SWARM CONTROL IN THE PRESENCE OF FAULTY NODES

Trust Consensus Method

Abstract

Controlling robots in a swarm motion to achieve set tasks is a highly interesting field with a lot of research work being done. These can be of exceptional use during rescue missions where time is a crucial factor and human intervention is not possible. During strenuous working conditions, it is possible for a robot to malfunction. This project implements two basic approaches to control robot swarms and then isolate a faulty robot using the method of trust consensus.

Atul Shrotriya

atul.shrotriya@uta.edu

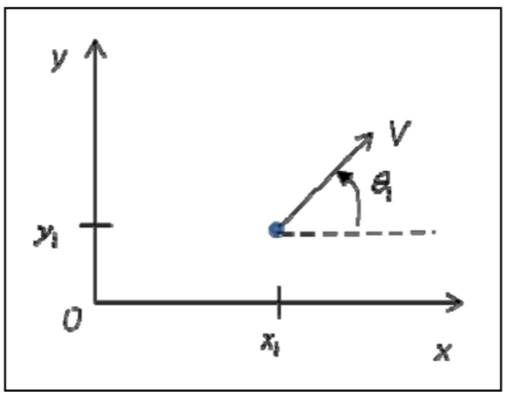
Network and Dynamics

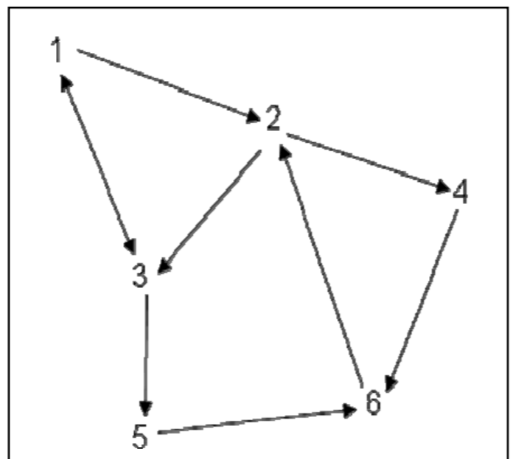
Every node has the following dynamics

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

Where $[x_i(t), y_i(t)]$ represents 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= 1 m/sec





Graph for formation control.

6 nodes are taken in a strongly connected communication graph structure as shown above.

The continuous time local voting protocol given below is used to determine the heading for each node.

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

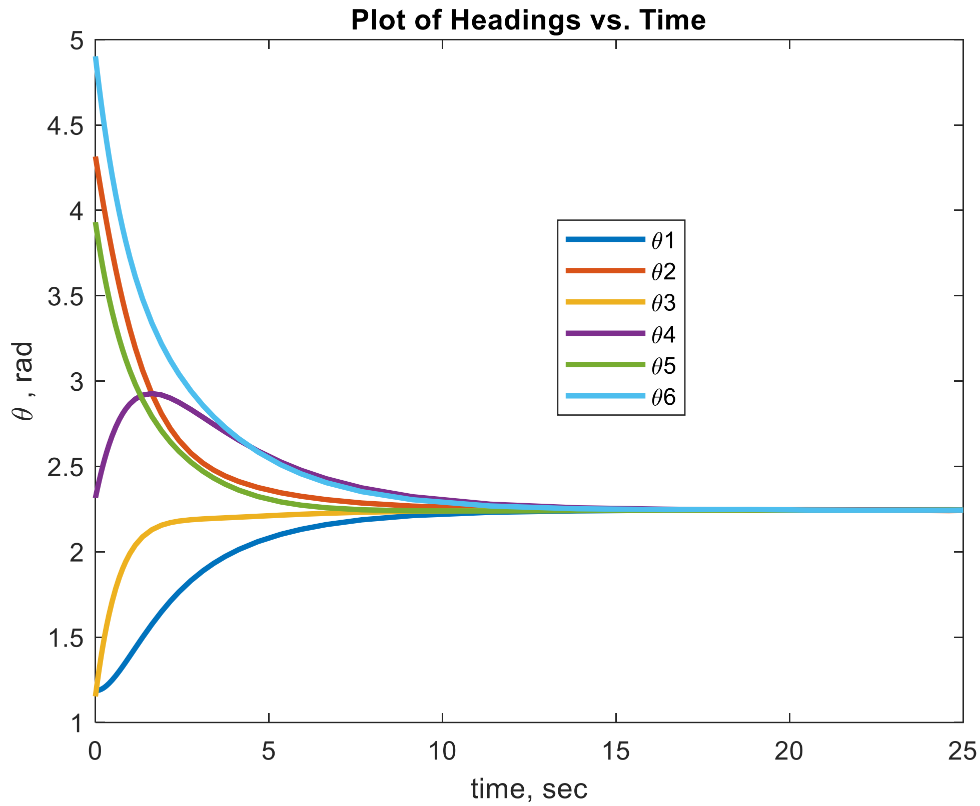
Initial headings are set randomly and all edge weights are taken as 0.5 which leads to a convergence at the average of all initial headings.

Method - 1

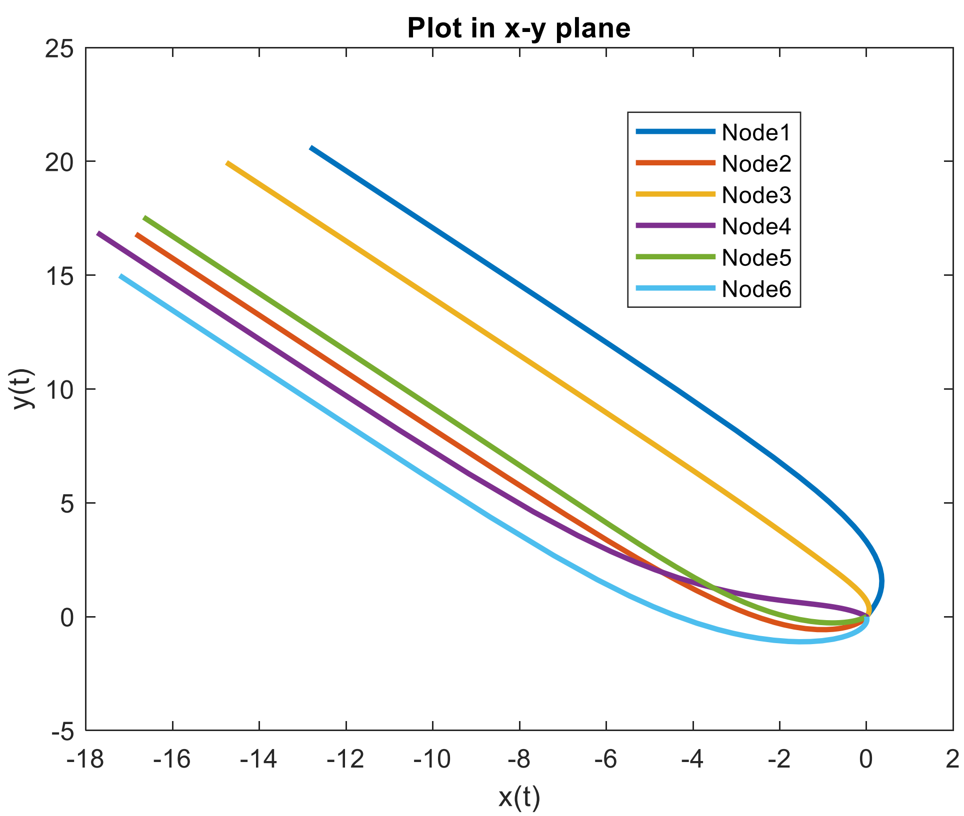
```
function method1
a=0.5; V=1; init=rand(1,6).*(2*pi);
tin=[0 25]; %time interval
A=[0 0 0.5 0 0 0;0.5 0 0 0 0 0.5;0.5 0.5 0 0 0 0;0 0.5 0 0 0 0;0 0 0.5 0 0 0;0 0 0 0.5 0.5 0];
D=[0.5 0 0 0 0 0;0 1 0 0 0 0;0 0 1 0 0 0;0 0 0 0.5 0 0;0 0 0 0 0.5 0;0 0 0 0 0 1];
L=D-A;
[t,x]=ode23(@equations,tin,[init zeros(1,12)]);
function dx=equations(t,x)
 dx=zeros(18,1);
 dx(1:6)=-L*x(1:6);
 dx(7:12)=V.*cos(x(1:6));
 dx(13:18)=V.*sin(x(1:6));
end
theta=x(:,1:6);
figure()
plot(t,theta, 'LineWidth',2)
figure()
plot(x(:,7:12),x(:,13:18),'LineWidth',2)
end
```

Method - 2

```
function method2
a=0.5; V=1; init=rand(1,6).*(2*pi);
tin=[0 25]; %time interval
[t,x]=ode23(@equations,tin,[init zeros(1,12)]);
function dx=equations(t,x)
 dx=zeros(18,1);
 dx(1)=a*(x(3)-x(1));
 dx(2)=a*(x(1)-x(2))+a*(x(6)-x(2));
 dx(3)=a*(x(1)-x(3))+a*(x(2)-x(3));
 dx(4)=a*(x(2)-x(4));
 dx(5)=a*(x(3)-x(5));
 dx(6)=a*(x(4)-x(6))+a*(x(5)-x(6));
 dx(7:12)=V.*cos(x(1:6));
 dx(13:18)=V.*sin(x(1:6));
end
theta=x(:,1:6);
figure(1)
plot(t,theta,'LineWidth',2)
figure(2)
plot(x(:,7:12),x(:,13:18),'LineWidth',2)
end
```



Plot of Headings vs. Time
θ , rad
time, sec
θ1
θ2
θ3
θ4
θ5
θ6
5
4.5
4
3.5
3
2.5
2
1.5
1
0
5
10
15
20
25



Plot in x-y plane
y(t)
x(t)
Node1
Node2
Node3
Node4
Node5
Node6
25
20
15
10
5
0
-5
-18
-16
-14
-12
-10
-8
-6
-4
-2
0
2

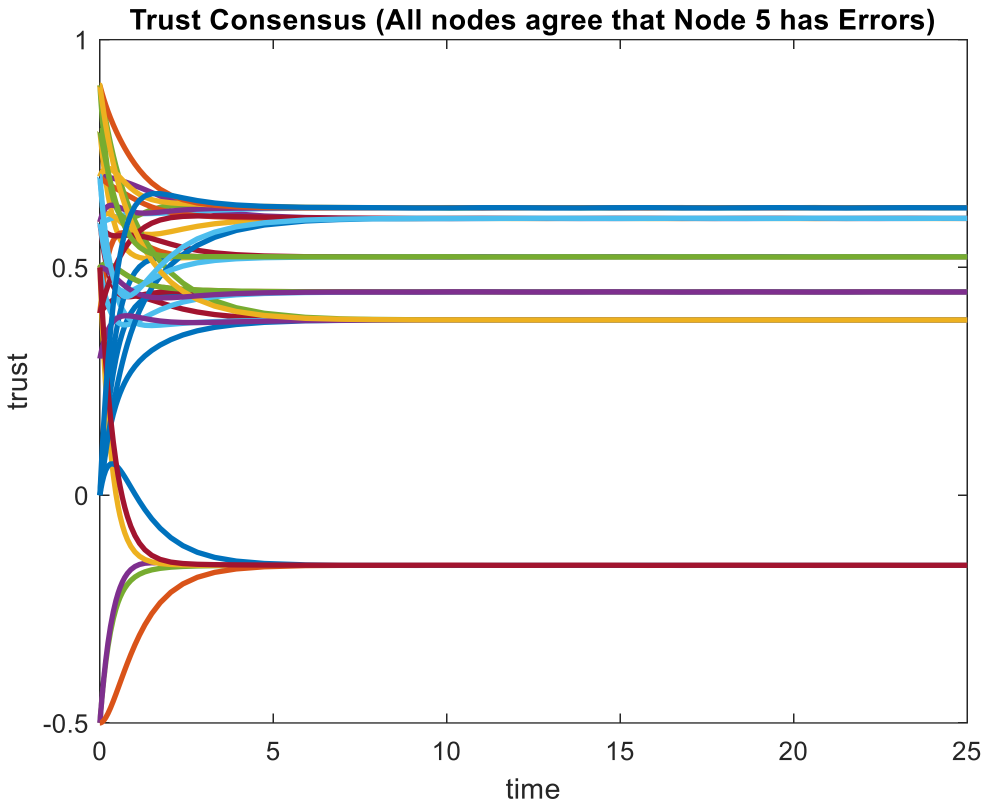
If Node 5 is malicious

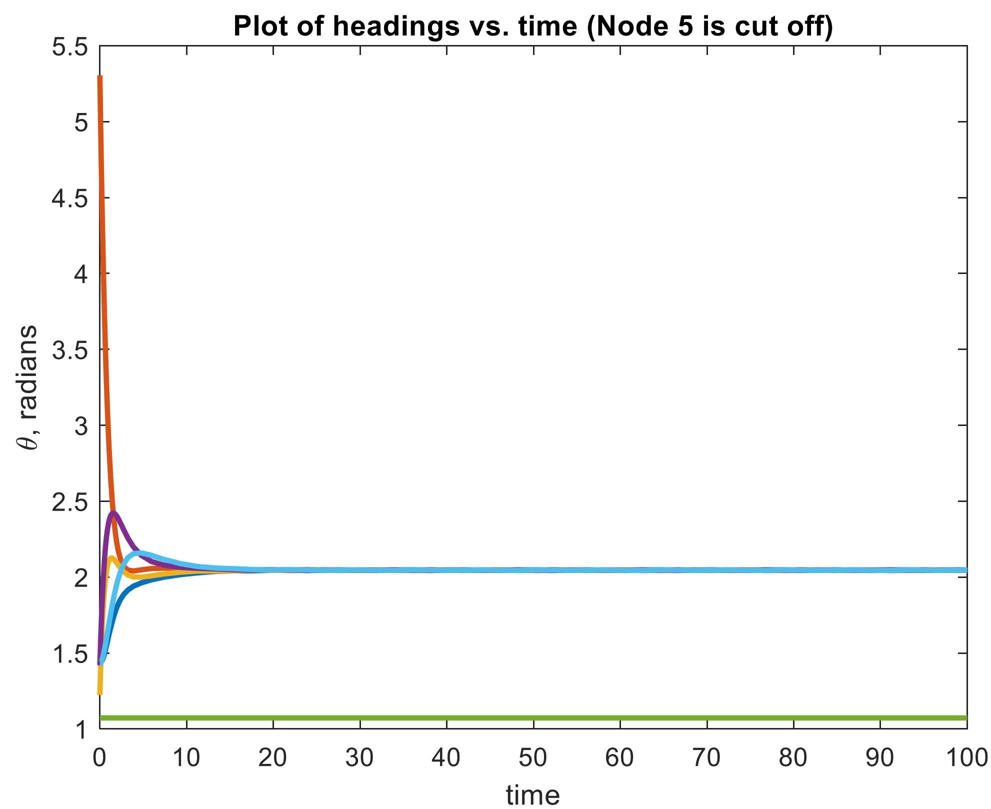
Nodes 1, 2 and 4 have negative trust for Node 5 initially. The below given MATLAB code can be modified in case Node 5 has more than one outgoing connections.

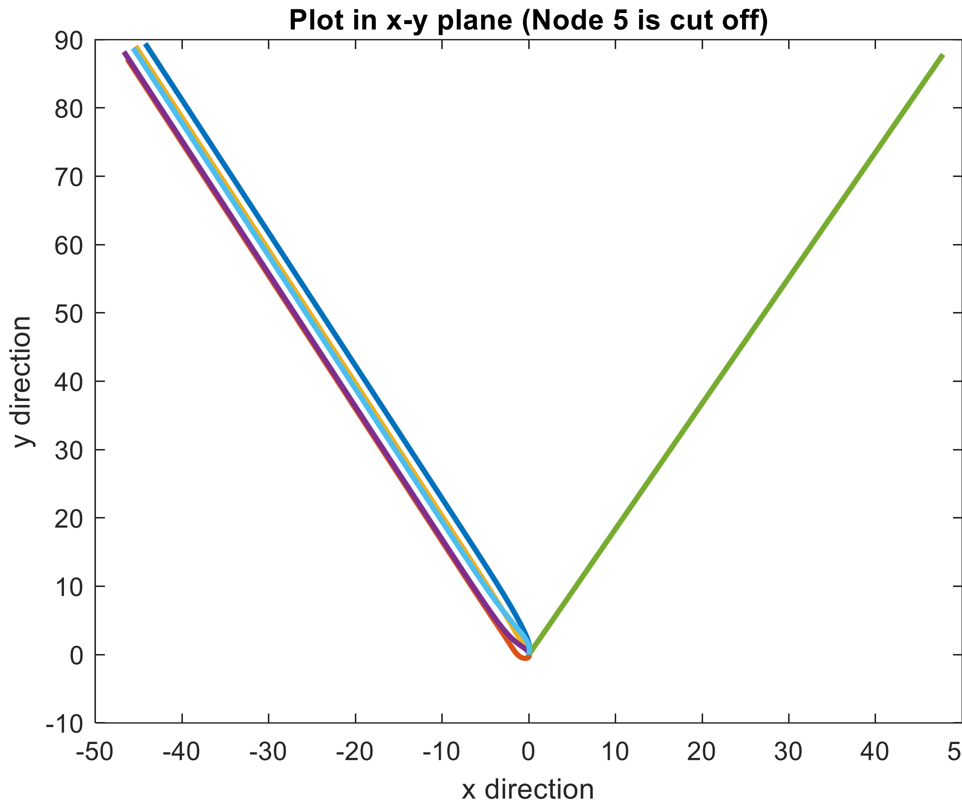
```
function trustconsensus
itrust(:,1)=[0;0.5;0.8;0.7;-0.5;0.6];
itrust(:,2)=[0.4;0;0.4;0.6;-0.5;0.9];
itrust(:,3)=[0.7;0.6;0;0.7;0.5;0.6];
itrust(:,4)=[0.9;0.5;0.6;0;-0.5;0.7];
itrust(:,5)=[0.3;0.5;0.6;0.4;0;0.9];
itrust(:,6)=[0.9;0.5;0.8;0.7;0.5;0];
init=[itrust(:,1);itrust(:,2);itrust(:,3);itrust(:,4);itrust(:,5
);itrust(:,6)];
tin=[0 25];
[t,x]=ode23(@equations,tin,init);
function dx=equations(t,x)
 dx=zeros(36,1);
 dx(1:6)=x(13:18)-x(1:6);
 dx(7:12)=x(1:6)+x(31:36)-x(7:12)-x(7:12);
 dx(13:18)=x(1:6)+x(7:12)-x(13:18)-x(13:18);
 dx(19:24)=x(7:12)-x(19:24);
 dx(25:30)=x(13:18)-x(25:30);
 dx(31:36)=x(19:24)+x(25:30)-x(31:36)-x(31:36);
end
figure()
plot(t,x)
Z1=x(:,1:6);
Z2=x(:,7:12);
Z3=x(:,13:18);
Z4=x(:,19:24);
Z5=x(:,25:30);
Z6=x(:,31:36);
finval=length(Z1(:,1));
trustcon=[Z1(finval,:);Z2(finval,:);Z3(finval,:);Z4(finval,:);Z5
(finval,:);Z6(finval,:)];
a=0.5; V=1; init=rand(1,6).*(2*pi);
tint=[0 100]; %time interval
A=[0 0 0.5 0 0 0;0.5 0 0 0 0 0.5;0.5 0.5 0 0 0 0;0 0.5 0 0 0 0;0
0 0.5 0 0 0;0 0 0 0.5 0.5 0];
D=[0.5 0 0 0 0 0;0 1 0 0 0 0;0 0 1 0 0 0;0 0 0 0.5 0 0;0 0 0 0
0.5 0;0 0 0 0 0 1];
for i=1:1:6
    for j=1:1:6
        if trustcon(i,j)<0
            ErrN(i)=j;
```

```
        end
    end
end
if range(ErrN)==0
    ErrorNode=ErrN(1);
end
for i=1:1:6
    if A(i,ErrorNode)>0
        DC=i;
    end
end
D(DC,DC)=D(DC,DC)-0.5;
A(:,ErrorNode)=[0 0 0 0 0 0];
A(ErrorNode,:)=[0 0 0 0 0 0];
D(ErrorNode,ErrorNode)=[0];
% D(:,ErrorNode)=[];
L=D-A;
[T,th]=ode23(@equation,tint,[init zeros(1,12)]);
function dth=equation(T,th)
    dth=zeros(18,1);
    dth(1:6)=-L*th(1:6);
    dth(7:12)=cos(th(1:6));
    dth(13:18)=sin(th(1:6));
end
figure()
plot(T,th(:,1:6))
figure()
plot(th(:,7:12),th(:,13:18))
end
```

Plots Shown on Next Page


Trust Consensus (All nodes agree that Node 5 has Errors)
trust
time
1
0.5
0
-0.5
0
5
10
15
20
25


Plot of headings vs. time (Node 5 is cut off)
θ, radians
time
5.5
5
4.5
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70
80
90
100



References

1. Dr. Frank L. Lewis, Professor at Electrical Engineering Department, The University of Texas at Arlington. "Distributed Decision and Control". Spring 2021 semester.
2. S. Zheng, T. Jiang and J. S. Baras, "Robust State Estimation under False Data Injection in Distributed Sensor Networks," *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Miami, FL, 2010, pp. 1-5, doi: 10.1109/GLOCOM.2010.5685223.