EE 5329 Homeworks
Spring 2018

Updated: Monday, April 02, 2018

DO NOT DO HOMEWORK UNTIL IT IS ASSIGNED.
THE ASSIGNMENTS MAY CHANGE UNTIL ANNOUNCED.

- For full credit, show all work.
- Some problems require hand calculations. In those cases, do not use MATLAB except to check your answers.

It is OK to talk about the homework beforehand.

BUT, once you start writing the answers, MAKE SURE YOU WORK ALONE.

The purpose of the Homework is to evaluate you individually, not to evaluate a team.

Cheating on the homework will be severely punished.

The next page must be signed and turned in at the front of ALL homeworks submitted in this course.

Homework Pledge of Honor

On all homeworks in this class - YOU MUST WORK ALONE.

Any cheating or collusion will be severely punished.

It is very easy to compare your software code and determine if you worked together It does not matter if you change the variable names.

Please sign this form and include it as the first page of all of your submitted homeworks.

....………………………………………………………………........

Typed Name: ____

Pledge of honor:

"On my honor I have neither given nor received aid on this homework."

Signature: ____________________________________

EE 5329 Homework 1

State Variable Systems, Computer Simulation

1. Simulate the van der Pol oscillator $y''+\alpha(y^2-1)y'+y=0$ using MATLAB for various ICs. In each case below, first use ICs of y(0)=0, y'(0)= 0.5. Then use ICs of y(0)=4, y'(0)= 4. Plot y(t) vs. time *t* and also the phase plane plot y'(t) vs. y(t).
   a. For α= 0.05.
   b. For α= 0.9.

2. Do MATLAB simulation of the Lorenz Attractor chaotic system. Run for 150 sec. with all initial states equal to 0.5. Plot states versus time, and also make 3-D plot of x_1, x_2, x_3 using PLOT3(x1,x2,x3).

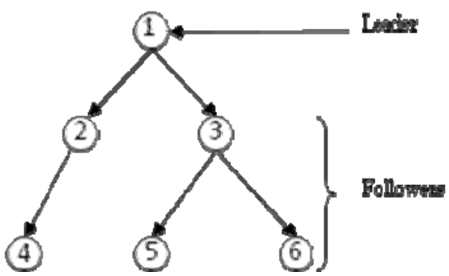
$$\dot{x}_1 = -\sigma(x_1 - x_2)$$
$$\dot{x}_2 = rx_1 - x_2 - x_1x_3$$
$$\dot{x}_3 = -bx_3 + x_1x_2$$

use σ= 10, r= 28, b= 8/3.

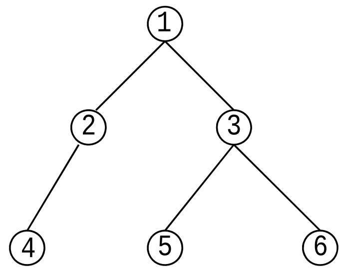
EE 5329 Homework 2

Graph Laplacian Eigenvalues

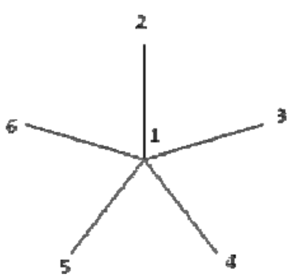
For the following graphs, take all edge weights equal to 1/2. (a) Write the adjacency matrix A and the graph Laplacian L. (b) Find the eigenvalues of L and plot in the complex s-plane. (c) Compare the Fiedler e-vals λ_2. (d) Find the left eigenvector w_1 of L for $\lambda_1 = 0$. (e) Find the Fiedler e-vector v_2.



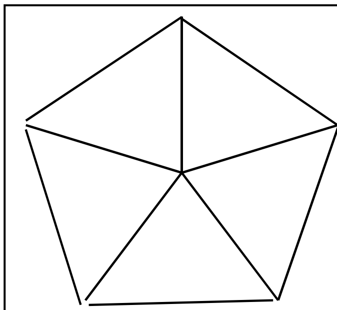
1. Directed Tree graph. (Formation graph)



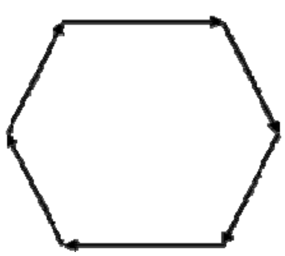
2. Undirected Tree graph.



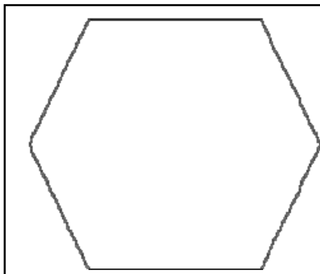
3. Star $K_{1,5}$



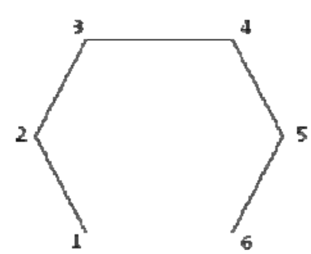
4. Star with extra edges



5. Directed 6-cycle (6-periodic graph)



6. 6-cycle C^6 (2-regular graph).



7. Path P^5.

EE 5329 Homework 3

Consensus and Graph Eigenvalues

1. Continuous-Time Consensus

Simulate the continuous-time consensus protocol

$$\dot{x}_i = u_i = \sum_{j \in N_i} a_{ij}(x_j - x_i).$$

for all the graphs on Homework 2. Take all edge weights equal to 1. For each case, plot all the states versus time.

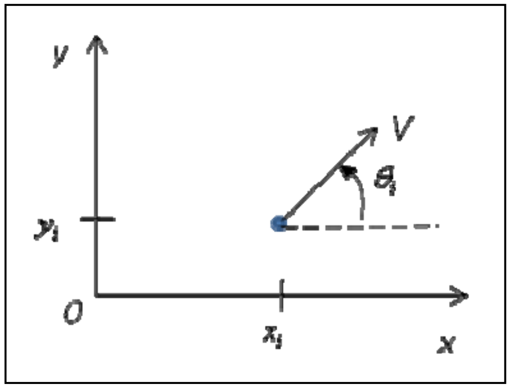
2. Consensus for Formation Control

Let each node have the vehicle dynamics given by

$$\dot{x}_i = V \sin\theta_i$$

$$\dot{y}_i = V \cos\theta_i$$

with $(x_i(t), y_i(t))$ the position and $\theta_i(t)$ the heading. This corresponds to motion in the *(x,y)* plane with velocity *V* as shown. All nodes have the same velocity *V*.

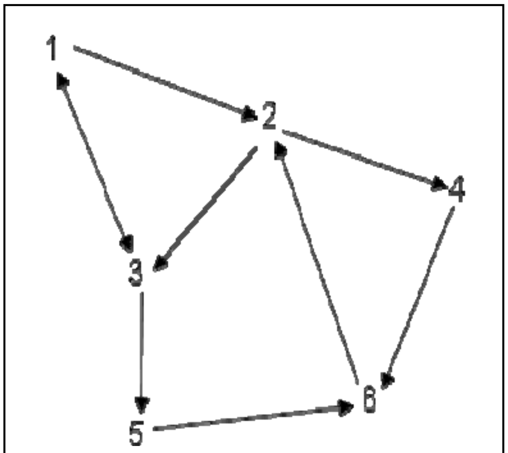


Take a flock of 6 nodes with the strongly connected communication graph structure shown. Run the continuous-time local voting protocol on the headings of the nodes as

$$\dot{\theta}_i = \sum_{j \in N_i} a_{ij}(\theta_j - \theta_i)$$

Take all the edge weights equal to 1. Set the initial headings random between $0 - 2\pi$ rads.

Now, you have 3 states running at each node: $\theta_i(t)$, $(x_i(t), y_i(t))$. Plot the six headings vs time. They should reach the weighted average consensus. Plot the six positions $(x_i(t), y_i(t))$ of the nodes in the plane vs time. They should eventually all go off in the same direction.



Graph for formation control.

EE 5329 Homework 4

Discrete-time Consensus

1. **Discrete-Time Consensus**
   Simulate the DT consensus protocol using the normalized form protocol

$$x_i(k+1) = x_i(k) + \frac{1}{d_i+1}\sum_{j\in N_i} a_{ij}(x_j(k) - x_i(k)) = \frac{1}{d_i+1}\left(x_i(k) + \sum_{j\in N_i} a_{ij}x_j(k) \right)$$

on graphs 3 and 7 from Homework 2. Take all edge weights equal to 1//2. For each case, plot all the states versus time. Start with random initial condition states between -1, +1.

EE 5329 Exam 1 – Takehome exam
Formation Control

1. **Formation control with desired position offsets.**

The Newton's motion dynamics for motion in the plane are given by

$$\dot{x}_i = v_i \qquad \dot{x}_0 = v_0$$
$$\dot{v}_i = u_i \qquad \dot{v}_0 = u_0$$

with vector position $x_i \in R^2$, velocity $v_i \in R^2$, and acceleration input $u_i \in R^2$. Take $x_i = [p_i \ \ q_i]^T$ where $(p_i(t), q_i(t))$ is the position of node *i* in the *(x,y)*-plane.

Define the formation control protocol as position and velocity local neighborhood feedback

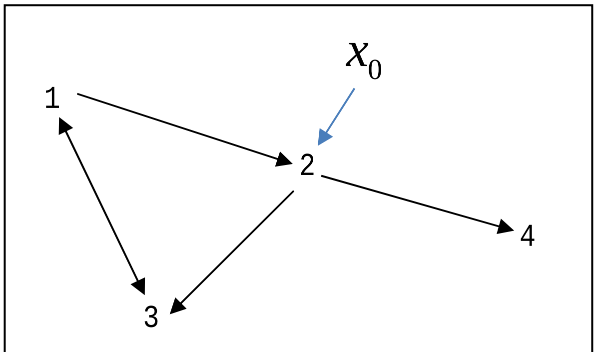
$$u_i = cK_p \left[\sum_{j \in N_i} a_{ij}(x_j - \Delta_j - x_i + \Delta_i) + g_i(x_0 - x_i + \Delta_i) \right]$$
$$+ cK_d \left[\sum_{j \in N_i} a_{ij}(\mathrm{v}_j - v_i) + g_i(\mathrm{v}_0 - v_i) \right]$$

with coupling gain $c > 0$, and Δ_i the constant desired position offset in the formation of node *i* from the leader node. The pinning gains g_i are only nonzero for a small subset of the nodes in the graph. Select PD control gains $K_p = I_2, K_d = \gamma I_2$.

Simulate the above systems. Take the formation graph shown with 4 agents, which contains a spanning tree. Take edge weights and pinning gains as ½. Select γ large enough to get stability.

The desired position offsets are 2-vectors to specify the desired *x* and *y* position of node *i*. Take the desired position offsets as the 4 corners of a square around the leader

$$\Delta_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \Delta_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \Delta_3 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \Delta_4 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$



Communication Graph

The leader has the same dynamics as the agents with $u_0 = 0$, $x_0(0) = (0,0), v_0(0) = (1,1)$. Select random initial positions and velocities for the agents in the square $[2,-2] \times [2,-2]$. Simulate and plot positions in the 2-D plane. Verify that formation consensus is reached.

EE 5329 Homework 5
Mobile Robot Control & Potential Fields

1. **Potential Field.** Use MATLAB to make a 3-D plot of the potential fields described below. You will need to use plot commands and maybe the mesh function. The work area is a square from (0,0) to (13,13) in the (x,y) plane. The goal is at (10,10). There are obstacles at (3,2) and (6,7). Use a repulsive potential of K_i / r_i for each obstacle, with r_i the vector to the *i*-th obstacle. For the target use an attractive potential of $K_T r_T$, with r_T the vector to the target. Adjust the gains to get a decent plot. Plot the sum of the three potential fields in 3-D.

2. **Potential Field Navigation.** For the same scenario as in Problem 1, a mobile robot starts at (0,0). The front wheel steered mobile robot has dynamics

$$\dot{x} = V\cos\varphi\cos\theta$$
$$\dot{y} = V\cos\varphi\sin\theta$$
$$\dot{\theta} = \frac{V}{L}\sin\varphi$$

with (x,y) the position, θ the heading angle, *V* the wheel speed, *L* the wheel base, and ϕ the steering angle. Set *L*= 4.

   a. Compute forces due to each obstacle and goal. Compute total force on the vehicle at point *(x,y)*.
   b. Design a feedback control system for force-field control. Sketch your control system.
   c. Use MATLAB to simulate the nonlinear dynamics assuming a constant velocity V and a steerable front wheel. The wheel should be steered so that the vehicle always goes downhill in the force field plot. Plot the resulting trajectory in the *(x,y)* plane.

3. **Extra Credit - Swarm/Platoon/Formation.** Do what you want to for this problem. The intent is to focus on some sort of swarm or platoon or formation behavior, not the full dynamics. Therefore, take 6 vehicles each with the simple point mass (Newton's law) dynamics

$$\ddot{x} = F_x / m$$
$$\ddot{y} = F_y / m$$

with *(x,y)* the position of the vehicle and F_x, F_y the forces in the *x* and *y* direction respectively. The forces might be the sums of attractive forces to goals, repulsive forces from obstacles, and repulsive forces between the agents.

Make some sort of interesting plots or movies showing the leader going to a desired goal or moving along a prescribed trajectory and the followers staying close to him, or in a prescribed formation. Obstacle avoidance by a platoon or swarm is interesting.

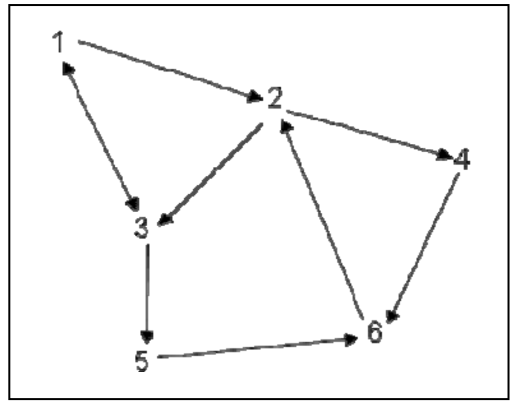
EE 5329 Homework 6
Time-varying Graph Topologies

1. Discrete-Time Consensus with Varying Graph Topology

We want to simulate DT consensus for the normalized protocol systems

$$x_i(k+1) = x_i(k) + \frac{1}{d_i+1}\sum_{j\in N_i} a_{ij}(x_j(k) - x_i(k))$$

$$= \frac{1}{d_i+1}\left(x_i(k) + \sum_{j\in N_i} a_{ij}x_j(k) \right)$$



The global form of this is

$$x(k+1) = x(k) - (I+D)^{-1}Lx(k)$$

$$= (I+D)^{-1}(I+A)x(k) \equiv Fx(k)$$

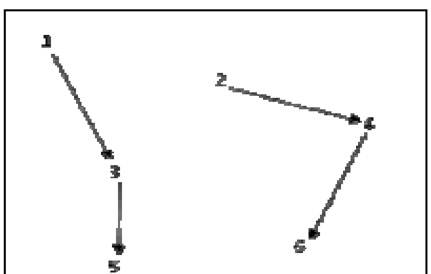
a. Simulate this algorithm for the strongly connected digraph shown above. Start with random initial conditions between -1, +1. Take edge weights of 1.

b. Shown below are three disconnected graphs whose union is the original strongly connected graph.

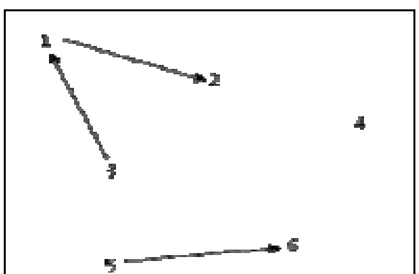
Simulate the DT protocol for the case of time-varying protocols

$$x(k+1) = F(k)x(k) = (I+D(k))^{-1}(I+A(k))x(k)$$

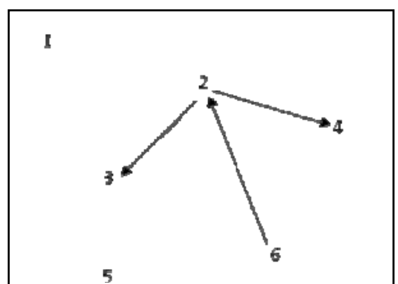
where *F(0)* corresponds to graph G1, *F(1)* to Graph G2, *F(2)* to G3, and then this cycle 123 123 is repeated. Use the same initial conditions as in part 2a. Plot the states vs. time. Verify that consensus is achieved. Are the consensus values the same as in Problem 2a?



Graph G1



Graph G2



Graph G3

c. Repeat for a different order of switching between the three graphs. Use the cycle 12222233333 12222233333 ad infinitum. Use the same initial conditions as in part 2a.

Compare to part a. Is the consensus value the same?

2. Gossip Algorithms

Use the same strongly connected graph as in problem 1. Simulate the gossip algorithm for average consensus. That is, at each step, select an edge at random. Update the states of the two nodes joined by the edge according to

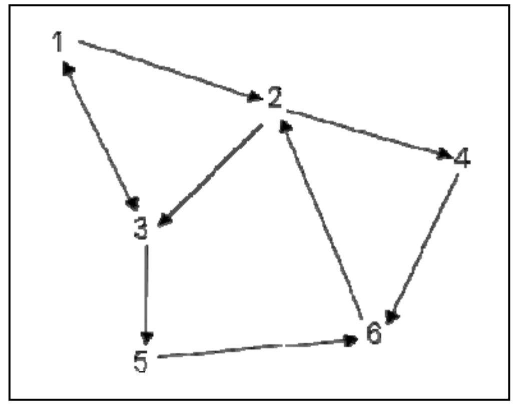
$$x_i = x_i + \frac{x_j - x_i}{2}, \qquad x_j = x_j - \frac{x_j - x_i}{2}$$

Start with random initial conditions. Plot the states vs. time. Verify that consensus is reached. The consensus value should be the average of the initial states.

EE 5329 Exam 2
Parallel Algorithms

1. Parallel Solution of Equations

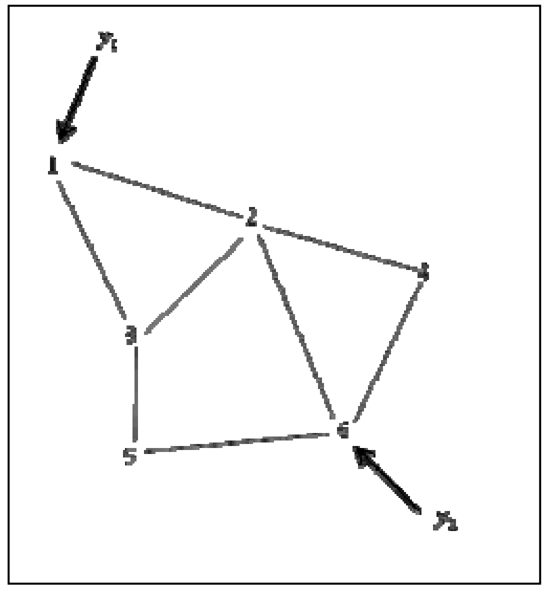
We want to solve the equation $Lx = b$ where



$$L = \begin{bmatrix} 2 & 0 & -1 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 & 0 & -1 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix} , \quad b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

a. Invert L to solve the equation.
b. Use the Jacobi method to solve the equation.

Note that this is the Laplacian matrix from the book Example 2.1 with a 1 added to element (1,1). See Fig. 1. What does it mean?



Parallel Processing to Solve Electric networks

2. Undirected graph

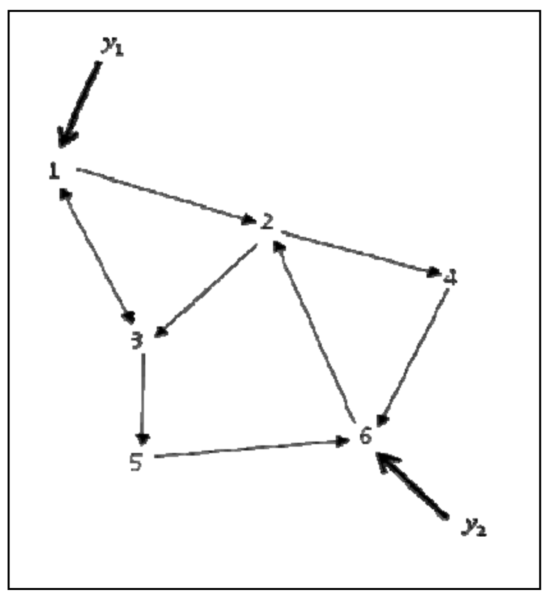
Consider the electric circuit shown in Fig. 2 where all the edges represent conductances of 1 unit (ohm^{-1}) and there are two external voltage sources $y_1 = 10v, y_2 = 0v$. Use the algorithm

$$\dot{x}_i = \sum_{j \in N_i} a_{ij}(x_j - x_i) + b_{ik}(y_k - x_i)$$

to compute the steady-state voltages induced at each node. Plot the voltages vs. time.

3. Undirected graph, only one V source

Repeat for the case where $y_1 = 10v$ and there is no source y_2. Note that this is not the same as $y_2 = 0v$. Plot the voltages vs. time. Verify that all nodes reach consensus at the value $y_1 = 10v$.



4. Directed Graph.

Repeat for the directed graph shown in Fig. 3, where all edges have weights of 1. Plot the voltages vs. time. They should reach steady-state. What does it mean to have a circuit with directed edges? Does Kirchhoff's current law hold at steady-state- the sum of currents into each node equals zero?