Centro de Investigación y de Estudios
Avanzados del Instituto Politécnico Nacional
Unidad Guadalajara

Tarea 4. Seguimiento de consenso y topologías cambiantes

Presentado por

Jesús Alejandro Díaz Hernández

Presentado para el curso de

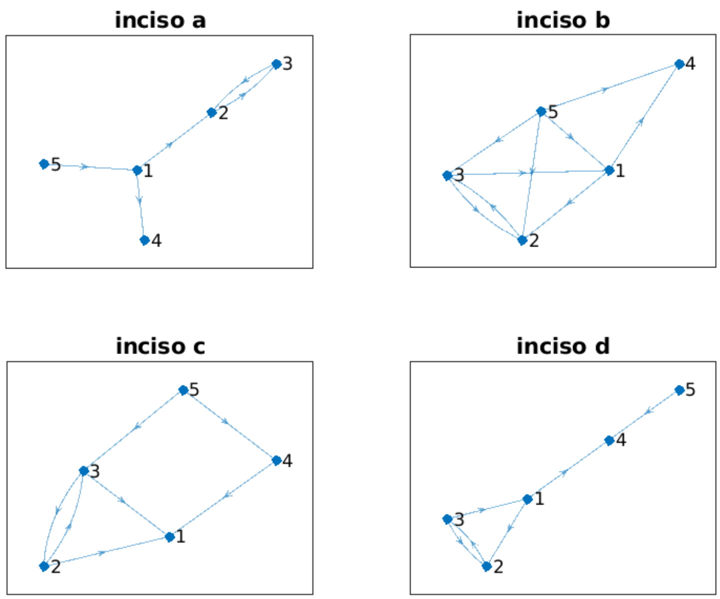
Tópicos avanzados de control 2

Curso impartido por: Héctor Manuel Becerra Fermín
Profesor

Guadalajara, Jalisco 5 de junio del 2024

Pregunta 1.-

Para referencia los grafos son los que se muestran a continuación



Inciso a)

Las matrices Laplacianas son: inciso a)

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

inciso b)

$$\begin{bmatrix} 2 & 0 & -1 & 0 & -1 \\ -1 & 3 & -1 & 0 & -1 \\ 0 & -1 & 2 & 0 & -1 \\ -1 & 0 & 0 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

inciso c)

$$\begin{bmatrix} 3 & -1 & -1 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 2 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

inciso d)

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

b)

Los eigenvalores de la matriz laplaciana negativa son: inciso a)

$$\begin{bmatrix} -0.3820 \\ -2.6180 \\ -1 \\ -1 \\ 0 \end{bmatrix}$$

inciso b)

$$\begin{bmatrix} -2 \\ -1 \\ -3 \\ -3 \\ 0 \end{bmatrix}$$

inciso c)

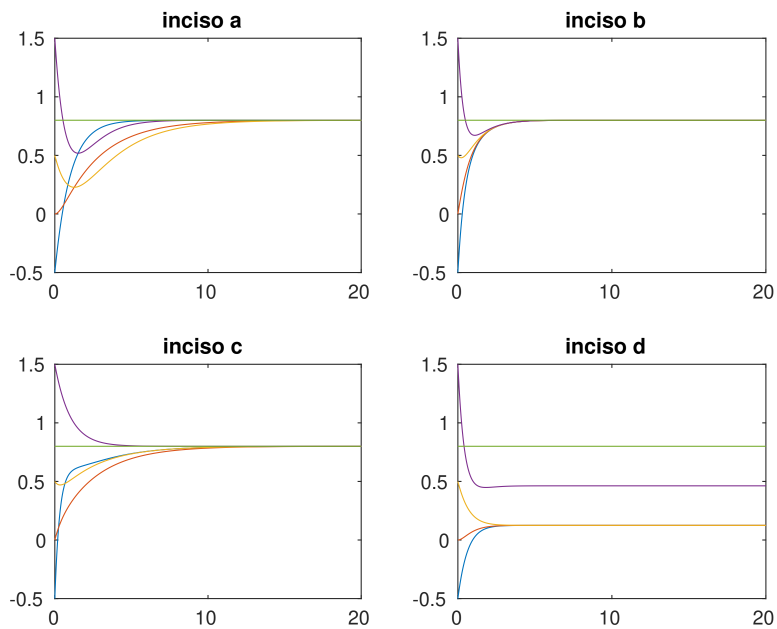
$$\begin{bmatrix} -3 \\ -0.3820 \\ -2.6180 \\ -1 \\ 0 \end{bmatrix}$$

inciso d)

$$\begin{bmatrix} -2 \\ 0 \\ -2 \\ -2 \\ 0 \end{bmatrix}$$

c)

Reproduciendo el control del algoritmo 4 podemos reproducir los resultados de la figura 2 del artículo, pero para una referencia de 0.8.

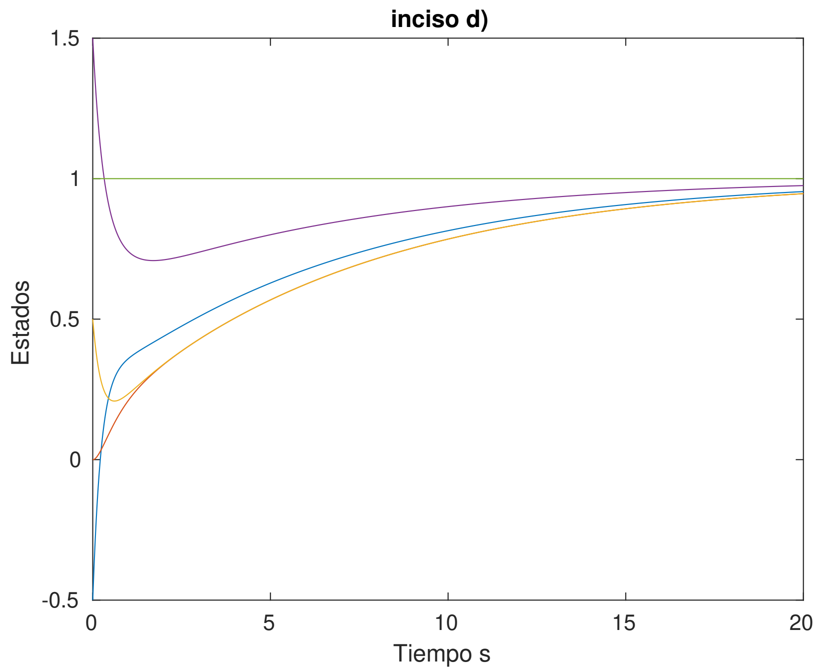


Como se puede ver a mayor conectividad algebraica, mayor rápido es la convergencia al punto de referencia. Además, la conectividad algebraica de cero en el inciso d) habla sobre la presencia del nodo aislado, por lo que no habrá consenso.

d)

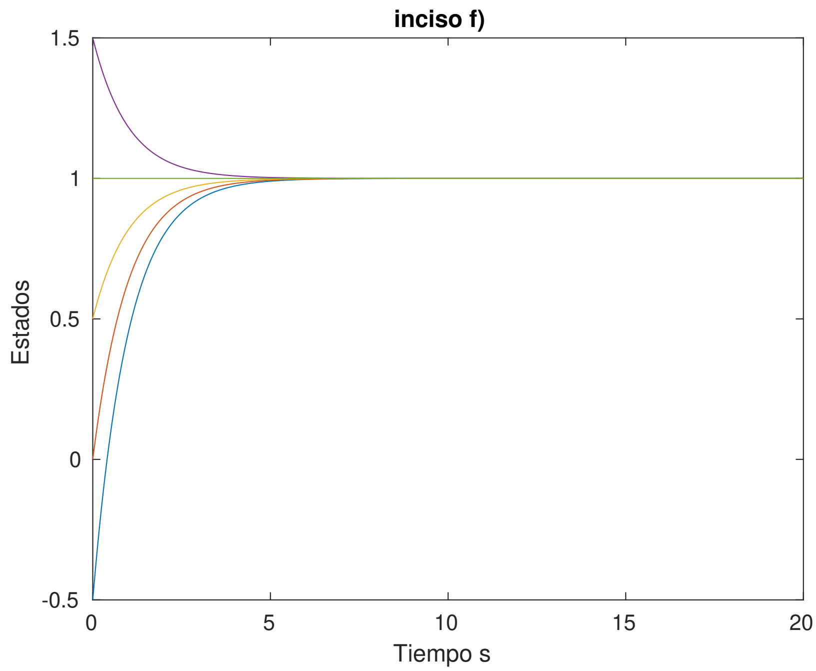
Como ahora el nodo 4 no está aislado, y existe un spanning tree, entonces se esperaría consenso. Esto es mostrado en el inciso siguiente pero para una referencia de 1.

e)



f)

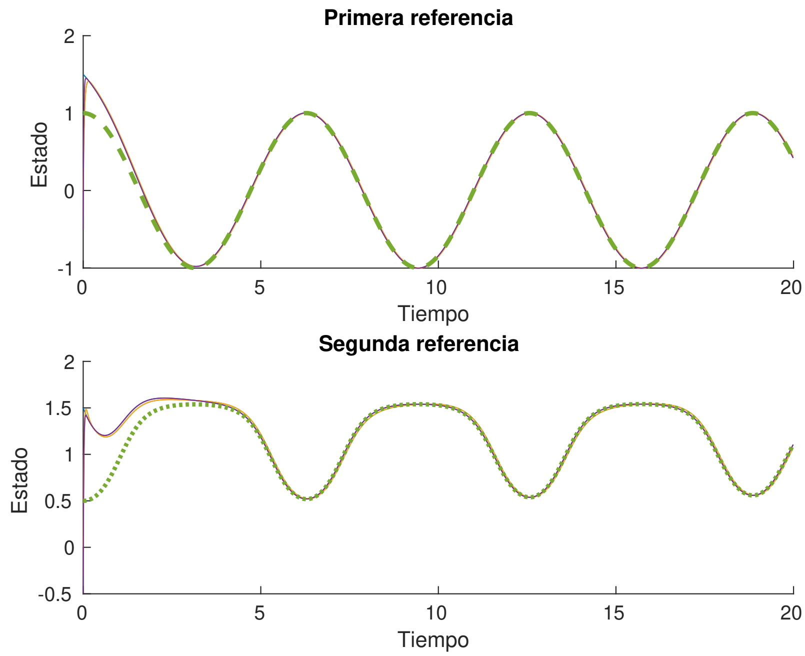
Considerando el grafo b), pero quitando la comunicación entre agentes, es decir, solo el nodo de referencia les envía información, se obtiene el siguiente resultado



Pregunta 2.-

punto 1.-

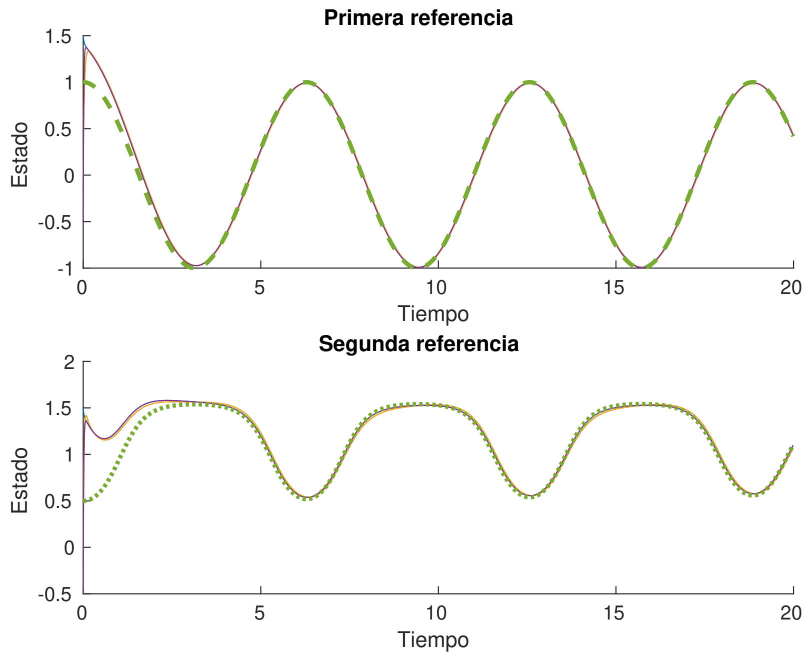
Se puede reproducir los grafos dirigidos que corresponden a la figura 6, es el mismo que el grafo de la pregunta anterior inciso a), y utilizando el algoritmo 8, se puede llegar al siguiente consenso



Sin embargo, el artículo no establece valores para las ganancias así que use matrices de unos tanto para k como para α, y matrices de 30 para las γ

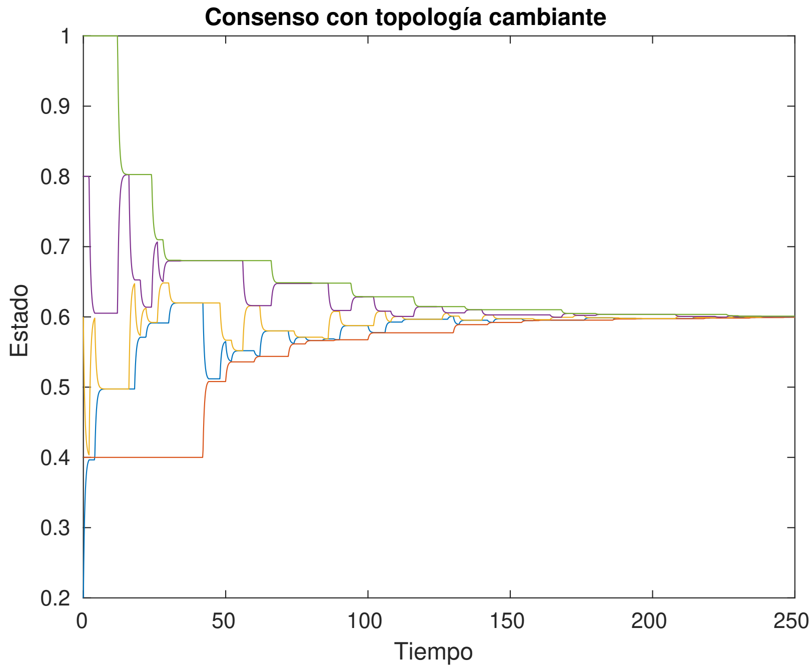
punto 2.-

Considerando el grafo como no dirigidose obtienen los siguientes resultados



Pregunta 3.-

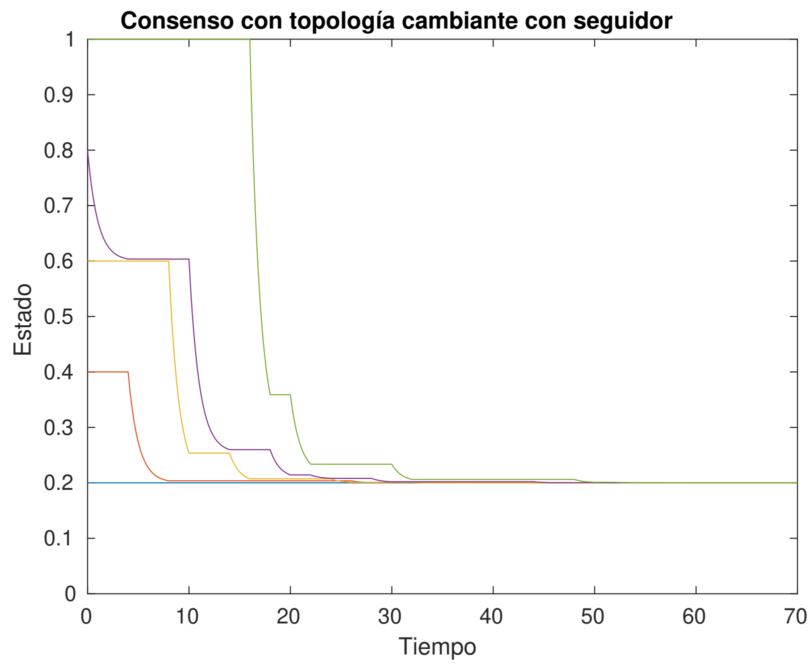
Considerando las condiciones iniciales $x_0 = \left[0.2; 0.4; 0.6; 0.8; 1\right]$ y con una ley de control con un periodo de muestreo de $0.1s$, cambiando la topología cada 2 segundos, además, cambiando los grafos de la diapositiva 9 por grafos no dirigidos, se obtienen los siguientes resultados



Los cuales convergen al promedio de las condiciones iniciales.

Pregunta 4.-

Para integradores simples con las mismas condiciones iniciales que la pregunta anterior, pero considerando el caso líder seguidor, es decir, $\mathcal{G}_5^l \triangleq \mathcal{G} \backslash \mathcal{G}_{5(1)}$, y ahora considerando los grafos dirigidos como en las diapositivas, se obtiene el siguiente resultado



Pregunta 5.-

En cuanto a los algoritmos de consenso con referencia, la implementación más sencilla se da en el caso de seguimiento de referencias constantes. Sin embargo, cuando se trata de referencias variantes en el tiempo, la correcta elección de las ganancias es crucial. El consenso se logra en ambos escenarios (referencias constantes y variantes en el tiempo), pero la selección adecuada del algoritmo es fundamental para asegurar el seguimiento eficaz de la referencia deseada.

Respecto al consenso con topologías cambiantes, se observó que el consenso puede lograrse tanto con un esquema de líder-seguidor como sin él. La definición del intervalo de tiempo para cada cambio de topología es un factor clave que influye significativamente en la velocidad de convergencia del sistema. Una correcta elección del tiempo de cambio entre topologías puede mejorar notablemente la rapidez con la que se alcanza el consenso.

Anexo (código usado)

Pregunta 1

```
clc
clearvars
close all
%Matriz de grado
```

```
%Primer punto
D1=[1 0 0 0 0;
    0 2 0 0 0;
    0 0 1 0 0;
    0 0 0 1 0;
    0 0 0 0 0];
D2=[2 0 0 0 0;
    0 3 0 0 0;
    0 0 2 0 0;
    0 0 0 2 0;
    0 0 0 0 0];
D3=[3 0 0 0 0;
    0 1 0 0 0;
    0 0 2 0 0;
    0 0 0 1 0;
    0 0 0 0 0];
D4=[1 0 0 0 0;
    0 2 0 0 0;
    0 0 1 0 0;
    0 0 0 2 0;
    0 0 0 0 0];



%Matriz de adyacencia
%Primer punto
A1=[0 0 0 0 1;
    1 0 1 0 0;
    0 1 0 0 0;
    1 0 0 0 0;
    0 0 0 0 0];
A2=[0 0 1 0 1;
    1 0 1 0 1;
    0 1 0 0 1;
    1 0 0 0 1;
    0 0 0 0 0];
A3=[0 1 1 1 0;
    0 0 1 0 0;
    0 1 0 0 1;
    0 0 0 0 1;
    0 0 0 0 0];
A4=[0 0 1 0 0;
    1 0 1 0 0;
    0 1 0 0 0;
    1 0 0 0 1;
    0 0 0 0 0];



%hago el objeto grafo, pero aun no lo ploteo
G1=digraph(A1');
G2=digraph(A2');
G3=digraph(A3');
G4=digraph(A4');

%Nodos
n=size(A1,1);
```

```
%
%Laplaciano
L1=D1-A1
L2=D2-A2
L3=D3-A3
L4=D4-A4
%

%Caso especial inciso d)
L=[3 -1 -1 -1 0;
   -1 2 -1 0 0;
   -1 -1 2 0 0;
   -1 0 0 2 -1;
   0 0 0 0 0];
%Caso especial inciso f)
Lf=[1 0 0 0 -1;
    0 1 0 0 -1;
    0 0 1 0 -1;
    0 0 0 1 -1;
    0 0 0 0 0];

%Calculo de los eigenvalores
e1=eig(-L1);
e2=eig(-L2);
e3=eig(-L3);
e4=eig(-L4);
svd(-L1)
svd(-L2)
svd(-L3)
svd(-L4)
%
%Hago el producto Kronecker
m=1; %dimensiones que estoy considerando
Im=eye(m);
Kron=kron(L,Im);   %Este es para el inciso d)
Kronf=kron(Lf,Im); %Este es para el inciso f)
Kron1= kron(L1,Im);
Kron2= kron(L2,Im);
Kron3= kron(L3,Im);
Kron4= kron(L4,Im);
%
% %% Esta parte es para la simulacion del sistema

% %Valores iniciales
ref=1; %La referencia que quiero que sigan
xi0=[-0.5;0;.5;1.5;ref];%Guardo los valores iniciales para
    referencia
xi=xi0;xif=xi0;xi1=xi0;xi2=xi0;xi3=xi0;xi4=xi0;xi5=xi0;%Uso estos
    valores en la aproximacion de Euler
%
%periodo de muestreo
Dt=0.01;
tiempo=20; %segundos
iteraciones=tiempo/Dt;

%Simulo el sistema
for k=1:iteraciones
```

```
    %Aproximacion de Euler con Gamma_kron
    xi(:,k+1)=xi(:,k)-Dt*(Kron*xi(:,k));      %Inciso d)
    xif(:,k+1)=xif(:,k)-Dt*(Kronf*xif(:,k)); %Inciso f)
    xi1(:,k+1)=xi1(:,k)-Dt*(Kron1*xi1(:,k));
    xi2(:,k+1)=xi2(:,k)-Dt*(Kron2*xi2(:,k));
    xi3(:,k+1)=xi3(:,k)-Dt*(Kron3*xi3(:,k));
    xi4(:,k+1)=xi4(:,k)-Dt*(Kron4*xi4(:,k));
end


%Vectores de tiempo para graficar
t=linspace(0,tiempo,iteraciones+1);

figure
subplot(2,2,1)
plot(t,xi1);
title('inciso a')
subplot(2,2,2)
plot(t,xi2);
title('inciso b')
subplot(2,2,3)
plot(t,xi3);
title('inciso c')
subplot(2,2,4)
plot(t,xi4);
title('inciso d')

figure
plot(t,xi);
title('inciso d)')
xlabel('Tiempo s')
ylabel('Estados')

figure
plot(t,xif);
title('inciso f)')
xlabel('Tiempo s')
ylabel('Estados')

figure
subplot(2,2,1)
plot(G1);
title('inciso a')
subplot(2,2,2)
plot(G2);
title('inciso b')
subplot(2,2,3)
plot(G3);
title('inciso c')
subplot(2,2,4)
plot(G4);
title('inciso d')
```

Pregunta 2

```
clc
```

```
clearvars
close all

% Parametros
num_agents = 4; % Numero de agentes
T = 20; % Tiempo de simulacion
Dt = 0.01; % Paso de tiempo
iteraciones = T / Dt;

 % Matriz de adyacencia
A1 = [0 0 0 0;
      1 0 1 0;
      0 1 0 0;
      1 0 0 0;];
% Matriz de adyacencia segundo caso
% A1=[0 1 0 1;
%     1 0 1 0;
%     0 1 0 0;
%     1 0 0 0];

K = ones(num_agents); % Matriz de coeficientes k_{ij}
alpha = ones(num_agents, 1); % Vector de coeficientes alpha_i
gamma = 50*ones(num_agents, 1); % Vector de coeficientes gamma_i

% Inicializacion
xi = zeros(num_agents, iteraciones); % Estados de los agentes
xi_dot = zeros(num_agents, iteraciones); % Derivadas de los estados

% Inicializacion de los estados de los agentes
xi(:, 1) = [1.5;.5;0;-0.5]; % Estados iniciales estados

% Primera referencia
xi_r1 = cos(0:Dt:(iteraciones-1)*Dt); % Primera referencia variante
     con el tiempo

% Dinamica de la primera referencia
f1 =  @(t, xi_r) -sin(t); % Derivada de la referencia

% Simulacion con la primera referencia
xi1 = xi; % Copia de los estados iniciales
for k = 1:iteraciones-1
    t = (k-1) * Dt;

    % Derivada de la referencia en el tiempo actual
    xi_r1_dot = f1(t, xi_r1(k));

    for i = 1:num_agents
        if i == 1 % Vehiculo con acceso a la referencia
            u_i = xi_r1_dot - sum(A1(i, :) .* K(i, :) .* (xi1(i, k)
                 - xi1(:, k)')) - alpha(i) * (xi1(i, k) - xi_r1(k))
                ;
        else % Otros vehiculos
            sum_gk = sum(A1(i, :) .* K(i, :));
            if sum_gk == 0
                u_i = 0;
            else
                u_i = (1 / sum_gk) * sum(A1(i, :) .* K(i, :) .* (
```

```
                    xi_dot(:, k)' - gamma(i) * (xi1(i, k) - xi1(:, k)')));
            end
        end
        xi_dot(i, k) = u_i;
    end

    % Actualizacion de los estados usando la aproximacion de Euler
    xi1(:, k+1) = xi1(:, k) + Dt * xi_dot(:, k);
end

% Segunda referencia
xi_r2 = zeros(1, iteraciones);
xi_r2(1) = 0.5; % Estado inicial de la segunda referencia

% Dinamica de la segunda referencia
f2 = @(t, xi_r) sin(t) * sin(2 * xi_r); % Derivada de la segunda referencia

% Simulacion con la segunda referencia
xi2 = xi; % Copia de los estados iniciales
for k = 1:iteraciones-1
    t = (k-1) * Dt;

    % Derivada de la referencia en el tiempo actual
    xi_r2_dot = f2(t, xi_r2(k));

    % Actualizacion de la segunda referencia
    xi_r2(k+1) = xi_r2(k) + Dt * xi_r2_dot;

    for i = 1:num_agents
        if i == 1 % Vehiculo con acceso a la referencia
            u_i = xi_r2_dot - sum(A1(i, :) .* K(i, :) .* (xi2(i, k) - xi2(:, k)')) - alpha(i) * (xi2(i, k) - xi_r2(k));
        else % Otros vehiculos
            sum_gk = sum(A1(i, :) .* K(i, :));
            if sum_gk == 0
                u_i = 0;
            else
                u_i = (1 / sum_gk) * sum(A1(i, :) .* K(i, :) .* (xi_dot(:, k)' - gamma(i) * (xi2(i, k) - xi2(:, k)')));
            end
        end
        xi_dot(i, k) = u_i;
    end

    % Actualizacion de los estados usando la aproximacion de Euler
    xi2(:, k+1) = xi2(:, k) + Dt * xi_dot(:, k);
end

% Ploteo de resultados
time = 0:Dt:(iteraciones-1)*Dt;

figure;

```

```
% Subplot para la primera referencia
subplot(2, 1, 1);
hold on;
for i = 1:num_agents
    plot(time, xi1(i, :));
end
plot(time, xi_r1, '--', 'LineWidth', 2); % Primera referencia
xlabel('Tiempo');
ylabel('Estado');
title('Primera referencia');
hold off;

% Subplot para la segunda referencia
subplot(2, 1, 2);
hold on;
for i = 1:num_agents
    plot(time, xi2(i, :));
end
plot(time, xi_r2, ':', 'LineWidth', 2); % Segunda referencia
xlabel('Tiempo');
ylabel('Estado');
title('Segunda referencia');
hold off;
```

Pregunta 3

```
clc
clearvars
close all

%Laplacianos
L1=[1 0 -1 0 0;
    0 0 0 0 0;
    -1 0 1 0 0;
    0 0 0 0 0;
    0 0 0 0 0];

L2=[1 -1 0 0 0;
    -1 1 0 0 0;
    0 0 0 0 0;
    0 0 0 0 0;
    0 0 0 0 0];

L3=[1 0 -1 0 0;
    0  0 0 0 0;
    -1 0 1 0 0;
    0  0 0 0 0;
    0  0 0 0 0];

L4=[0 0 0 0 0;
    0 0 0 0 0;
    0 0 1 -1 0;
    0 0 -1 1 0;
    0 0 0 0 0];

L5=[0 0 0 0 0;
```

```
    0 0 0 0 0;
    0 0 0 0 0;
    0 0 0 1 -1;
    0 0 0 -1 1];

%
%Hago el producto Kronecker
m=1; %dimensiones que estoy considerando
Im=eye(m);
Kron1= kron(L1,Im);
Kron2= kron(L2,Im);
Kron3= kron(L3,Im);
Kron4= kron(L4,Im);
Kron5= kron(L5,Im);
%
% %% Esta parte es para la simulacion del sistema

% %Valores iniciales
xi0=[0.2;0.4;0.6;0.8;1];%Guardo los valores iniciales para
    referencia
xi=xi0;%Uso estos valores en la aproximacion de Euler
%
%periodo de muestreo
Dt=0.01;
tiempo=100; %segundos
iteraciones=tiempo/Dt;

%Simulo el sistema
for k=1:iteraciones
     % Seleccion aleatoria de la topologia cada dos segundos
    if mod(k, 200) == 1
        rand_index = randi([1, 5]);
    end

    % Seleccion del producto de Kronecker correspondiente
    switch rand_index
        case 1
            Kron = Kron1;
        case 2
            Kron = Kron2;
        case 3
            Kron = Kron3;
        case 4
            Kron = Kron4;
        case 5
            Kron = Kron5;
    end
    %Aproximacion de Euler con Gamma_kron
    xi(:,k+1)=xi(:,k)-Dt*(Kron*xi(:,k));
end


%Vectores de tiempo para graficar
t=linspace(0,tiempo,iteraciones+1);


figure
```

```
plot(t,xi);
xlabel('Tiempo');
ylabel('Estado');
title('Consenso con topologia cambiante')
```

Pregunta 4

```
    clc
clearvars
close all

%Laplacianos

L2=[0 0 0 0 0;
    -1 1 0 0 0;
    0 0 0 0 0;
    0 0 0 0 0;
    0 0 0 0 0];

L3=[0 0 0 0 0;
    0 0 0 0 0;
    -1 0 1 0 0;
    0 0 0 0 0;
    0 0 0 0 0];

L4=[0 0 0 0 0;
    0 0 0 0 0;
    0 0 0 0 0;
    0 0 -1 1 0;
    0 0 0 0 0];

L5=[0 0 0 0 0;
    0 0 0 0 0;
    0 0 0 0 0;
    0 0 0 0 0;
    0 0 0 -1 1];

%
%Hago el producto Kronecker
m=1; %dimensiones que estoy considerando
Im=eye(m);
Kron2= kron(L2,Im);
Kron3= kron(L3,Im);
Kron4= kron(L4,Im);
Kron5= kron(L5,Im);
%
% %% Esta parte es para la simulacion del sistema

% %Valores iniciales
xi0=[0.2;0.4;0.6;0.8;1];%Guardo los valores iniciales para
    referencia
xi=xi0;%Uso estos valores en la aproximacion de Euler
%
%periodo de muestreo
Dt=0.01;
tiempo=70; %segundos
```

```
iteraciones=tiempo/Dt;

%Simulo el sistema
for k=1:iteraciones
     % Seleccion aleatoria de la topologia cada dos segundos
    if mod(k, 200) == 1
        rand_index = randi([1, 4]);
    end

    % Seleccion del producto de Kronecker correspondiente
    switch rand_index
        case 1
            Kron = Kron2;
        case 2
            Kron = Kron3;
        case 3
            Kron = Kron4;
        case 4
            Kron = Kron5;
    end
    %Aproximacion de Euler con Gamma_kron
    xi(:,k+1)=xi(:,k)-Dt*(Kron*xi(:,k));
end


%Vectores de tiempo para graficar
t=linspace(0,tiempo,iteraciones+1);


figure
plot(t,xi);
xlabel('Tiempo');
ylabel('Estado');
title('Consenso con topologia cambiante con seguidor')
ylim([0 1])
```