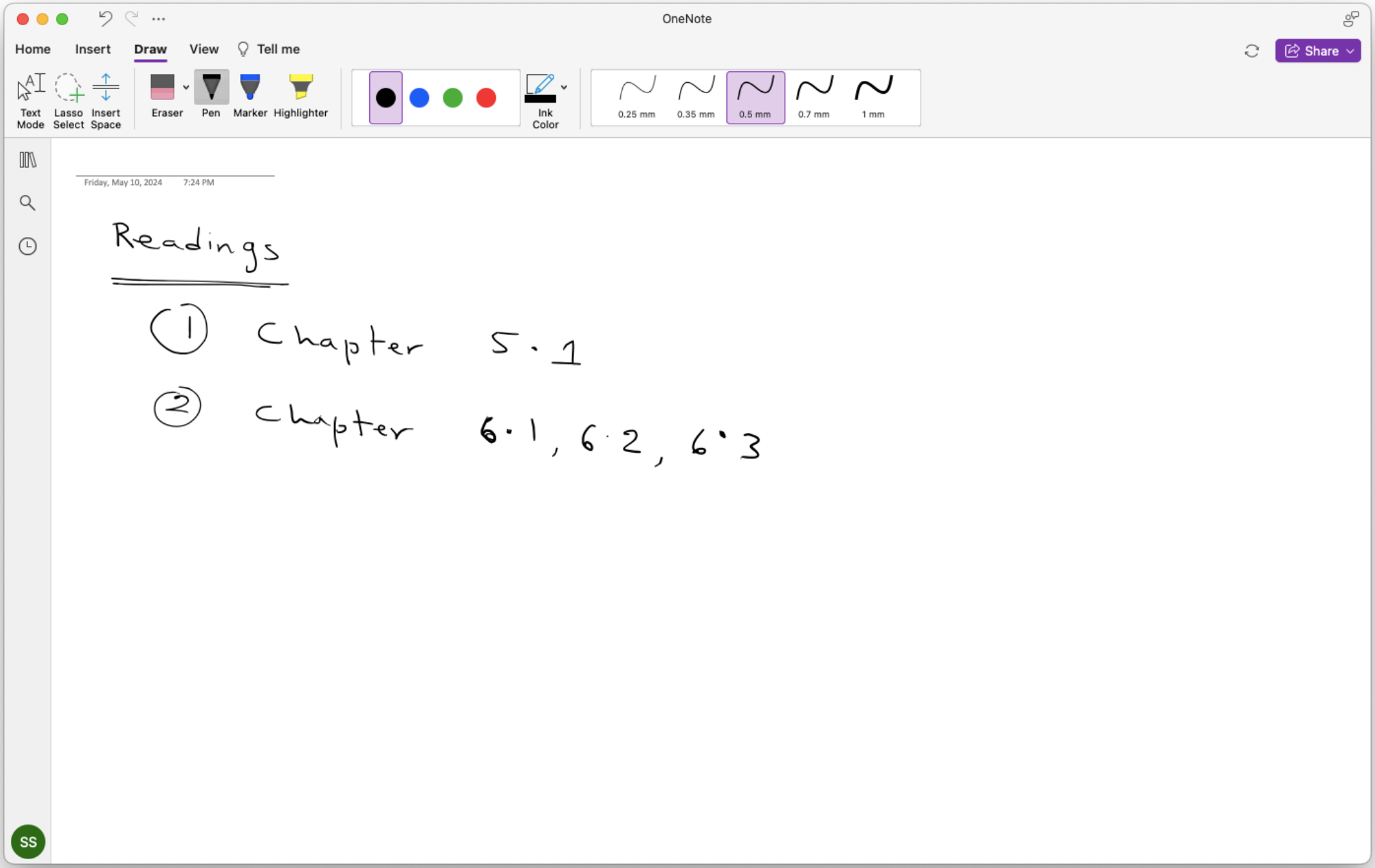

OneNote
Home
Insert
Draw
View
Tell me
Share
Text Mode
Lasso Select
Insert Space
Eraser
Pen
Marker
Highlighter
Ink Color
0.25 mm
0.35 mm
0.5 mm
0.7 mm
1 mm
Friday, May 10, 2024 7:24 PM
Readings
1 Chapter 5.1
2 Chapter 6.1, 6.2, 6.3
SS

OneNote

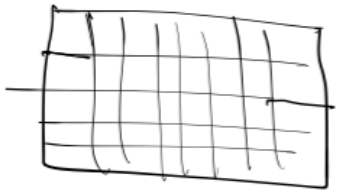
Home Insert Draw View Tell me Share

Text Mode | Lasso Select | Insert Space | Eraser | Pen | Marker | Highlighter | Ink Color | 0.25 mm | 0.35 mm | 0.5 mm | 0.7 mm | 1 mm

Friday, May 10, 2024 7:41 PM

Chapter 5 (Recap)

Grid based search motion planning

1.  → Graph → Search Graph for a path best (A* or Dijkstra's)
   - (a) Heavy computation
   - (b) Can handle static obstacles
2.  Differential Flatness $Ax=b \Rightarrow x=A^{\dagger}b$
   - (a) Does not include obstacle avoidance

SS

OneNote

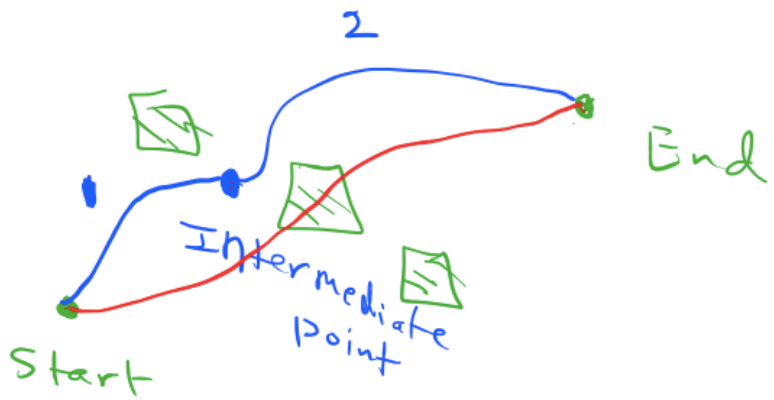
Home Insert Draw View Tell me

Share

Text Mode | Lasso Select | Insert Space | Eraser | Pen | Marker | Highlighter | Ink Color | 0.25 mm | 0.35 mm | 0.5 mm | 0.7 mm | 1 mm

Friday, May 10, 2024 7:54 PM

In differential flatness for static obstacles we can compute piecewise paths.



SS

OneNote

Home Insert Draw View Tell me Share

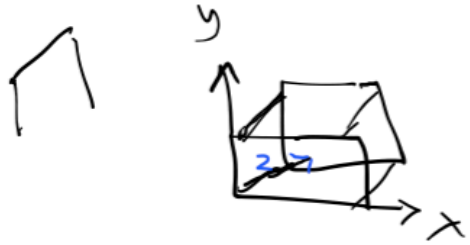
Text Mode | Lasso Select | Insert Space | Eraser | Pen | Marker | Highlighter | Ink Color | 0.25 mm | 0.35 mm | 0.5 mm | 0.7 mm | 1 mm

Friday, May 10, 2024 7:45 PM

Example : Drone or Quad copter

↳ moves in X, Y, Z space

↳ Not very practical in 3D space



↳ Not very practical for real time systems

→ Differential flatness is probably a more practical approach

SS

Friday, May 10, 2024 8:03 PM

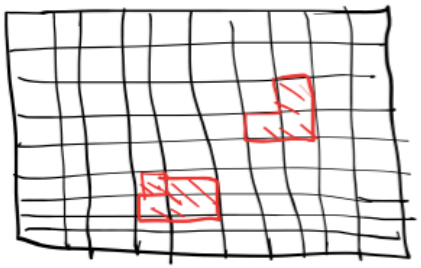
Sampling based motion planning

Instead of creating an exhaustive graph of the C_{free} space, we will sample randomly a number of nodes & create the graph from those nodes.

1. Probabilistic Road Map (PRM)
2. Rapidly Exploring Random Trees (RRT)

① PRM

Set $n = 50$ or 100



(1) Sample a node q_i from a random distribution

(2) Collision checker: Check if $q_i \in C_{free}$ (↓ keep)

If $q_i \in C_{obs}$ → discard

(3) Create the graph using the sampled nodes.

(4) Run A* or Dijkstra's on this graph to find the best path

OneNote

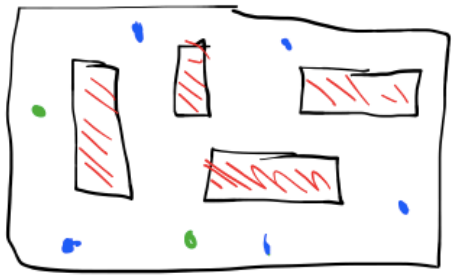
Home Insert Draw View Tell me Share

Text Mode | Lasso Select | Insert Space | Eraser | Pen | Marker | Highlighter | Ink Color | 0.25 mm | 0.35 mm | 0.5 mm | 0.7 mm | 1 mm

Friday, May 10, 2024 8:20 PM

PRM is a multi-query planner.

Example : You want to make warehouse delivery robot



Warehouse

① Using PRM method, you can create a graph

② Then search this graph for different start & end point.

In PRM method we are assuming the obstacles are static

SS

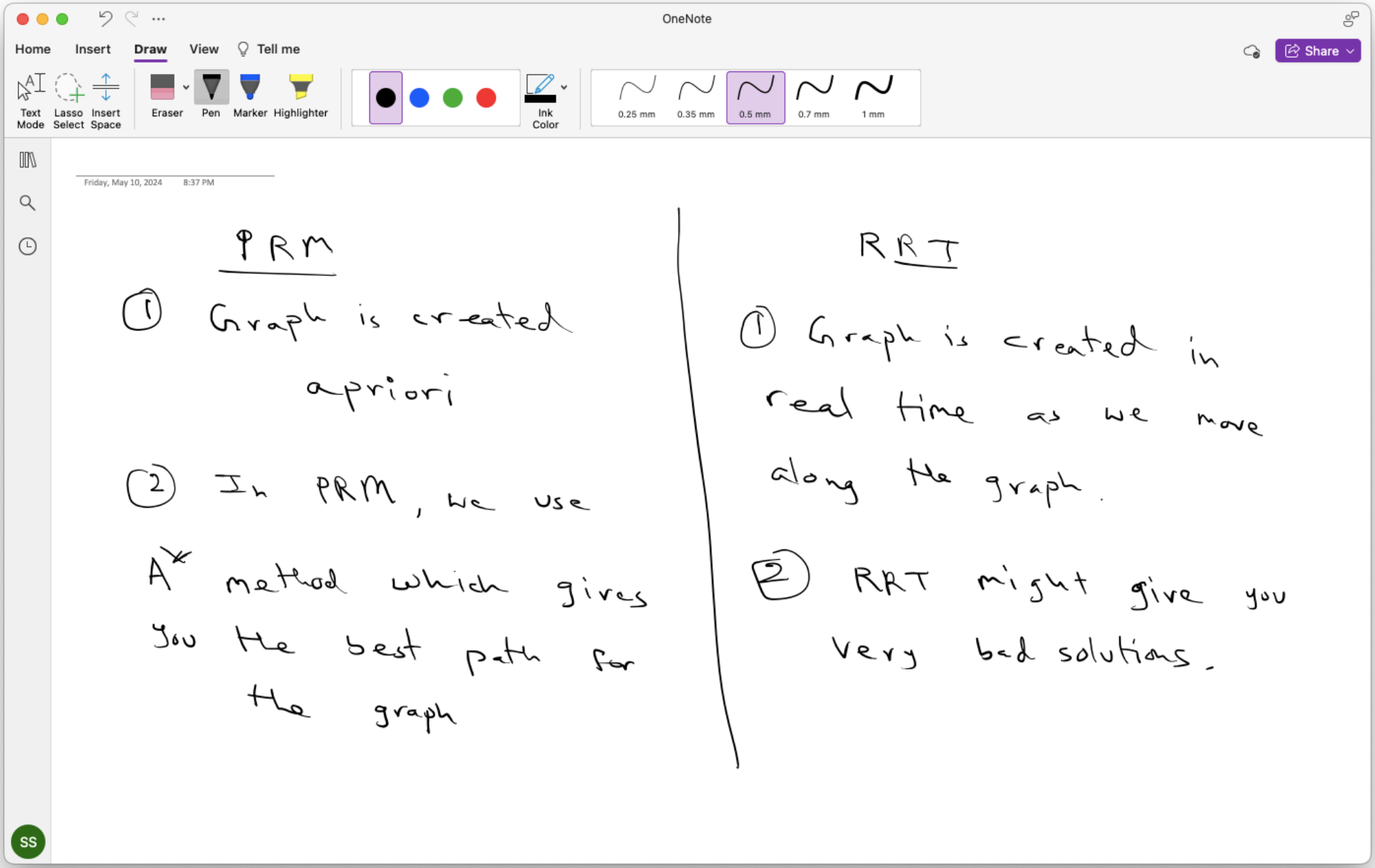
But in many cases, in robotics we have dynamic obstacles.

For example: other vehicles, pedestrians.

In these cases we can use RRT (a real time method)

RRT:

1. Sample a node q_i
2. Check if $q_i \in C_{free}$ → keep otherwise discard
3. Create edge & move along the graph
4. Until we reach q_G


OneNote
Home
Insert
Draw
View
Tell me
Share
Text Mode
Lasso Select
Insert Space
Eraser
Pen
Marker
Highlighter
Ink Color
0.25 mm
0.35 mm
0.5 mm
0.7 mm
1 mm
Friday, May 10, 2024 8:37 PM
PRM
1 Graph is created apriori
2 In PRM, we use A* method which gives you the best path for the graph
RRT
1 Graph is created in real time as we move along the graph.
2 RRT might give you very bad solutions.
SS