

UAS sistem pengaturan Formasi dan kolaborasi

Muhammad Azriel Rizqifadiilah - 6022221047



Departemen teknik elektro

fakultas teknologi elektro dan informatika cerdas

INSTITUT TEKNOLOGI SEPULUH NOPEMBER 2023

1. In the long term (as t → ∞), the system is expected to reach a steady state, where the rate of change of each follower's state is zero. This means that for each follower, the influences from its neighbors balance out.

Setting each equal to zero and solving, we get:

Solving this system of equations, we can get:

These are the steady state values of , , and .

Regarding the convergence rates and the influence of , , and :

The terms α1, α2, and α3 are the weights that the followers place on the difference between their current state and the state of their neighbors. If αi is larger, then follower fi will adjust its state more quickly in response to a difference with its neighbors.

1. Consider an edge tension.

Determine associated with negative gradient flow . Also, assuming the graph is undirected and connected at all times, what will the agents’ states be at infinite time?

We know that *the edge tension* in the problem can be written as

so that the derivative is against

and when a negative value is selected

and

If or the initial graph is assumed to be an undirected graph and connected all the time, the agents will achieve convergence on the values if each agent using *control law* above.

1. Consider agents consisting of leaders and followers all with scalar dynamics placed at at . Let the dynamics of each leader with state be.

For some positive weight . Moreover, assume that the followers are executing.

Suppose each agent has a radius (radius will be used for designing proximity graph). What are the values of and in order for the agents to stay connected from origin initial value to ?

From the state and we can see that the leader and follower have quite similar dynamics, it's just that the leader dynamics get additional inputin the form of the leader's distance to the target multiplied by the weighting. Therefore, to choose a value and should not be bound by any conditions. However, generally the number of leaders is less than the followers so

The selection is also so, we do not need to limit where the desired goal position is. But the election of will affect to . Such a position can make a very large value, especially if the value is also large. It can cause great value addition to the leader will leave his followers.

Thus, it is necessary to design a rule where the value and are inversely proportional. When the distance to the target is still far away, the value is made small. Conversely, as the distance to the target gets closer, the value will get bigger. As a result, it is no longer a weighting whose value is constant but becomes a variable.

Then, we need to know that connectivity can be maintained during if and only if where is the range sensor on the agent. Values can be adjusted to the agent's specifications. One solution is to provide control law as in problem number 2.

This method is better known as the hysteresis protocol.

1. Let be the rows of the 4 x 4 identity matrix in the observation scheme for a four-node sensor network, observing state . It assumed that the nodes from a cycle graph and that is a zero-mean, unit variance, gaussian noise. Choose the weighting matrix and step size which satisfied the condition for stability.

There are 4 nodes or agents that are accounted for by a measurement model.

value will always be updated through the equation.

We can change equation above to

Where is the sum of the identity matrix to the Laplacian matrix of the graph ( is a graph whose relationships between *nodes*).

and itself can be obtained from

It can be seen from the explanation above that and will affect the response of the nodes. The selection of the value of the two cannot be done carelessly. To achieve convergent values, it is necessary.

Trying to use variation

Simulate in MALTAB, with initial random initial state and number iteration 25. Result state



Trying to use variation

Simulate in MALTAB, with initial random initial state and number iteration 25. Result state



1. Determine the control signal u[k] which can stabilize the floating nodes. Show the code and give some comments on the evolution of states of the agents.

% number of agents

N = 5;

% adjacency matrix for a cycle graph

A = [0 1 0 0 1;

1 0 1 0 0;

0 1 0 1 0;

0 0 1 0 1;

1 0 0 1 0];

% input node and floating nodes

input\_node = 1;

floating\_nodes = 2:N;

% Laplacian matrix

L = diag(sum(A)) - A;

% Af and Bf matrices

Af = L(floating\_nodes, floating\_nodes);

Bf = L(floating\_nodes, input\_node);

% time steps

T = 50;

% control signal

u = zeros(T, 1);

u(1) = 1; % initial control signal

% initial states

xf = randn(N-1, T); % initial states of floating nodes

% output

y = zeros(T, 1);

% simulation

for k = 1:T-1

% update states of floating nodes

xf(:,k+1) = xf(:,k) + 0.1\*(-Af\*xf(:,k) - Bf\*u(k));

% calculate output

y(k) = -Bf' \* xf(:,k);

% update control signal

u(k+1) = -y(k); % feedback control

end

Result states node, control signal, and output illustration.







1. Design a simple code in MATLAB to convert a position of 7 agents in a two dimension into both Voronoi diagram and proximity graph. Input of the program should be just position , and the output should be both Voronoi diagram and the proximity graph in two separate figures. Show the code.

clc; clear; clf

% positions of the 7 agents

pos = [3 2; 5 3; 6 4; 5 9; 6 8; 7 2; 1 8];

% check if pos is a 7x2 matrix

if size(pos, 1) ~= 7 || size(pos, 2) ~= 2

error('Input should be a 7x2 matrix where each row is an (x, y) coordinate of an agent.');

end

% Voronoi Diagram

figure(1);

voronoi(pos(:,1), pos(:,2));

title('Voronoi Diagram');

% Proximity Graph

figure(2);

DT = delaunayTriangulation(pos(:,1), pos(:,2));

triplot(DT);

title('Proximity Graph');



