COS 495 – Lecture 23
Autonomous Robot Navigation

Instructor: Chris Clark

Semester: Fall 2011

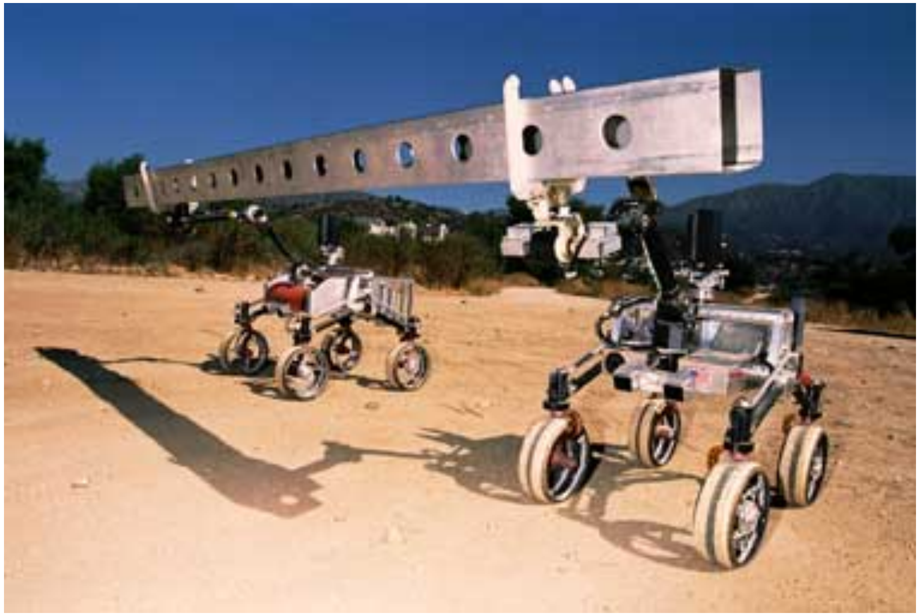
Figures courtesy of Siegwart & Nourbakhsh

Multi-Robot Systems: Outline

1. Motivation
2. Application Examples
3. Taxonomies
4. Motion Planning

Motivation

- Force Multiplication



NASA Planetary Outpost - JPL

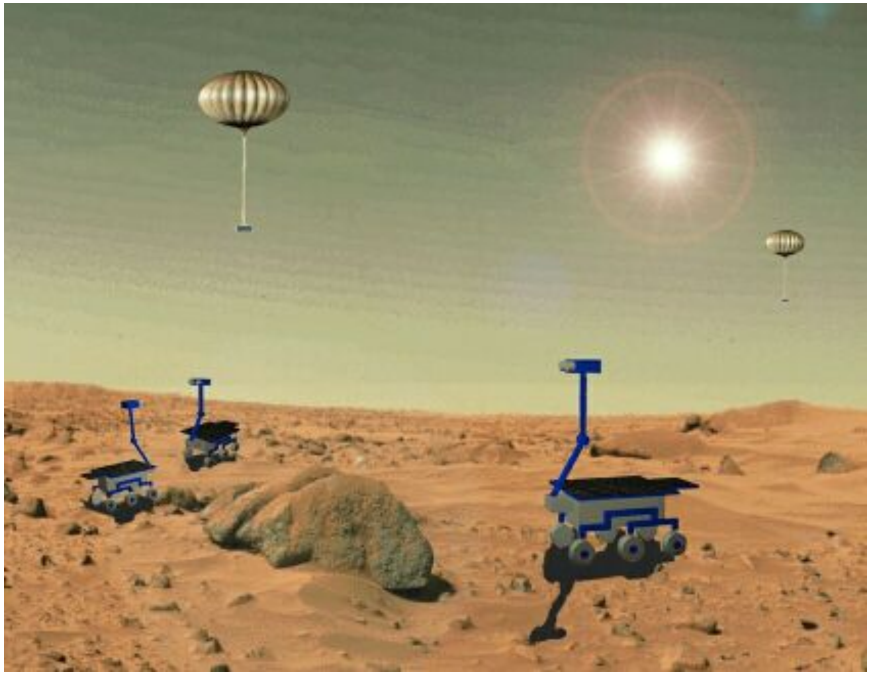
Motivation

- Simultaneous Presence

Security Robot - iRobot

Motivation

- Redundancy/Fault Tolerance



MARS Explorations - Matsuoka 2002

Motivation

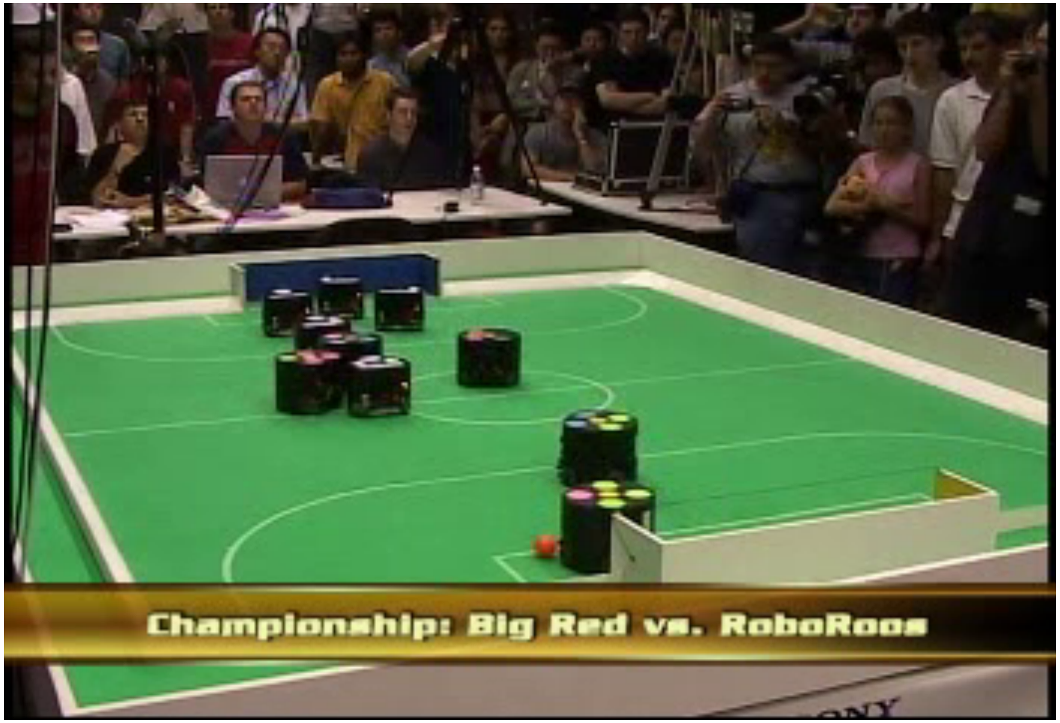
- Ideal Applications?:
  - For *R* robots, increase performance factor by greater than *R*.
  - Example: Applications that cannot be accomplished by only a single robot.

Multi-Robot Systems: Outline

1. Motivation
2. Application Examples
3. Taxonomies
4. Motion Planning

Application Examples

- Competitions

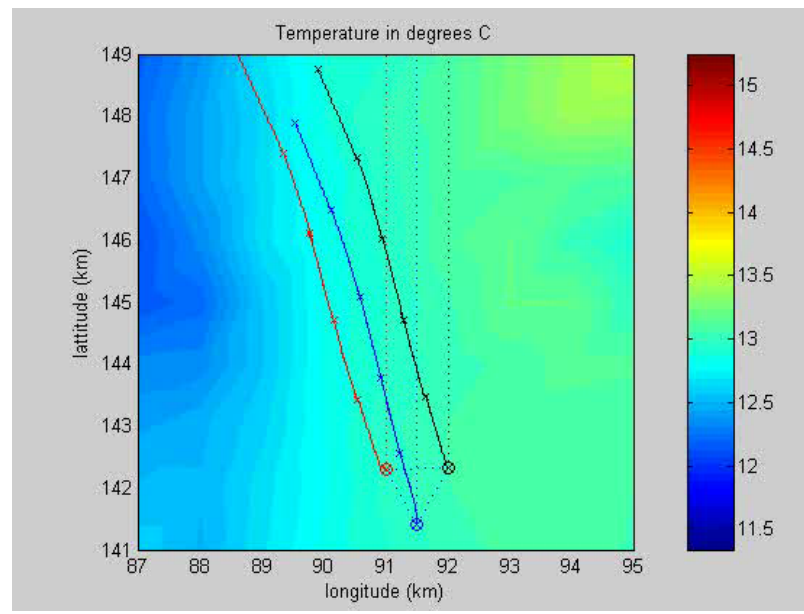


2003 RoboCup in Padua, Italy

Application Examples

- Underwater Sensing

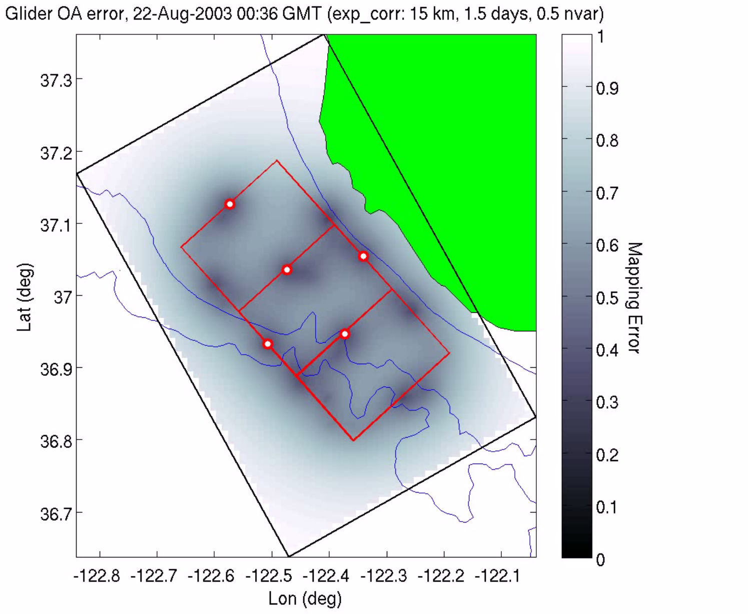
Gliders from the Autonomous Ocean Sampling Network II, Naomi Leonard, 2003



Application Examples

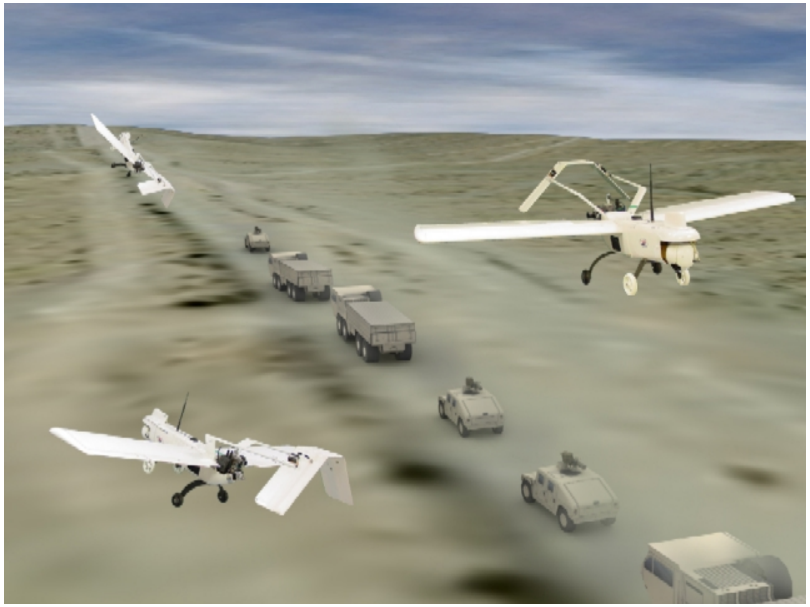
- Underwater Sensing

Adaptive Sampling & Prediction, Naomi Leonard

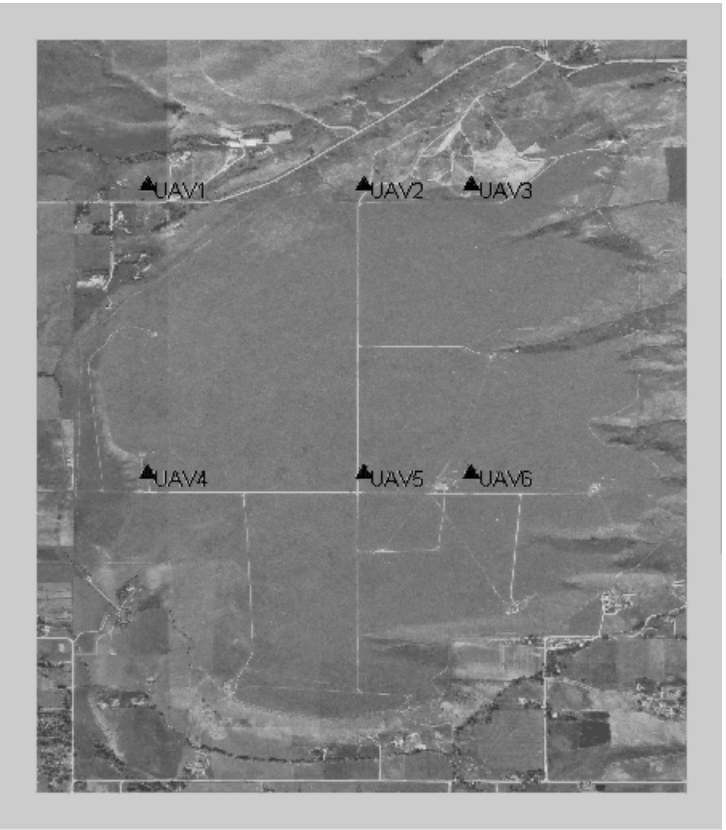


Application Examples

- Unmanned Aerial Vehicles



Eric Frew and MLB,



Multi-Robot Systems: Outline

1. Motivation
2. Application Examples
3. Taxonomies
    1. MRS Taxonomies
    2. Classifying an example system
4. Motion Planning

Taxonomies

- Taxonomies provide a classification system.
- We need taxonomies to
  - Allow us to compare different MRS
  - Identify the key issues in MRS
  - Identify trade-offs that can occur in MRS
- This is very important in a field where methods are application specific.

Taxonomies (Dudek, et. Al.)

- Communication
- Control Distribution
- Group Architecture
- Benevolence vs. Competitiveness
- Coordination & Cooperation
- Size
- Composition

Communication

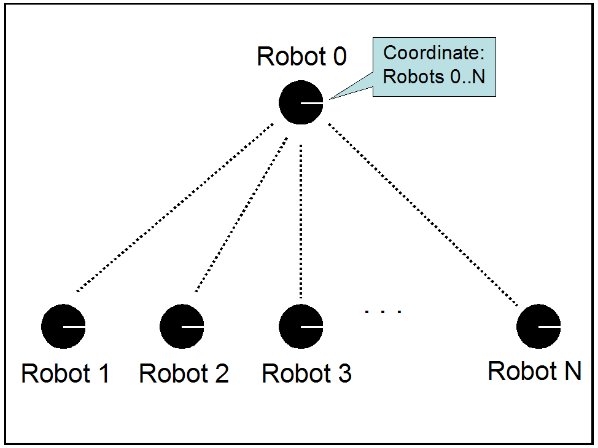
- Topology
  - broadcast
  - address
  - tree
  - graph
- Range
  - none
  - near
  - infinite
- Bandwidth
  - infinite
  - motion dependent
  - low
  - zero

Control Distribution

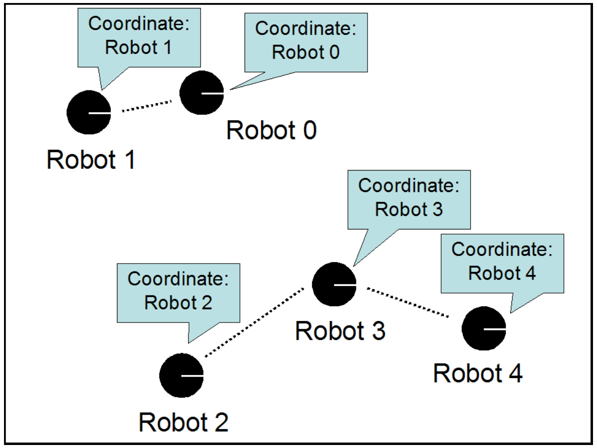
- Centralized
  - All control processing occurs in a single agent.
- Decentralized
  - Control processing is distributed among the agents.
- Hierarchies
  - Use groups of centralized systems.

Group Architectures (Cao et. Al.)

- Group Architectures are defined by the combination of control distribution and communication topology.
- Simply a different method of classification



Centralized



Decentralized

Benevolence vs. Competitiveness (Stone & Veloso)

- Benevolence
  - Robots are working together
- Competitiveness
  - Robots are competing for resources
  - Possibly wishing to harm one another
  - (not covered in our class)

Coordination & Cooperation

- Coordination
  - When many robots share common resources (e.g. workspace, materials), they must coordinate their actions to resolve conflicts (e.g. collision).
- Cooperation
  - Many systems strive to incorporate cooperation – where robots are working together towards common goals.
  - Cooperation requires coordination.

Size

- Define size of the MRS:
  - single robot
  - pair of robots
  - Limited number of robots
  - Infinite number of robots
- Scalability
  - Describes how amenable the system is to adding more robots.
  - Can result in a continuous degradation in performance as opposed to discrete.

Size

- Performance
  - We can characterize the performance of a system based on the number of robots
  - E.g. The number of tasks that can be accomplished in 1 hour.
- Interference
  - Given limited resources, there is often a plateau or even decrease in performance once a certain threshold of robots is reached.

Composition

- Homogeneous
  - All robots in the system have similar functionality and hardware.

- Heterogeneous
  - Robots have varying functionality and hardware.
  - Affects maneuverability, tasks achievable, control possibilites, …
  - Can lead to robots having “roles”

Classifying an Example System

- The Robot Scout System:
  - Used for sensing dangerous/hostile environments

Classifying an Example System

- Classifying he Robot Scout System based on our taxonomies:
  - Communication
    - Wireless RF
    - Broadcast with addresses
    - Near range
    - High bandwidth
  - Control Distribution
    - Hierarchical
  - Coordination and Cooperation
    - Both, but not autonomous

Classifying an Example System

- Classifying he Robot Scout System based on our taxonomies (cont’):
  - Benevolence vs. Competitiveness
    - Benevolent
  - Size
    - Limited (10)
    - Scalable within hierarchies, but not wrt autonomy since more operators required.
  - Composition
    - Heterogeneous

Multi-Robot Systems: Outline

1. Motivation
2. Application Examples
3. Taxonomies
4. Motion Planning
    1. Coupled and Decoupled Planning
    2. Decoupled Approaches

MRMP

- The two main approaches in MRMP are:

1. *Coupled Planning* - Plan for all robots at once
2. *Decoupled Planning* – Plan for robots one at a time

MRMP

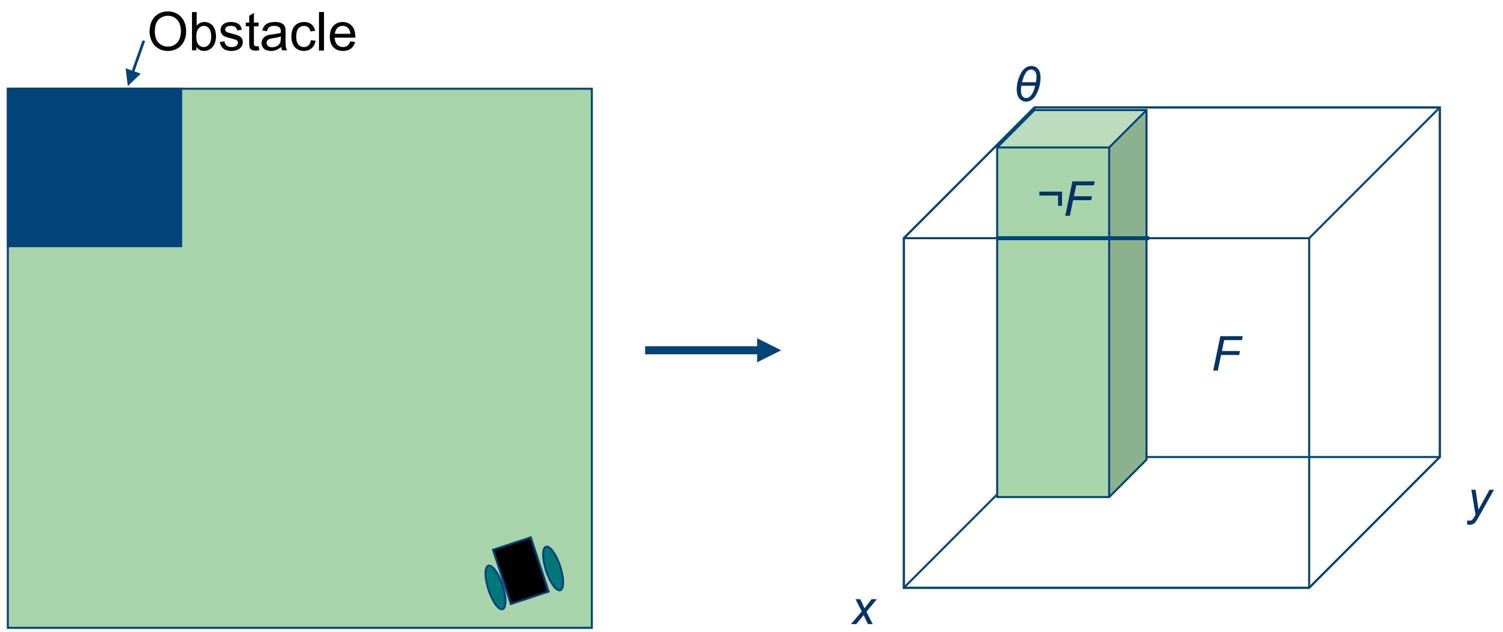
- In both approaches, time must be considered in the configuration space.
  - Whether a robot can occupy a space depends on if the space is occupied.
  - The occupancy of a space now varies with time because several robots are moving through the space.

MRMP

- In Coupled Planning, the composite configuration space of all robots is searched.
- In Decoupled planning, several searches of individual robot configuration spaces are conducted.

MRMP: The Configuration Space

- Previous Example: Mobile Robot



Workspace

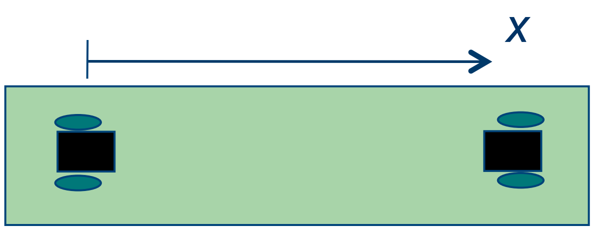
Configuration Space

MRMP:
The Configuration Space

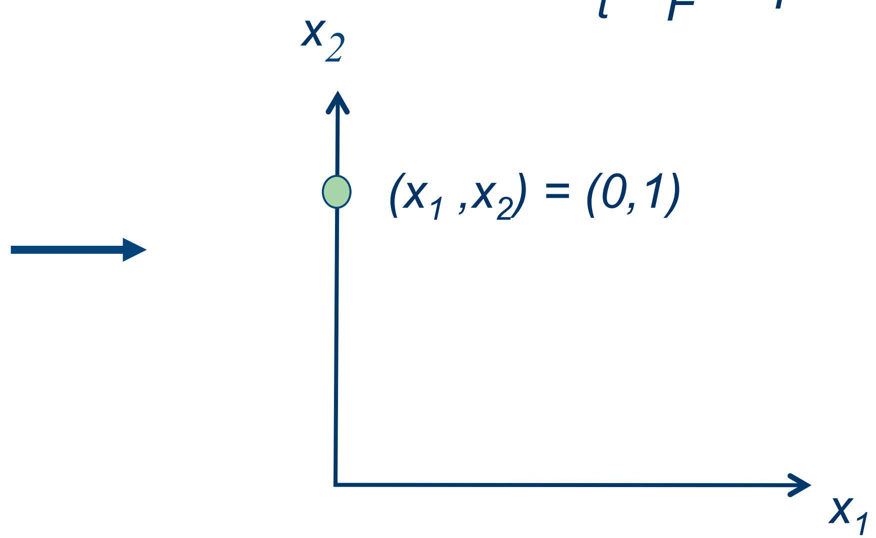
- New Example:
  - One Dimensional Multi-Robot System

¬F

t F F



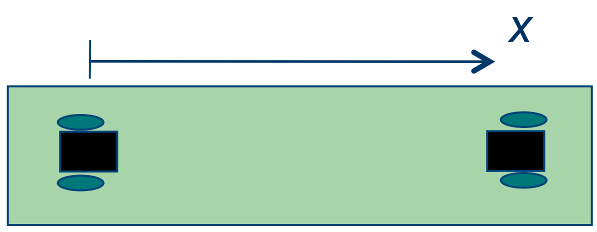
Workspace



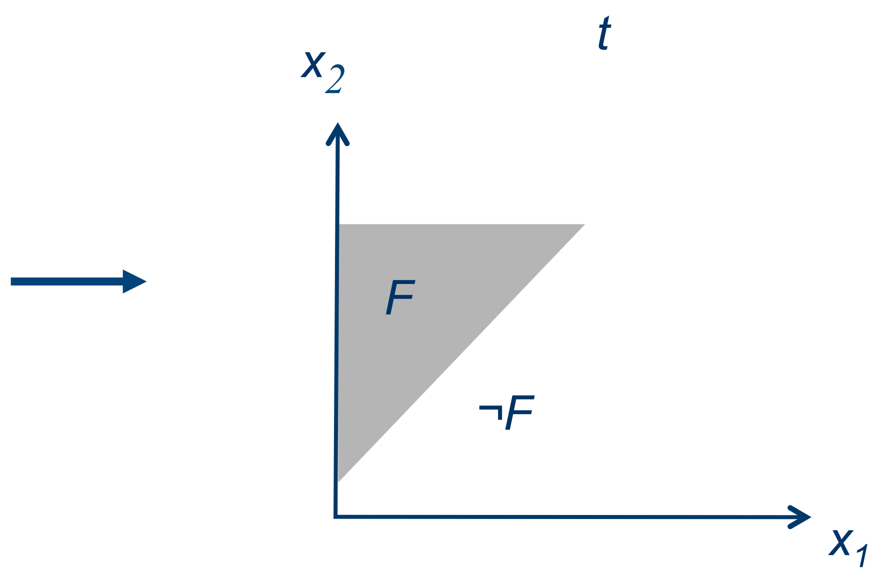
Initial Configuration

MRMP: The Configuration Space

- New Example:
  - One Dimensional Multi-Robot System



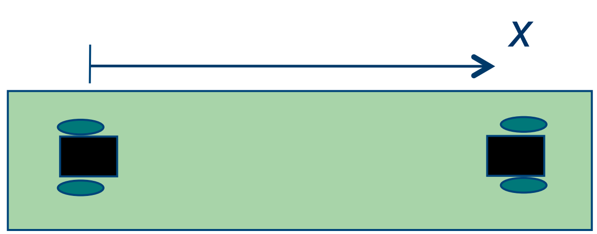
Workspace



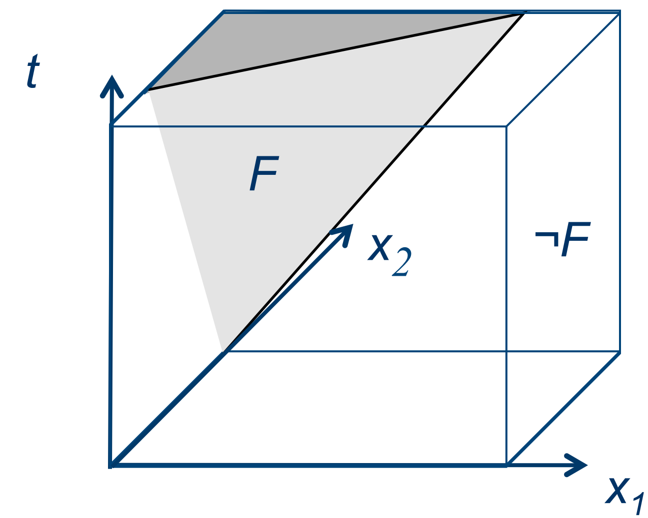
Possible Configurations

MRMP: The Configuration Space

- New Example:
  - One Dimensional Multi-Robot System



Workspace



Configuration Space including time

MRMP
Coupled Planning

- In the new example, the configuration space provided is the *composite* configuration space of both robots.
  - To search this space is to use “coupled” planning.
  - This can be done using any of the algorithms for single robot systems.
  - This can be time-consuming for many robots.

MRMP
Decoupled Planning

- In contrast, one could use a decoupled approach and search individual robot configuration spaces

MRMP
Decoupled Planning

- After developing a trajectory for each robot, the trajectories must be coordinated to make sure there are no collisions.
  - The individual robot trajectory planning can be done using any of the algorithms for single robot systems.
  - Several ways to handle coordination.
  - This can be much quicker than coupled planning.
  - This is generally not complete.

MRMP Overview

- Coupled Planning
  - Complete
  - Slower
  - Possibly Optimal

- Decoupled Planning
  - Not Complete
  - Fast
  - Not Optimal

Multi-Robot Systems: Outline

1. Motivation
2. Application Examples
3. Taxonomies
4. Motion Planning
    1. Coupled and Decoupled Planning
    2. Decoupled Approaches

Decoupled Planning

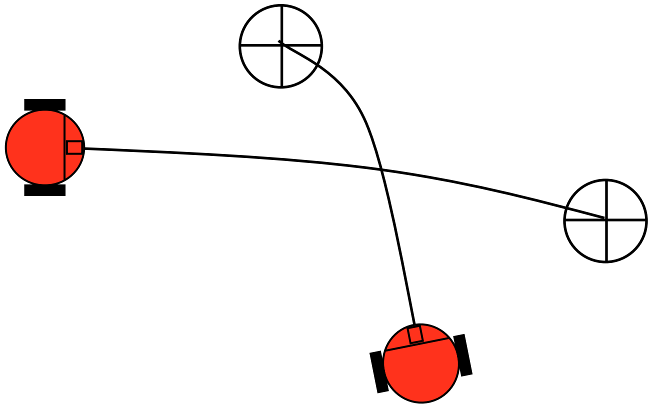
- Decoupled Motion Planning Approaches
  1. Velocity Tuning
  2. Coordination Diagram
  3. Priority based planning
  4. Implementation

Velocity Tuning

- Overview
  1. Construct independent robot paths that are collision free of obstacles.
  2. Modify the velocities of robots following their paths to ensure that robots will not collide.

Velocity Tuning

- Example
  - Despite intersecting, the following pair of paths are velocity tunable.



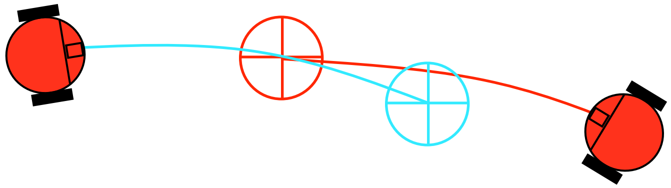
Velocity Tuning

- **Theorem:** Any pair of robot paths can be velocity-tuned to provide collision-free trajectories if the 3 following conditions hold:

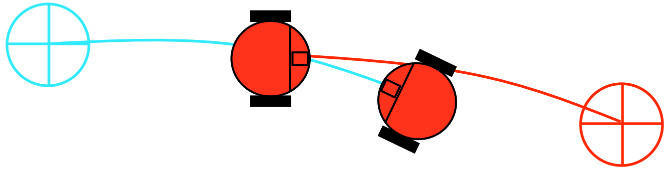
1. Both robots goal locations do not lie on the other robot's path.
2. Both robots start locations do not lie on the other robot's path.
3. Each robot's goal and start locations do not lie on the other robot's path.

Velocity Tuning

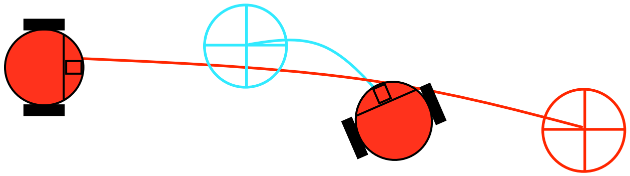
1. Both robots goal locations do not lie on the other robot's path.



2. Both robots start locations do not lie on the other robot's path.

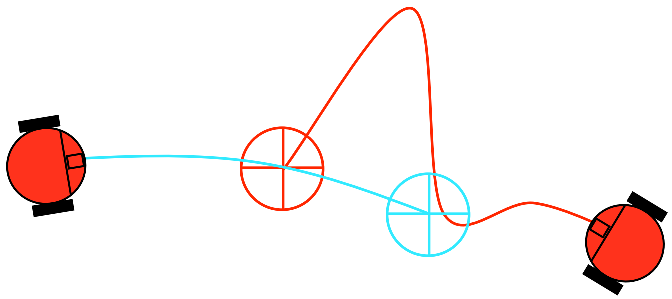


3. Each robot's goal and start locations do not lie on the other robot's path.



Velocity Tuning

- These conditions are only *sufficient*, not *necessary.*
- For example, consider the following trajectory pair in which robot goals do lie on the other robot's path:



Velocity Tuning

- Given these conditions, we have a quick and efficient check to see if trajectories are velocity tuneable.
- We can check if two trajectories are velocity tuneable, then construct appropriate time parameterizations.

Decoupled Planning

- Decoupled Motion Planning Approaches
  1. Velocity Tuning
  2. Coordination Diagram
  3. Priority based planning
  4. Implementation

Coordination Diagram

- Originally presented by O' donell & Lozano-Perez:

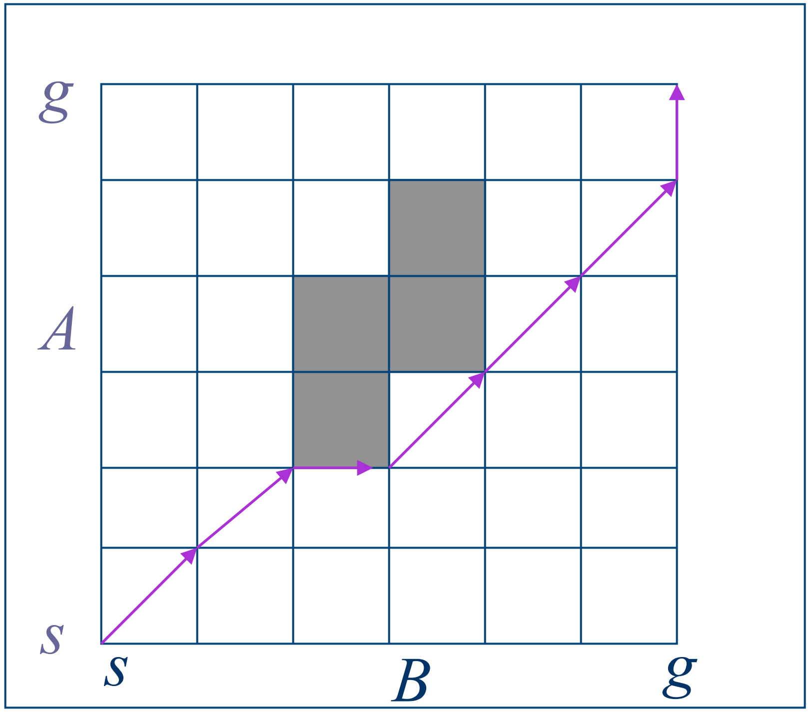
  "Deadlock-Free & Collision-Free Coordination of Two Robot Manipulators"

Coordination Diagram

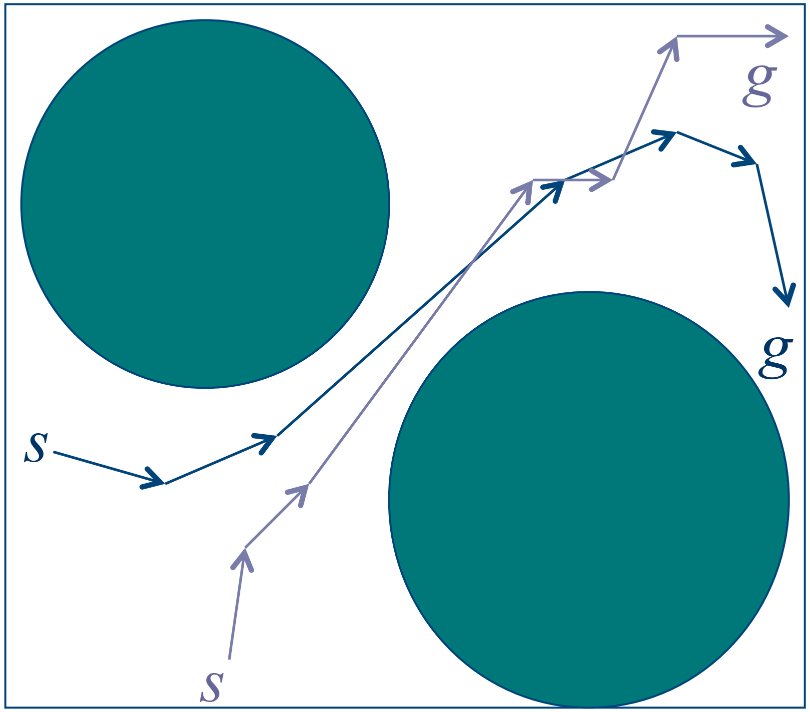
- Task:
  - Coordinate trajectories of 2 robots
- Method:
  - Plan a path for each robot independently
  - Let the path comprise of many path segments
  - Coordinate asynchronous execution of the path segments
- Problems with Coordination:
  - Avoid collisions and “deadlock”

Coordination Diagram

- Task Completion Diagram with "Greedy" algorithm

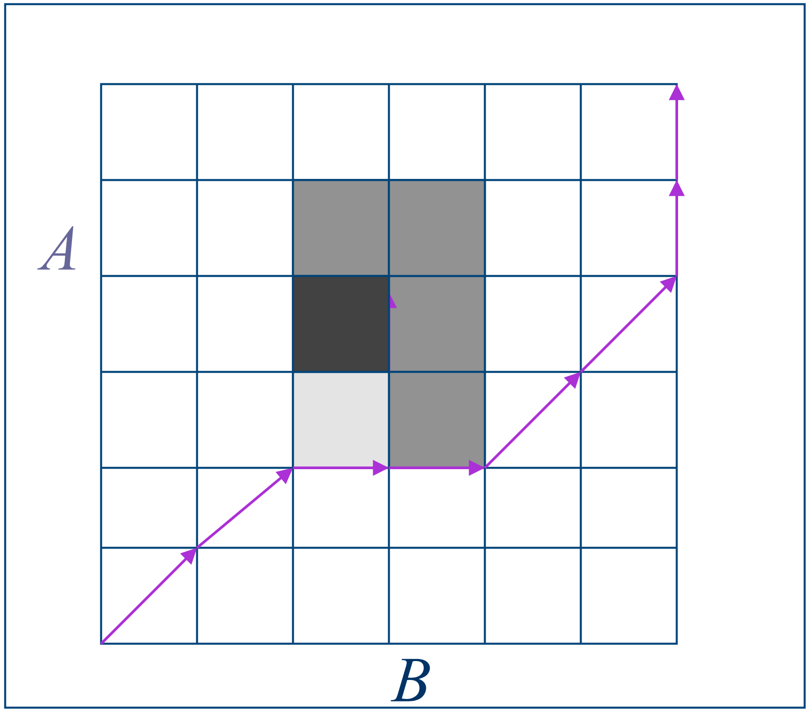


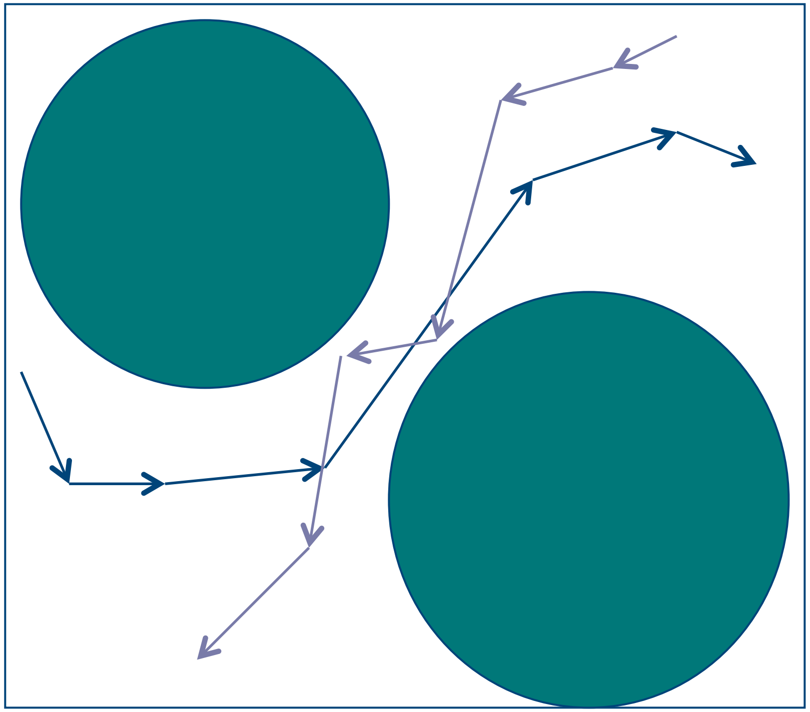
- Sample path



Coordination Diagram

- Removing deadlocks: The *SW-closure*





Coordination Diagram

- Remarks:
  - Removed Deadlock for completeness
  - Increased parallelism for optimality
  - Can we plan for n > 2 robots?

Decoupled Planning

- Decoupled Motion Planning Approaches
  1. Velocity Tuning
  2. Coordination Diagram
  3. Priority based planning
  4. Implementation

Priority Based Planning

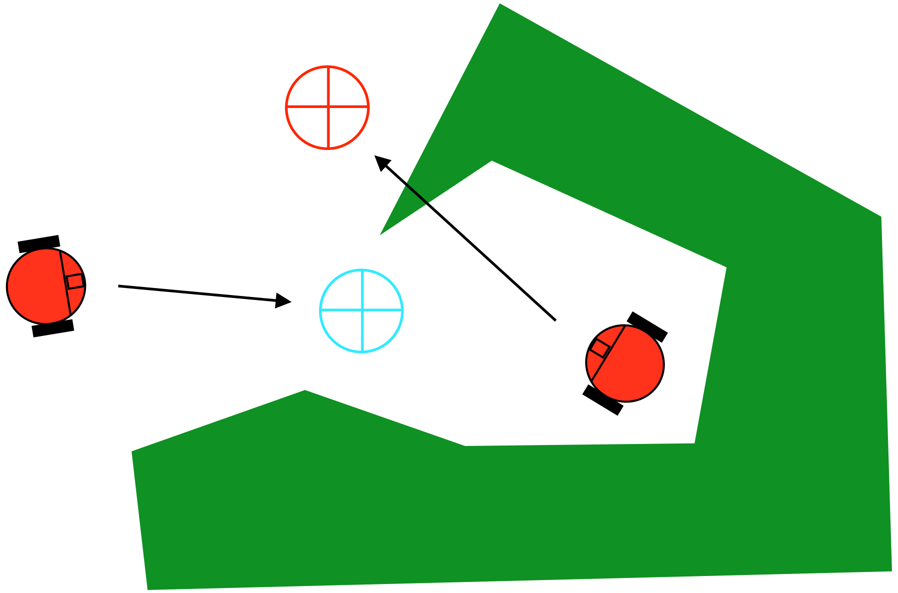
- Robots sequentially construct trajectories.
- As each robot constructs its trajectory, it will use previously constructed trajectories as obstacles to avoid.

Priority Based Planning

- Example: Three robots where robot 0 has highest priority and robot 2 has the lowest:
  - Construct robot 0’s trajectory.
  - Construct robot 1’s trajectory, considering robot 0 as an obstacle to avoid.
  - Construct robot 2’s trajectory, considering robot 0 and robot 1 as obstacles to avoid.

Priority Based Planning

- The priority order is of critical importance
  - For example: inside robot *needs* priority

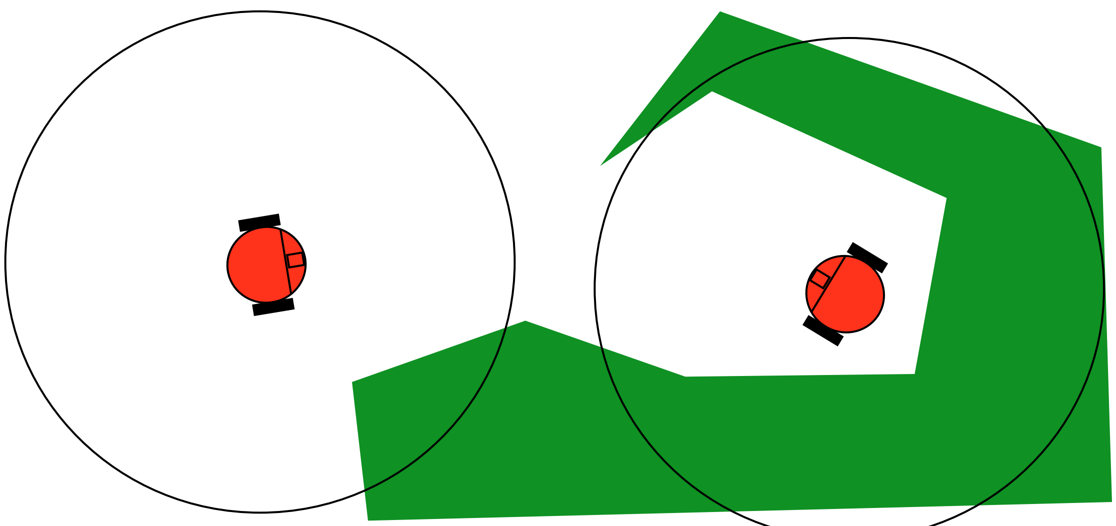


Priority Based Planning

- Static vs. Dynamic Priority Systems
  - Static: priorities stay constant over time.
  - Dynamic: priorities change over time, either to reflect each individual robot's current value to a mission, or the degree of planning difficulty.

Priority Based Planning

- Determining priorities dynamically
  - Can determine each robot's degree of planning difficulty based on the amount of occupied space surrounding the robot.



Priority Based Planning

- Centralized Case: in central planner

```
for i=0..numRobots
  assign robot i priority number p[i]
  where p is an integer

for i=0..numRobots
  construct traj for robot p[i], using
  robots p[0]..p[i-1] as obstacles to
  avoid
```

Priority Based Planning

- Decentralized Case: for robot i

```
Broadcast robot i's priority bid
Receive priority bids
Determine robot i's priority
Receive traj's from robots of higher
  priority
Construct traj using received robots
  traj's as obstacles to avoid
Broadcast trajectory to other robots of
  lower priority.
```

Priority Based Planning

- Simulations
  - Vary number of robots, static obstacles, & dynamic obstacles.
  - Randomly generate start/ goal configurations
  - Use a Probabilistic Road Map (PRM) Planner to construct trajectories.

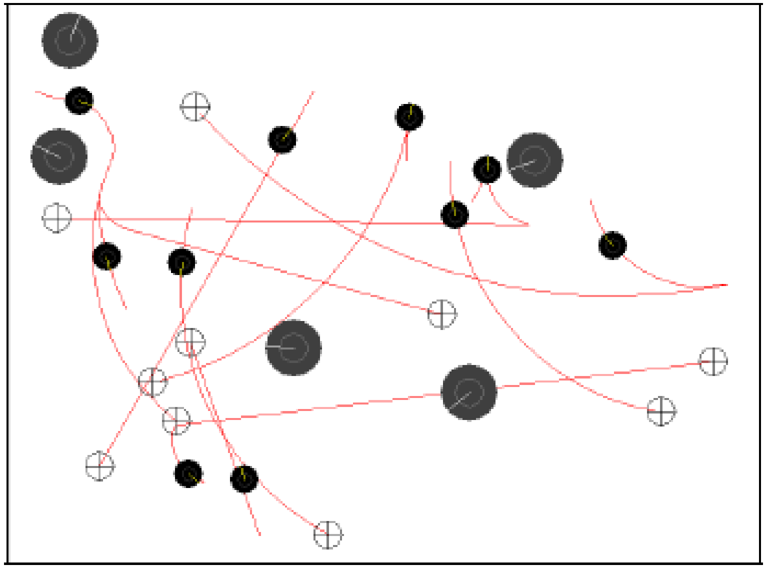


Figure 9b) - Simulation at time T2:
Rovers after constructing their first plans

Priority Based Planning

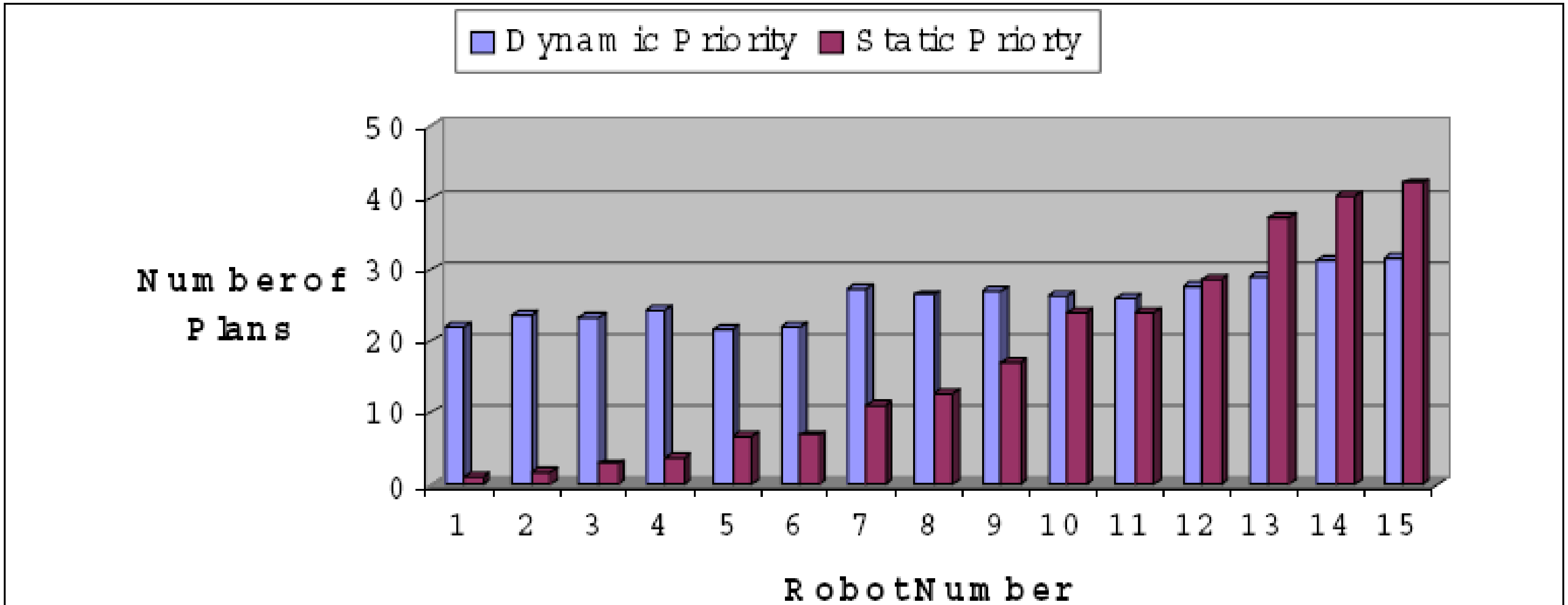
- Results
  - On-line planning can be achieved.
  - Obstacles are harder to avoid than robots.

Table 1 - Simulation Results

Experiment Set	1	2	3
Robots	5	10	15
Stationary Obstacles	5	5	0
Moving Obstacles	5	0	0
Average Plan Time (ms)	38.57	96.36	4.66
Average Maximum Plan Time (ms)	102.47	276.52	44.44

Priority Based Planning

- Results
  - More robots doing more planning
  - Reduced max. number of robots planned for.



Priority Based Planning

- Results
  - Dynamic Priority System decreases planning times because trajectories need to consider fewer robots.

Table 2 – Comparing Static and Dynamic Priority Systems

Experiment Set	Static Priority System	Dynamic Priority System
Robots	15	15
Stationary Obstacles	0	0
Moving Obstacles	0	0
Average Plan Time (ms)	4.66	1.03
Average Maximum Plan Time (ms)	44.44	15.06