ENG-466_DIS_Notes

> *Lecture notes by Yujie He*
>
> *Last updated on 2021/4/11*

Lecture1-Lecture4

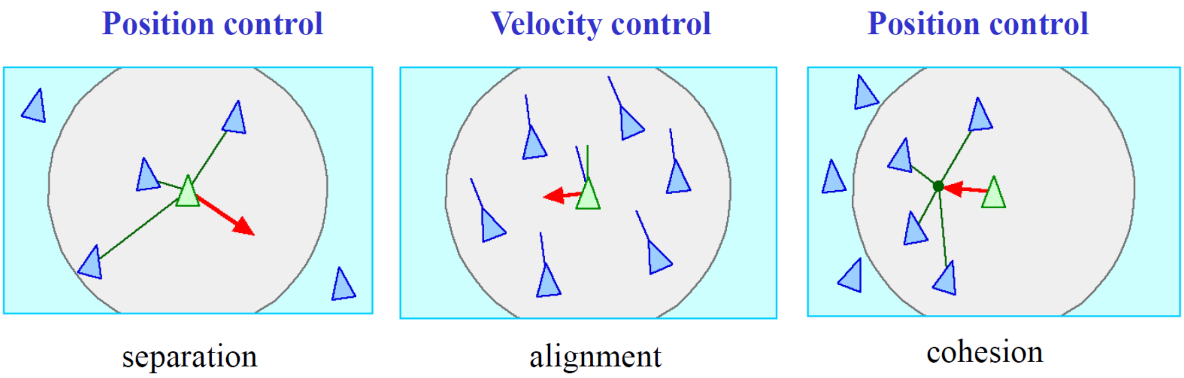
🚧 To be updated!!!

Lecture5-More on Localization Methods and an Introduction to Collective Movements

🚧 To be updated!!!

Boids' Flocking Rules

1. **Separation**: **avoid collision** with nearby
2. **Alignment**: attempt to **match velocity (speed and direction)** with nearby
3. **Cohesion**: attempt to **stay close to** nearby



Boids' Sensory System (ideal)

- **Local, almost omni-directional** sensory system
- **Perfect relative range and bearing system**
- **Immediate response**: perception-to-action loop
- **Homogeneous** system
- **"Natural" nonlinearities**: negative exponential of the distance

Lecture6-Collective Movements in Multi-Robot Systems

6.1 Flocking for Multi-Robot Systems

> *robots to stay together while navigating in the environment as a group.*

- applications: lawn-mowing, vacuum cleaning, security patrolling, overage and mapping, search and exploration in hazardous environment

6.1.1 Differences between digital and physical world

6.1.1.1 A Real On-Board Sensory System for Flocking

compared to Boids' Sensory System (ideal)

- In general, for real robots
  - **Noise** in the range and bearing measurement, communication
  - **Homogeneous system impossible**-small manufacturing discrepancies -> calibration might be the key for an efficient system
  - **Immediate response impossible**-limited computational and sensory capacity
  - **Identifier** for each teammate possible but scalability issues-assigning ID to each robot
  - **Non holonomicity** of the vehicles-e-puck robot design

- More specifically, for **local range and bearing systems**
  - Depending on the system used for range and bearing-**occlusion possible (line of sight)**
  - **Nonlinearities** determined by the underlying technology-need control to compensate
  - **Estimation and filtering take time**: Second order variables, **velocity**, **estimated** with 2 first order measures, position, takes time; noisier signal also takes more time to filter

6.1.2 Examples

6.1.2.1 Kelly's Flocking (1996)

- Separation and cohesion only, but no alignment
- Migration script replaced by **leadership**
- using onboard IR system to local communication and sonar (ultrasound) for avoiding with robot and wall at fast speed of 10 Hz

6.1.2.2 Hayes's Flocking (2002) at Caltech

- a group of 10 Moorebot
- Separation, cohesion, and alignment
- Range & bearing using **off-board system** with overhead **camera** and **LAN radio** channel
- Different platforms (lab robots, UGVs)

6.2 Formations for Multi-Robot Systems

6.2.1 Behavior-based control strategies

> *from Balch & Arkin, 1998 [pdf]*

- **Absolute coordinate** system assumed but positional **error** considered (GPS, dead reckoning)
- **Fully networked** system but **transmission delays** considered (and formation traveling speed adapted4
- **Motor-schema-based** formation control

  move-to-goal, avoid-static-obstacle, avoid-robot, and maintain-formation

6.2.1.1 Formation Taxonomy

- Based on formation shape: line, column, diamond, wedge
- Based on the reference structure (ex. on wedge): **unit-center**-referenced, **leader**-referenced, **neighbor**-referenced

6.2.1.2 Fredslund & Matarić (2002)

> *inspiring for course project*

- Overview
  - **Neighbored-referenced** arch based on a **on-board relative positioning**
  - **single leader** always
  - **extreme low formation** speed (2cm/s)
  - Tested on **4 different formations mentioned before** + **switching** between them
  - Each robot has an ID under global network (broadcasted regularly)

- a function of the formation + order in the chain (ID-based rules)
- relative range and bearing to another robot is calculated
- **Hardware for inter-robot relative positioning**-Combined use of **Laser Range Finder (LRF)** and **pan camera**
    - Relative range: LRF
    - Relative angle: through the camera pan angle; neighboring robot kept in the center (also for a robustness sake)
    - Neighboring robot ID via **colored visual beacon**

6.2.1.3 Pugh et al. (2009) at EPFL

> *Paper in pdf; Spec can be found in Week 4 slides*

- Khepera III robots
    - Range: 3.5 m
    - Update frequency 25 Hz with 10 neighboring robots (or 250 Hz with 1)
    - Line-Of-Sight method
    - Measure range & bearing can be coupled with standard RF channel (e.g. 802.11) for heading assessment
- **Formation Taxonomy**
    - **Neighbor-referenced** control using an **on-board relative positioning** system
    - Approach: **potential field control** (similar to Balch 1998)
    - Formations can be divided into **two categories**
        1. **Location**-based (position-based): maintain relative fixed location between teammates (x heading)
        2. **Heading**-based (pose-based): maintain relative fixed location and headings

           subset-**leader heading-based**: only the pose of leader as reference
- **Formation Localization Modes**
    - Mode 1: No relative positioning
    - Mode 2: Relative positioning -> **location**-based formation

      observe teammates with relative positioning module
    - Mode 3: Relative positioning with communication -> **Heading**-based formation

      observe and **share information with leader robot** using relative positioning and **wireless radio** (sending the heading between robots)

  Notes:

    - **Pose-based formations** can also be achieved by using local localization systems that also deliver full pose of the neighbor without communication (e.g., **multi-markers or shape detection + vision**)
- **Sample Results**
    - **Setup**: Diamond formation movement; Robot speed at the speed of 10 cm/s; Update rate at about 10-15 Hz
    - **Metric**: **average position error** for the 4 robots **respect to the prescribed diamond shape** measured with an overhead camera system
    - Conclusion
        - Without the relative positioning, the error will increase along the time compared to that using relative positioning
        - Without communication, the error is about two times than that with communication

6.2.2 Graph/Consensus-based (consensus-based) control strategies

- Motivation: **Graph-theory** to reconfigure, avoid obstacles, control cohesion or formation

6.2.2.1 Graph Definitions

- Graph $G = (V, E)$, Vertex Set, Edge Set

  ❗ can be oriented (directed), only assume undirected graphs in this lecture

- Matrixes

Incidence Matrix	Weight Matrix	Laplacian Matrix
$\mathcal{I} \in \mathbb{R}^{\Vert\mathcal{V}\Vert \times \Vert\mathcal{E}\Vert}$	$\mathcal{W} \in \mathbb{R}^{\Vert\mathcal{E}\Vert \times \Vert\mathcal{E}\Vert}$	$\mathcal{L} \in \mathbb{R}^{\Vert\mathcal{V}\Vert \times \Vert\mathcal{V}\Vert}$
$\mathcal{I}(i,j) = \begin{cases} -1, & \text{if } e_j \text{ leaves } n_i \\ 1, & \text{if } e_j \text{ enters } n_i \\ 0, & \text{otherwise} \end{cases}$	$\mathcal{W}(i,j) = \begin{cases} w_i, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}$	$\mathcal{L} = \mathcal{I} \cdot \mathcal{W} \cdot \mathcal{I}^\mathsf{T}$
Arbitrarily choosing an orientation for any edge for undirected graph	indicate the weight associated with the edge; larger value means more important edge	if any element value (degree of node) not equal to 1, that the matrix is called weighted Laplacian matrix
$I = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$	$W = \begin{bmatrix} w_1 & 0 & 0 & 0 & 0 \\ 0 & w_2 & 0 & 0 & 0 \\ 0 & 0 & w_3 & 0 & 0 \\ 0 & 0 & 0 & w_4 & 0 \\ 0 & 0 & 0 & 0 & w_5 \end{bmatrix}$	$L = \begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & -1 & 2 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}$

💡 Ref: https://www.cis.upenn.edu/~cis515/cis515-14-graphlap.pdf

6.2.2.2 The *Rendezvous* (会合) Problem

- **Formulation** (1D)
  - Each node is given a state x_i as time tends to infinity
  - Final consensus value **not pre-established** but consensus framework (e.g., variable type, range) is **shared** and defined a priori
  - Example-5 robots R_i moving on 1D measuring distances d_{ij}
  - 💡Note: to construct a consensus-based control law; don't need a fully-connected graph in real-world task
- **Solution**
  - use the Laplacian matrix: $\dot{x}(t) = -\mathcal{L}x(t)$, which is is equivalent to

    $\dot{x}_i = \sum_{\mathbf{R}_j \in \mathcal{N}_i} w_{ij} \left(x_j - x_i\right)$

    where $\mathbf{R}_j$ and $\mathcal{N}_i$ indicate the robot j and neighborhood of robot i.
  - If $w_{ij} > 0$, graph connected, rendezvous is guaranteed.
- **Generalization** (2D or more): solve the problem for each dimension separately
- **Holonomicity** for *Rendezvous*
  - definition: total number of degree of freedom = number of controllable degree of freedom
  - From the point of view of mobility: a mobile robot is holonomic if it can **move in any direction at any point in time.**
  - **Considerations**
    - Laplacian method gives the **direction vector** at each point in time
- **holonomic** robots we can simply go in that direction while **non-holonomic need to transform the direction vector** in something useable by the robots given their mobility constraints
  - **Solution for non-holonomicity**

- (Using **global** localization system) Derive transformation: total degrees of freedom (DOFs) -> controllable DOFs, and eventually to actuator control via the inverse kinematic model
    - all about finding the right function f such that $\begin{bmatrix} u \\ \omega \end{bmatrix} = f(\dot{x}, \dot{y})$
        - ❗ rendez-vous is **not supposed to find consensus on the full pose, only position**
    - procedure: The motion is directed toward the goal; speed is proportional to the distance to that goal (nonlinear but proportional controller)
- (Using **relative/local** localization system) Apply relative range and bearing

Global	Relative
transform the global coordinates to local coordinates	
associate the controller DoF with local coordinates	$u = K_u e \cos\alpha$ $w = K_w \alpha$

6.2.2.3 Reconfiguring

- Motivation: exploit the consensus for robots to reach prescribed points from random initialization
- Configurations Using a Bias
    - by adding a bias vector, we can modify the state (or assumed position): $\dot{x} = -\mathcal{L}(x(t) - \mathcal{B})$
    - achieve in decentralized way using relative positioning -> Each robot solves the Laplacian equation taking as x and y the relative coordinates of the other robots

6.2.2.4 Real robots example [Falconi et al., ICRA 2010]

> *Gowal S., Falconi R., and Martinoli A., "Local Graph-based Distributed Control for Safe Highway Platooning". Proc. of the 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, October 2010, Taipei, Taiwan, pp. 6070-6076.*

6.2.2.5 Adding obstacle avoidance/cohesion control

- Can we do? Yes, we can change $\mathcal{W}$ in $\mathcal{L} = \mathcal{I} \cdot \mathcal{W} \cdot \mathcal{I}^\intercal$
- **Obstacle avoidance**
    1. **propagate** the **position** of the detected **obstacle** to **other robots**
    2. Each robot **updates** its neighbors list if necessary by **adding a repulsive agent**
    - **Positive** weights will **attract** vehicles together while **negative** weights will create a **repulsion** mechanism

6.2.2.6 Graph-Based Formation Control [Falconi et al., ICRA 2010]

> *https://doi.org/10.1109/IROS.2010.5649318*

- to enable a group of robots (the followers) to follow a robotic leader
- we can also modify the control law (**PI controller**)

  For example, single leader moving **at a constant velocity**, we can **add an integral term**

$$u = K_u e \cos\alpha + K_I \int_0^t e \, \mathrm{d}t$$
$$w = K_w \alpha$$

Task	Result

6.2.2.7 Additional Estimation layer [Falconi et al., 2013]

> *https://link.springer.com/chapter/10.1007/978-3-319-00065-7_25*
>
> *PhD Thesis: https://infoscience.epfl.ch/record/187744*

to deal with noisy localization

Lecture7-Multi-Level Modeling Methods for Swarm Robotic Systems

🚧

Misc.

- range and bearing 距离与方位
- flocking 群
- formation control 编队控制