MECH 6V29: Multiagent Robotic Systems- HW 5

Jonas Wagner

2022, April 20^{th}

Contents

Preliminary Notes

a) Definitions

Definition 1. *Graph $G(V, E)$ is constructed with vertex set*

$$V = \{v_1, v_2, \ldots, v_n\}$$

of n discrete vertices and edge set

$$E = \{e_1, \ldots, e_m\} \subseteq V \times V$$

consisting of m edges $e_{k=(i,j)} = (v_i, v_j)\forall_{k=1,\ldots,m}$ connecting vertices v_i and v_j.

Definition 2. *Let $V = \{v_1, v_2, \ldots, v_n\}$ be vertices. Δ-Disk Graphs are constructed for a particular Δ such that*

$$(v_i, v_j) \in E \iff \|v_i, v_j\| \leq \Delta$$

Definition 3. *Let $V = \{v_1, v_2, \ldots, v_n\}$ be vertices. A Gabriel Graph is defined as $G(V, E)$ in which*

$$\forall_{v_i, v_j \in V}(v_i, v_j) \in E \iff \forall_{v_k \in V} v_k \notin D(v_i, v_j)$$

where $D(a, b)$ is the closed disc with diameter between (a, b). In other words, a disk constructed from two adjacent vertices should not contain any other vertices.

Definition 4. *Let graph $G(V, E)$ with $V = \{v_1, \ldots, v_n\}$ and $E \subseteq V \times V$.*

a. $G(V, E)$ is considered undirected if

$$(v_i, v_j) \in E \iff (v_j, v_i) \in E$$

otherwise, $G(V, E)$ is considered directed.

b. An undirected graph $G(V, E)$ is considered connected if there exists a path between any two vertices.

c. An undirected graph $G(V, E)$ is considered planer if the graph can be drawn on a plane without any edges crossing.

Definition 5. *Let graph $G(V, E)$ with $V = \{v_1, \ldots, v_n\}$ and $E \subseteq V \times V$.*

a. The degree matrix $\Delta \in \mathbb{R}^{n \times n}$ is a diagonal matrix defined as

$$\Delta := \begin{bmatrix} \deg(v_1) & & & \\ & \deg(v_2) & & \\ & & \ddots & \\ & & & \deg(v_n) \end{bmatrix}$$

b. The adjacency matrix $A \in \mathbb{R}^{n \times n}$ is a symmetric matrix $(A = A^T)$ defined s.t.

$$A = [a_{ij}] \ : \ a_{ij} \begin{cases} 1 & (v_i, v_j) \in V \\ 0 & (v_i, v_j) \notin V \end{cases}$$

c. *The incidence matrix* $D \in \mathbb{R}^{n \times m}$ *is defined as*

$$D = [d_{ij}] \;:\; d_{ij} \begin{cases} 1 & (v_i, -) \in e_j \\ -1 & (-, v_i) \in e_j \\ 0 & otherwise \end{cases}$$

d. *The Laplacian matrix* $L \in \mathbb{R}^{n \times n}$ *is a symmetric* $(L = L^T)$ *and strictly semi-positive definite* $(L \succeq 0)$ *is defined as*

$$L := \Delta - A = DD^T$$

and

$$L = \begin{bmatrix} \deg(v_1) & -a_{12} & -a_{13} & \cdots & -a_{1n} \\ -a_{21} & \deg(v_2) & -a_{23} & \cdots & -a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_{n1} & -a_{n2} & -a_{n3} & \cdots & \deg(v_n) \end{bmatrix}$$

e. *For a weighted graph* $G(V, E, W)$, *the diagonal weighted matrix* $W \in \mathbb{R}^{m \times m}$ *is defined as*

$$W = [w_{ij}] \forall_{ij \in E}$$

were w_{ij} *are the corresponding weights for* $e_{ij} = (v_i, v_j)$.

Definition 6. *Consider directed graph* G_F.

a. G_F *is Rigid if certain interagent distances are maintained then all interagent distances are maintained when the formation moves smoothly.*

b. G_F *is Constraint Consistent if the directed graph is able to maintain the specified interagent distances.*

c. G_F *is considered persistent if and only if* G_F *is Rigid and Constraint Consistent.*

Definition 7. *Let* $G(V, E, l, \Sigma)$ *be a labeled graph with vertices* V, *edges* E, $\Sigma = \{\sigma_1, \ldots, \sigma_p\}$ *be a set of labels, and* $l : V \to \Sigma$ *be a function relating each vertices to labels.*

A graph grammar is a set of rules for changing l *and* E. *Each rule,* r_i, *takes labeled subgraph* L_i *and changes that subgraph in* R_i. *i.e.*

$$r_i : L_i \longrightarrow R_i$$

Problem 1

State a summary of **Notes 16.1-17.4**, (which Include the topics of Lloyd's algorithm, network controllability (upper and lower bounds)) preferably by creating a concept map diagram (flow diagram). The whole purpose is to make sure that we are clear about the bigger picture, and reiterate why are we doing and discussing the specific topics in the class. Do not merely write the topics, instead create connections between topics to clarify the flow of information.

a) Big Picture Chart

TO DO: Draw graph

Problem 2

Problem

A MATLAB Code:

All code I write in this course can be found on my GitHub repository:
https://github.com/jonaswagner2826/MECH6V29_MultiagentRoboticSystems