UCLA EE183DA Lab4 Summarization of Key Contribution

Trajectory Planning

RRT Algorithm

- Disadvantage: Path generated is not optimal

Algorithm 1: Pseudo Code for RRT Algorithm [1][3]

```
INITIALIZATION

$\mathbb{V} = \text{InsertNode}(x_i, \mathbb{V})$ ;                     // Initialize $\mathbb{V} = \{x_i\}$
$\mathbb{E} = \text{InsertEdge}(\emptyset, \mathbb{E})$ ;                     // Initialize $\mathbb{E} = \{\emptyset\}$
$\mathbb{T} = (\mathbb{V}, \mathbb{E})$ ;                                    // Tree = (Vertices,Edges)

RRT ALGORITHM

while $x \neq x_f$ do
    $x_{rand} \leftarrow \text{Sample}(x)$ ;                               // Sample $\in \mathbb{C}$
    $x_{nearest} \leftarrow \text{Nearest}(\mathbb{V}, x_{rand})$ ;                // Closest $x \in \mathbb{V}$
    $x_{new} \leftarrow \text{Drive}(x_{nearest}, x_{rand})$ ;    // Drive from $x_{nearest}$ to $x_{rand}$
    if No Collision then
        InsertNode $(x_{min}, x_{new}, \mathbb{V})$ ;                     // Add $x_{new}$ to $\mathbb{V}$
        InsertEdge $(p_{|x_{min}}, p_{new}, \mathbb{E})$ ;                 // Add $p_{new}$ to $\mathbb{E}$
    return $\mathbb{T}$
```

RRT* Algorithm

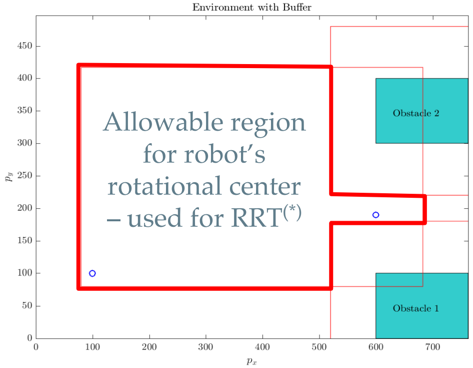
- Advantage: Improve path quality by introducing tree rewiring and neighbor search
- Disadvantage: Longer execution time and slower path convergence rate

Algorithm 2: Pseudo Code for RRT* Algorithm [1][3]

```
INITIALIZATION

$\mathbb{V} = \{x_i\}$ ;                                  // Initialize $\mathbb{V}$ = Add $x_i$ to $\mathbb{V}$
$\mathbb{E} = \{\emptyset\}$ ;                                      // Initialize $\mathbb{E} = \emptyset$
$\mathbb{T} = (\mathbb{V}, \mathbb{E})$ ;                                  // Tree = (Vertices,Edges)

RRT* ALGORITHM

while $x \neq x_f$ do
    for $i = 0$ to $i = N$ do
        $x_{rand} \leftarrow \text{Sample}(x)$ ;                           // Sample $\in \mathbb{C}$
        $x_{nearest} \leftarrow \text{Nearest}(\mathbb{V}, x_{rand})$ ;            // Closest $x \in \mathbb{V}$
        $x_{new} \leftarrow \text{Drive}(x_{nearest}, x_{rand})$ ;  // Drive from $x_{nearest}$ to $x_{rand}$
        if No Collision then
            $x_{near} \leftarrow \text{Near}(\mathbb{V}, x_{new}, |\mathbb{E}|)$ ;      // Return nearby $x \in \mathbb{V}$
            $x_{min} \leftarrow \text{SelectParent}(x_{near}, x_{nearest}, x_{new})$ ;  // Select best parent
             $x_{new} \in x_{near}$
            InsertNode $(x_{min}, x_{new}, \mathbb{V})$ ;                 // Add $x_{new}$ to $\mathbb{V}$
            InsertEdge $(p_{|x_{min}}, p_{new}, \mathbb{E})$ ;             // Add $p_{new}$ to $\mathbb{E}$
            $\mathbb{T} \leftarrow \text{Rewire}(\mathbb{T}, x_{near}, x_{min}, x_{new})$ ;     // Rewire the tree
    return $\mathbb{T}$
```

Robot Boundary Model

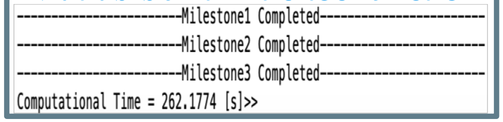
Buffer given to the box and all obstacles to avoid collision



Allowable region for robot's rotational center – used for RRT(*)

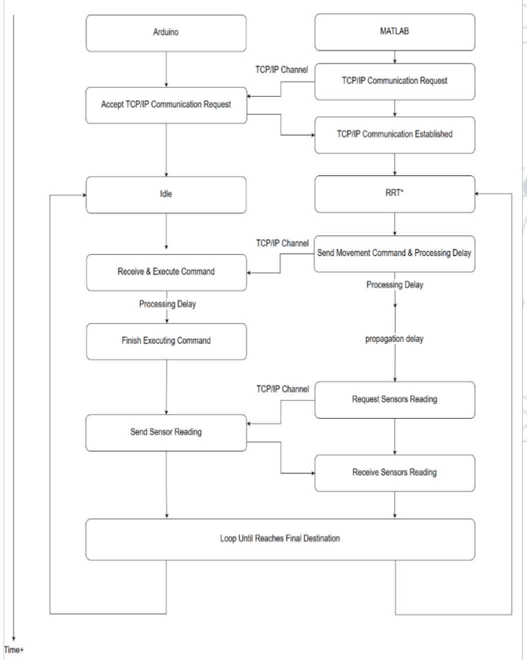
Computational Efficiency

Matlab built in tic-toc function



```
-----------------------Milestone1 Completed-----------------------
-----------------------Milestone2 Completed-----------------------
-----------------------Milestone3 Completed-----------------------
Computational Time = 262.1774 [s]>>
```

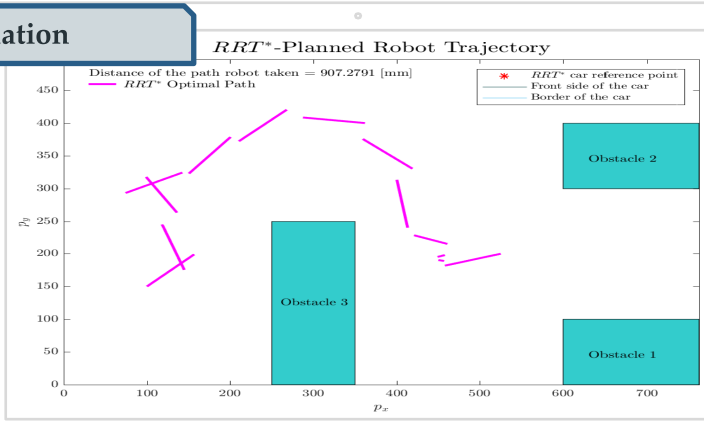
Matlab – Arduino Wireless Communication



Evaluation

Goal

- 8 performance cases are chosen and done for different evaluation discussions



Trajectory Plot

- RRT* optimal car trajectory shown
- Distance of the path robot taken shown

Iou-Sheng (Danny) Chang ▸ UID: 804743003