UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

Multi-robot Systems

RO51 - Introduction to Mobile Robotics

Zhi Yan

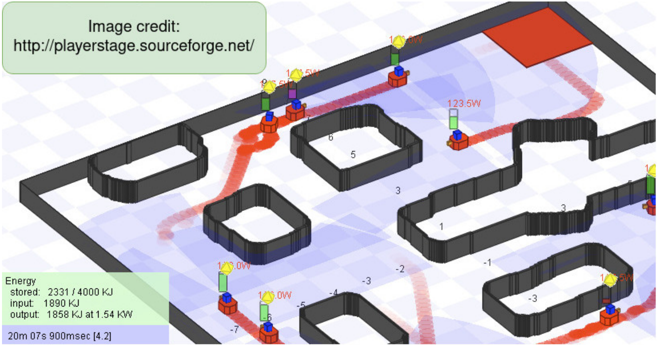
June 5, 2024

https://yzrobot.github.io/

www.utbm.fr

What?

- As the name suggests, a multi-robot system refers to an **organized** unit consisting of two or more robots.
- "**Organized**" usually means that there is some kind of **collaboration** between robots: need for **coordination** mechanism.



Why?

Using a multi-robot system (MRS) has several potential advantages over a single-robot system:

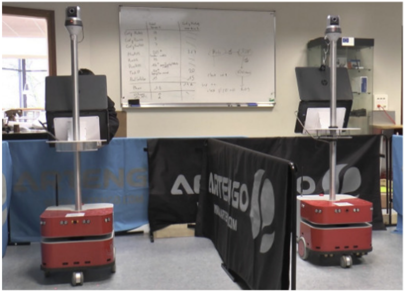
- A MRS has a better spatial distribution.
- A MRS can achieve better overall system performance (e.g. faster).
- A MRS introduces robustness that can benefit from data fusion and information sharing among the robots, and fault-tolerance that can benefit from information redundancy.
- A MRS can have a lower cost (a set of simple robots vs. a single powerful robot).
- A MRS can exhibit better system reliability, flexibility, scalability and versatility.

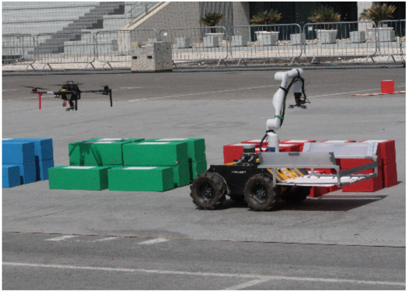
How?

- **Robot team design**: problem-oriented/demand-oriented (i.e. what robots should be included, geometry, kinematics, sensors, etc.).
- **Coordination**: multiple robots working together should generate collective actions/thoughts to benefit the work.
- **Communication** among robots.
- **Planning** for MRS (c.f. Lecture 7).
- **Decision-making**: who's in charge?

Robot team: homogeneous vs. heterogeneous

- In **homogeneous** robot teams, the capabilities of the individual robots are identical (physical structures do not need to be the same).
- In **heterogeneous** robot teams, the capabilities of robots are different, whereby robots can be specialized for specific tasks.





Inherent problem: resource conflict

- Generally, if multiple requests targeting the same resource arrive simultaneously, **resource conflict** will occur.
- In MRS, resource conflict typically arises when multiple robots need to share:
  - space,
  - manipulable object,
  - communication media.

Resource conflict: space

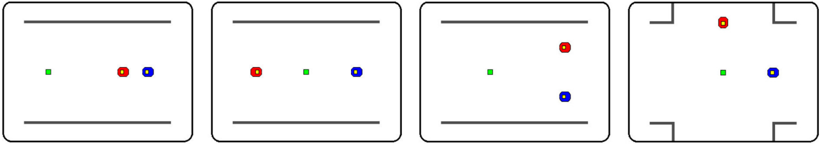
Problems caused by space conflicts:

- collision
- deadlock
- congestion

Resource conflict: space

Four typical cases of **congestion** due to mutual exclusion of waypoints in the field of multi-mobile robots (small green squares represent waypoints):

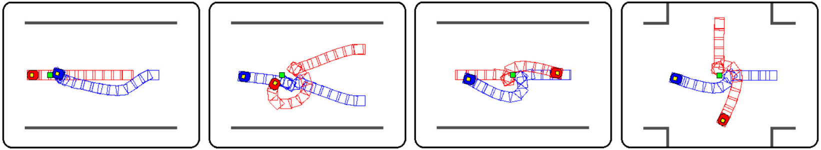
(a) same direction and same path
(b) different directions but same path
(c) same direction but different paths
(d) different directions and different paths



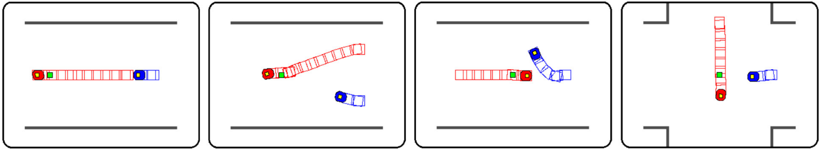
(a) (b) (c) (d)

Resource conflict: space

Without coordination (e.g. selfish):

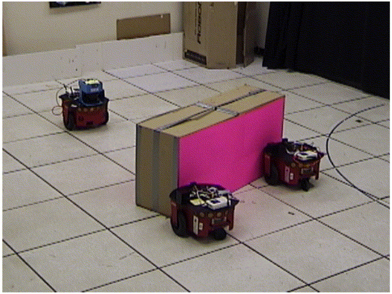


With coordination (e.g. one-by-one):



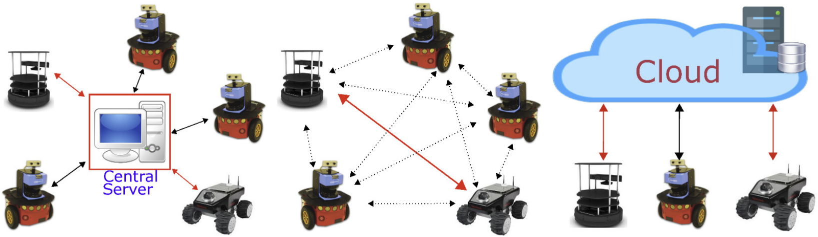
Resource conflict: object

When multiple robots manipulating an object together => There is an essential need for coordination since they usually have the same goal.



Resource conflict: media

Similar to space conflicts, communication media conflicts are usually caused by limited communication channel and bandwidth.



Coordination: static vs. dynamic

- **Static coordination**:
  - Also known as deliberative coordination or offline coordination.
  - Refer to the adoption of a convention prior to engaging in the task, e.g.:
    - "keep right"
    - "stop at intersection"
    - "keep sufficient space between yourself and the robot in front of you"
    - etc.
- **Dynamic coordination**:
  - Also known as reactive coordination or online coordination.
  - Occur during the execution of a task, based on the analysis and synthesis of collected information, and can be:
    - **explicit**: there is an active agreement between the robots, or
    - **implicit**: kind of tacit understanding that does not involve an initiative.

Communication: explicit vs. implicit

- Communication, as **a means of coordination**, often manifests as a rational behavior in MRS.
- Robots can share sensory data, positioning information, individual understanding and intent, and more with their peers in the system.
- Two communication modes:
    - **Explicit mode**: at least one of the sender and receiver of information is known to the other (e.g. unicast or broadcast over the network).
    - **Implicit mode**: the sender and receiver of the information are unaware of each other (e.g. markers or biochemical information left in the environment).

Planning (c.f. Lecture 7)

- The optimal planning for a MRS is typically an *NP-hard* problem. => It may be more realistic to find a "good", but perhaps not the best, planning.
- In MRS, **task planning** is primarily designed to solve the problem of which robot should execute which task, which involves:
  - Task decomposition: break down a task into several subtasks that can be performed by individual robots.
  - Task allocation: (reasonably) assign subtasks to different robots.
- In MRS, **motion planning** should not only avoid collisions between robots, but also further consider reducing any possible interference between them (c.f. space conflict).

Planning (c.f. Lecture 7)

Multi-robot motion planning, beyond the complexity of single robot motion planning, usually reduces the continuous motion planning problem to a discrete graph search problem by identifying some canonical states and paths in free space.

	Cell decomposition approach	Potential field approach	Roadmap approach	
			Voronoi diagram	Sampling-based method
Strength	optimal paths can be found	efficient and easy to implement	resulting paths tend to maximize clearance	efficient for high-dimensional problems
Weakness	strongly depends on the grid resolution of the world	easy to fall into local minima	inefficient and complex to implement	solutions often sub-optimal
Completeness	complete			probabilistically complete
Performance	computational complexity depends on the number of points			rate of convergence depends on the use of local planner
Main applied field	multi-robot area coverage	multi-robot formation control	multi-robot exploration	industrial manipulators

Decision-making: centralized vs. decentralized

- **Decision-making**: what should the robot do (with or without information updates)?
- In MRS, the decision-making can be:
  - **Centralized**: there is a central agent (can be a computer or a robot) that has global awareness of the entire MRS and is able to communicate with all robots.
    - More suitable for small robotic teams.
    - Suffer from uncertainties such as dynamic environments, communication failures, etc.
    - The performance and reliability of MRS is highly dependent on the capability and health of the central agent.
  - **Decentralized**: can be further divided into two categories:
    - **Distributed**: no central agent, all robot decisions are fully autonomous.
    - **Hierarchical**: one or more local central agents which organize robots into clusters. => A form between centralized and distributed

Case study

Multi-robot exploration

- Motivation/Objective: deploy a team of robots to complete the exploration of an unknown space in the **shortest possible time** (video)[1] <= an optimization problem
- Robot team composition: homogeneous (2x nonholonomic mobile robots (1x 2D lidar, 2x wheel encoders, 2x pneumatic bumpers, etc.))
- Resource conflict: space
- Coordination: dynamic
- Communication: explicit
- Decision-making: decentralized

[1] https://github.com/yzrobot/mrs_testbed

Case study

Multi-robot exploration

Communication:

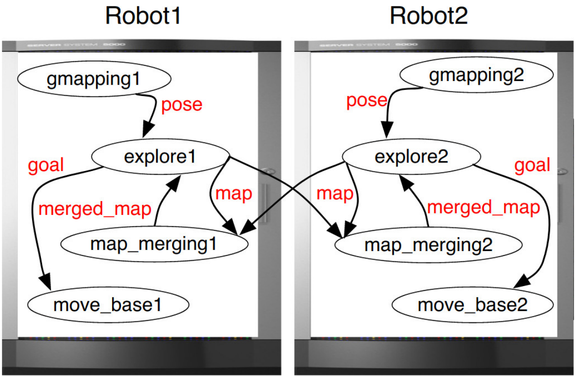
- Different robots should have different IPs in a networked MRS.
- Robots can communicate with each other via WiFi.
- Technically, the clocks of the robots in the team should be synchronized.

Case study

Multi-robot exploration

Coordination:

- Map merging: each robot shares the explored map with other robots in the team.



Case study

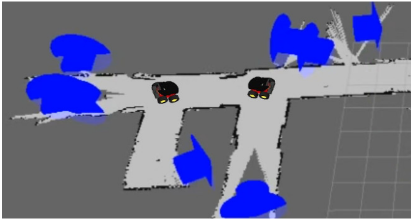
Multi-robot exploration

Suppose a central agent is added to transform the decision-making mechanism into a centralized one, consider the coordination problem from a planning perspective: what can we do (better)?

Task planning:

- A group of robots
- A set of frontiers

Which robot should explore which frontier? (c.f. bipartite graph)



Case study

Multi-robot exploration

Hungarian algorithm:

- H.W. Kuhn, **The Hungarian Method for the Assignment Problem**, *Naval Research Logistics Quarterly*, 1955.
- A classical algorithm for solving the maximum matching problem in bipartite graphs.
- The time complexity of the original algorithm was $O(n^4)$, but subsequent improved versions reached $O(n^3)$.

Consider assigning a set of tasks to a set of machines, each with a different cost to complete the task:

- By applying the Hungarian method, we hope to be able to accomplish this set of tasks with **minimal cost** (i.e. find the optimal solution to a given problem).

Case study

Multi-robot exploration

The Hungarian algorithm[2] can be summarized into the following four steps:

1. Build an $n \times n$ cost matrix, where each entry in the matrix is the cost of a machine to complete a task.
2. Calculate a reduced cost matrix by subtracting from each element the minimal element in its row. Then, do the same with the minimal element in each column.
3. Find the minimal number of horizontal and vertical lines required to cover all zeros in the matrix. In case exactly n lines are required, the optimal assignment is given by the zeros covered by the n lines. Otherwise, continue with Step 4.
4. Find the smallest nonzero element in the reduced cost matrix that is not covered by a horizontal or vertical line. Subtract this value from each uncovered element in the matrix and add this value to each element in the reduced cost matrix that is covered by a horizontal and a vertical line. Continue with Step 3.

[2] https://yzrobot.github.io/research/libhungarian.tar.gz

Case study

Multi-robot exploration

For frontier assignment in multi-robot exploration:

- Construct a cost matrix $n \times n$, where n corresponds to the number of frontiers (or robots). The entries in the matrix correspond to the length of the planned path from the robot's current position to the exploration target (i.e., the frontier).
- Use the Hungarian algorithm to find the best frontier allocation scheme.

Question: What if the number of frontiers and the number of robots are not equal? (the implementation of the above Hungarian method requires that the number of tasks and the number of machines are equal)

- Use "dummy robots" (with unlimited cost) or duplicate existing frontiers to implement a square matrix.

Case study

Multi-robot exploration

Motion planning:

- A group of robots
- A set of frontiers
- A set of waypoints

How to avoid multiple robots heading towards the same waypoint concurrently? (through which to travel to different frontiers)

=> Add traversable waypoints nearby.

Case study

Multi-robot exploration

Probabilistic Roadmap (PRM):

- L. Kavraki *et al.*, **Probabilistic Roadmaps for Path Planning in High-dimensional Configuration Spaces**, *IEEE Transactions on Robotics and Automation*, 1996.
- A classical (sampling-based) method for solving the robot path planning problem in high-dimensional spaces.
- Unlike incremental heuristic search (e.g. A* and D*) and topological map (e.g. Voronoi diagram), the running time of sampling-based methods does not grow exponentially with the dimension of the configuration space (i.e. C-space).

Case study

Multi-robot exploration

The original PRM algorithm[3] can be summarized into the following three steps:

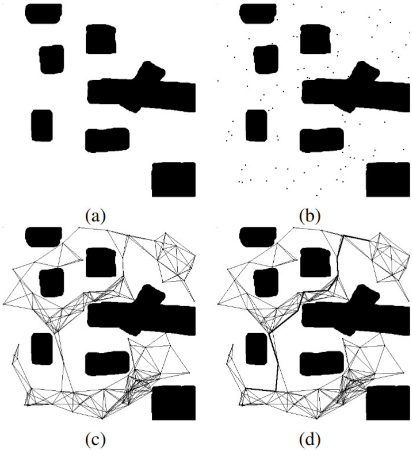
1. Randomly sample a sufficient number of points in C-space and keep any points that do not collide with obstacles. This creates a set of points in free C-space.
2. Apply a local planner (e.g. a line planner), trying to connect pairs of samples that are relatively close to each other. This creates a graph data structure called a roadmap.
3. Try to connect the start and target configurations by querying the roadmap. If successful, use standard graph search methods (e.g. A*) to search the graph for a path from the start point to the target.

[3] https://yzrobot.github.io/research/PRM.tar.gz

Case study

Multi-robot exploration

Generation of a probabilistic roadmap:



Summary

- Multi-robot systems (MRS): what, why, and how.
- MRS should contribute positively to the task: coordination is key.
- Inherent problem: resource conflict.
- Technical issues of concern for MRS: communication, planning, decision-making, and more.
- A case study: exploration with a team of robots.
- Extended reading: `http://journals.sagepub.com/doi/full/10.5772/57313`

The end

Thank you for your attention!

Any questions?