EE 5329 Homeworks
Spring 2013

Updated: Monday, March 11, 2013

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

EE 5329 Homework 0

DO NOT TURN IN.
This homework is not assigned, but make sure you know how to do it using MATLAB function ODE23.

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. Plot y(t) vs. t and also the phase plane plot y'(t) vs. y(t). Use y(0)=0.2, y'(0)= 0.
   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 0.3. 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.

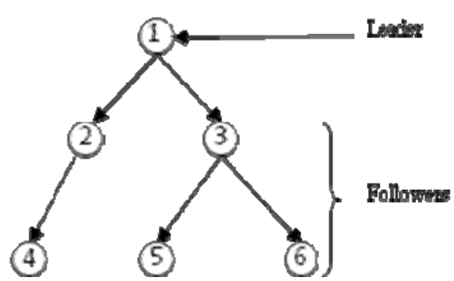
3. A system has transfer function $H(s) = \dfrac{s+4}{s^2+4s+13}$

   a. Use MATLAB to make a 3-D plot of the magnitude of H(s)
   b. Use MATLAB to make a 3-D plot of the phase of H(s)
   c. Use MATLAB to draw magnitude and phase Bode plots

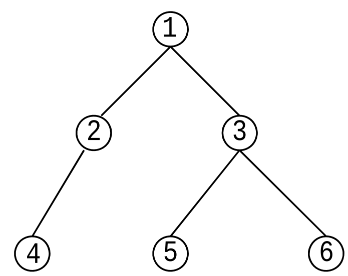
EE 5329 Homework 1

Graph Laplacian Eigenvalues

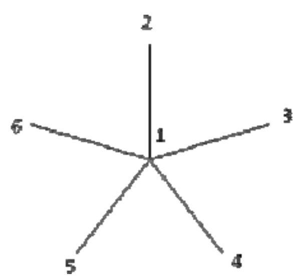
For the following graphs, take all edge weights equal to one. (a) Write the adjacency matrix A and the graph Laplacian L. (b) Find the eigenvalues of L and plot in the complex s-plane. (c) Compare the Fiedler e-vals λ_2. (d) Find the left eigenvector w_1 of L for $\lambda_1 = 0$. (e) Find the Fiedler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



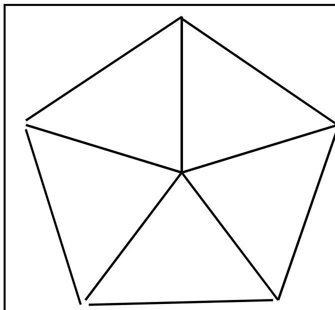
1. Directed Tree graph. (Formation graph)

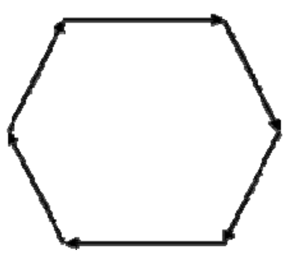


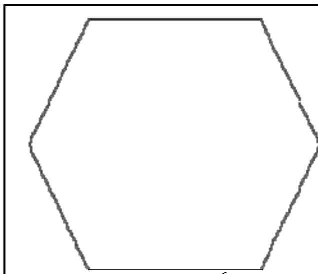
2. Undirected Tree graph.

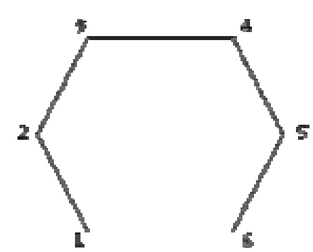


3. Star $K_{1,5}$


4. Star with extra edges


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


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



7. Path P^5.

DO NOT DO

(c) Find the eigenvalues of the normalized graph Laplacian $\bar{L} = D^{-1}L$ and plot in the complex *s*-plane.

EE 5329 Homework 2

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 graphs 1, 3, 5, 7 on homework 1. Take all edge weights equal to 1. For each case, plot all the states versus time.

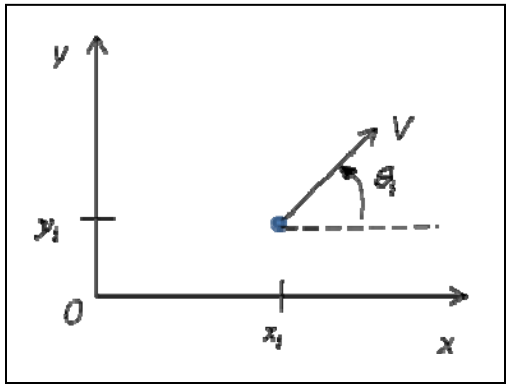
2. Discrete-Time Consensus

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 1 and 5 from homework 1. Take all edge weights equal to 1. For each case, plot all the states versus time.

3. Consensus for Formation Control



Let each node have the vehicle dynamics given by

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

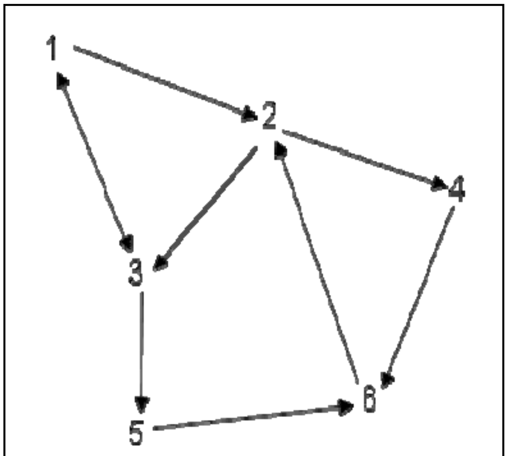
with $(x_i(t), y_i(t))$ the position and $\theta_i(t)$ the heading. This corresponds to motion in the *(x,y)* plane with velocity *V* as shown. All nodes have the same velocity *V*.

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 3
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 (12,12) in the (x,y) plane. The goal is at (10,10). There are obstacles at (4,6) and (6,4). 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 force 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

$$\dot{x} = V \cos\phi \sin\theta$$

$$\dot{y} = V \cos\phi \cos\theta$$

$$\dot{\theta} = \frac{V}{L} \sin\phi$$

with (x,y) the position, θ the heading angle, V the wheel speed, L the wheel base, and ϕ the steering angle. Set L= 2.

   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. **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 5 vehicles each with the simple point mass (Newton's law) dynamics

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

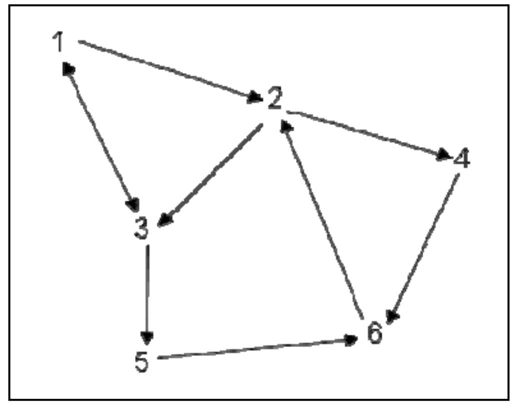
$$\ddot{y} = F_y / m$$

with (x,y) the position of the vehicle and F_x, F_y the forces in the x and y direction respectively. 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 Takehome Exam 1

Switched Topologies and Second-order Consensus

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

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

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

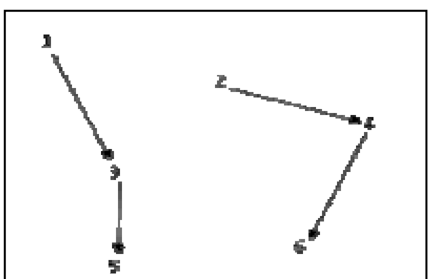
Use the strongly connected digraph shown.

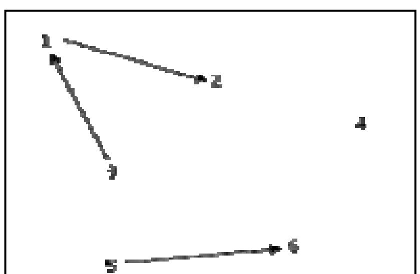
Shown below are three disconnected graphs whose union is the original strongly connected graph.

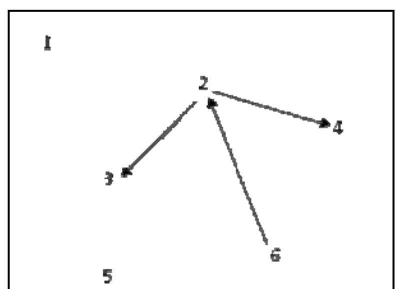
a. Simulate the DT protocol for the case of time-varying protocols

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

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


Graph G1


Graph G2


Graph G3

b. Repeat for a different order of switching between the three graphs. Use the cycle 12222233333 12222233333 ad infinitum.

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.

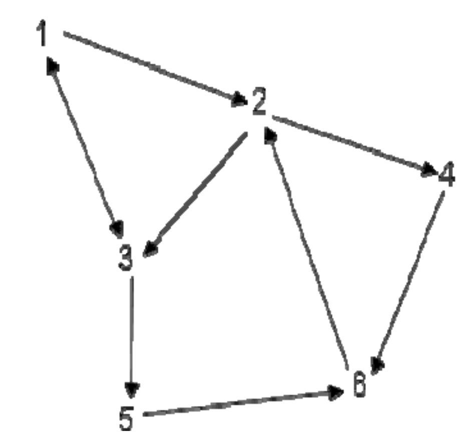
3. Second-Order Consensus

1. Consider a formation control graph where each node has the vehicle dynamics given by Newton's law

$$\dot{x}_i = v_i$$
$$\dot{v}_i = u_i$$

This corresponds to motion of all nodes along a line. Let each agent use the second-order protocol

$$u_i = c\sum_{j\in N_i} a_{ij}(x_j - x_i) + c\gamma \sum_{j\in N_i} a_{ij}(v_j - v_i)$$
$$= \sum_{j\in N_i} ca_{ij}\left((x_j - x_i) + \gamma(v_j - v_i)\right)$$

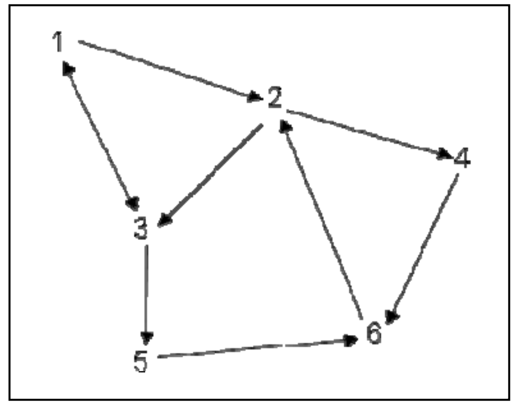


Graph for Problem 1.

Consider 6 nodes in the graph shown.

a. Find the graph Laplacian eigenvalues.
b. Simulate the motion of the nodes for gains $c\gamma^2$ that satisfy the condition Theorem 1 in the class notes. Plot $x_i(t)$ vs. t and $v_i(t)$ vs. t and show that consensus is reached in both position and velocity. Select random initial states $x_i(0)$ within *[0,1]* and $v_i(0) = 0, \forall i$.
c. Repeat part b for nonzero initial velocities $v_i(0)$ also randomly selected within *[0,1]*.

EE 5329 Homework 4
Parallel Algorithms

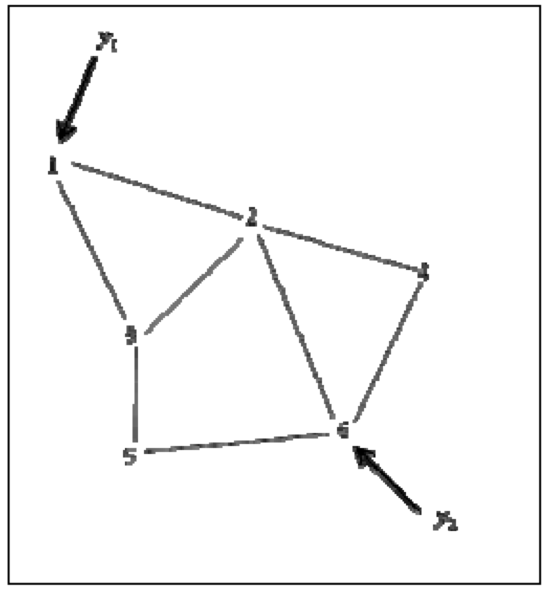


1. Parallel solution of equations

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

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

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



Note that this is the Laplacian matrix from Example 1 in Chapter 1 of the text notes online, with a 1 added to element (1,1). See Fig. 1. What does it mean?

Parallel Processing to Solve Electric networks

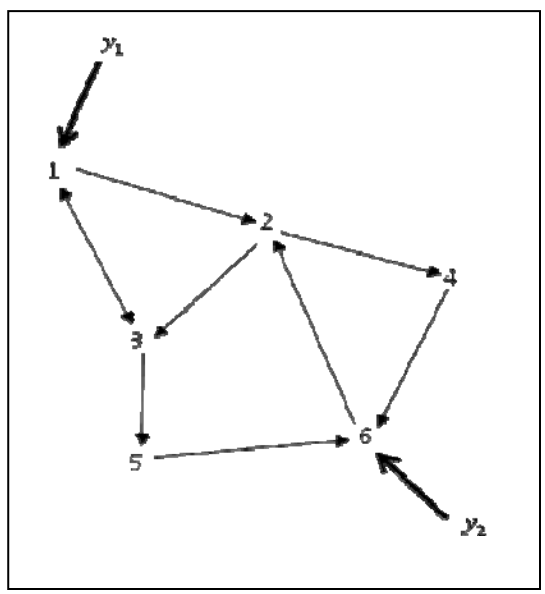
2. Undirected graph

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?

EE 5329 Takehome Exam 2
Graph Theory

Plot Basic Graph Types

Generate the following graphs. Make good plots using MATLAB. You may use Matlab Graph Theory tool box available at the following site
www.mathworks.com/matlabcentral/fileexchange/4266
However, the graphs are easily made without using that toolbox if you prefer.

1. Write MATLAB code to plot N points in the square [0,10]x[0,10] in the (x,y)-plane. Select a uniform distribution of points in each coordinate. Make plots for N= 20, 50, 100.

2. Write MATLAB code to construct an ad hoc wireless network. This is a distance graph whose edges depend on the distance between nodes. Use your code from problem 1 to place N=20 agents randomly in a 10x10 grid in the (x,y)-plane. Now, plot edges between points if they are closer than d to each other. Make plots for d= 0.1, 0.5, 1. Do we observe a phase transition? That is, is there a value of d above which the graph is connected in probability? Change N and d if you need to.

3. Write MATLAB code to generate a random graph with N=100 vertices by using Erdős–Rényi model. Take various values of m, the allowed number of edges, and probability p of having a particular edge. Observe that for some probability value p, as m increases the graphs are connected in probability. Give a few typical graphs produced.

Extra credit- A problem with random graphs is that the edges are randomly placed in the graph, unlike the distance graph where the edges are easy to see because they are between neighboring nodes. In the random graph, the edges are a jumble of short and long routes, like a plate of spaghetti. Can you make code to reposition the nodes in the (x,y)-plane to view better the connectivity properties of the graph?

4. Write MATLAB code to generate a small world graph with N vertices by using Watts and Strogatz model. Start with a k-regular graph of N nodes in a ring. Then, run through the edges, and rewire each one with probability p to another random node. Take various values of N, k, and probability p of rewiring an edge. Plot a few typical graphs produced.

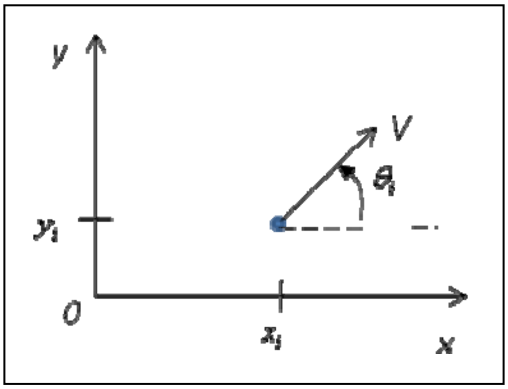
DO NOT DO- Compute the diameter of the graph for each case. Plot diameters vs. p for a given N and k.

5. Write MATLAB code to generate a random graph with N vertices by using Barabási–Albert scale-free model. The total number of nodes to be connected is N. Start with m0 nodes. Each new node that enters the graph connects to m existing nodes, with the probability of connecting to node i given by $p_i = d_i / vol(G)$ where the volume of the graph is $vol(G) = \sum_{i=1,N} d_i$. Plot the result for various values of N, m0, m. Observe the graph shapes in comparison with previous problems. Show a few typical graphs produced.

Extra credit- Make code to reposition the nodes in the plane to view better the connectivity properties of the graph.

DO NOT WORK BEYOND THIS POINT
THESE NEXT HWKS HAVE NOT YET BEEN ASSIGNED AND MAY CHANGE

Consensus for Motion Control



4. Let each node have the vehicle dynamics given by

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

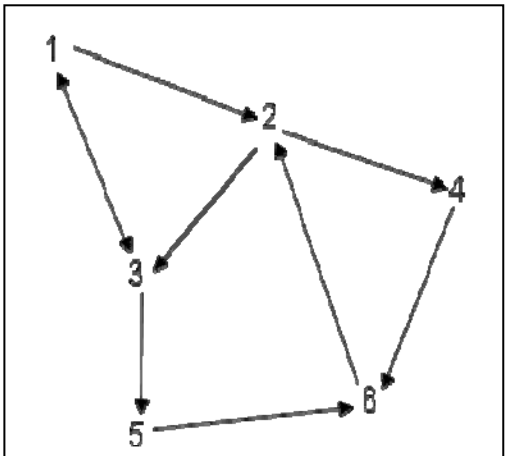
with $(x_i(t), y_i(t))$ the position and $\theta_i(t)$ the heading. This corresponds to motion in the *(x,y)* plane with velocity *V* as shown. All nodes have the same velocity *V*.

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

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

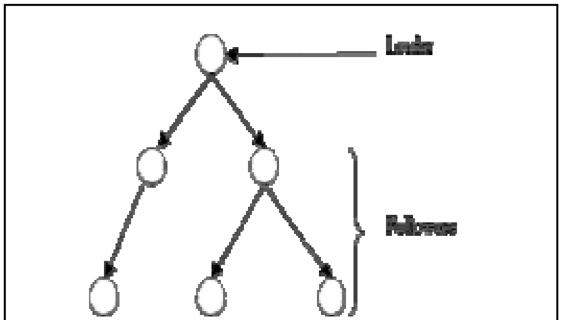
Take all the edge weights equal to one. Set the initial headings random between $0 - 2\pi$ rads.

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



Graph for Problem 1.

5. Tree formation. Repeat for the tree communication graph structure shown. Everyone should converge to the heading of the leader.



Problem 2. Tree Formation with leader.

6. Repeat problem 1 where both the velocities and headings of the nodes are different. Then, run also a local voting protocol for the velocities as

$$\dot{V}_i = \sum_{j \in N_i} a_{ij}(V_j - V_i)$$

Take all the edge weights equal to one.

Now you have 4 dynamic equations. Simulate and plot vs time the headings, velocities, and *(x,y)* positions.

EE 5329 Homework 4

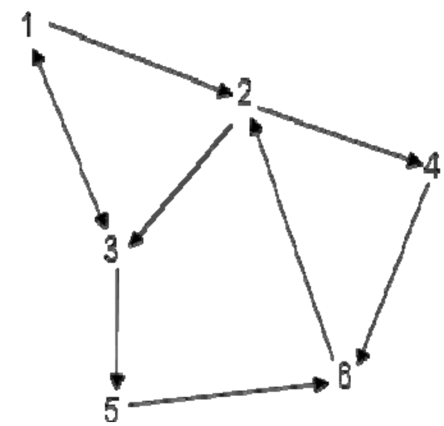
Second-Order Consensus

2. Consider a formation control graph where each node has the vehicle dynamics given by Newton's law

$$\dot{x}_i = v_i$$
$$\dot{v}_i = u_i$$

This corresponds to motion of all nodes along a line. Let each agent use the second-order protocol

$$u_i = c\sum_{j\in N_i} a_{ij}(x_j - x_i) \; + c\gamma \sum_{j\in N_i} a_{ij}(v_j - v_i)$$
$$= \sum_{j\in N_i} ca_{ij}\left((x_j - x_i) + \gamma(v_j - v_i)\right)$$



Graph for Problem 1.

Consider 6 nodes in the graph shown.

   d. Find the graph Laplacian eigenvalues.
   e. Simulate the motion of the nodes for gains $c\gamma^2$ that satisfy the condition Theorem 1 in the class notes. Plot $x_i(t)$ vs. t and $v_i(t)$ vs. t and show that consensus is reached in both position and velocity. Select random initial states $x_i(0)$ within *[0,1]* and $v_i(0) = 0, \forall i$.
   f. Repeat part b for nonzero initial velocities $v_i(0)$ also randomly selected within *[0,1]*.
   g. Repeat part b for control gains $c\gamma^2$ that do not satisfy the condition Theorem 1. Plot $x_i(t)$ vs. *t* and $v_i(t)$ vs. *t* and show that consensus is not reached.

3. **Vehicle Strings.** Important today in automated traffic platooning is the tree graph shown where all vehicles move in a line and follow a leader. Take the dynamics and protocols as in problem 1.

Graph for Problem 2.

   a. Find the graph Laplacian eigenvalues.
   b. Do problem 1 part c. Verify convergence to the position and velocity of the leader node.
   c. Repeat for control gains $c\gamma^2$ that do not satisfy the condition of Theorem 1. Plot $x_i(t)$ vs. *t* and $v_i(t)$ vs. *t* and show that consensus is not reached.
   d. Compare the graph eigenvalues to the eigenvalues in Problem 1. Would it be easier or more difficult to reach consensus in the vehicle string than in the graph of problem 1? Think in terms of the Fiedler eigenvalue as well as the condition of Theorem 1.

The gain condition of Theorem 1 is closely related to the notion of vehicle string stability from the literature.

EE 5329 Homework 5

Formation Control

1. Formation control with desired position offsets.

In hwk 4 we simulated second-order consensus for scalar Newton's law dynamics at each node. Simulate now Newton's motion dynamics for motion in the plane

$$\dot{x}_i = v_i$$
$$\dot{v}_i = u_i$$

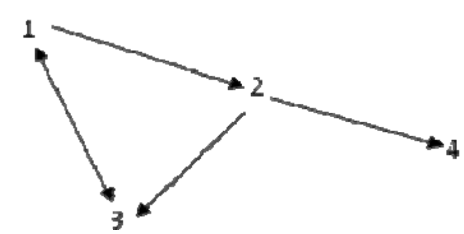
with vector position $x_i \in R^2$, velocity $v_i \in R^2$, and acceleration input $u_i \in R^2$. Take $x_i = [p_i \;\; q_i]^T$ where $(p_i(t), q_i(t))$ is the position of node *i* in the *(x,y)*-plane.

Use the formation control protocol

$$u_i = \dot{v}_0 + k_v(v_0 - v_i) + k_p(x_0 + \Delta_i - x_i) + c\sum_{j \in N_i} a_{ij}\left((x_j - \Delta_j) - (x_i - \Delta_i)\right) + c\gamma \sum_{j \in N_i} a_{ij}(v_j - v_i)$$
$$= \dot{v}_0 + k_v(v_0 - v_i) + k_p(x_0 + \Delta_i - x_i) + \sum_{j \in N_i} ca_{ij}\left((x_j - \Delta_j) - (x_i - \Delta_i) + \gamma(v_j - v_i)\right)$$

with leader node dynamics $\dot{x}_0 = v_0$ with initial position $x_0(0) = [p_0(0) \;\; q_0(0)]^T = [0 \;\; 0]^T$ and $v_0 = [1 \;\; 1]^T = const$.

Take the formation graph shown, which contains a spanning tree. Note that the leader adds a fifth node.



Graph for hwk 5

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

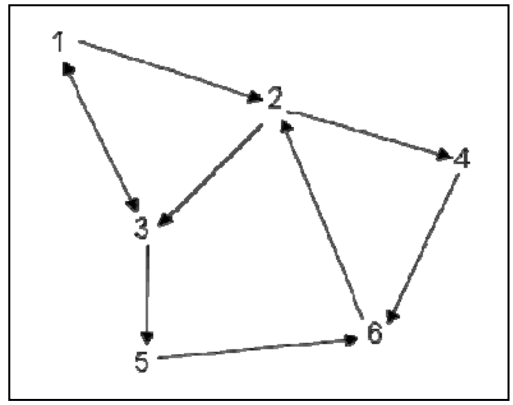
$$\Delta_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \Delta_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \Delta_3 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \Delta_4 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Simulate and plot positions in the 2-D plane vs. time. Verify that formation consensus is reached.

EE 5329 Takehome Exam 1

Switched Topologies

4. Discrete-Time Consensus with Varying Graph Topology



In hwk 2 problem 2 we simulated 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

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

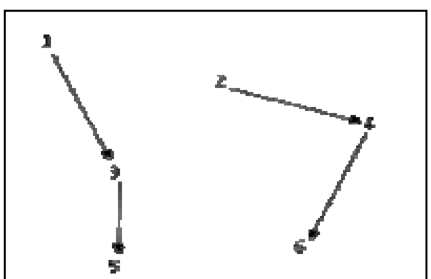
The graph used was the strongly connected digraph shown.

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

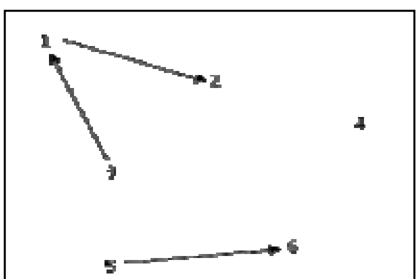
c. Simulate the DT protocol for the case of time-varying protocols

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

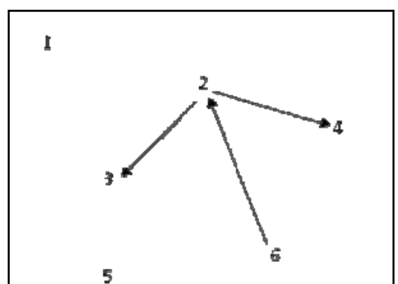
where *F(0)* corresponds to graph G1, *F(1)* to Graph G2, *F(2)* to G3, and then this cycle 123 123 is repeated. Start with random initial conditions. Plot the states vs. time. Verify that consensus is achieved. Are the consensus values the same as in Hwk 2 problem 2?



Graph G1



Graph G2



Graph G3

d. Repeat for a different order of switching between the three graphs. Use the cycle 12222233333 12222233333 ad infinitum.

Compare to part a.

5. 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.

6. Parallel solution of equations

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

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

c. Invert L to solve the equation.

d. Use the Jacobi method to solve the equation.

Note that this is the Laplacian matrix from Example 1 in Chapter 1 of the text notes online, with a 1 added to element (1,1). What does it mean?

EE 5329 Takehome Exam 2

Course Modules for a Distributed Decision and Control Class

Type up a 5 page, smooth, single-spaced set of course module notes for next year's class in microsoft word doc file form. Include explanations of basic ideas, development of concepts, equations.

You must go beyond what was given in class notes and make a final smooth form that shows you have understood the material and gone beyond what was done in class. DO NOT go through a huge literature search. If you want to look at a few other papers, that is OK, but not required. Wikipedia is a great first place to look, or the papers I gave you already.

You can interpret this as a first step towards your final report if you would like to make the topics related.

I put some doc files online at the end of the course outline that you can use and augment for your project if you like.

Your grade will depend on: (1) clarity, thoroughness, completeness. (2) how much new material you include not in my notes, or how much you improved or reorganized my notes to get a better presentation.

Choose one of the following topics for your course module:

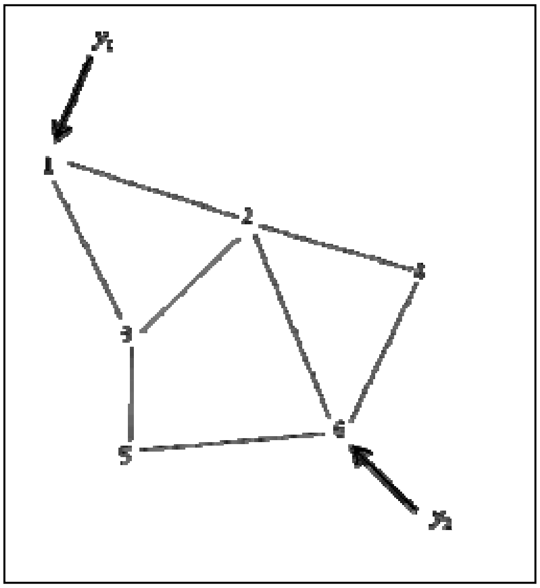
- Basic graphs: random, small world, scale free, Tadic world wide web, (others?)
  Papers are listed online in the course outline.
- Inertial systems with second order consensus
  e.g. second-order inertial systems $M_i\ddot{q}_i + N(q_i, \dot{q}_i) + d_i = \tau_i$
  There is a good paper by Lee and Spong about this.
- Graph structure- diameter, volume, girth, radius, shortest path, cuts
- Voltage networks, distributed circuits, and resistance distance (and passivity?)
- Reynolds rules and flocks, herds, swarms, schools. Couzin paper on animal leadership.
  Papers are listed online.
- Kuramoto Oscillators, Vicsek Model. Papers are listed online.

DO NOT WORK BEYOND THIS POINT
THESE NEXT HWKS HAVE NOT YET BEEN ASSIGNED AND MAY CHANGE

EE 5329 Homework 6
Spring 2011

Parallel Processing to Solve Electric networks

1. Undirected graph



Consider the electric circuit shown 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

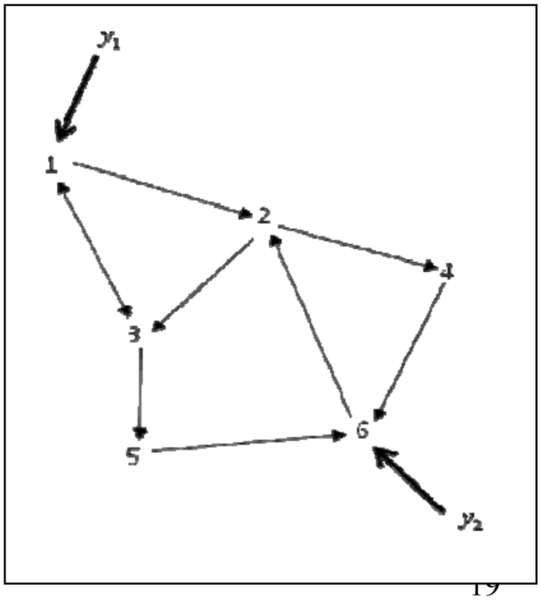
$$\dot{x}_i = \sum_{j \in N_i} a_{ij}(x_j - x_i) + b_{ik}(y_k - x_i)$$

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

2. 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 $y_1 = 10v$.

3. Directed Graph.

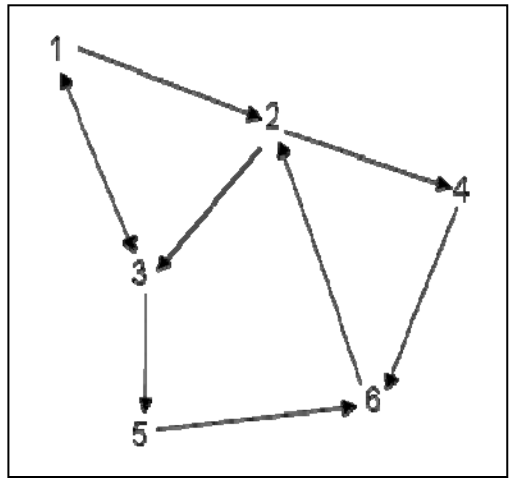


Repeat for the directed graph shown, where all edges have weights of 1. Plot the voltages vs. time. They should reach steady-state. What does it mean to have a circuit with directed edges? Does Kirchhoff's current law hold at steady-state- the sum of currents into each node equals zero?

EE 5329 Homework 4
Spring 2011

Trust Consensus for Motion Control

Here we are going to run two consensus algorithms at each node, one for trust consensus and one for heading consensus. Then, on top of that a motion dynamics at each node.



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.

The continuous-time local voting protocol on the headings of the nodes is

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

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

The value $\xi_{ij}(t)$ is the trust that node *i* has for node *j*. Define

$$\xi_i = \begin{bmatrix} \xi_{i1} \\ \xi_{i2} \\ \vdots \\ \xi_{iN} \end{bmatrix} \in R^N$$

as the trust vector of node *i*. It catalogs the trust node *i* has for each node in the network. The trust consensus algorithm is $\dot{\xi}_i = \sum_{j\in N_i} a_{ij}(\xi_j - \xi_i)$ so that $\dot{\xi} = -(L \otimes I_N)\xi$.

1. **Case of all trusts positive.** For the 6-node strongly connected graph. Initialize the trust values randomly between 0 and 1. Run the algorithm. Plot the 6 trust vectors (each of length 6) vs time. Plot headings vs. time. Plot motion vs. time in (x,y) plane.

2. Repeat for the tree graph structure. Everyone should come to the consensus given by the leader's trusts and heading.

3. **Case of negative trust.** Negative trust $\xi_{ij}(t)$ means node *i* does not trust node *j*. Then, he will try to get away from the heading of node *j*. Repeat the simulation when node 5 has negative trust values of -1 on all other nodes. All other nodes initialize randomly between 0 and 1/2. What happens? How can you get rid of the bad behavior?

EE 5329 Homework 9
Graph Theory

Graph Theory Home Work

Q1: Prove that a tree has either one center or two adjacent centers.

Q2: Let G be a graph with minimum degree $\delta > 1$ prove that it cannot be a tree.

Q3: We know about center vertex in a graph. Try to define some other critical point in a tree. How is it critical? Generalize the concept for a general graph. Give reference if it is taken from the existing literature.

Q4: Select suitably large number of vertices and some suitable value of k to make a k regular graph. Then rewire every link with some probability p use any reasonable analogy take a rational of your choice. Give plot of the graph after rewiring and calculate the diameter of the graph. You may use Matlab Graph Theory tool box available at www.mathworks.com/matlabcentral/fileexchange/4266

Q5: We have 25 agents each one is randomly allowed to make 1 to 5 edges. Choose any suitable random distribution, like uniform, normal, triangular or binomial to assign the maximum number of edges to each node. Now define an analogy for an agent to construct edges. Run the simulation for 125 time slots allowing one agent to take a construction/destruction of edge action at each step and discuss the graphs generated with respect to connectivity, diameter etc. What real life structure they look like?

Q6: Take any one graph generated in Q5 as the initial graph. Now redistribute the number of allowed edges in between 1 to 3. Define an analogy to make/break edges. Run the simulation for some suitable time allowing each agent to settle for its admissible number of edges and discuss the graphs generated.