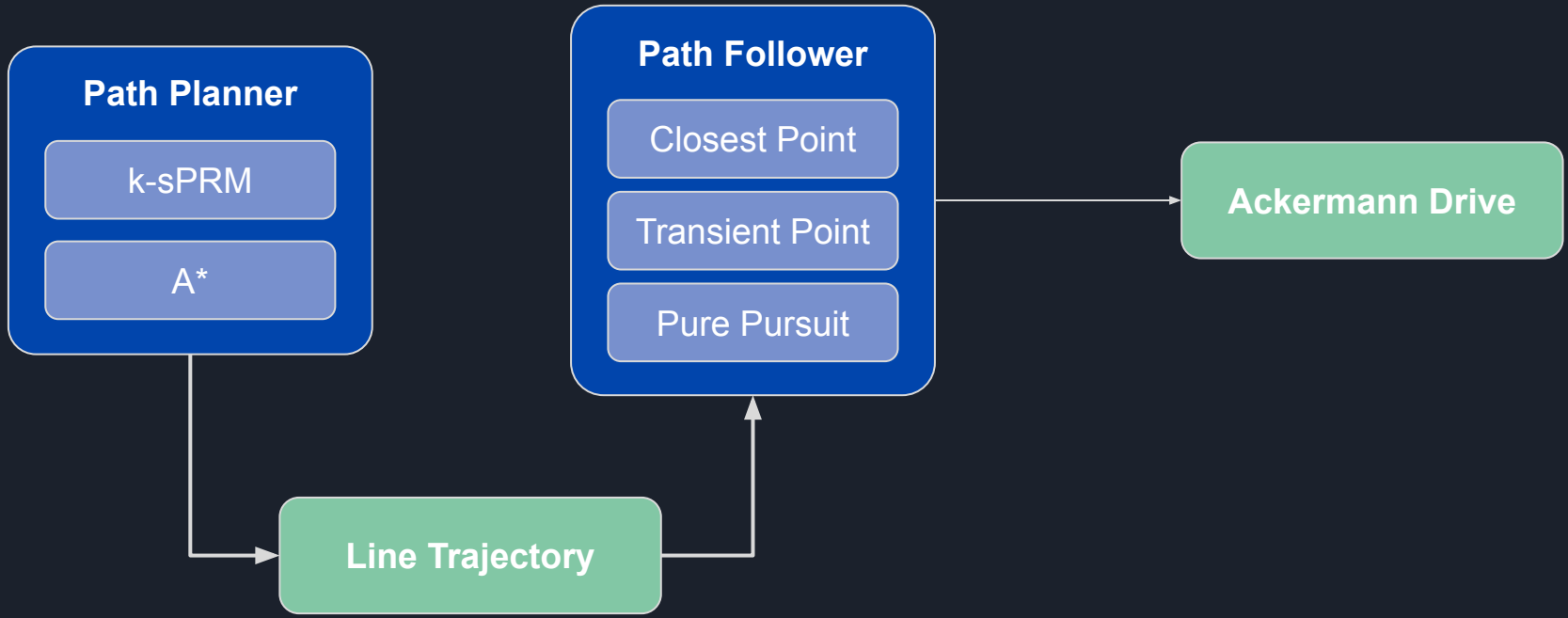
RSS Lab 6

Team Bird-Planes

Joshua, Isaac, Lilly, Mario

Path Planning Roadmap



Module 1:
Path Planning

Occupancy in Pixel Frame and Map Frame

1. Rotation

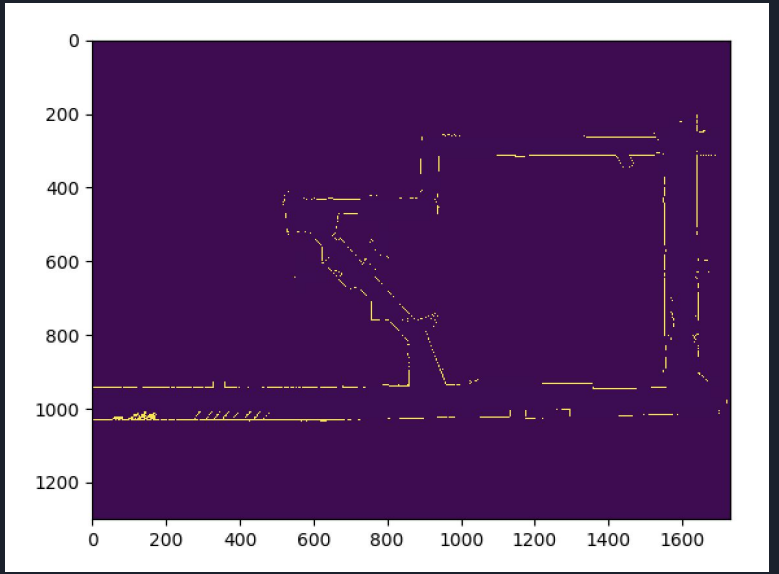
$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

2. Translation

$$(x_1, y_1) = (x_r + x_t, y + y_t)$$

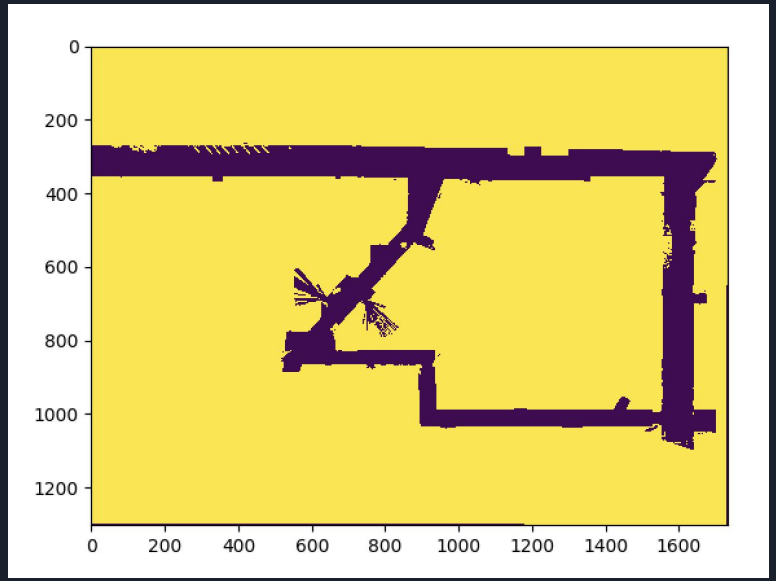
3. Scale

$$(u, v) = \left(\frac{x_1}{resolution}, \frac{y_1}{resolution}\right)$$

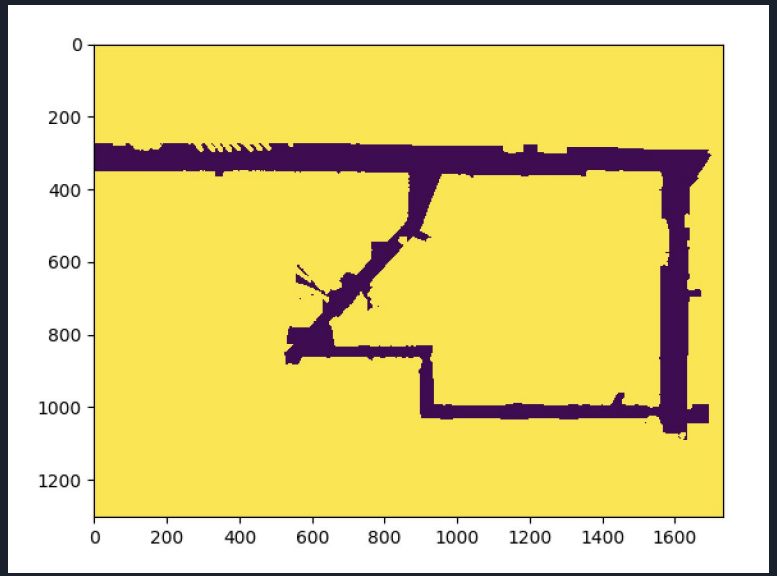


Given Occupancy Grid

Dilation creates a buffer between the robot and walls.

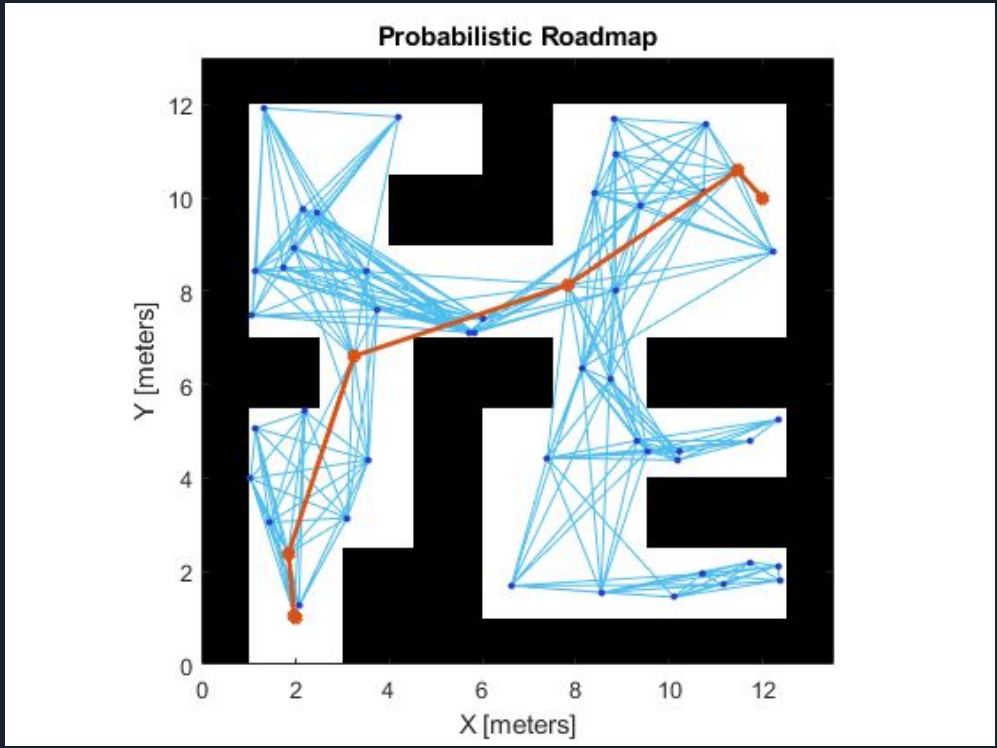


Work Space



Configuration Space

Probabilistic Roadmap (PRM)



A **simplified PRM** method samples nodes from the occupancy grid

```
k_sPRM = {
     node: [
          neighbor_1,
          ...,
          neighbor_k
     ],
     ...
}
```

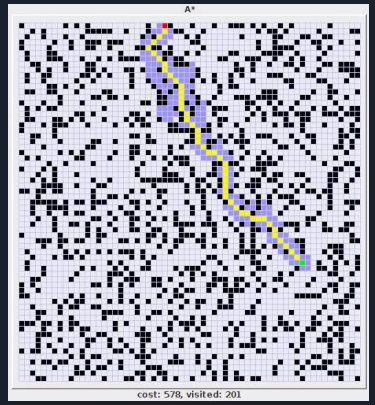
Algorithm	*Complete*	*Optimal*	*Converges*	*Complexity*
PRM	Yes	No	Yes	O(n log n)
sPRM	Yes	Yes	Yes	$O(n^2)$
k-sPRM	***No****	**Yes**	**Yes**	**O(n log n)***

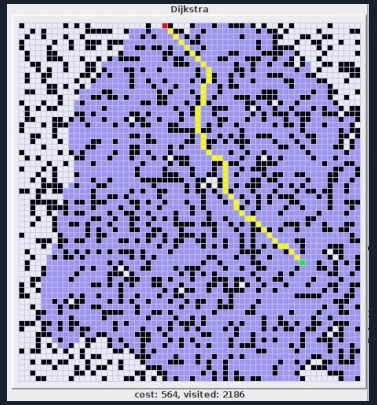
**For # of neighbors k << n, k-sPRM is not complete, but has time complexity O(n log n)*

A* Algorithm uses a heuristic to optimize its search path

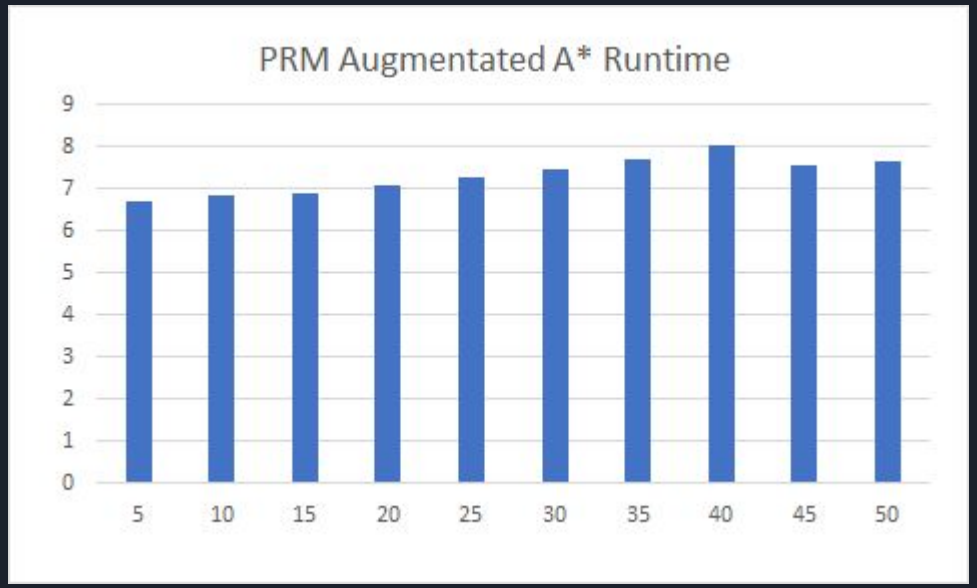
$$f(n) = g(n) + h(n)$$

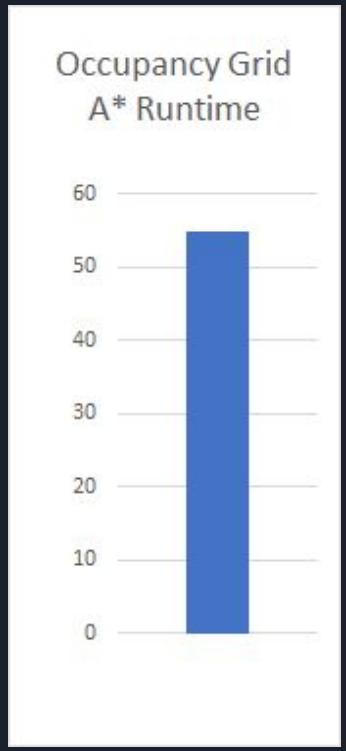
- $f(n)$ = total estimated cost of path through node n
- $g(n)$ = cost so far to reach node n
- $h(n)$ = estimated cost from n to goal





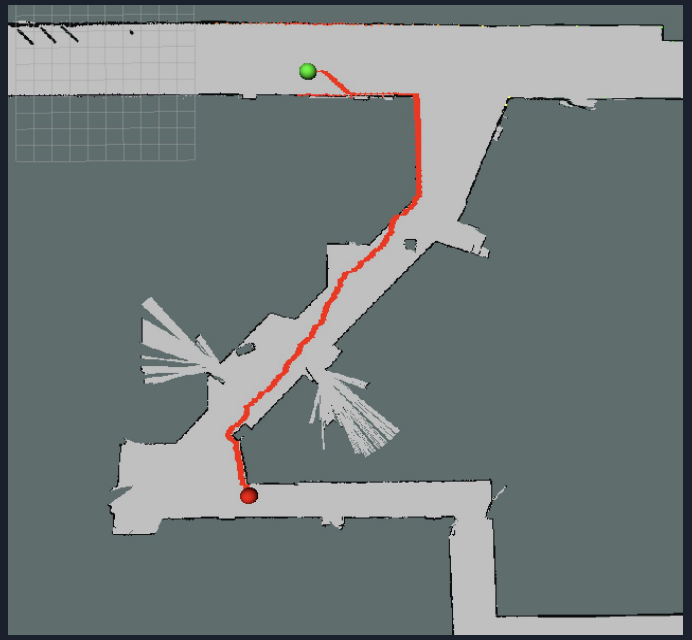
Tuned PRM vs. Occupancy Grid



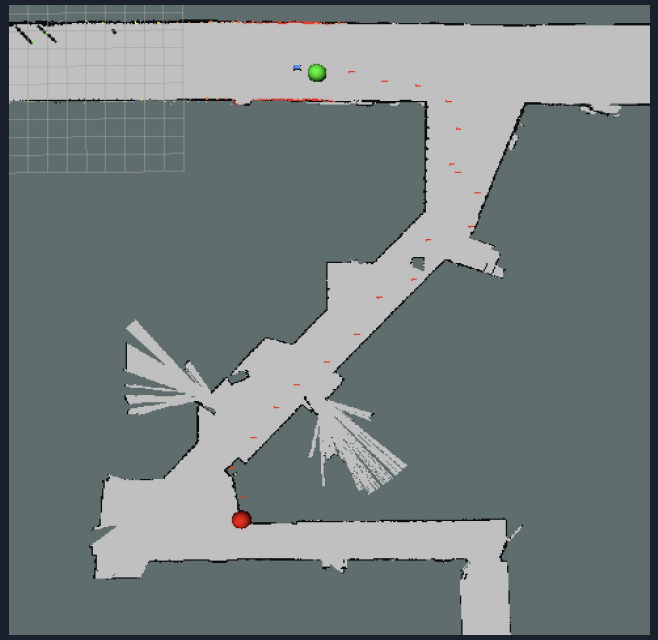


k-sPRM was generated from a random sample of N/100 nodes (n = 3k...5k)

Validating Path Finding



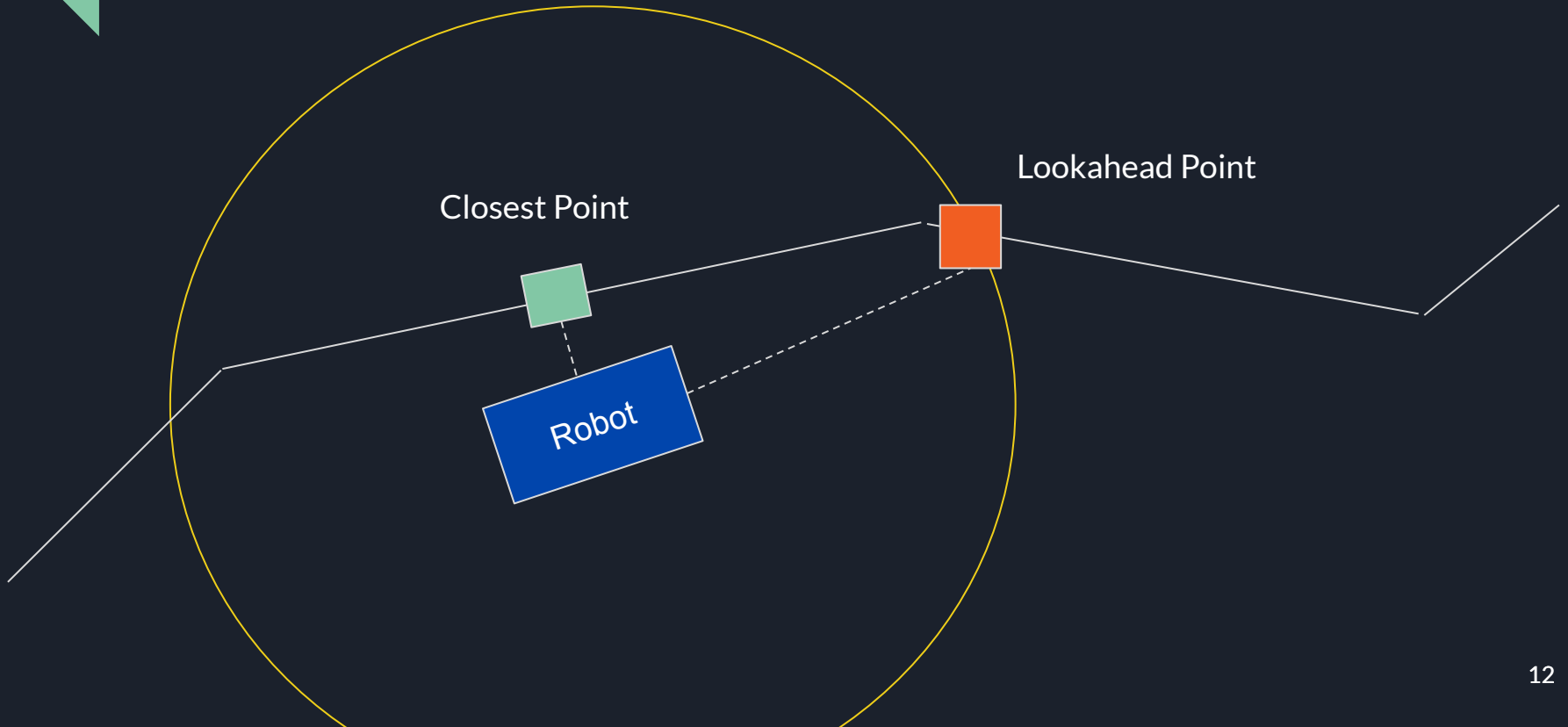
Without PRM



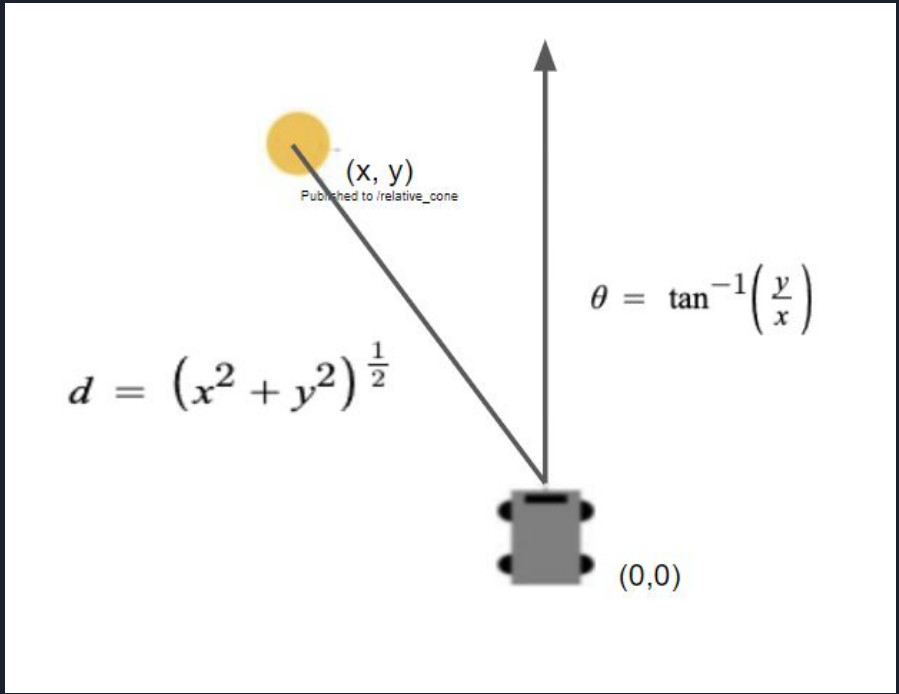
With PRM

Module 2:
Trajectory Follower

Transient Goal is chosen along Trajectory

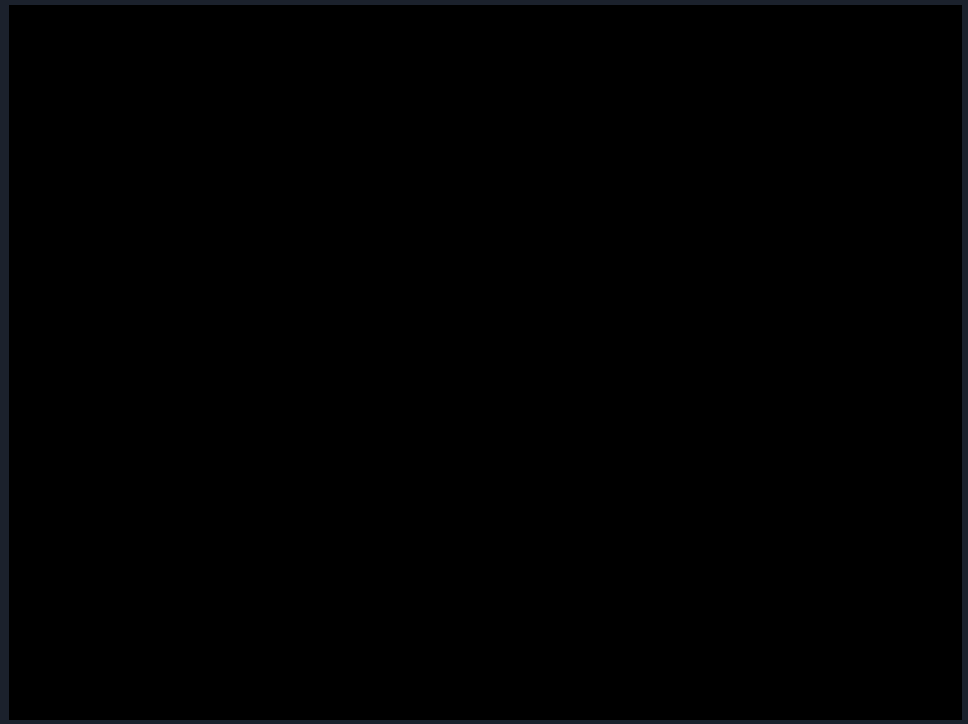


Pure Pursuit follows Transient Goal



$$\delta = \tan^{-1}\left(\frac{2L\sin\eta}{L_1}\right)$$

Path Following (using GT Odometry)



*Pure Pursuit publishes 25 Hz (and odometry at 50 Hz), the **closest point** updates at ~3 Hz

Integration & Next Steps

1. Troubleshoot localization errors
2. Tune random sampling
    - # of nodes ***n***, # of neighbors ***k***
3. Implement in Rviz using localization code
4. Tesse

Takeaways

Technical

- Path Planning Algorithms
  - Used and compared both sampling and search based algorithms
- Integration: Still needs some tuning

Communications

- Working in pairs >> Working alone
- Asynchronous communication (doc and comments)
- Nomenclature and naming conventions

Thank You

Questions?

References

- Sertac Karaman. “Sampling-based Algorithms for Optimal Motion Planning”. May 2011
- https://www.mathworks.com/help/examples/robotics/win64/PathPlanningExample_03.png