and third team? Explain why.

5 Compute the adjacency, degree and incidence matrices of the graph.

6 Compute the graph Laplacian. Is the graph connected? What property of the Laplacian is a necessary and sufficient condition for having a connected graph? Use this property to test if the graph is connected (use a computer if necessary, for example the linear algebra package of Numpy[1])

Exercise 2

Consider the same robotic problem as in Exercise 1. Now we are interested in designing the sensing graph associated to the robot mission. The drones can see the ground robots but the ground robots cannot see the drones. On the ground, the wheeled robots are moving in a line, one robot following the other and a robot can only see the robot that is in front of it. As before, the three teams of ground robots cannot see each other. The drones are flying in a triangle formation, with one drone in front and two in the back: the two robots in the back can see the front robot but they cannot see each other and the front drone cannot see the two robots in the back.

1 From the specification above, draw the sensing graph describing ”who can see who”. Label the vertexes with the robot types.

2 Write down the graph in terms of a set of edges and vertexes.

3 What type of graph is this? (explain)

4 What is the longest path in the graph? Can you find cycles in the graph?

5 Compute the adjacency, in-degree and incidence matrices of the graph.

6 Compute the graph Laplacian. Is the graph weakly connected? Is it strongly connected?

7 Does the graph contain a rooted-out branching? Then, what can you say about the eigenvalues of the Laplacian? Compute them using a computer. Is the graph balanced? Justify your answers.

[1] https://docs.scipy.org/doc/numpy/reference/routines.linalg.html

Exercise 3

Consider the undirected graph shown in Figure 1. Without any computations:

1 What can you say about the eigenvalues of the graph Laplacian?

2 If we were to remove the edge between 4 and 5, how would the eigenvalues change?

3 What is the sum of the rows of the Laplacian matrix of the graph shown in Figure 1? and the columns? Can you explain the result?

4 Using the answer to the previous question, provide an eigenvector and a left eigenvector of the graph Laplacian and their associated eigenvalues. Prove that they are indeed eigenvectors and left eigenvectors[2].

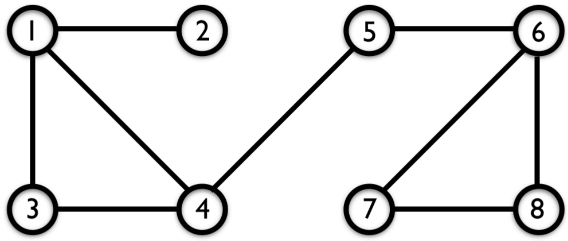

Figure 1:

Exercise 4

In python

1 Write a function *get_laplacian* that takes as input a list of edges (i.e. each element of the list is itself a list that contains the tail and head vertices defining the edge), the number of vertices and a Boolean flag on whether the graph is directed or not and that returns the graph Laplacian as a numpy array. Vertices are numbered starting at 0. For example:

```
# list of edges for a graph containing 3 nodes: 0, 1 and 2
E = [[0,1],[1,2],[2,0]]
n_vertices = 3

# construct an undirected graph
laplacian = get_laplacian(E, n_vertices, False)
print(laplacian)

# construct a directed graph
laplacian = get_laplacian(E, n_vertices, True)
print(laplacian)
```

would print first

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \tag{1}$$

for the first output which is a the Laplacian for the undirected graph and

$$\begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \tag{2}$$

which is the Laplacian for a directed graph given the set of edges.

2 Test this function to compute the Laplacian of the graphs computed in Exercises 1 to 3.

3 Using this function compute the two smallest and the two largest eigenvalues of the cycle graph C_4, C_{10} and C_{100}. You can use sorting functions[3] and linear algebra functions to compute eigenvalues[4]. You are also advised to automatically create the set of edges E (e.g. using a for loop).

[2] remember that an eigenvector v of L is such that $Lv = \lambda v$ where λ is the associated eigenvalue and that a left eigenvector w^T is such that $w^T L = \gamma w^T$, where γ is the associated left eigenvalue.

[3] https://docs.scipy.org/doc/numpy/reference/generated/numpy.sort.html

[4] https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.eigvals.html#numpy.linalg.eigvals

Exercise 5

Using the *get_laplacian* function computed previously, we will simulate, in matrix form, the consensus protocol for the temperature measurement example shown in class and whose communication graph is shown in Figure 2

- Write a function

  ```python
  def simulate_consensus(x_0, T, L, dt=0.001):
  ```

  that takes as input a vector of size n of conditions x_0, a desired integration time T, a graph Laplacian L and an optional integration time dt (which is 0.001 by default) and integrates the consensus protocol as the differential equations $\dot{x} = -Lx$ from $t = 0$ to $t = T$ and returns a vector t containing the time from 0 to T discretized every dt and a matrix (numpy.array) x of size $n \times \frac{T}{dt}$ that contains all n robot states at each instant of time (i.e. $x[i, j]$ contains the state of robot i at time $t[j]$). Use the Euler integration scheme seen in class to do the integration.

2 Simulate the consensus protocol for the initial temperature measurements

$$x_0 = [20, 10, 15, 12, 30, 12, 15, 16, 25] \tag{3}$$

Plot the state of every robot as a function of time in a single graph. What happens to the states when t goes to infinity?

3 Remove the edge between 2 and 4, and between 6 and 7. Will the consensus protocol still converge and if yes, will it converge faster or slower? Justify your answer by looking at the eigenvalues of the Laplacian matrices.

4 Starting from the graph in Figure 2, add an edge between 1 and 3, and one edge between 2 and 8 and remove an edge between 5 and 6. Will the consensus protocol still converge and if yes, will it converge faster or slower? Justify your answer by looking at the eigenvalues of the Laplacian matrices. Verify your answer by simulating the protocol and plot the state of every robot as a function of time in a single graph.

5 Starting from the graph in Figure 2, remove the edges between 5 and 6, 4 and 7, 4 and 6, 2 and 6. Will the consensus protocol still converge and if yes, will it converge faster or slower? Justify your answer by looking at the eigenvalues of the Laplacian matrices. Verify your answer by simulating the protocol and plot the state of every robot as a function of time in a single graph.

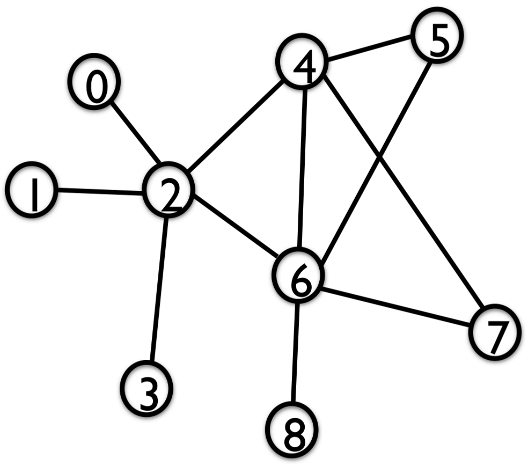

Figure 2: Communication network for the temperature measuring drones