PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-Based Planning[1]

Aleksandra Faust, Oscar Ramirez, Marek Fiser,
Kenneth Oslund, Anthony Francis, James Davidson, and Lydia Tapia

https://arxiv.org/pdf/1710.03937.pdf

Presented by:
Gaurav Kumar

Extension

Long-Range Indoor Navigation with PRM-RL

By

Anthony Francis, Aleksandra Faust, Hao-Tien Lewis Chiang, Jasmine Hsu, J. Chase Kew, Marek Fiser, and Tsang-Wei Edward Lee,

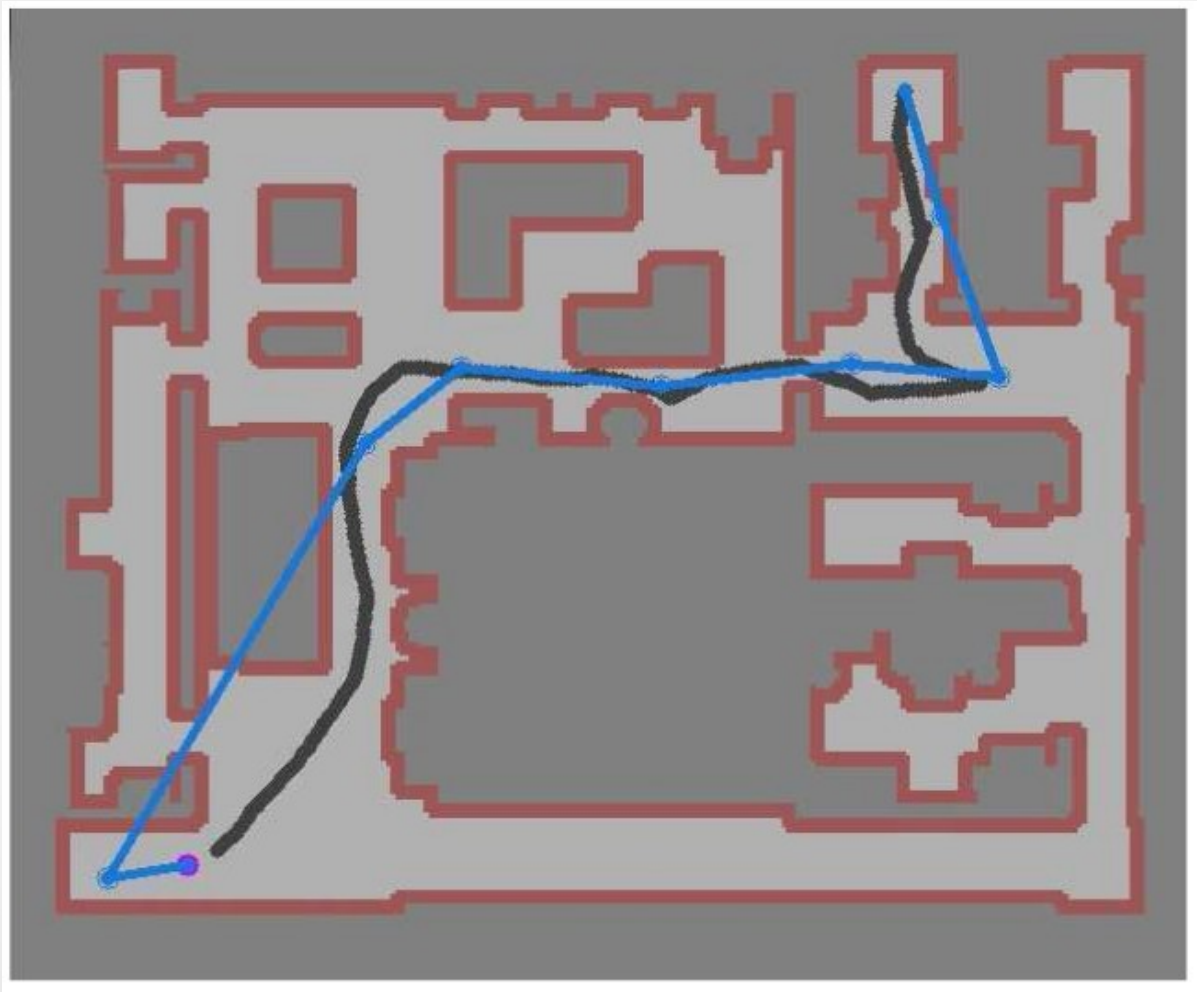
https://arxiv.org/abs/1902.09458

Index

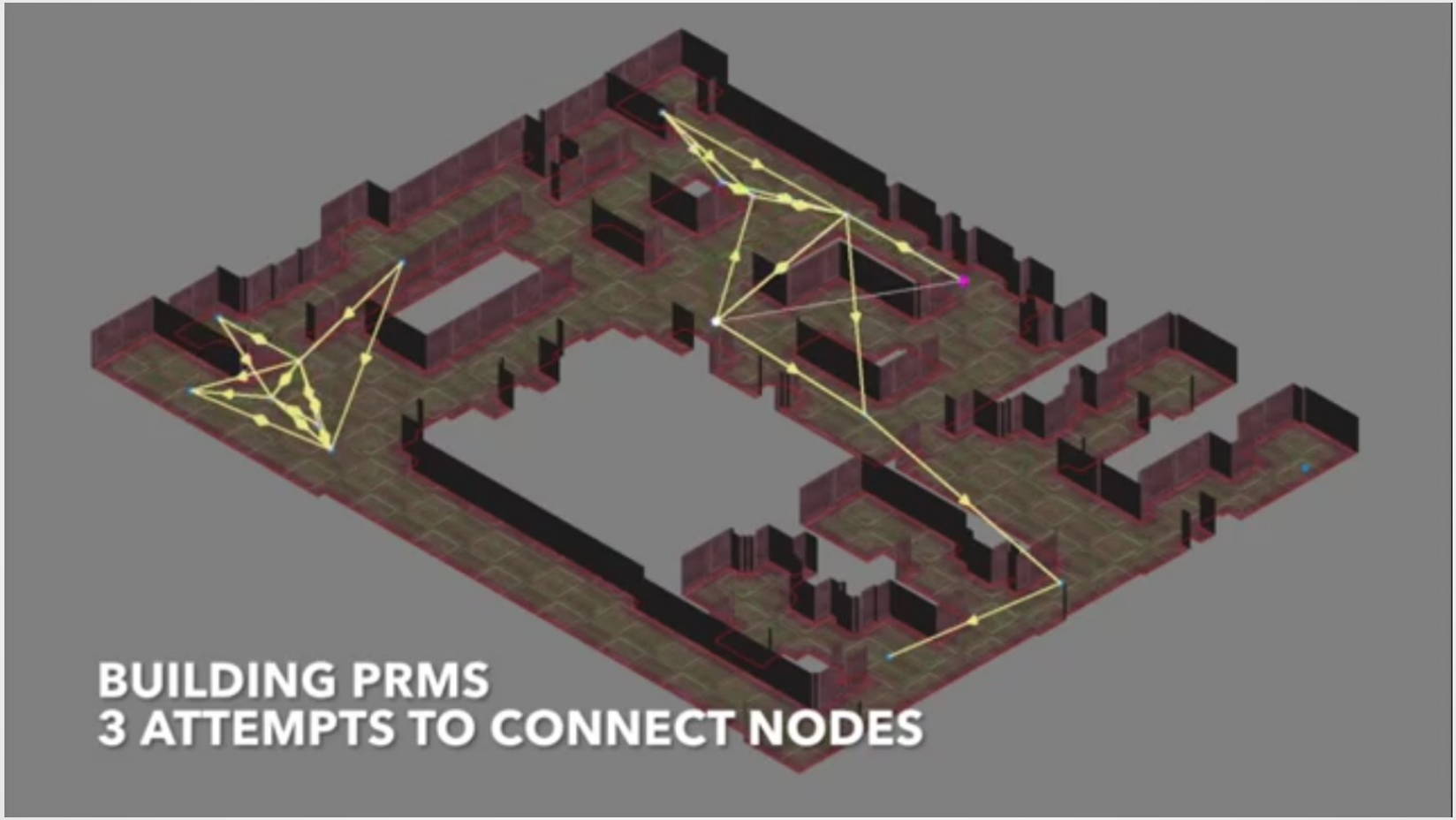
- Introduction
- Sampling-Based Planner
  - Probabilistic Roadmap
- Method (PRM Adaptation for RL)
- Task Environment
- Results

Introduction

Roadmap and local planner



Roadmap and local planner



Idea

- Sampling Based Planner

  Probabilistic Roadmaps[1][3]

  or

  Rapidly Exploring Random Trees

  - Expensive on its own
  - Approximate C-Space Topology via inexpensive local planner

- Reinforcement Learning Based Planners
  - Good at noisy dynamics
  - Fail in long term planning (specially sparse reward)

Idea

- Hierarchical method
  - For long-range navigation task
  - Combine sampling-based planning and RL
- Task division
  - Low-Level RL Controller – Environment Dynamics
  - High-Level PRM Controller – Navigation
- Compared Local planners
  - Straight Line (SL, also called line-of-sight)
  - Reinforcement Learning (RL)

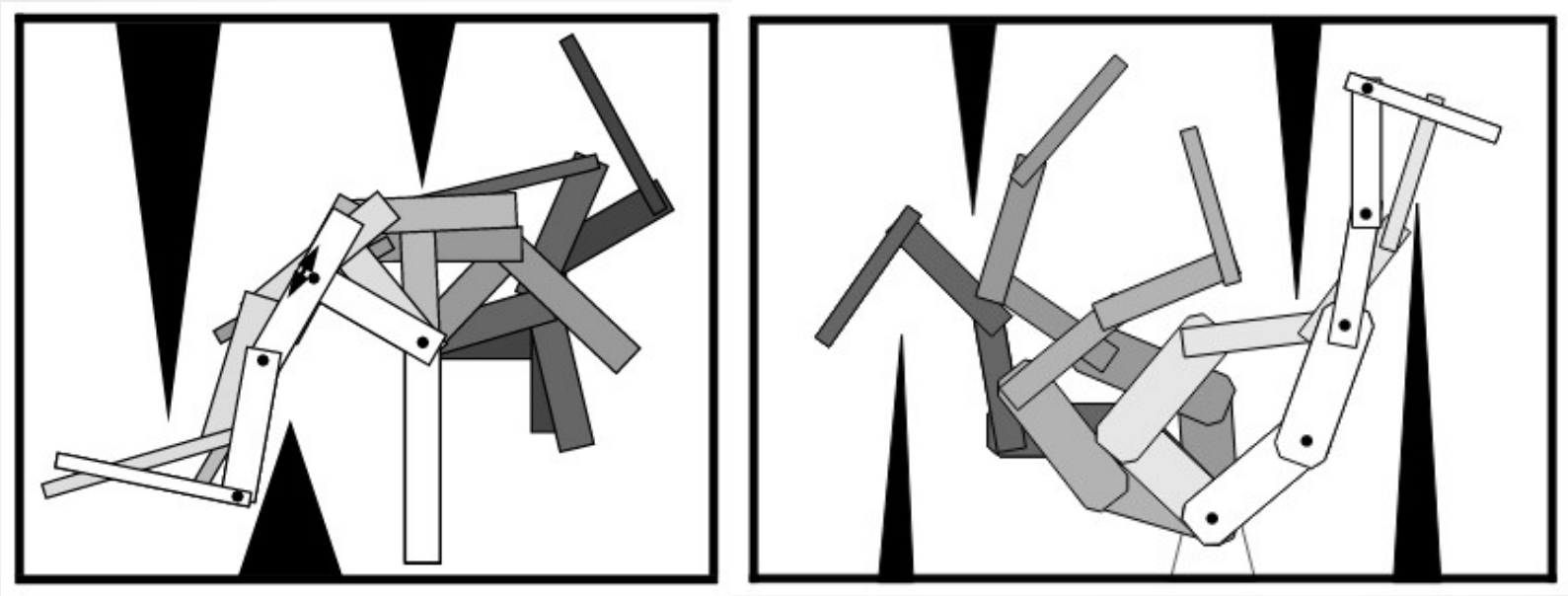
Sampling Based Planner (Probabilistic Roadmaps[3])

Why sampling based planner? [2]

- Grid based planner
  - Partition C-Space into grids
  - Fixed grid size (less scalability)
  - Expensive search (Curse of dimensionality)

- Sampling based planner
  - No fixed partition rather sampling of real-valued points
  - Probabilistic search
  - Probabilistically complete

What is C-Space ?



- It is a set of all the possible configurations in which a robot can go

Image taken from [3]

Why C-Space?

Seen in comparison of State-Space

State Space	Configuration Space (C-Space)
A state includes Task-Dependent State Observational State Internal Configuration State	Space of all possible internal configuration states. For this work, the configuration is a polar coordinate value on a map considering the agent as a point robot.
e.g. (x, y, z, dx/dt, dy/dt, dz/dt)	e.g. (x, y, z)
A state may be required to take task level decision in a system	A configuration may be required to build trajectory
Every state in State-Space has a projection on C-Space	C-Space is a subset of State-Space

Formal terms

- A roadmap is an undirected graph R = (N,E)
- C-Free is a subset of C-Space which is collision free
- A node in N represents a configuration from C-Free
- An edge in E represents a collision free path between two elements of N
- D is the distance metric for C-Space
- p(s) is the projection of state of the system from state-space to C-space

PRM Generation

- Basic PRM[3]
  - Sample a new point in C-Free

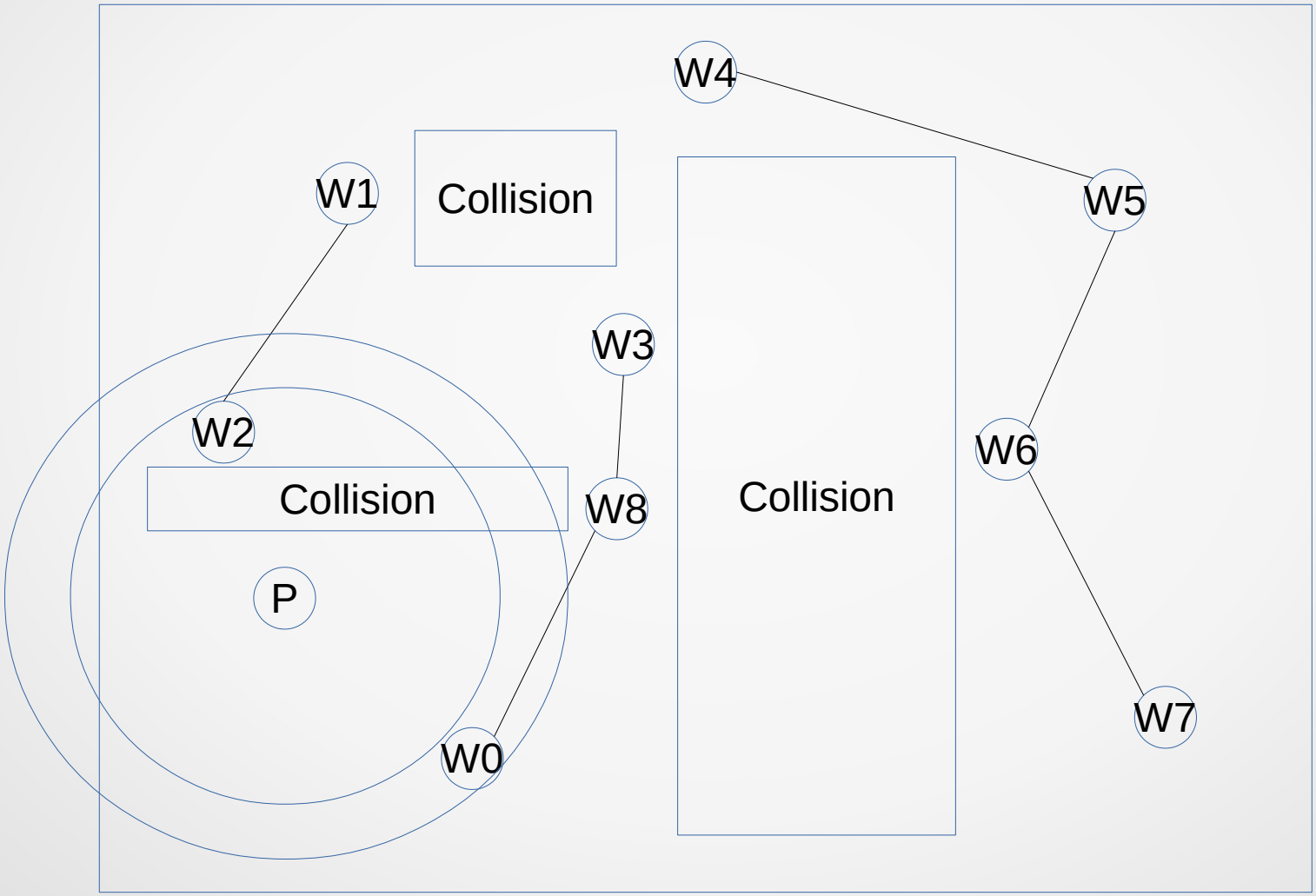
    add to the set of vertices of the graph
  - Connect this to a nearest neighbor in the existing graph

    evaluate neighbors in order of increasing distance via the Local planner
  - If connection is collision free, connect an edge too

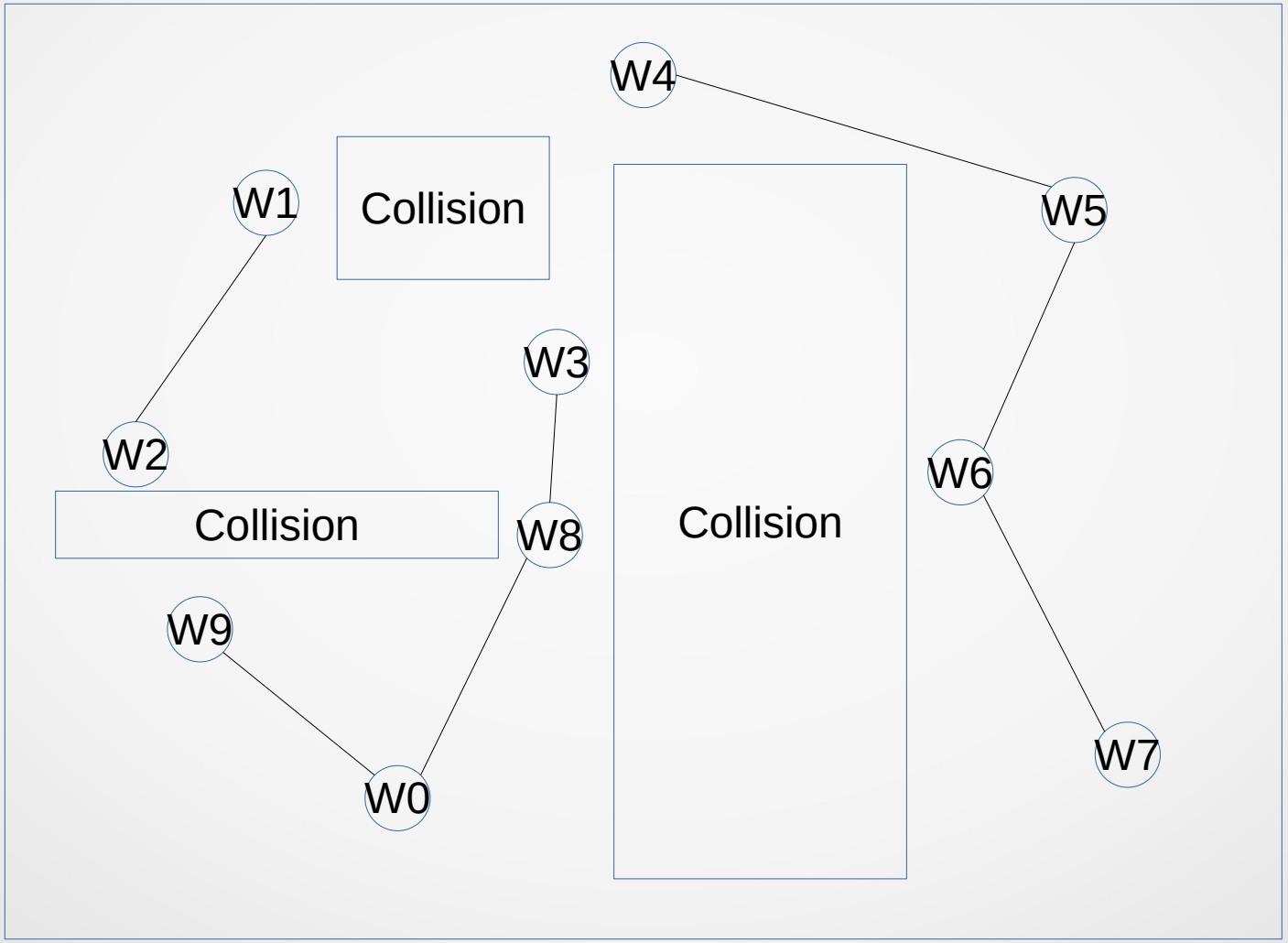
In the learning phase a probabilistic *roadmap* is constructed by repeatedly generating random free configurations of the robot and connecting these configurations using some simple, but very fast motion planner. We call this planner the *local planner*. The roadmap thus formed in the free configuration space (C-space [LP83]) of the robot is stored as an undirected graph R. The configurations are the nodes of R and the paths computed by the local planner are the edges of R. The learning phase is concluded by some postprocessing of R to improve its connectivity.

Text snippet from [3]

PRM Construction



PRM Construction

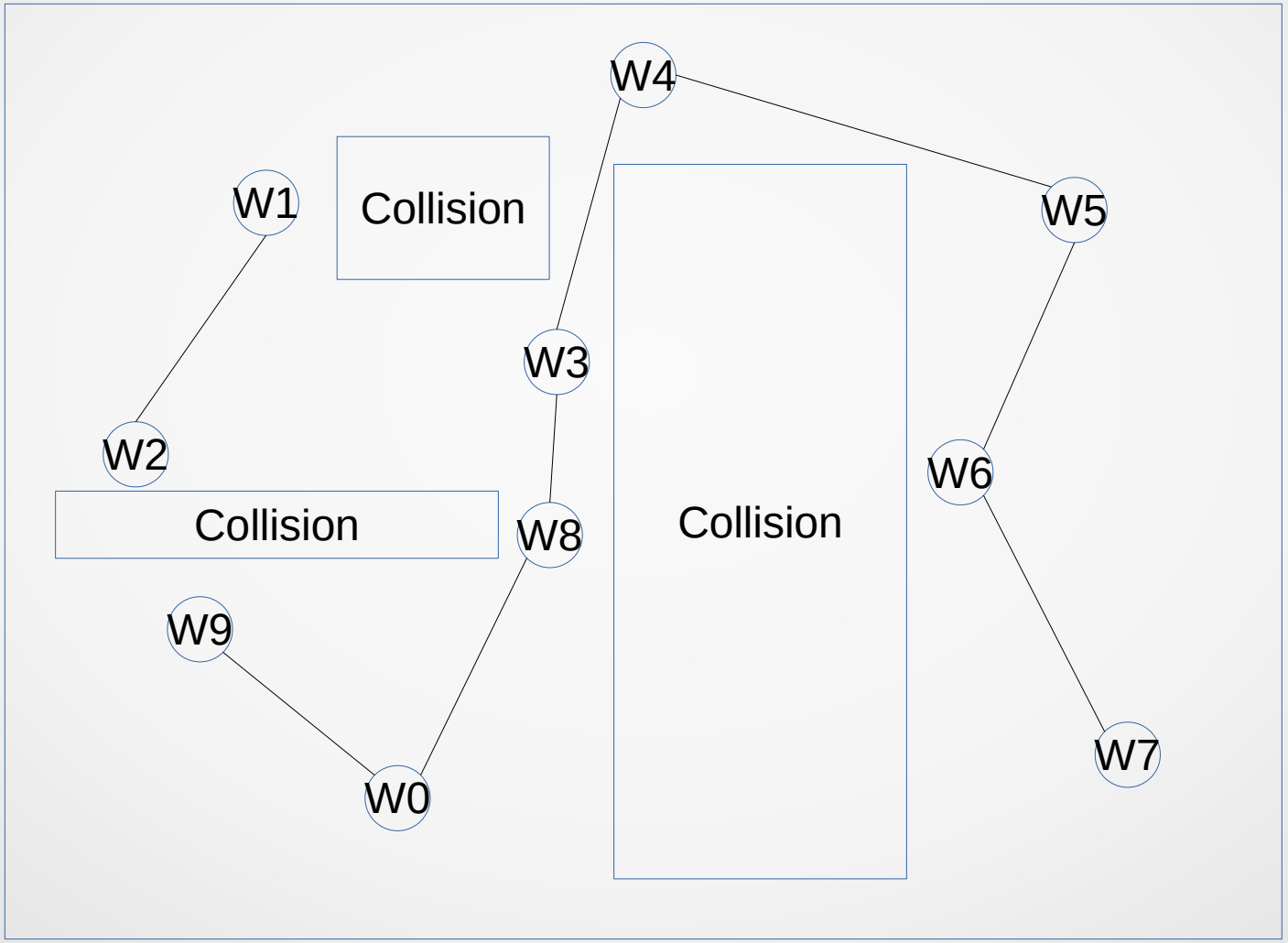


PRM Construction

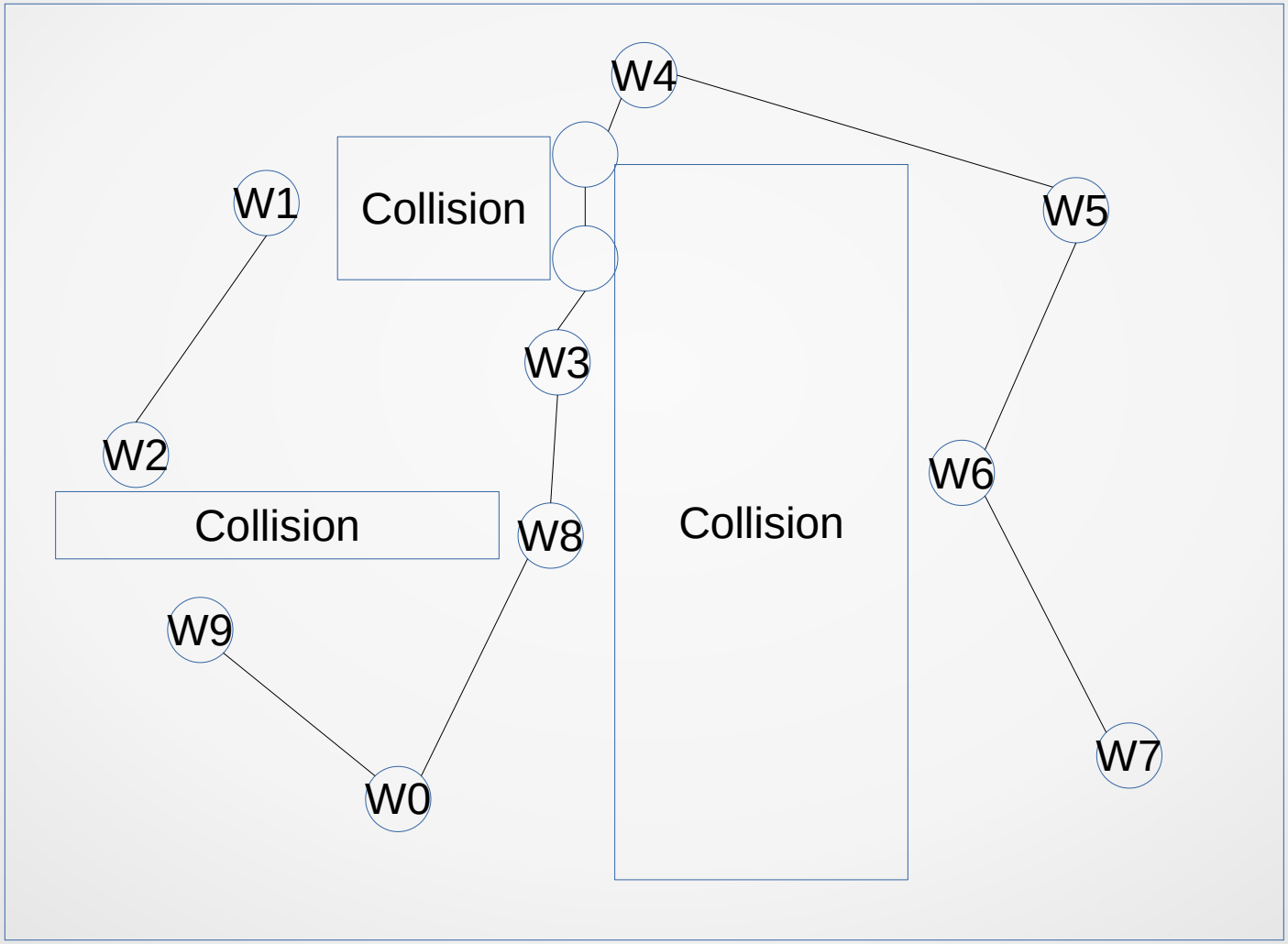
The learning phase consists of two successive steps, which we refer to as the construction and the expansion step. The objective of the former is to obtain a reasonably connected graph, with enough vertices to provide a rather uniform covering of free C-space and to make sure that most "difficult" regions in this space contain at least a few nodes. The second step is aimed at further improving the connectivity of this graph. It selects nodes of R which, according to some heuristic evaluator, lie in difficult regions of C-space and expands the graph by generating additional nodes in their neighborhoods. Hence, the covering of C_f by the final roadmap is not uniform, but depends on the local intricacy of the C-space.

Text snippet from [3]

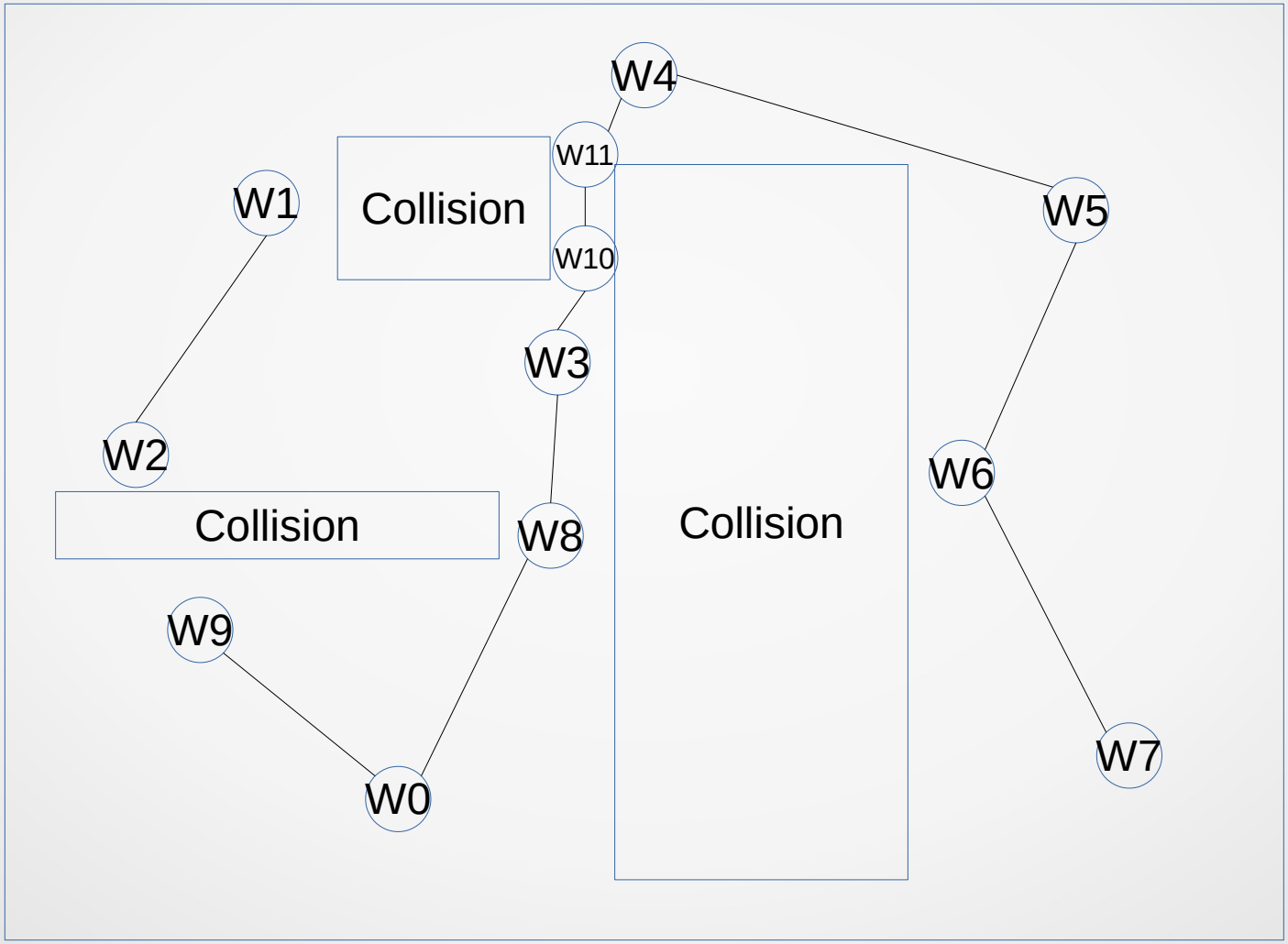
PRM Construction



PRM Construction



Generated PRM



PRM Generation

(1) $N \leftarrow \emptyset$

(2) $E \leftarrow \emptyset$

(3) **loop**

(4) $c \leftarrow$ a randomly chosen free configuration

(5) $N_c \leftarrow$ a set of candidate neighbors of c chosen from N

(6) $N \leftarrow N \cup \{c\}$

(7) **forall** $n \in N_c$, in order of increasing $D(c, n)$ **do**

(8) **if** $\neg same\_connected\_component(c, n) \wedge \Delta(c, n)$ **then**

(9) $E \leftarrow E \cup \{(c, n)\}$

(10) update R's connected components

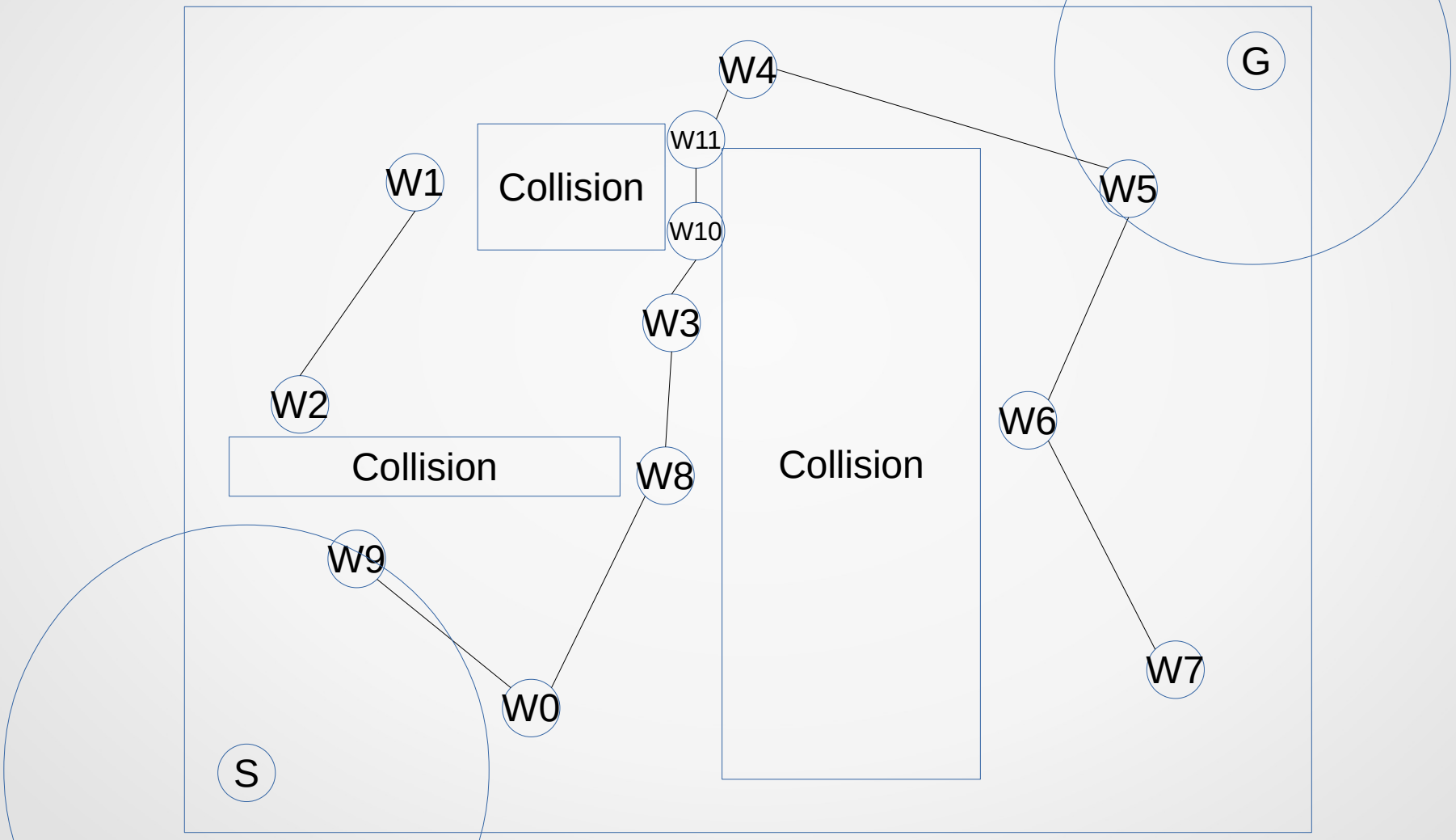
Text snippet from [3]

PRM Query

Following the learning phase, multiple queries can be answered. A *query* asks for a path between two free configurations of the robot. To process a query the method first attempts to find a path from the start and goal configurations to two nodes of the roadmap. Next, a graph search is done to find a sequence of edges connecting these nodes in the roadmap. Concatenation of the successive path segments transforms the sequence found into a feasible path for the robot.

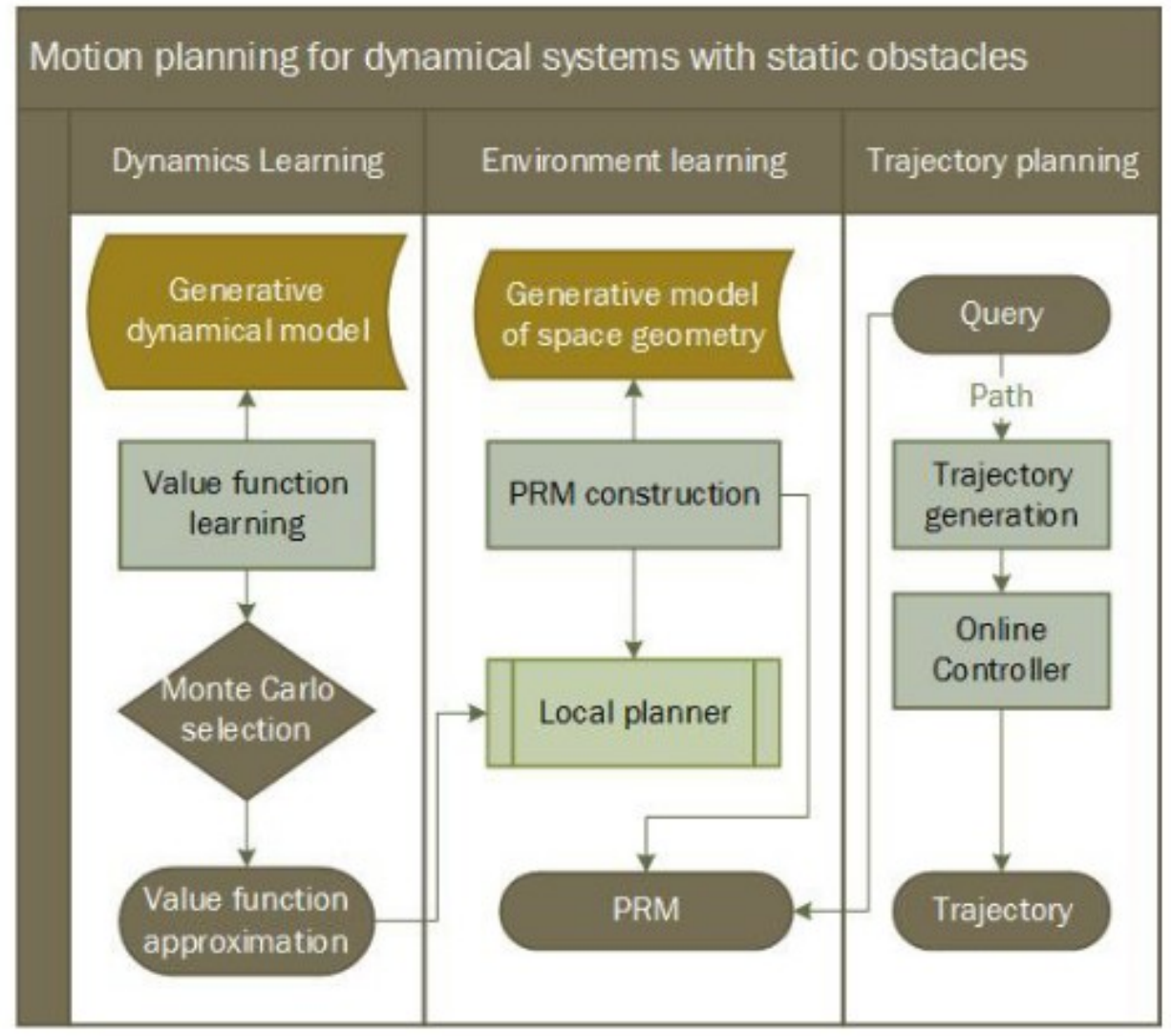
Text snippet from [3]

PRM Query


G
W4
W11
W1
Collision
W5
W10
W3
W2
W6
Collision
W8
Collision
W9
W7
W0
S

Method PRM-RL[2]

PRM-RL (Three Stage Process)



RL Agent Training (Office Space)

- State
  - 2 valued polar coordinate distance to goal
  - + 64 valued LIDAR* observation for collision
- Action
  - 2 valued vector of wheel speed
- Reward
  - +ve on goal state
  - -ve on collision
- Training
  - Deep Deterministic Policy Gradient[5] (DDPG)

We reward the agent for reaching the goal, with task specific reward shaping terms for each of the two tasks. For the indoor navigation task, we reward the agent for staying away from obstacles, while for the aerial cargo delivery task we reward minimizing the load displacement.

* Light Detection and Ranging

RL Agent Training (Aerial Space)

- State
  - 6 valued 3d position + 3d velocity vector
  - + 4 valued angular position and it's first derivative in spherical coordinate system
- Action
  - 3 valued vector of acceleration at center of mass
- Reward
  - Based on minimization of load displacement
- Training
  - Continuous Action Fitted Value Iteration [4] (CAFVI)

We reward the agent for reaching the goal, with task specific reward shaping terms for each of the two tasks. For the indoor navigation task, we reward the agent for staying away from obstacles, while for the aerial cargo delivery task we reward minimizing the load displacement.

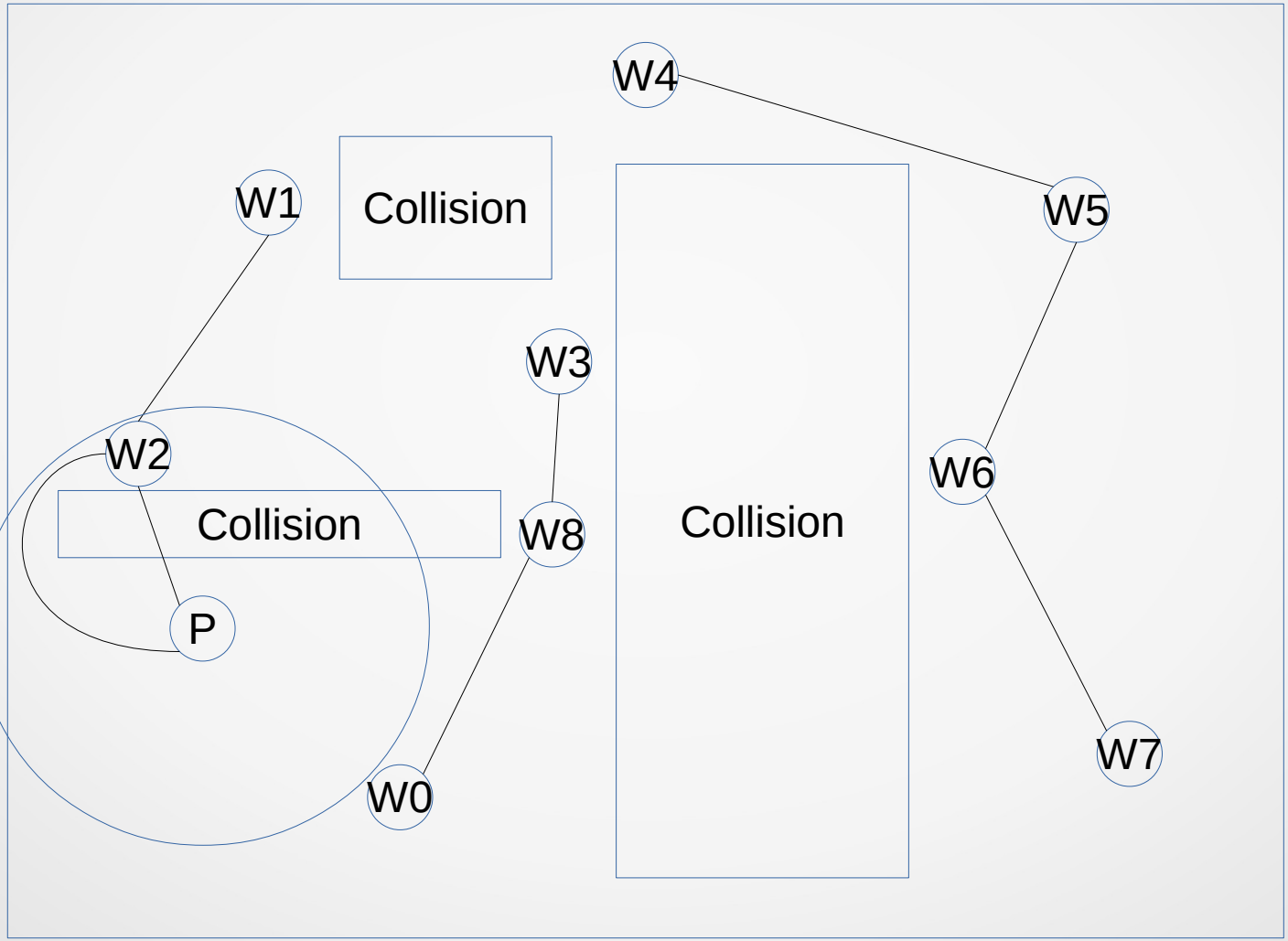
PRM Modifications (Office Space and Aerial Space)

- An inexpensive local planner can be a straight line planner.
  - For such a planner, a single trail is sufficient

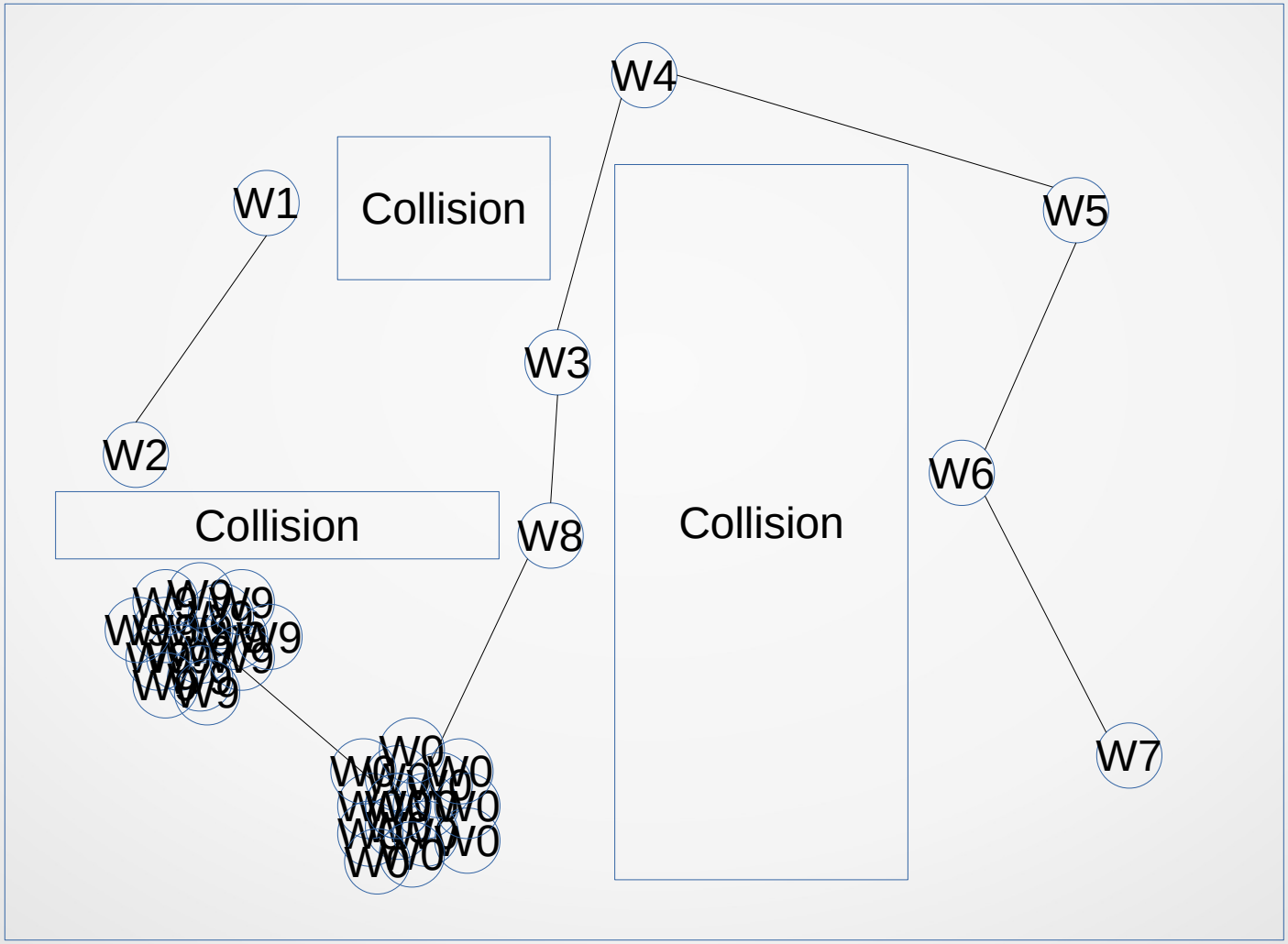
- With RL, we want to account for noise and complex dynamics
  - Multiple Trails are required
    and a threshold is applied to the success rate and and edge is added accordingly

  - Slight noise is added to the sampled start and goal position

PRM Construction



PRM Construction



Algorithm of adding a PRM edge

Input: $s, g \in C_{space}$: Start and goal.
Input: $p_{success} \in [0, 1]$ Success threshold.
Input: $num_{attempts}$: Number of attempts.
Input: ϵ: Sufficient distance to the goal.
Input: max_{steps}: Maximum steps for trajectory.
Input: $L(\boldsymbol{s})$: Task predicate.
Input: π: RL agent's policy.
Input: D Generative model of system dynamics.
Output: $add_{edge}, success_{rate}$, $length$

Line 8-13
represent a roll-out
via local planner to determine connectivity

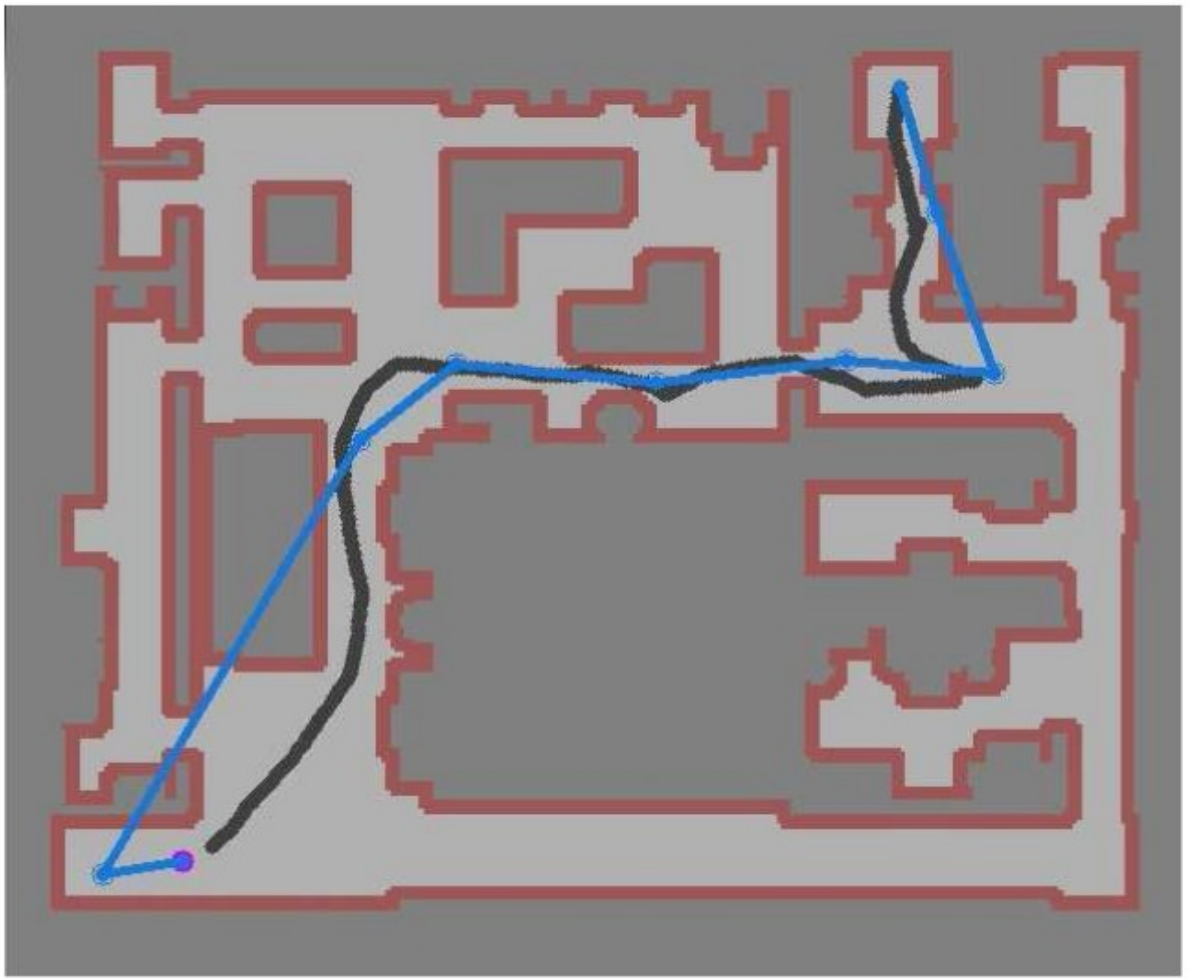
1: $success \leftarrow 0$, $length \leftarrow 0$
2: $needed \leftarrow p_{success} * num_{attempts}$
3: **for** $i = 1, \cdots num_{attempts}$ /* Run in parallel.*/ **do**
4: $\quad \boldsymbol{s}_s \leftarrow s.SampleStateSpace()$ // Sample from the
5: $\quad \boldsymbol{s}_g \leftarrow g.SampleStateSpace()$ // state space
6: $\quad success_{rate} \leftarrow 0$, $steps \leftarrow 0$, $\boldsymbol{s} \leftarrow \boldsymbol{s}_s$
7: $\quad length_{trial} \leftarrow 0$
8: $\quad$ **while** $L(\boldsymbol{s}) \wedge steps < max_{steps} \wedge \|p(\boldsymbol{s}) - p(\boldsymbol{s}_g)\| > \epsilon \wedge p(\boldsymbol{s}) \in$ C-free **do**
9: $\quad\quad \boldsymbol{s}_p \leftarrow \boldsymbol{s}$, $\boldsymbol{a} \leftarrow \pi(\boldsymbol{s})$
10: $\quad\quad \boldsymbol{s} \leftarrow D.predictState(\boldsymbol{s}, \boldsymbol{a})$
11: $\quad\quad num_{steps} \leftarrow num_{steps} + 1$
12: $\quad\quad length_{trial} \leftarrow length_{trial} + \|\boldsymbol{s} - \boldsymbol{s}_p\|$
13: $\quad$ **end while**
14: $\quad$ **if** $\|p(\boldsymbol{s}) - p(\boldsymbol{s}_g)\| < \epsilon$ **then**
15: $\quad\quad success \leftarrow success + 1$
16: $\quad$ **end if**
17: $\quad$ **if** $needed > success \wedge i > needed$ **then**
18: $\quad\quad$ **return** False, 0, 0 // Not enough success, we can terminate.
19: $\quad$ **end if**
20: $\quad length_{trial} \leftarrow length_{trial} + \|p(\boldsymbol{s}) - p(\boldsymbol{g})\|$
21: $\quad length \leftarrow length + length_{trial}$
22: **end for**
23: $length \leftarrow \frac{length}{success}$, $success_{rate} \leftarrow \frac{success}{i}$
24: **return** $success_{rate} > p_{success}, success_{rate}, length$

Task Environment

Environments

- PRM-RL evaluated over,
  - two navigation tasks
    with non-trivial robot dynamics
    - End-to-end differential drive indoor navigation
      in office environments
    - aerial cargo delivery
      in urban environments
      with load displacement constraints
  - both in simulation and on-robot

Office Environment



(a) Training environment - 23 m by 18 m

Office Environment

(b) Building 1 - 183 m by 66 m

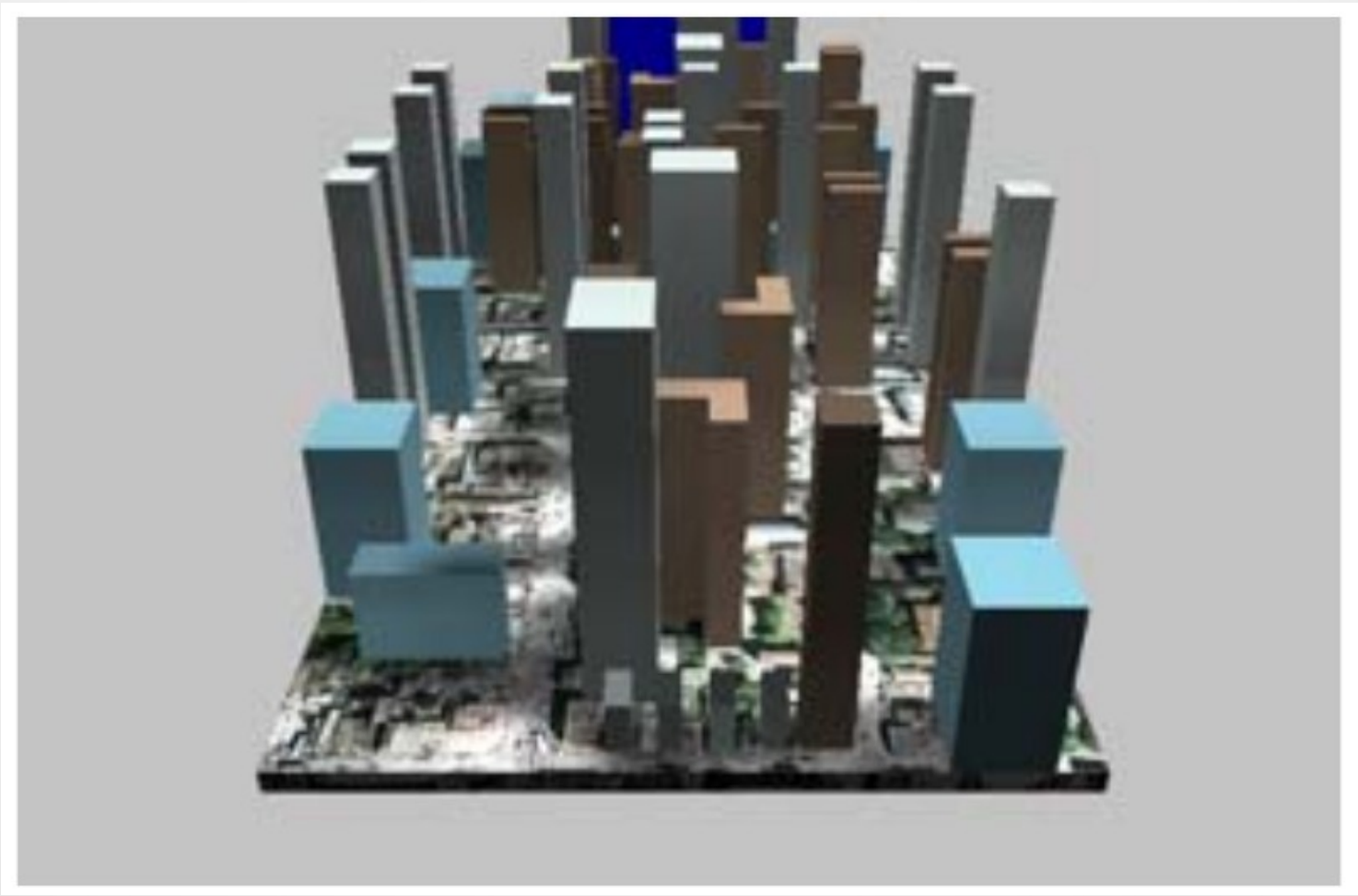
Office Environment

(c) Building 2 - 60 m by 47 m

Office Environment

(d) Building 3 - 134 m by 93 m

Urban Environment



Results

Performance

https://www.youtube.com/watch?v=_XiaL5W-5Lg

Indoor Navigation

- Cost of Building the Roadmaps
- Quality of the Planner Trajectories
- Actual Performance in Simulation
- Physical Robot Experiment Results

Cost of Building the Roadmaps

TABLE I

ROADMAP CONSTRUCTION SUMMARY DIFFERENT NODE SAMPLING DENSITIES (0.1, 0.2, AND 0.4 SAMPLES PER METER SQUARED). ENVIRONMENT, METHOD, SUCCESS RATE ON 100 QUERY EVALUATION, NUMBER OF NODES, NUMBER OF EDGES, NUMBER OF COLLISION CHECKS.

Sampling density	Method	Query success rate (%)			Nodes			Edges			Collision Checks		
		0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4	0.1	0.2	0.4
Training	PRM-RL	0.32	0.36	0.50	16	32	63	33	166	663	13009	38292	223898
	PRM-SL	0.06	0.09	0.29	16	32	63	15	123	464	914	3649	17264
Building 1	PRM-RL	0.14	0.31	0.43	436	871	1741	3910	15632	59856	1476931	5755744	23303949
	PRM-SL	0.06	0.17	0.15	436	871	1741	3559	13937	52859	156641	622257	2393841
Building 2	PRM-RL	0.17	0.22	0.38	116	232	463	403	1602	6833	294942	1174655	5218619
	PRM-SL	0.06	0.11	0.18	116	232	463	276	1190	5365	18297	72850	312859
Building 3	PRM-RL	0.19	0.39	0.56	441	881	1761	2962	11850	45623	1152524	4492144	17947728
	PRM-SL	0.07	0.11	0.08	441	881	1761	1852	7570	30267	97088	375304	1493816

- Edges and collision check depend upon local planner
- Detects more edges as RL does not depend upon straight line planning

Quality of the Planner Trajectories

TABLE II

EXPECTED PATH AND TRAJECTORY CHARACTERISTICS OVER 100 QUERIES. ENVIRONMENT, METHOD, ACTUAL AND EXPECTED SUCCESS PERCENT, NUMBER OF WAYPOINTS IN THE PATH, EXPECTED TRAJECTORY LENGTH IN METERS, AND DURATION IN SECONDS.

Environment	Method	Success (%)		Number of waypoints		Trajectory length (m)		Duration (s)	
		Actual	Expected	μ	σ	μ	σ	μ	σ
Training	PRM-RL	50	90	7.11	2.88	18.68	11.80	36.28	36.28
	PRM-SL	28	100	5.44	3.51	9.39	9.69	8.29	8.29
Building 1	PRM-RL	43	91	12.09	10.68	56.88	63.37	107.78	107.78
	PRM-SL	15	100	11.99	8.88	46.69	43.85	43.07	43.07
Building 2	PRM-RL	38	95	6.05	4.46	21.54	25.82	41.69	41.69
	PRM-SL	18	100	6.98	5.69	18.75	23.05	16.97	16.97
Building 3	PRM-RL	56	92	12.62	5.12	64.94	33.96	122.31	122.31
	PRM-SL	8	100	15.58	8.02	59.03	35.47	54.00	54.00

- SL has 100% success estimate due to single sample to decide connectivity
- SL has shorter trajectories and duration due to straight line planner

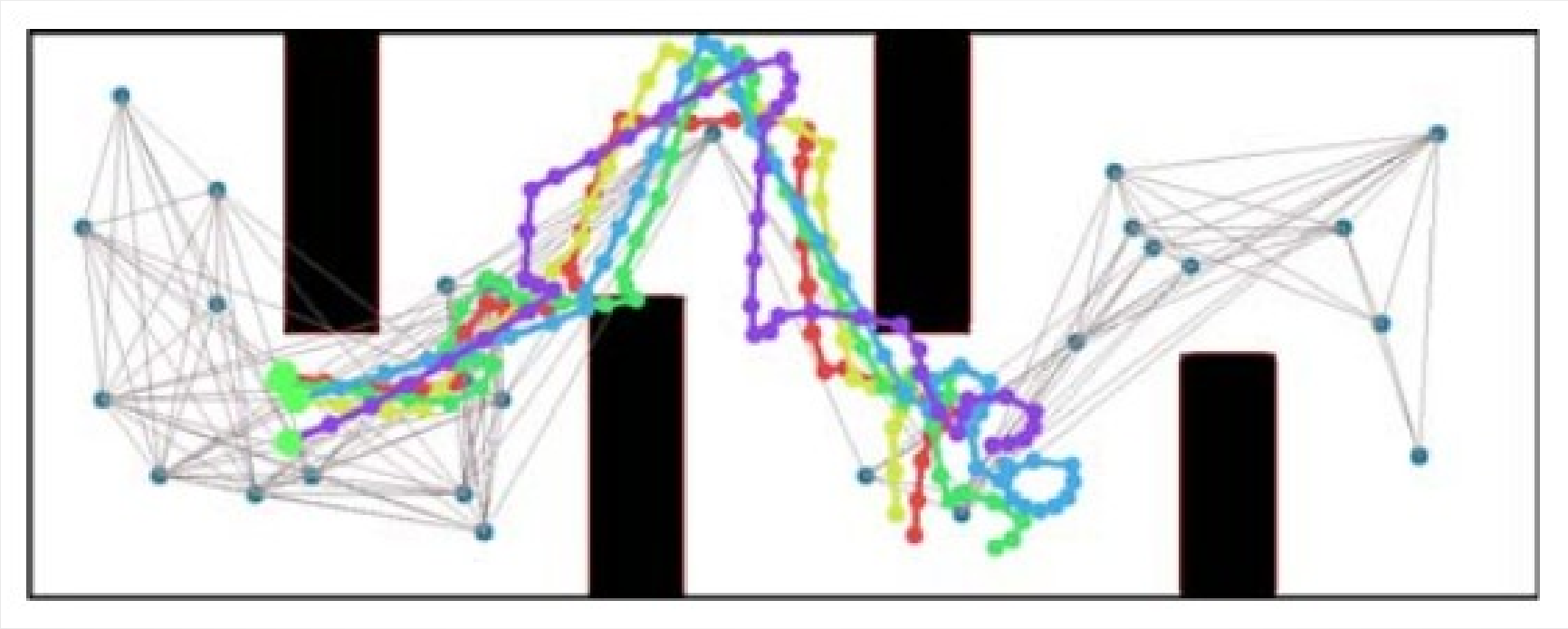
Actual Performance in Simulation

TABLE III

CHARACTERISTICS OF THE SUCCESSFUL AND UNSUCCESSFUL TRAJECTORIES. ENVIRONMENT, METHOD, ACTUAL AND EXPECTED SUCCESS PERCENT, NUMBER OF WAYPOINTS IN THE PATH, EXPECTED TRAJECTORY LENGTH IN METERS, AND DURATION IN SECONDS.

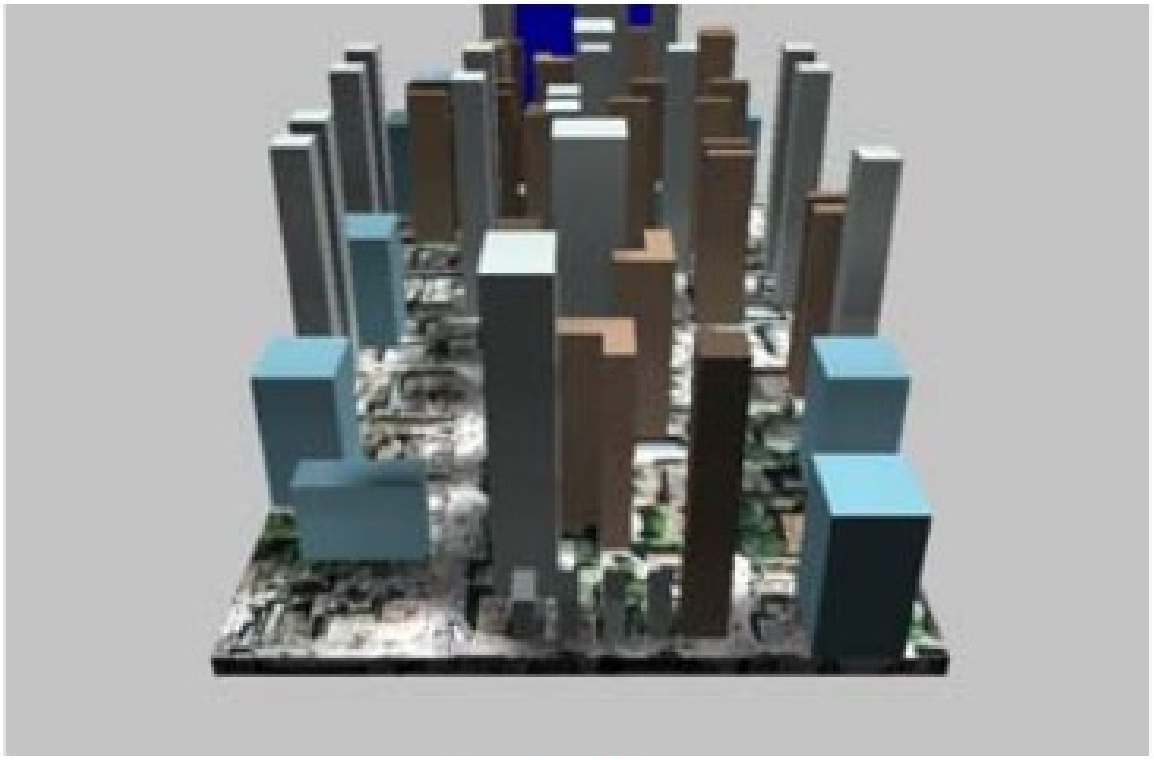
Environment	Method	Success (%)	Number of waypoints				Trajectory length (m)				Duration (s)			
			Successful		All		Successful		All		Successful		All	
			μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Training	PRM-RL	50	4.29	2.38	4.32	2.88	13.88	8.62	14.54	9.46	36.74	26.06	37.04	37.04
	PRM-SL	28	2.45	3.02	2.49	2.73	6.71	7.10	6.39	6.32	6.71	7.10	6.39	6.39
Building 1	PRM-RL	43	10.76	11.01	8.44	9.77	51.17	56.66	43.16	51.74	117.46	108.60	112.29	112.29
	PRM-SL	15	4.37	4.69	4.09	4.51	20.19	21.78	18.32	20.46	20.19	21.78	18.32	18.32
Building 2	PRM-RL	38	8.08	3.67	3.96	4.53	32.93	21.58	23.64	20.78	70.18	45.87	77.33	77.33
	PRM-SL	18	3.89	5.63	3.41	4.71	15.31	20.79	12.29	17.20	15.31	20.79	12.29	12.29
Building 3	PRM-RL	56	9.74	5.60	9.61	5.79	58.71	32.09	57.60	34.78	130.79	63.06	130.06	130.06
	PRM-SL	8	5.66	5.33	5.21	4.46	22.46	19.67	21.30	17.50	22.46	19.67	21.30	21.30

Physical Robot Experiment Results

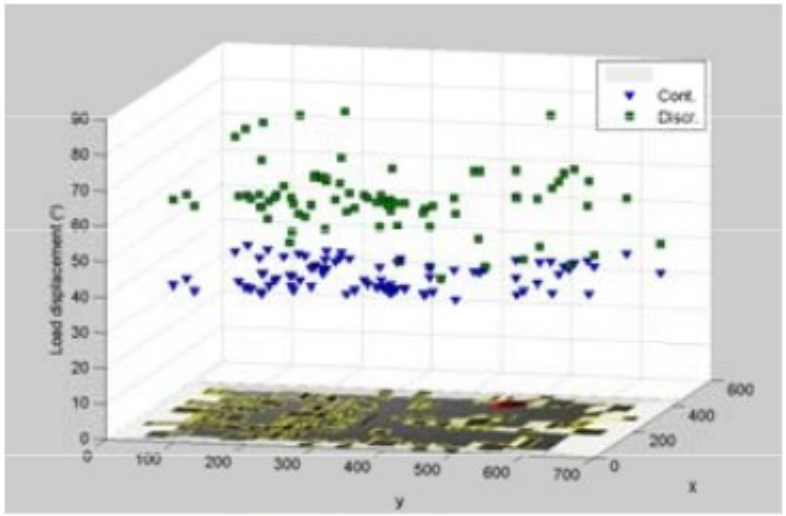


Aerial Cargo Delivery

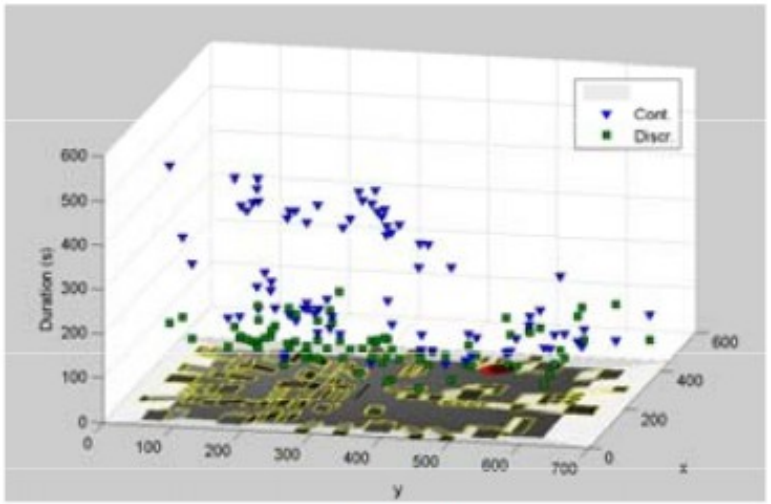
- Simulation Results
- Experimental Results



Simulation Results

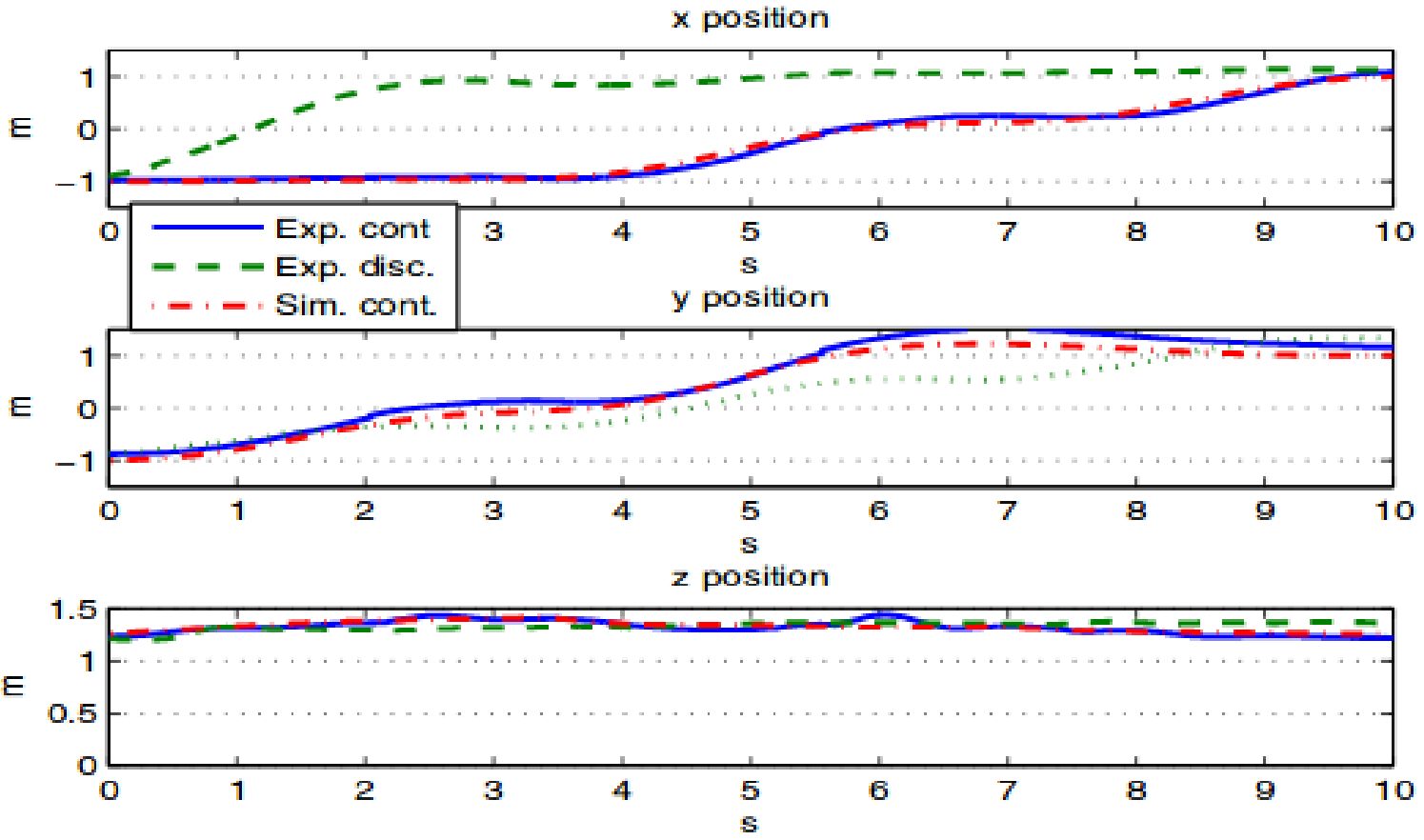


(b) Load displacement

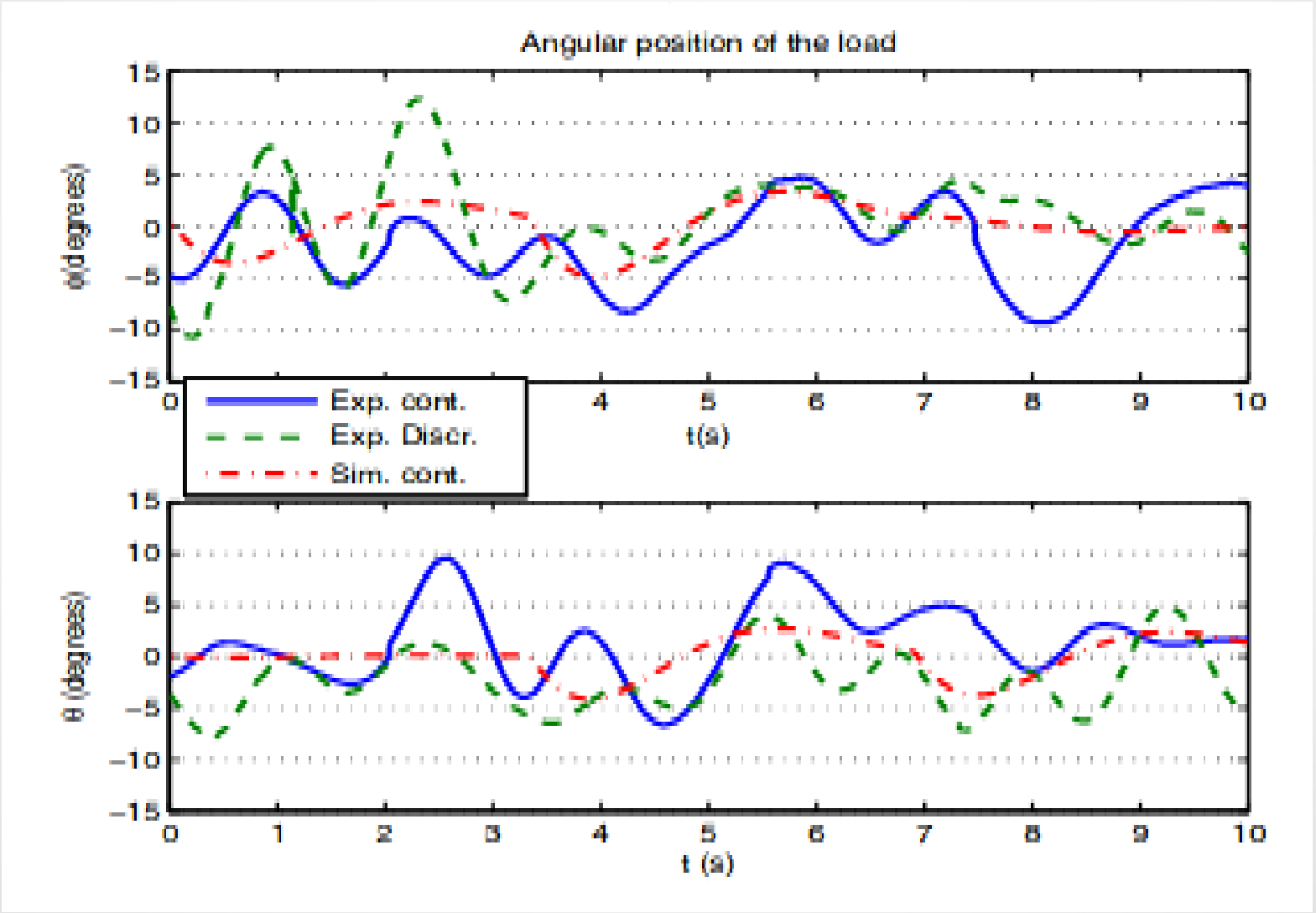


(c) Duration

Experimental Results 1



Experimental Results 2



Refrences

1. Aleksandra Faust, Oscar Ramirez, Marek Fiser, Kenneth Oslund, Anthony Francis, James Davidson, and Lydia Tapia, PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-based Planning
2. Luka Petrovi´c, Motion planning in high-dimensional spaces
3. L. E. Kavraki, P.˘Svestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robot. Automat., 12(4):566–580, August 1996.
4. Aleksandra Faust, Peter Ruymgaart, Molly Salmany, Rafael Fierroz, Lydia Tapia, Continuous Action Reinforcement Learning for Control-Affine Systems with Unknown Dynamics.
5. Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver and Daan Wierstra, Continuous control with deep reinforcement learning.

Thank You