ROB-GY 6333
Swarm Robotics

Homework 2

To make it easy for the TA to grade, please submit your homework as two attachments:

1 . Main document file in *.pdf format that contains all your answers, including any plots where requested

2 . Code used to generate your results as either a Jupyter notebook or MATLAB live script.

Note: My personal recommendation is that you write your entire homework as a stand-alone Jupyter notebook/MATLAB live script and then simply export it all as a single pdf. However, some of you may opt for a boutique, scrapbook approach of stitching together handwritten notes, screenshots, etc. As long as it is legible, it is fine.

As usual, you are encouraged to use Python, as assumed by the instructions below, but there is no penalty for using MATLAB. All code should be written such that it can be run by the TA directly with minimal effort. Include comments explaining how the functions work and how the code should be run if necessary.

Exercise 1

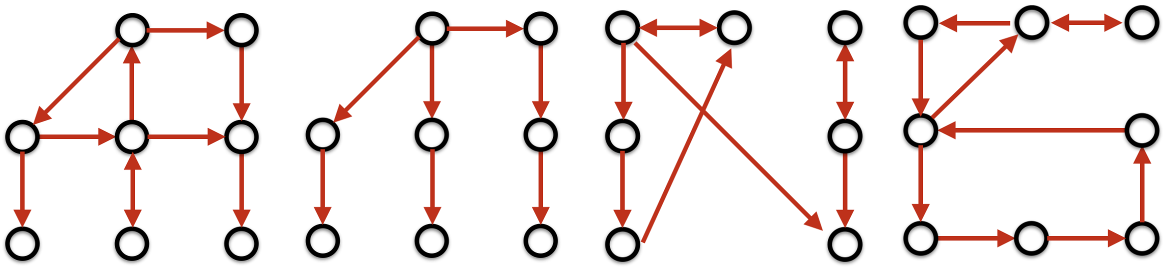
For directed graphs that contain a rooted-out branching, we showed that the consensus algorithm converged to $\mathbf{1}\mathbf{q}_1^T\mathbf{x}_0$, where $\mathbf{q}_1^T$ is a left eigenvector of L (the graph Laplacian) with left eigenvalue 0.

1 Prove that $\mathbf{q}_1^T\mathbf{x}(t)$ is a constant of motion. How can you interpret this result?

Consider the graphs in Figure 1.

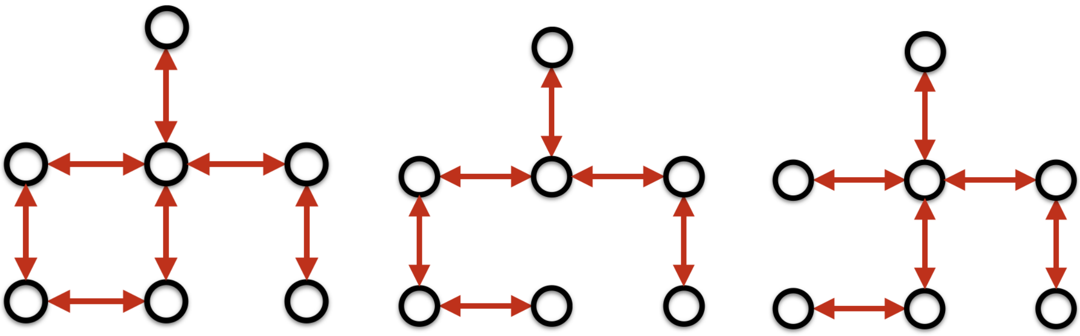
2 Which graph will converge to agreement? Why? Does it depend on the initial conditions?

3 For graphs converging to agreement, what will be the agreed value? Recommendation: write your own function *get_laplacian* to compute the Laplacian from any graph as in put, and compute the (left) eigenvectors to get the agreed value (as a function of the initial conditions).



(a) (b) (c) (d)

Figure 1



(a) (b) (c)

Figure 2

4 Write your own code to simulate the consensus algorithm on each graph starting from random initial conditions for the states of the vertices. Verify your answers from questions 2 and 3 numerically (i.e. in the case of predicted consensus convergence, compute the predicted converged value from point 3 and verify convergence). Plot the graph states as a function of time.

Exercise 2

Consider the following dynamical system

$$\begin{aligned} \dot{x} &= -x + y \\ \dot{y} &= -x - y \end{aligned}$$

1 What are the fixed points of the system? (justify)

2 Consider the following functions

$$\begin{aligned} V_1(x,y) &= \frac{1}{2}x^2 + y^2 \\ V_2(x,y) &= -x^2 + 100y^2 \\ V_3(x,y) &= (x - \frac{1}{2}y)^2 + \frac{7}{4}y^2 \end{aligned}$$

Is any of these functions a Lyapunov function for the system? (Explain) Can you draw any conclusions on the stability of the fixed point(s)?

Exercise 3

1 Consider the scalar system $\dot{x} = ax^p + g(x)$ where p is a positive integer and $g(x)$ satisfies $|g(x)| \leq k|x|^{p+1}$ in some neighborhood of the origin $x = 0$. Show that the origin is asymptotically stable if p is odd and $a < 0$ using a Lyapunov type argument.

2 Find a quadratic Lyapunov function to show that the origin is asymptotically stable for

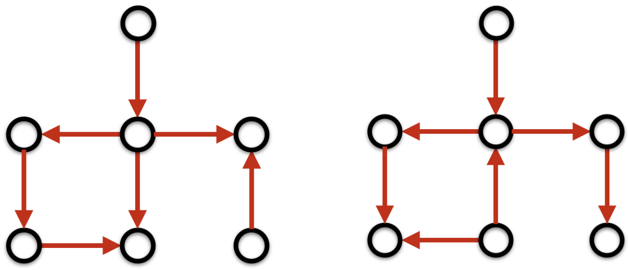
$$\dot{x} = -x + xy \qquad (1)$$
$$\dot{y} = -y \qquad (2)$$

Is the origin globally asymptotically stable? why?

Exercise 4

Consider the graphs in Figure 2.

1 For each graph, will the consensus protocol converge? Justify



(a) (b)

Figure 3

2 The mobile robots trying to reach consensus have their connectivity graph change due to their motion in the field. The graph is switching in a periodic manner from graph a) to graph b) to graph c) and then a) again. Will the agreement protocol converge on the system with switching graphs? Explain.

3 In Python, write a function

```
def simulate_consensus(x_0, T, L_list, switch_time, dt=0.001):
```

that takes as input a vector of size n of conditions x_0, a desired total simulation time T, a list of graph Laplacians *L_list*, a switching time *switch_time*, and an optional integration time dt (which is 0.001 by default) and integrates the consensus protocol when the graphs are switching every *switch_time* seconds. It returns as 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 for the simulation. (There is no benefit to using higher-order Runge-Kutta methods for this assignment, Euler will do fine).

4 Use the function above to simulate the system when switching between graphs happen every 2 seconds (use a random initial state). Compare the behavior when graphs are switching every 0.1 seconds (using the same initial state as before). Plot the behavior in each case.

Exercise 5

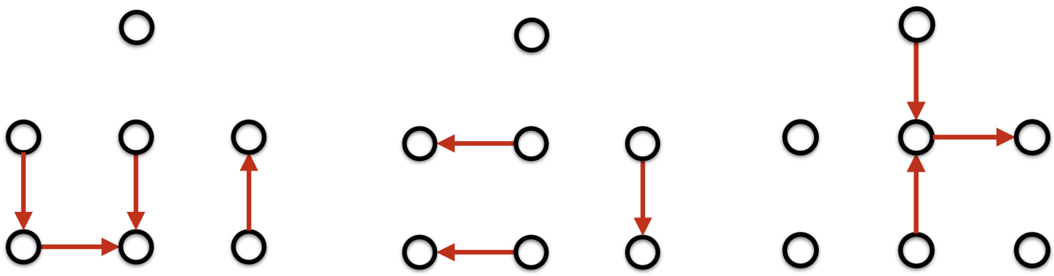
Answer the following four questions for the graphs shown in Figure 3 and Figure 4.

1 Will the consensus protocol converge for each graph shown in the figure? Explain why.

2 Will the consensus protocol converge is the graph is switching back and forth between these two graphs? Explain.

3 Using the Python function written in the previous exercise (and in the previous exercise series), simulate the consensus protocol for each graph separately (no switching) and then for the switching graphs case (assume switching happens every 0.5 second). Plot the behavior of the system.

4 Simulate the switching graph case when switching occurs every 0.1 seconds, every 0.3 seconds, every second and every 2 second. Observing the results, how is convergence affected by switching time?



(a) (b) (c)

Figure 4