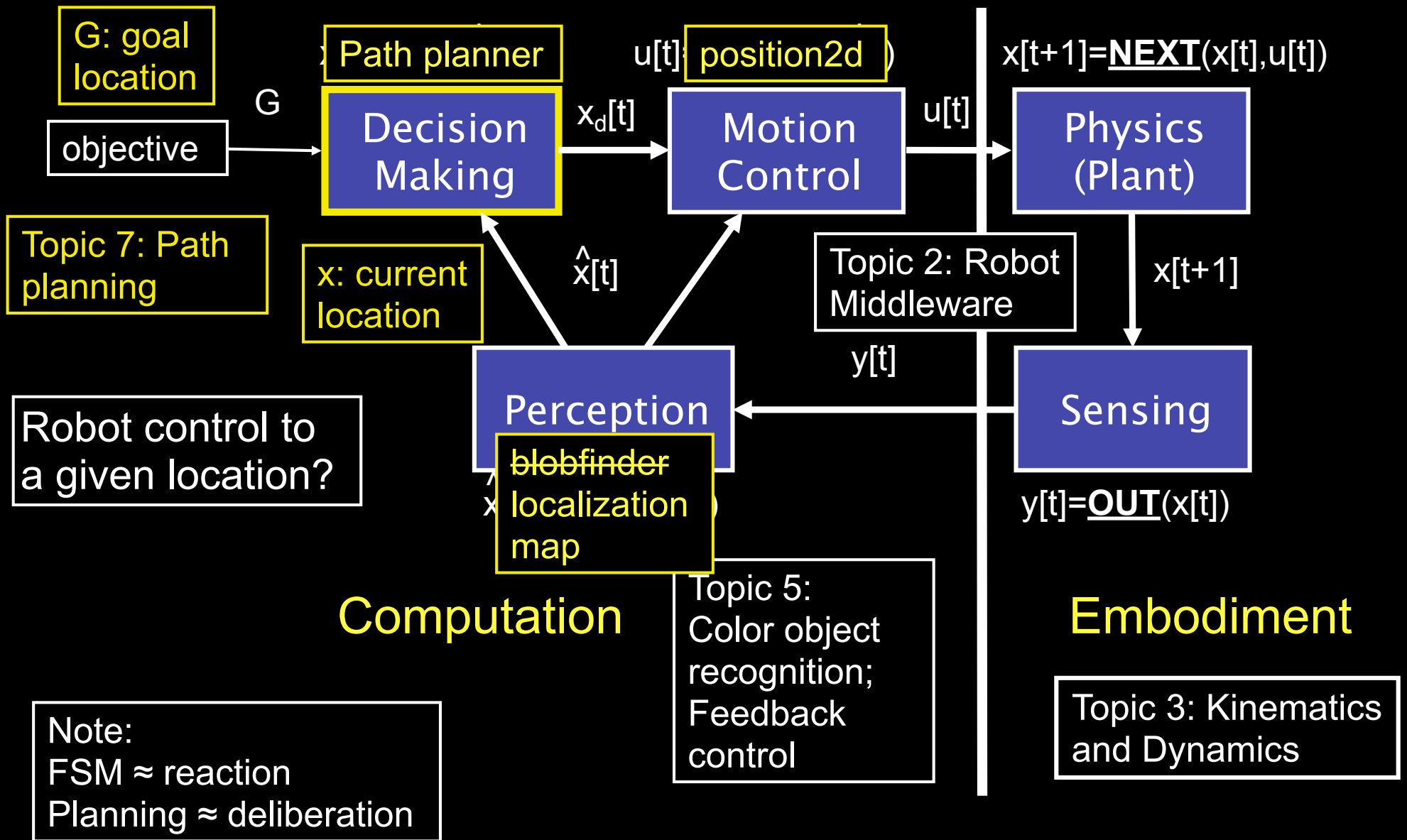


robot control loop

- someone please sketch on the board

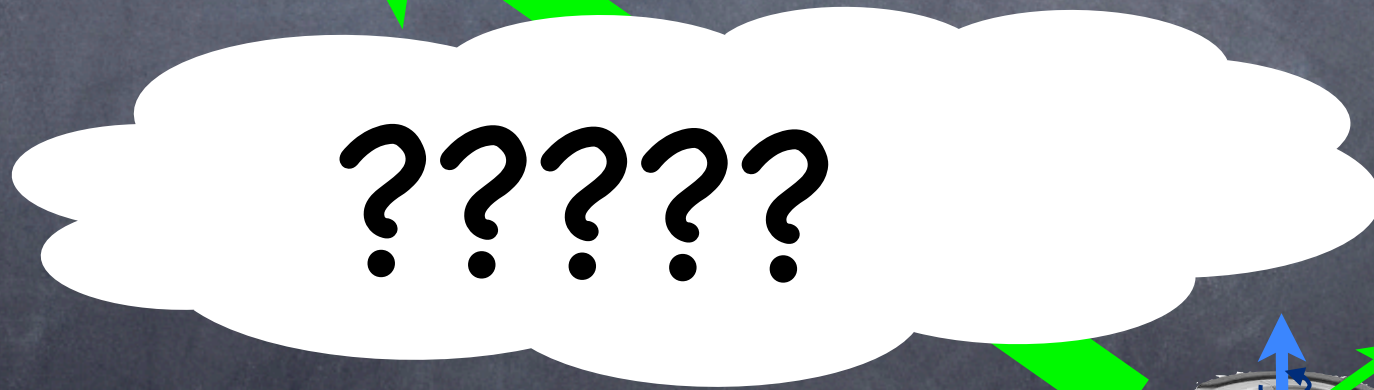
The Robot Control Loop



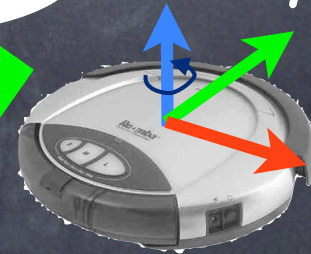
How to get from A to B?

B: Goal

Consider arbitrary
start and goal locations

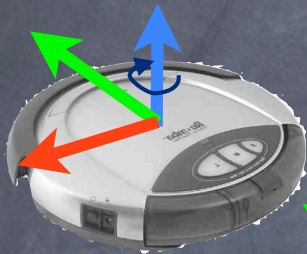


A: Start



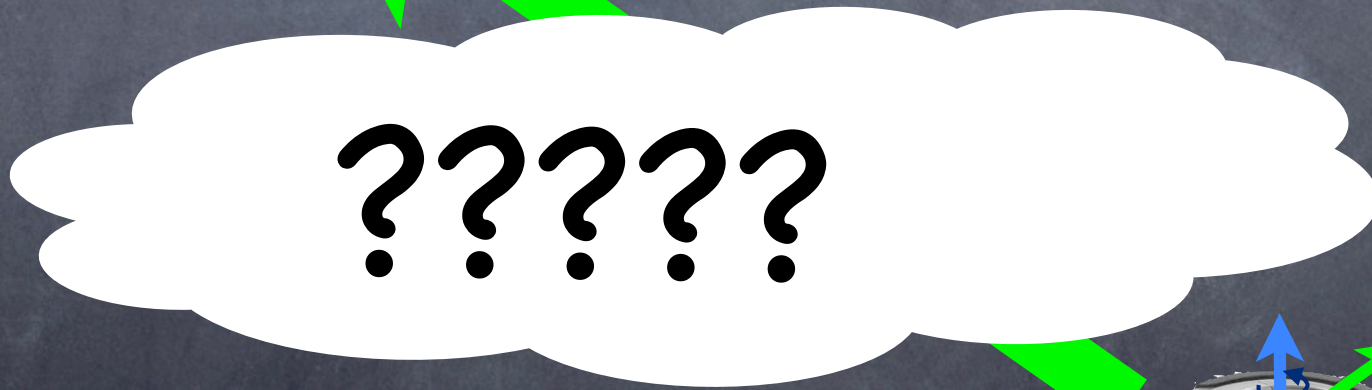
How to get from A to B?

B: Goal

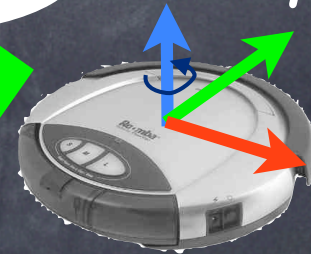


FSM (reactive)

Path planning (deliberative)



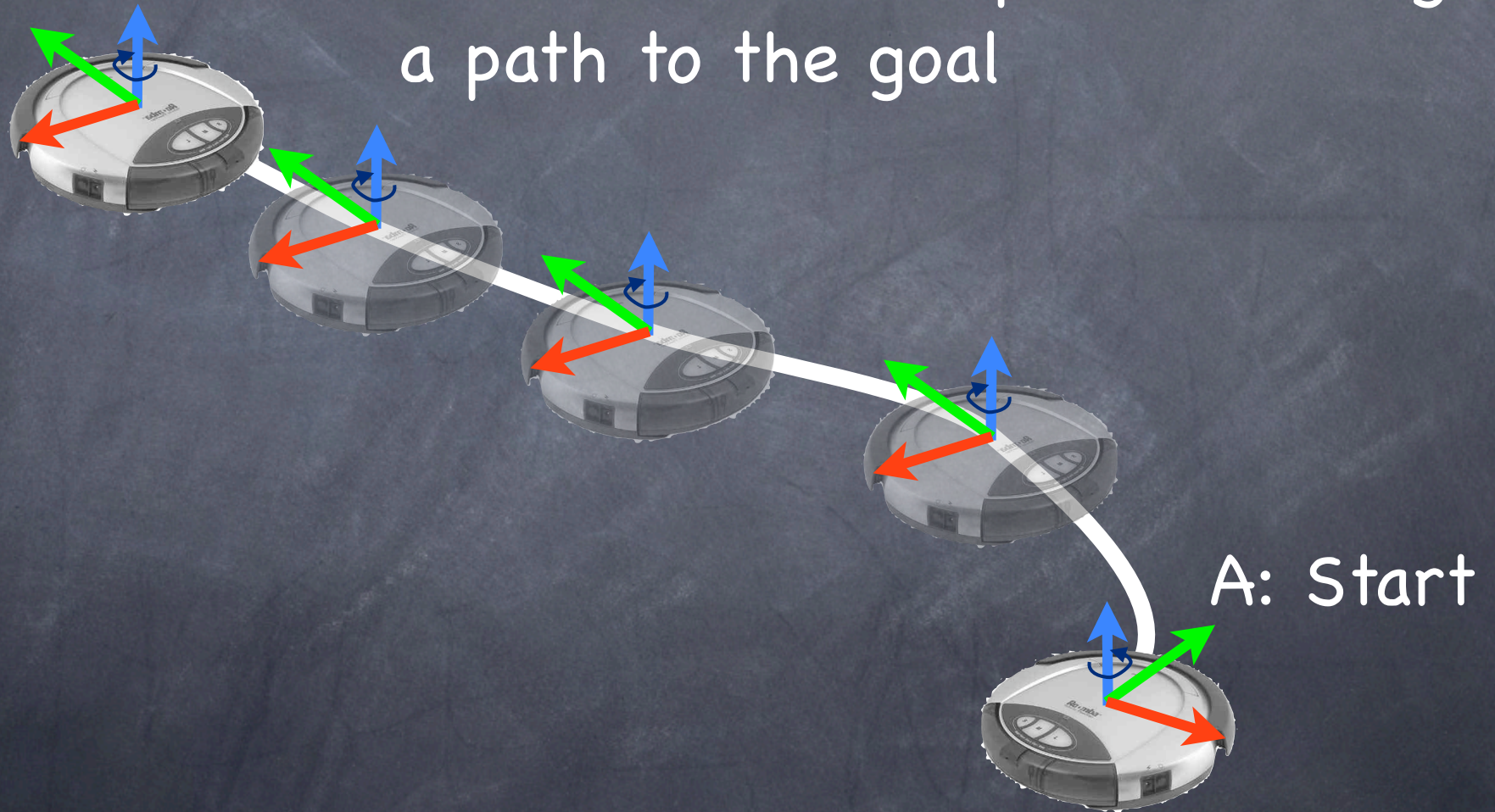
A: Start



Path Planning

B: Goal

Find intermediate poses forming a path to the goal

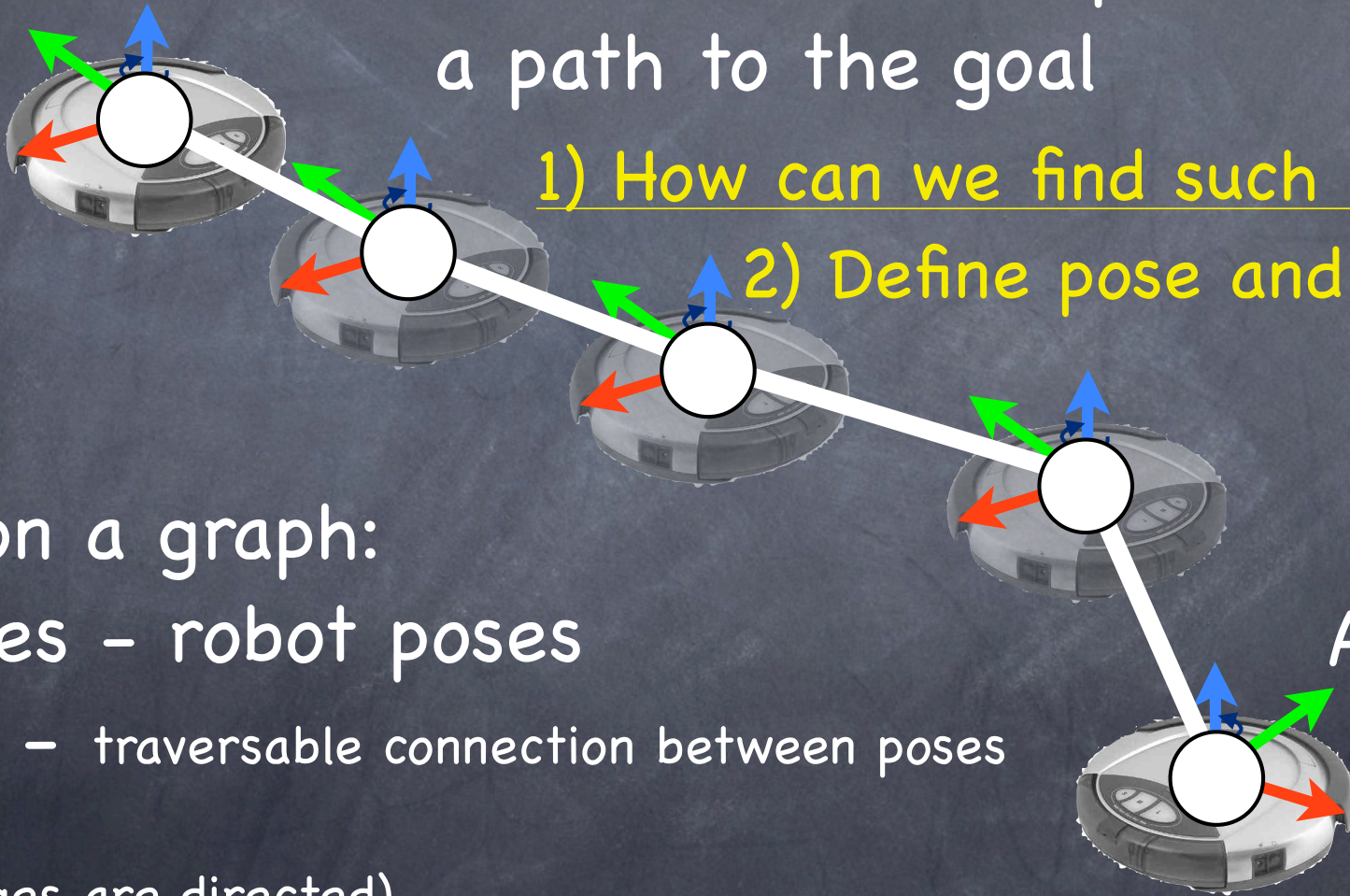


A: Start

Path Planning

B: Goal

Find intermediate poses forming a path to the goal



1) How can we find such paths?

2) Define pose and controls?

Path on a graph:

vertices - robot poses

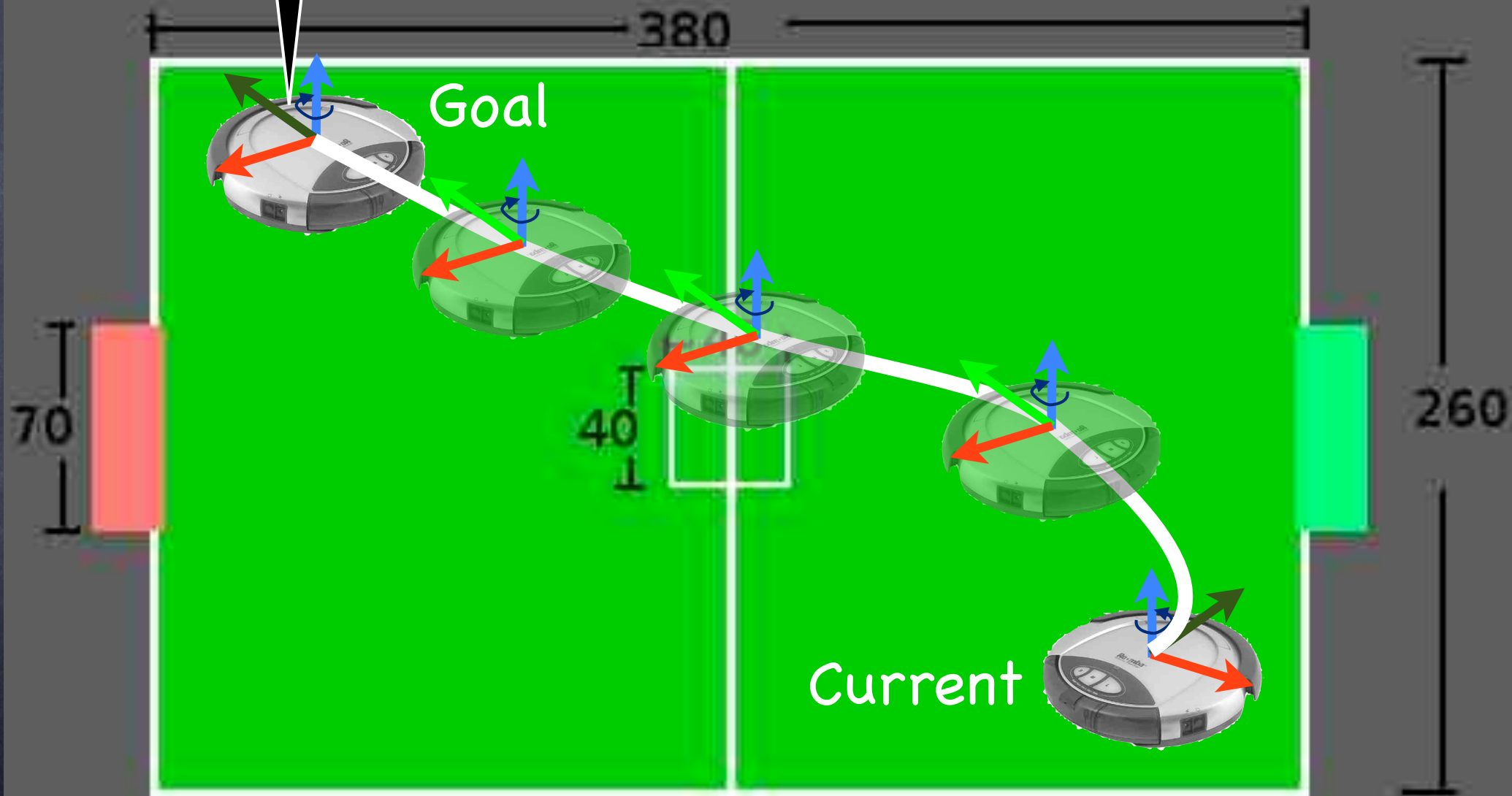
edges - traversable connection between poses

(note edges are directed)

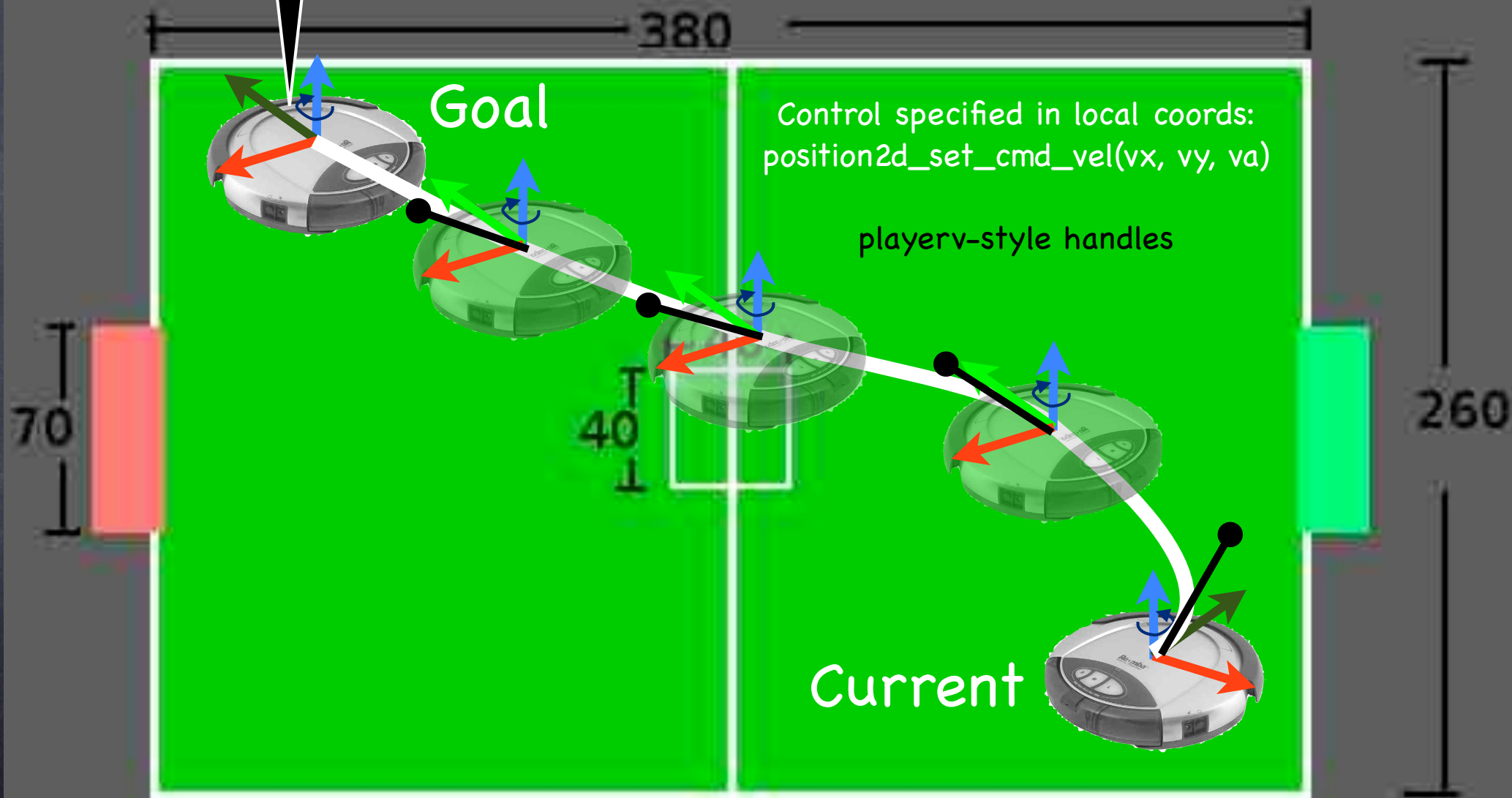
A: Start

Warning: illustrations are approximate

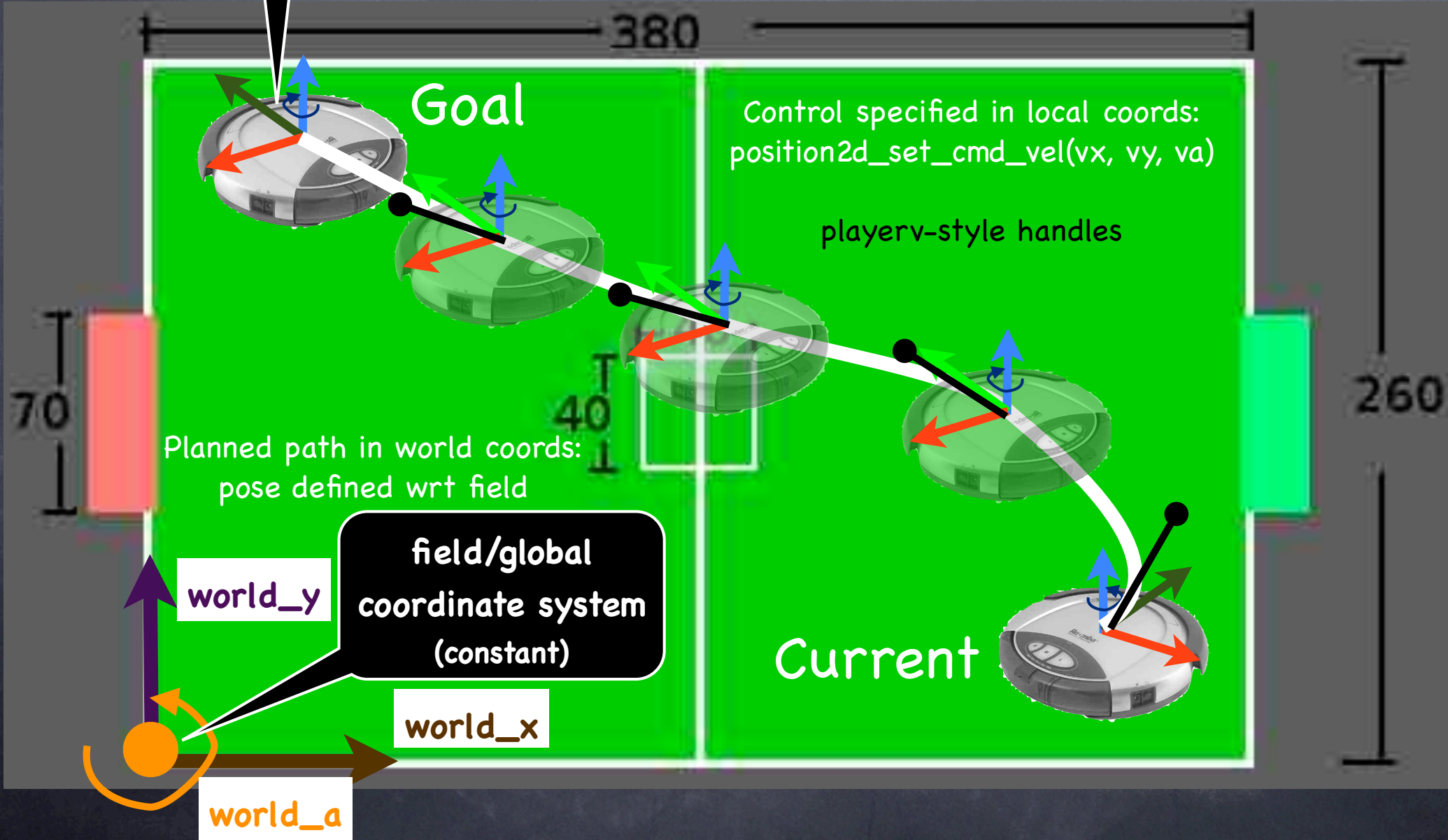
robot's local
coordinate system
(moves with robot)



robot's local
coordinate system
(moves with robot)



robot's local coordinate system (moves with robot)



Control specified in local coords:
`position2d_set_cmd_vel(vx, vy, va)`

playerv-style handles

Planned path in world coords:
pose defined wrt field

field/global coordinate system (constant)

world_y

world_x

world_a

Current

Goal

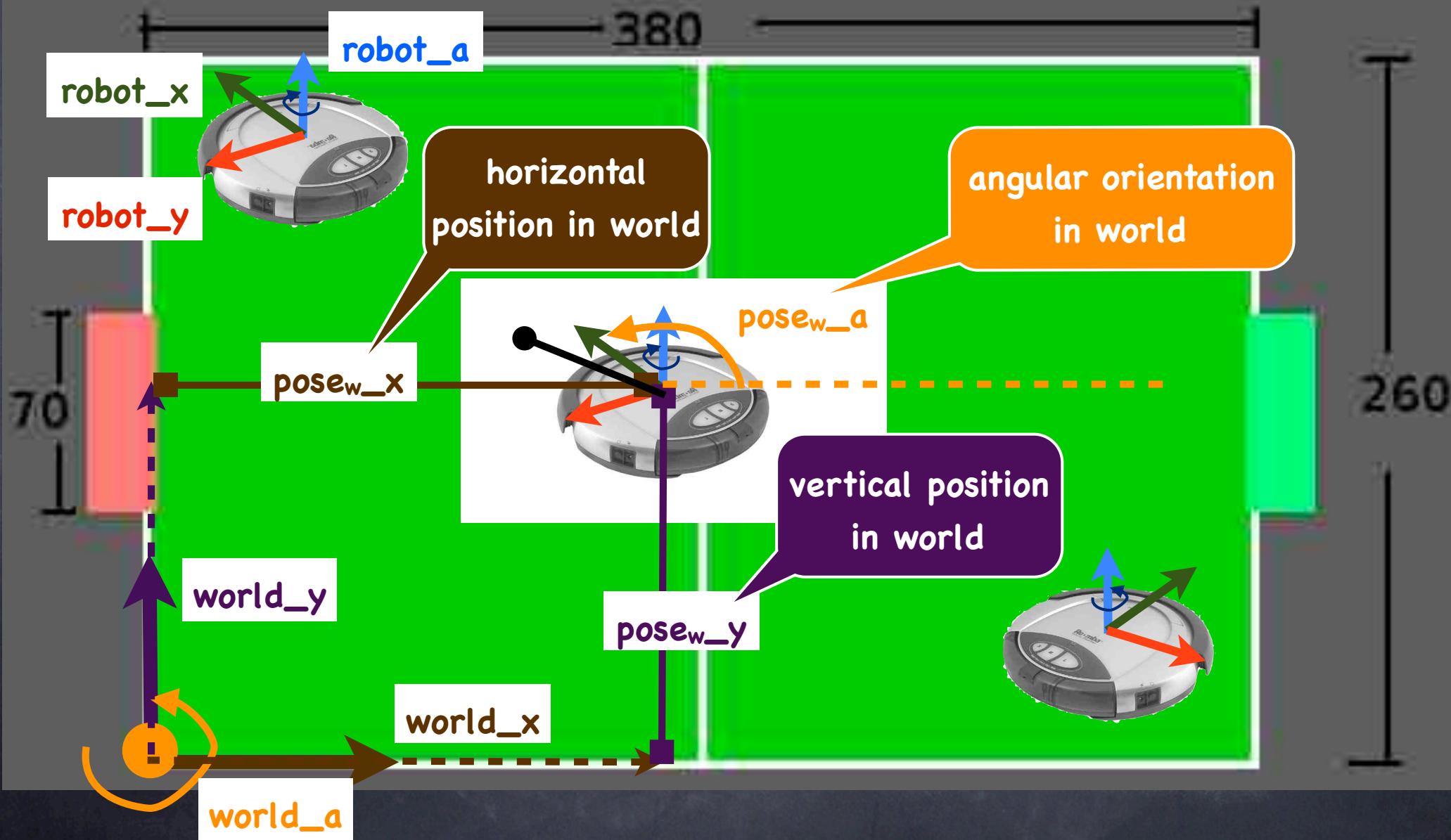
380

260

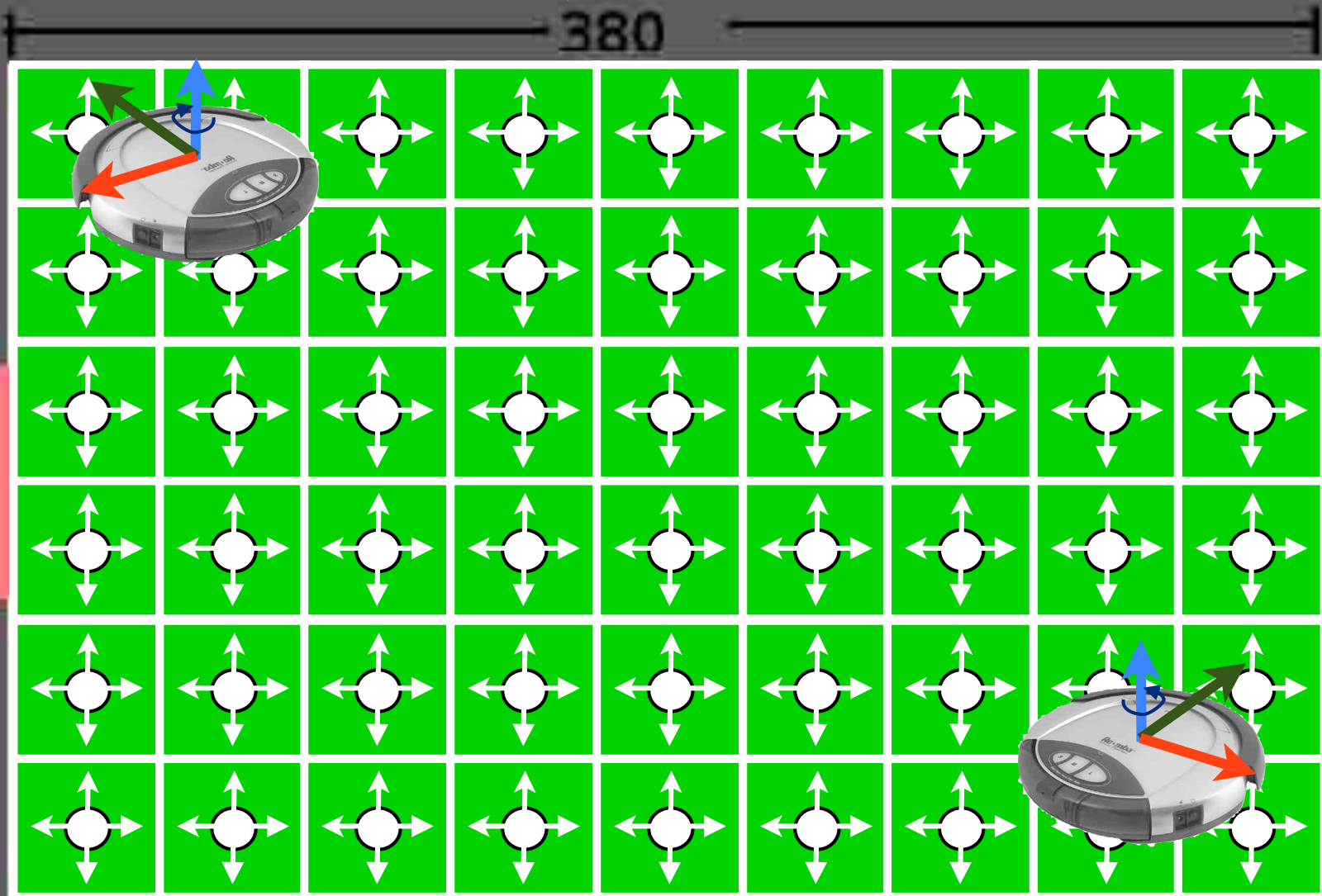
70

40

Poses defined w.r.t. the field in 3 dimensions:
2 position DOFs, 1 rotational DOF



Consider all possible poses as graph vertices
Edges connect adjacent (traversable) poses
How to find a valid path?



Approaches to path planning

- Search (fixed graph)

- DFS, BFS, Dijkstra, A*

What exactly are we searching for?

- Search (explore graph):

- Probabilistic Road Maps

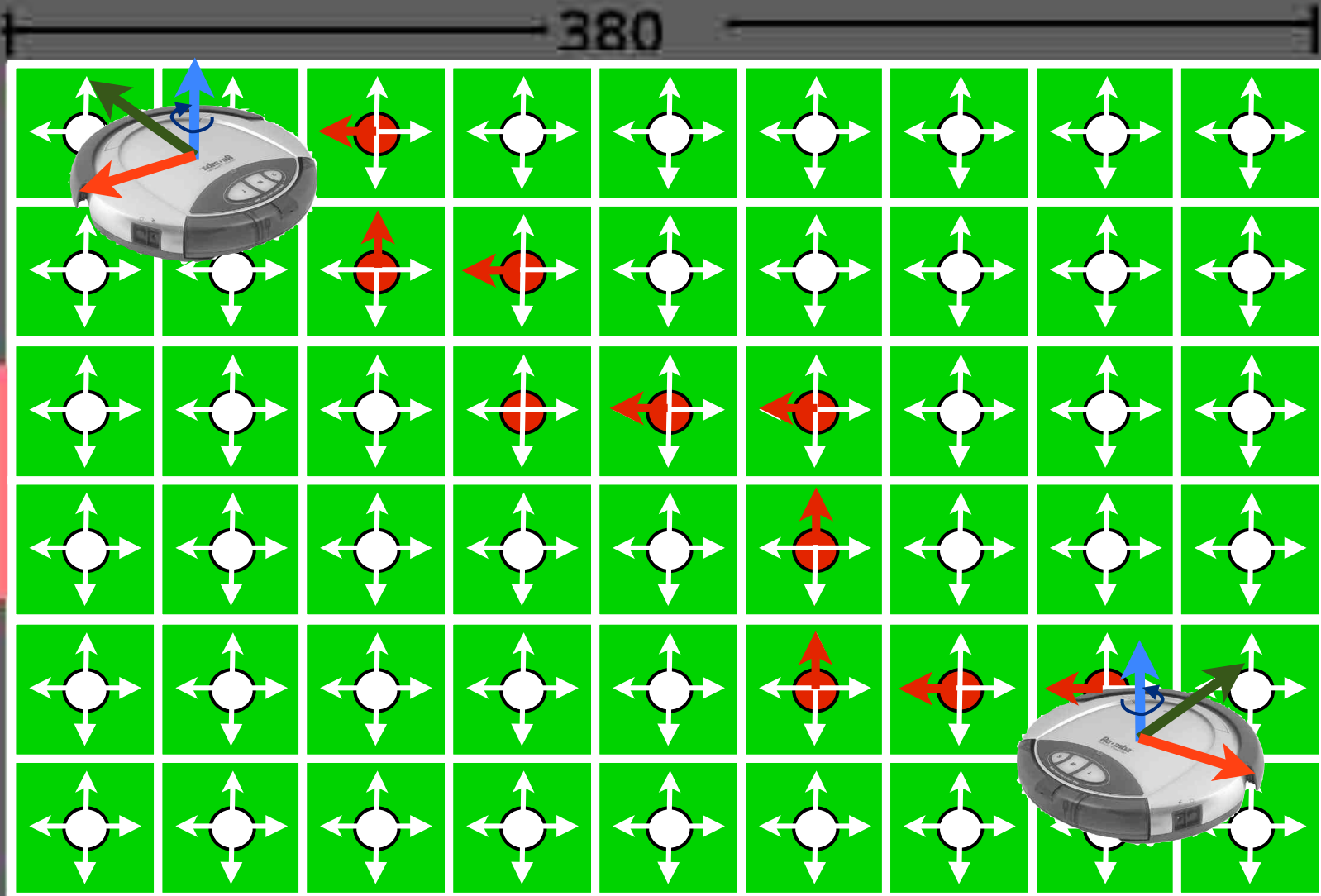
- Rapidly-exploring Random Trees

- Optimization (local search):

- Potential fields, gradient descent

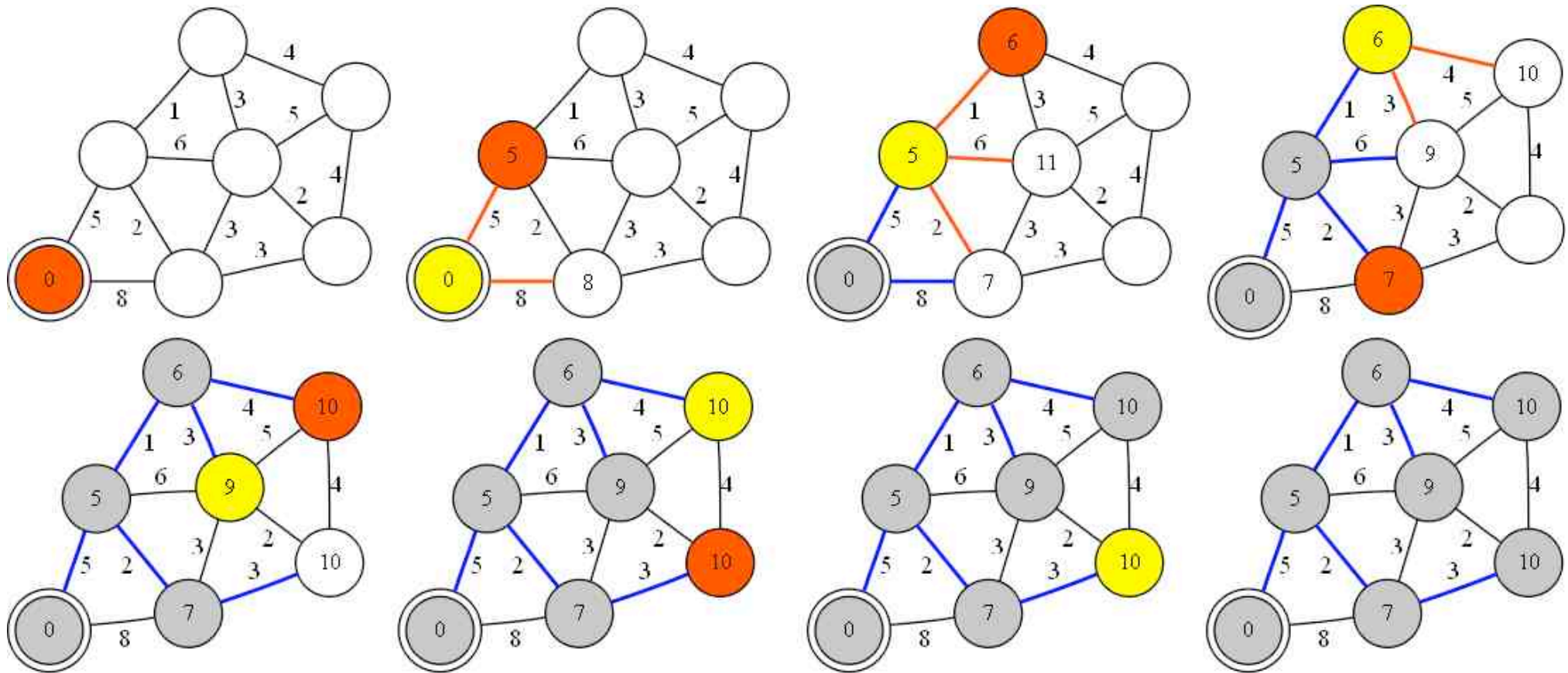
Grid Search

DFS vs. BFS vs.
Dijkstra vs. A*



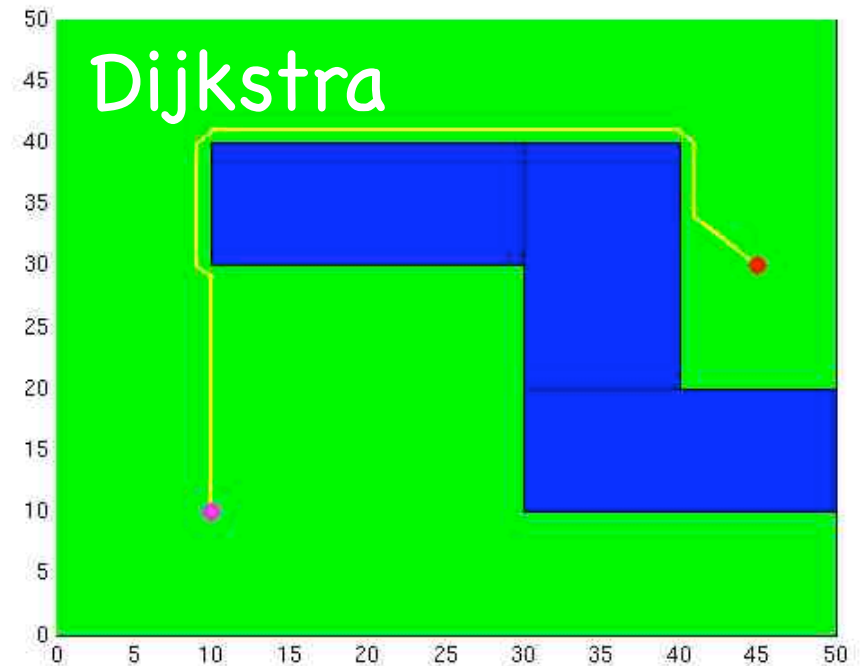
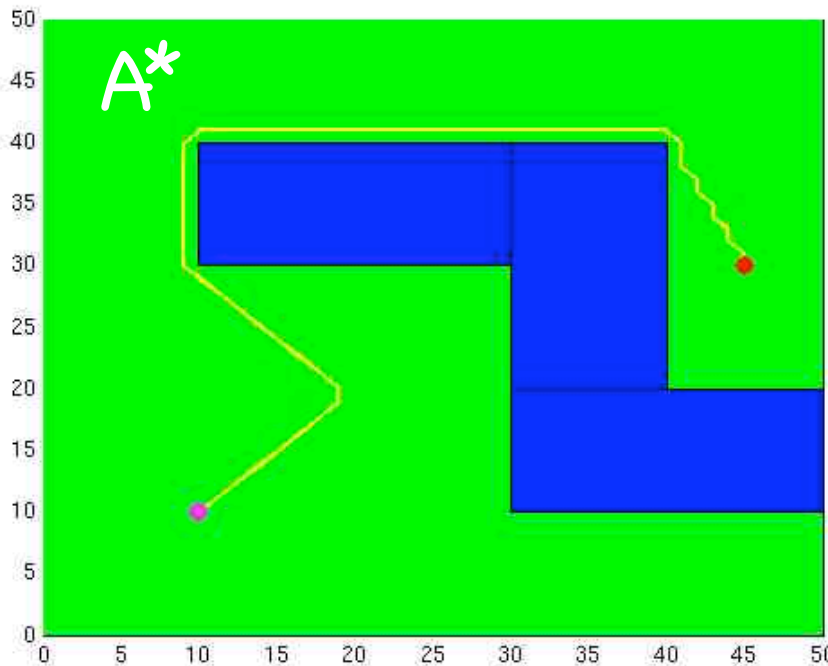
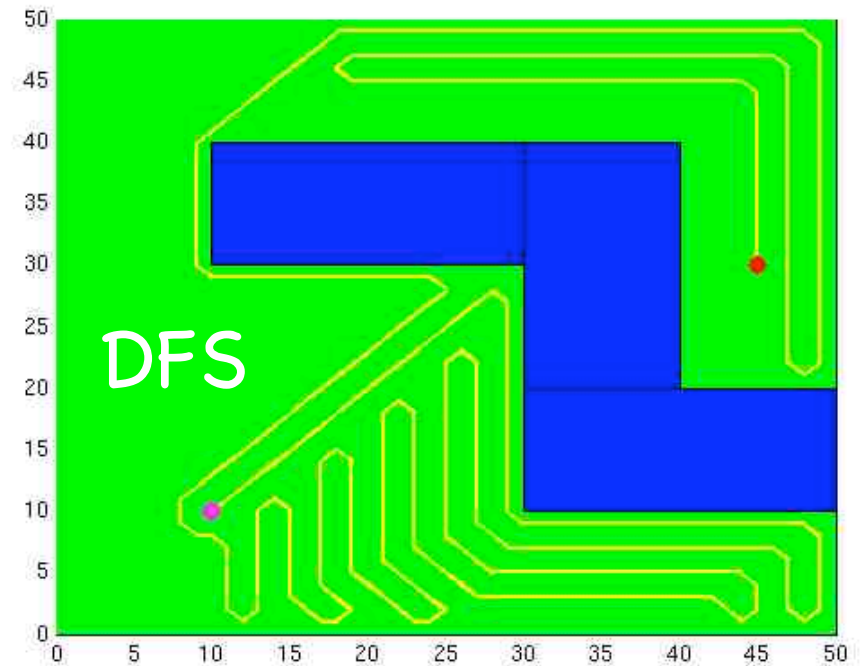
Dijkstra's Algorithm:

Given weighted graph and start node, find shortest path back to start for all nodes



Explore each node one-by-one, calculate path distance from explored tree

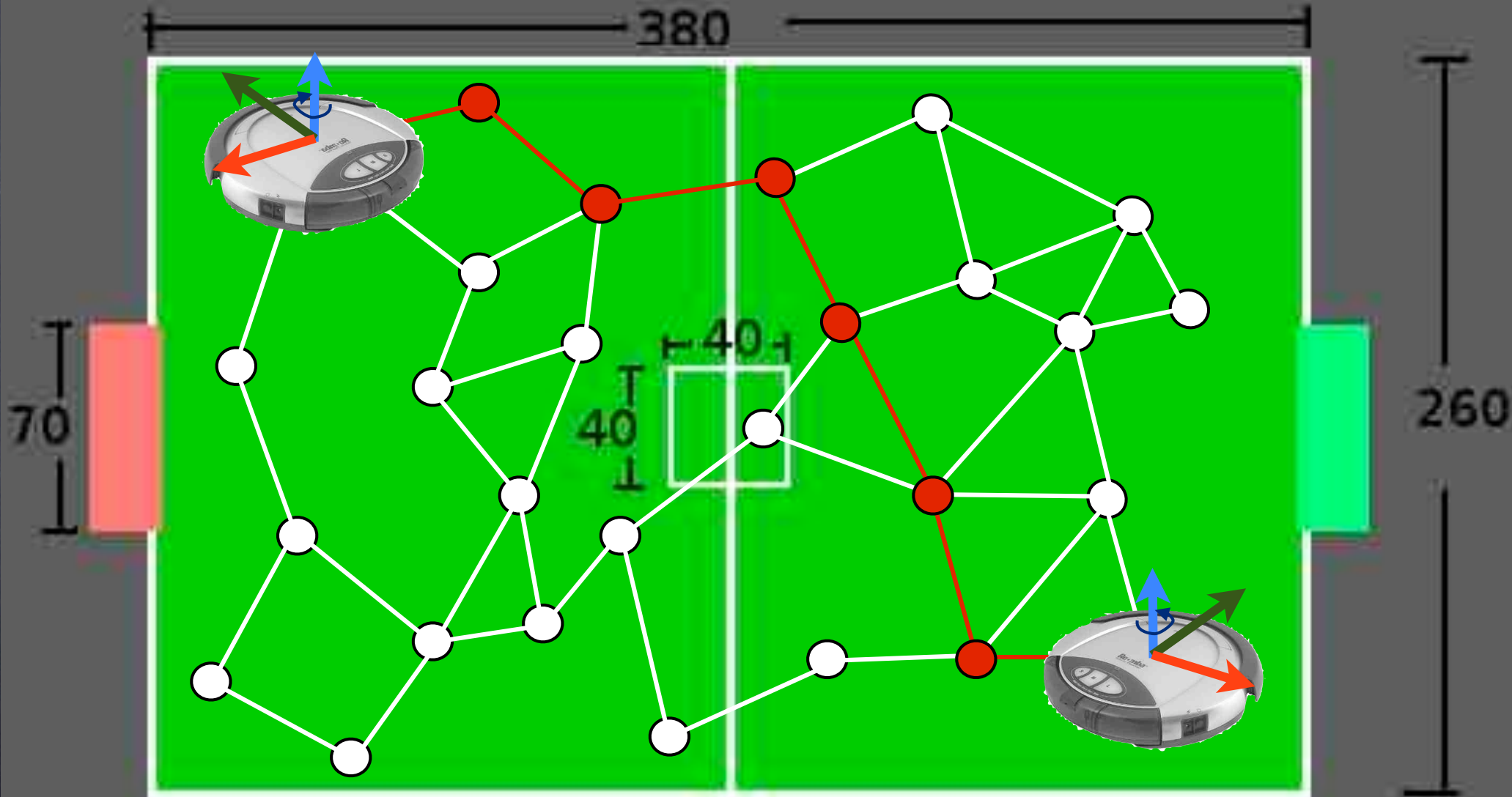
Matlab example:
`/course/cs148/pub/
pathplan_pub.m`



RRT

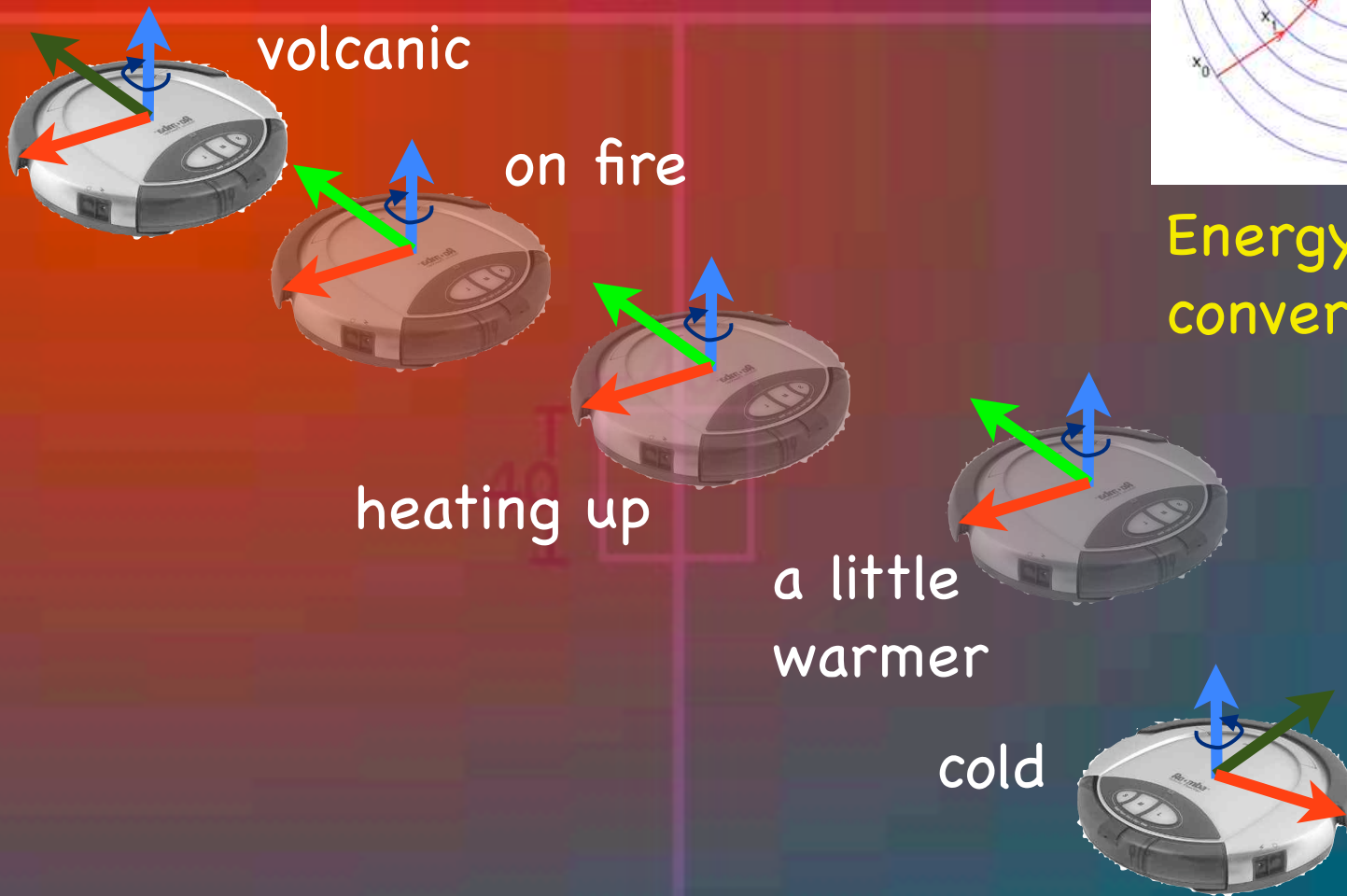
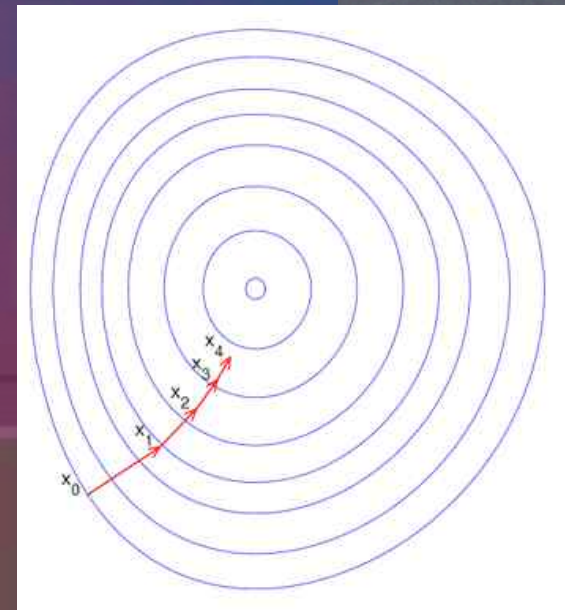
(more detail later)

Explore poses and connectivity;
Find shortest path in built graph



Potential field

(more detail later)



Energy potential converges at goal

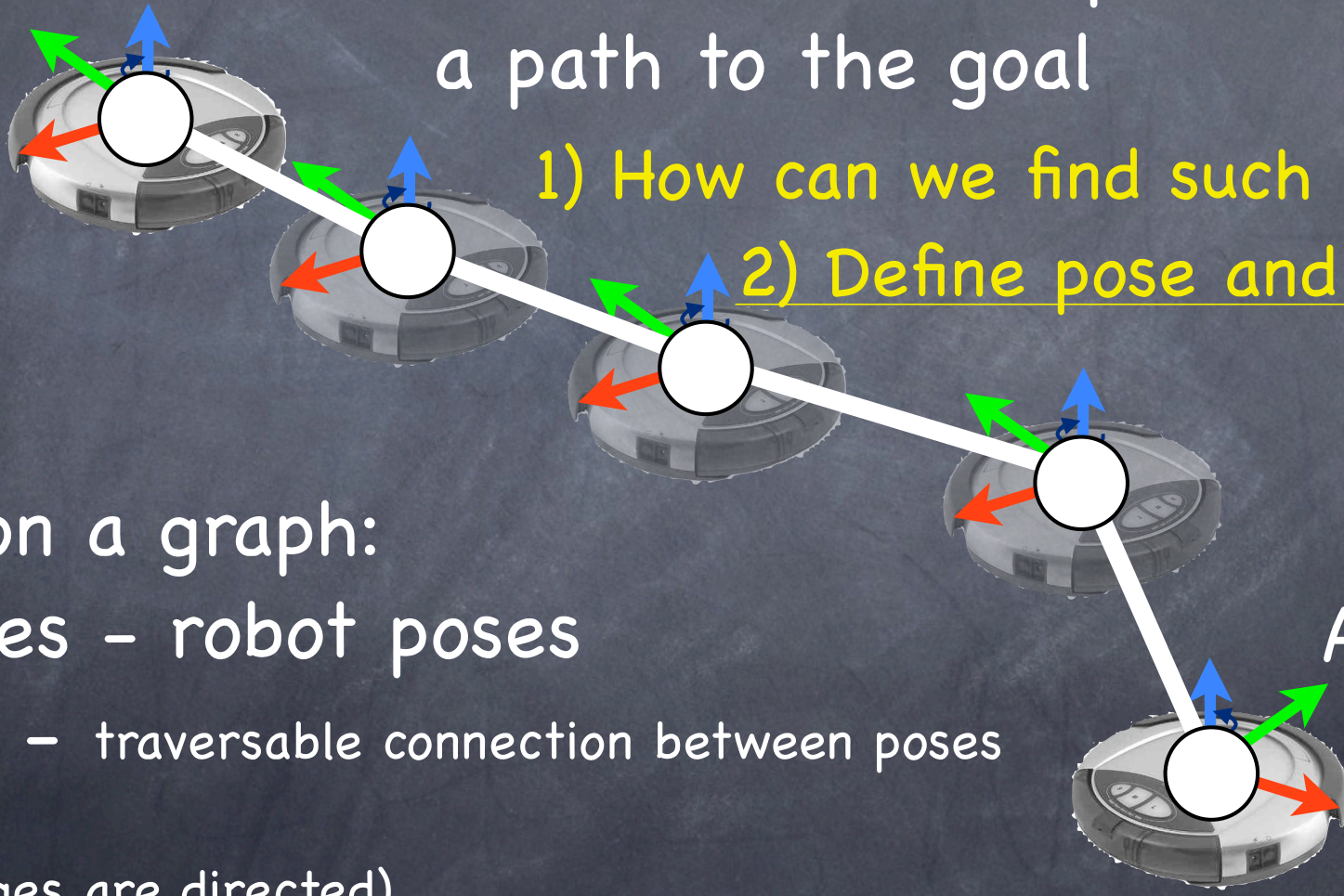
260

Similar to "getting warmer, getting colder" game

Path Planning

B: Goal

Find intermediate poses forming a path to the goal



1) How can we find such paths?

2) Define pose and controls?

Path on a graph:

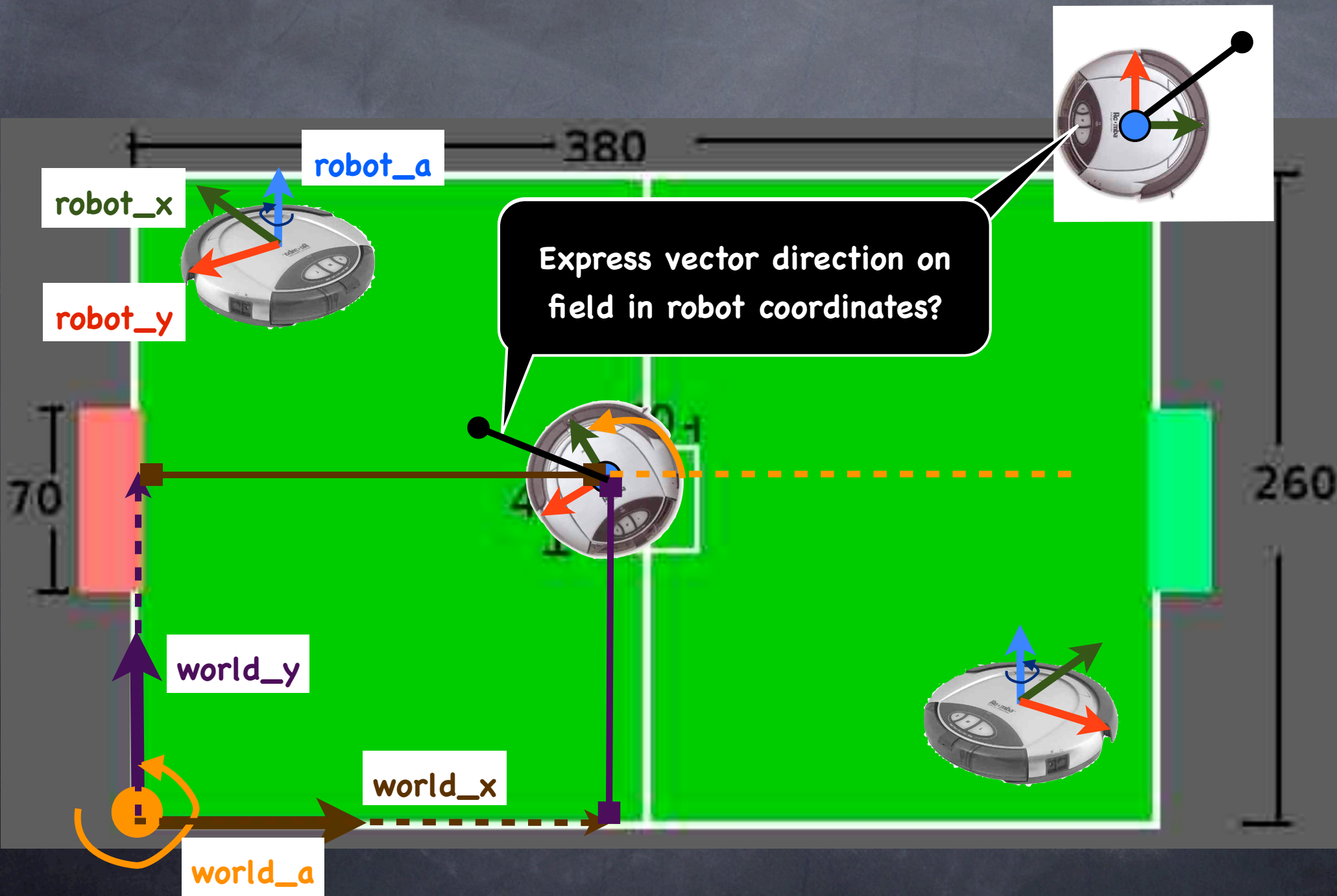
vertices - robot poses

edges - traversable connection between poses

(note edges are directed)

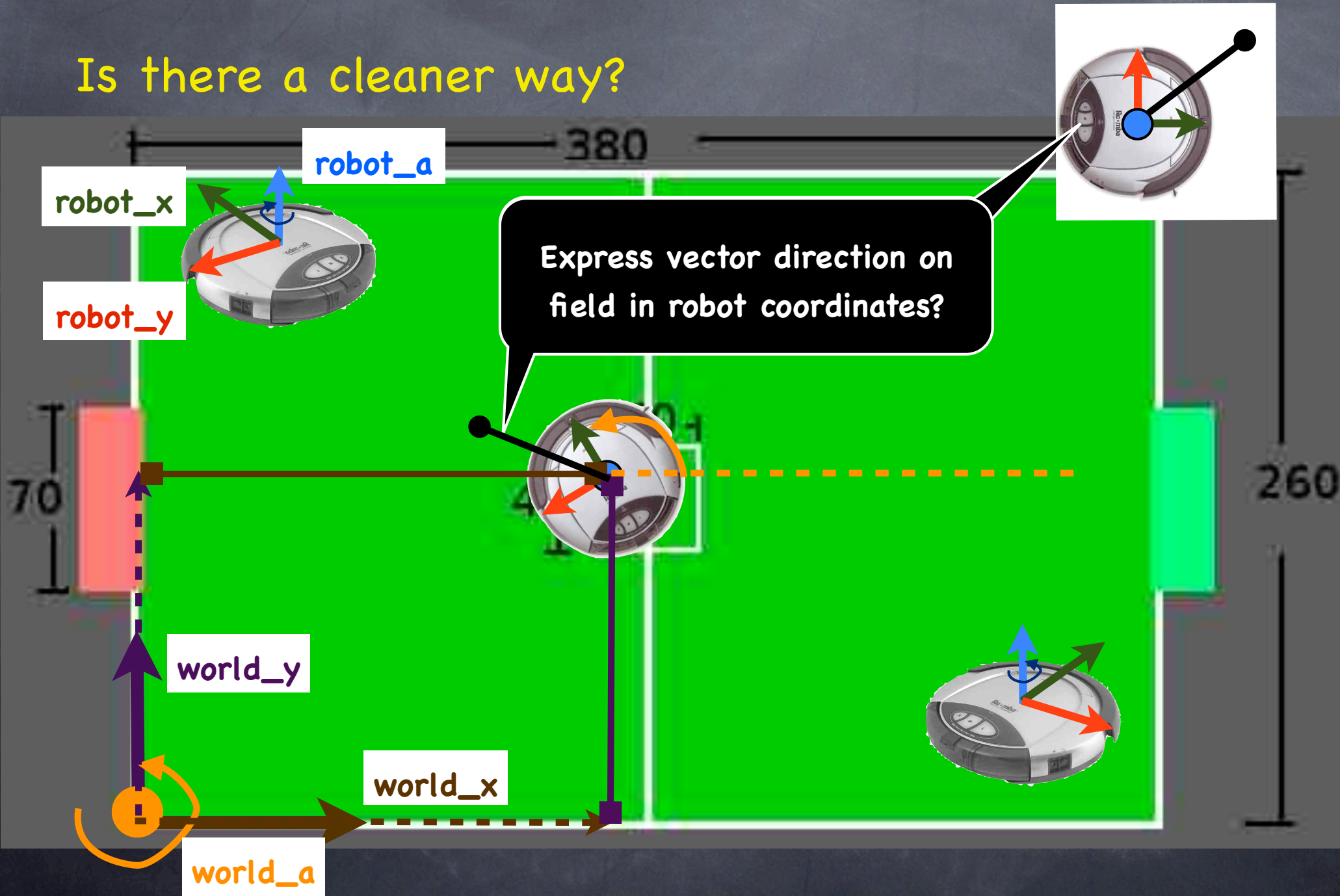
A: Start

Warning: matrix multiplication ahead



Simple answer rotate vector by $-\text{pose}_w_a$

Is there a cleaner way?



asgn3 milestone description

