Notes for Autonomous Robotic Networks

Professor Jason Isaacs

1 Introduction

Professor Isaacs is from Kentucky, and is the first member of his family to go to college. Got his Bacchelors in Physics, and then worked at a Control Systems Engineering company? He was interested for 2 years, but the lack of innovation caused him to go to UCSB to get his PhD in Control Systems.

Andrew Trough — his company Chessapeke Technology, a military contract corp, is paying for his Masters. He was an undergrad here and grew up in this area.

Matt — Bachelors in Physics in UCSD — didn't like it because he was

Dharini

Dharuv — wanted to be a professor.

Sing — animation, ee degree.

Vishanta — born in Shri Lanka — was an EE in central Africa. Moved to the US 16–17 years ago.

2 Graph Theory

We will be designing systems for groups of agents who have only local information. This information includes:

1. Communication
2. Sensing

Note that connections are not always circular over a distance, and are generally not associative. This defines the graphs that we use in our model. Generally, all to all communication is impossible. Connections may be directed, but we won't worry about that now.

2.1 Sensing

One example of sensing is a video feed. This is directional connection, but we won't discuss this too much. Other examples include

- Omnidirectional laser rangefinder.
- Tactile sensor.

- A Single ray.

This motivates these graph models.

- Nodes (verticies)
- Edges — connections.

Three Network Graphs

1. Static Network — If x_1 can communicate with x_2 they will always be able to communicate. This can be studied using Linear Systems (LTI)
2. Dynamic Networks — for mobile agents. If x_1 can communicate with x_2, this may be state dependent and time variant. Hybrid System (Non-linear stability)
3. Random Networks — from Packet Drops. Lyapunor Theory or Stochastic Stability.

We will cover

1. Consensus — global agreement among the agents. Given only local communication, can the entire group converge into a common value.
2. Formations. Can we do this with algorithms.
3. Assignments — task allocation. How do we assign roles to individuals autonomously. Concord algorithm — for the Traveling Salesman Problem (TSP) which is quite popular challenge for assignments.
4. Coverage.
5. Flocking / Swarming.
6. Social Networks. This is good for finding opinion dynamics.
7. Distributed Estimation. This is similar to consensus, but can be more general. It can mean more than averaging.

3 Graph Theory (for real this time)

Graphs are made up of vertices and edges. Let V be the vertices and E be the edges, and the graph then be $G(V,E)$. We now cover undirected graphs. We define the neighborhood $N(i)$ of i is defined as

$$N(i) \equiv \{v_j \in V | (v_i, v_j) \in E\} \subseteq V$$

Now, if $v_j \in N(i)$ then $v_i \in N(j)$. Then a path through the graph goes across edges. Similarly, a m-length path is exactly what you'd expect. We now define end points and interior points. If all vertices in the sequence are distinct, except the end points, then this is a cycle. It appears a path may contain a cycle, but it is not generally used that way.

Connectivity: a graph G is connected if for every pair of vertices $V(G)$ there is a path with them as end points.

Degree: for an undirected graph G, the degree for a given vertex $D(v_i)$ is the cardinality of the neighborhood set $N(i)$.

We will now define the degree matrix $\Delta(G)$ which is a diagonal matrix of the degrees.

$$\Delta(G) = \begin{bmatrix} d(v_1) & 0 & \cdots & 0 \\ 0 & d(v_2) & \cdots & 0 \\ 0 & 0 & \cdots & d(v_i) \end{bmatrix}$$

An $n \times n$ Adjacency Matrix is defined as

$$[A(G)]_{ij} = \begin{cases} 1 & \text{if}(v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

The graph Laplacian is defined as

$$L(G) = \Delta(G) - A(G)$$