Autonomous and Mobile Robots

Kinodynamic motion planning for steerable WMRs

Department of Computer, Control, and Management Engineering Antonio Ruberti

Sapienza
Università di Roma

Professor:
Giuseppe Oriolo

Supervisor:
Michele Cipriani

Students:
Dario Basile 1845115
Lorenzo Ciarpaglini 1813738
Luca Del Signore 2096442

Introduction

- In recent years mobile robots with steering wheels are widely used in autonomous navigation, logistics, and service robotics applications.

- Path planning in complex and dynamic environments requires advanced algorithms that ensure safety and precision.

- KRRT* (Kinodynamic Rapidly-exploring Random Tree Star) offers significant advantages over traditional methods in terms of computational efficiency and path quality

Differences with other mobile platforms

Car-like Robots:

- Low production cost
- High load capacity
- Lack of maneuverability

Omnidirectional Robots with unconventional wheels:

- High production cost
- Low load capacity
- High maneuverability

SWM Robots:

- Low production cost
- High load capacity
- Higher maneuverability

Contents

- Kinematic Model for an off-centered SWMR
- Kinodynamic RRT* motion planning
- Simulations and Results on different maps
- Analysis on efficiency and drawbacks

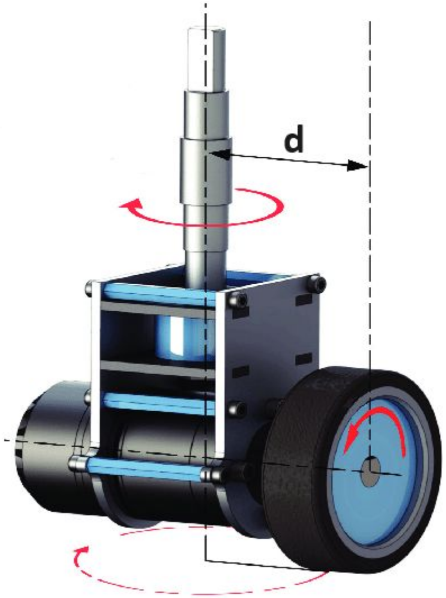
Assumptions

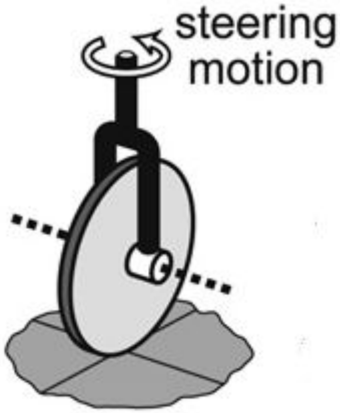
- Localization Precision in Simulation
- Absence of Friction or External Forces
- Pure Rolling Constraints
- Rigid Body Assumption
- A Completely Known Environment

Neobotix MPO-700

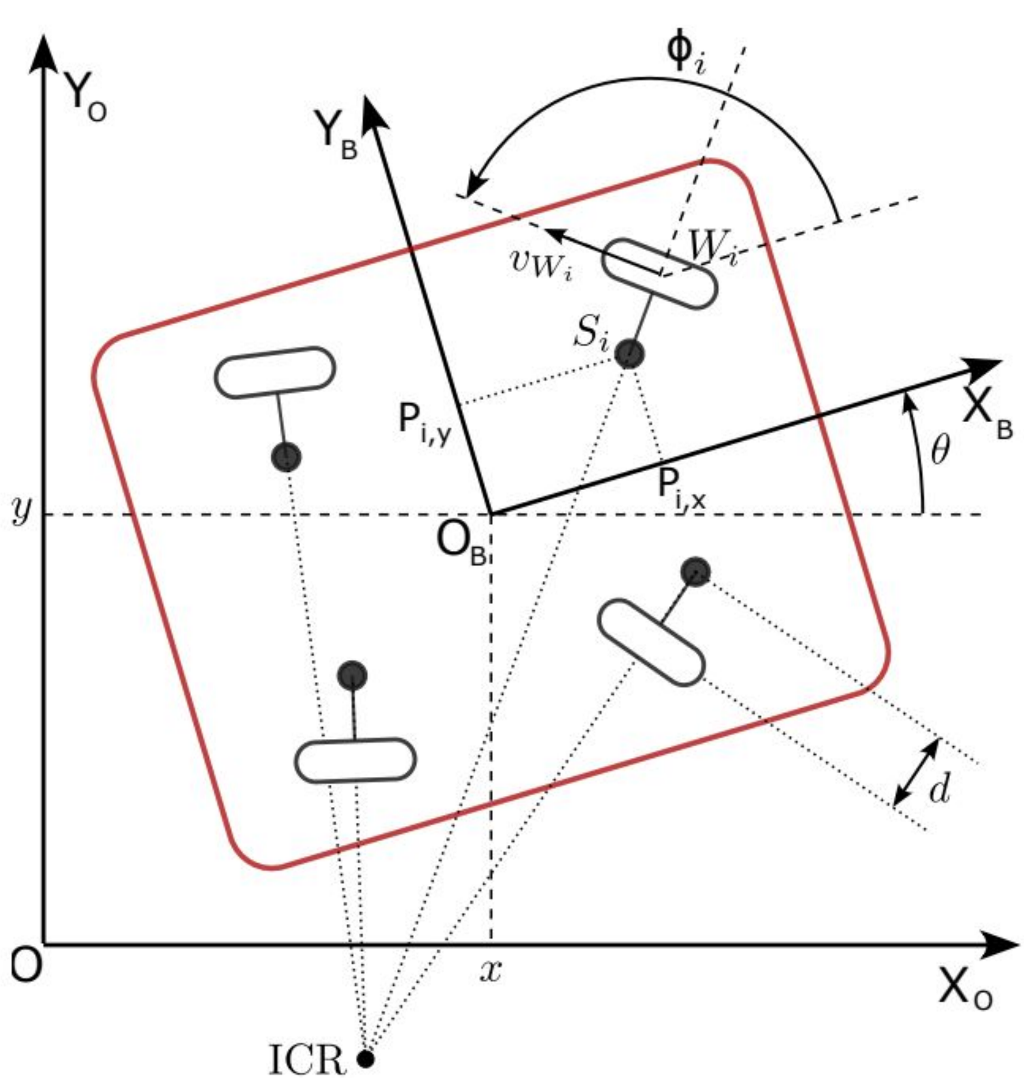
Neobotix MPO-700

- 4 steerable independent actuated wheels
- Omnidirectionality





Schematic model of MPO-700



$$\boldsymbol{\xi} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \in SE(2)$$

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{\xi}^T & \boldsymbol{\phi}^T \end{bmatrix}$$

Position of S_i

$$\mathbf{o}_{S_i} = \begin{bmatrix} x \\ y \end{bmatrix} + \mathbf{R}(\theta) \begin{bmatrix} P_{i,x} \\ P_{i,y} \end{bmatrix}$$

Position of W_i

$$\mathbf{o}_{W_i} = \mathbf{o}_{S_i} + \mathbf{R}(\theta + \phi_i) \begin{bmatrix} 0 \\ -d \end{bmatrix}$$

Model

Pfaffian constraints

Each wheels is subject to the pure rolling constraint

$$\begin{bmatrix} -\sin(\theta + \phi_i) \\ \cos(\theta + \phi_i) \end{bmatrix}^T \dot{\mathbf{o}}_{W_i} = 0$$

Implicit model

$$\begin{bmatrix} -\sin(\theta+\phi_1) & \cos(\theta+\phi_1) & \Delta_1 & 0 & \cdots & 0 \\ -\sin(\theta+\phi_2) & \cos(\theta+\phi_2) & \Delta_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -\sin(\theta+\phi_{n_s}) & \cos(\theta+\phi_{n_s}) & \Delta_{n_s} & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \vdots \\ \dot{\phi}_{n_s} \end{bmatrix} = 0$$

where

$$\Delta_i = P_{i,x}\cos(\phi_i) + P_{i,y}\sin(\phi_i)$$

Geometric constraints

- All axles of the wheels must intersect at a single point (ICR) or the platform will not move.
- ICR can be seen as a geometric constraint
- ICR must depend on the geometrical path $\boldsymbol{\xi}(t)$

Resolving i-th Pfaffian constraint for ϕ_i we obtain this geometric constraint:

$$\phi_i = h_i(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \arctan\left(\frac{-\sin\theta\dot{x} + \cos\theta\dot{y} + P_{i,x}\dot{\theta}}{\cos\theta\dot{x} + \sin\theta\dot{y} - P_{i,y}\dot{\theta}}\right)$$

A general kinematic model

The implicit kinematic model can be reduced

$$\begin{bmatrix} -\sin(\theta+\phi_1) & \cos(\theta+\phi_1) & \Delta_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \end{bmatrix} = 0$$

$$\boldsymbol{q} = \begin{bmatrix} x & y & \theta & \phi_1 \end{bmatrix}^T$$

To be used with these geometric constraints:

$$\phi_i = h_i(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}), \quad i = 2, \ldots, n_s$$

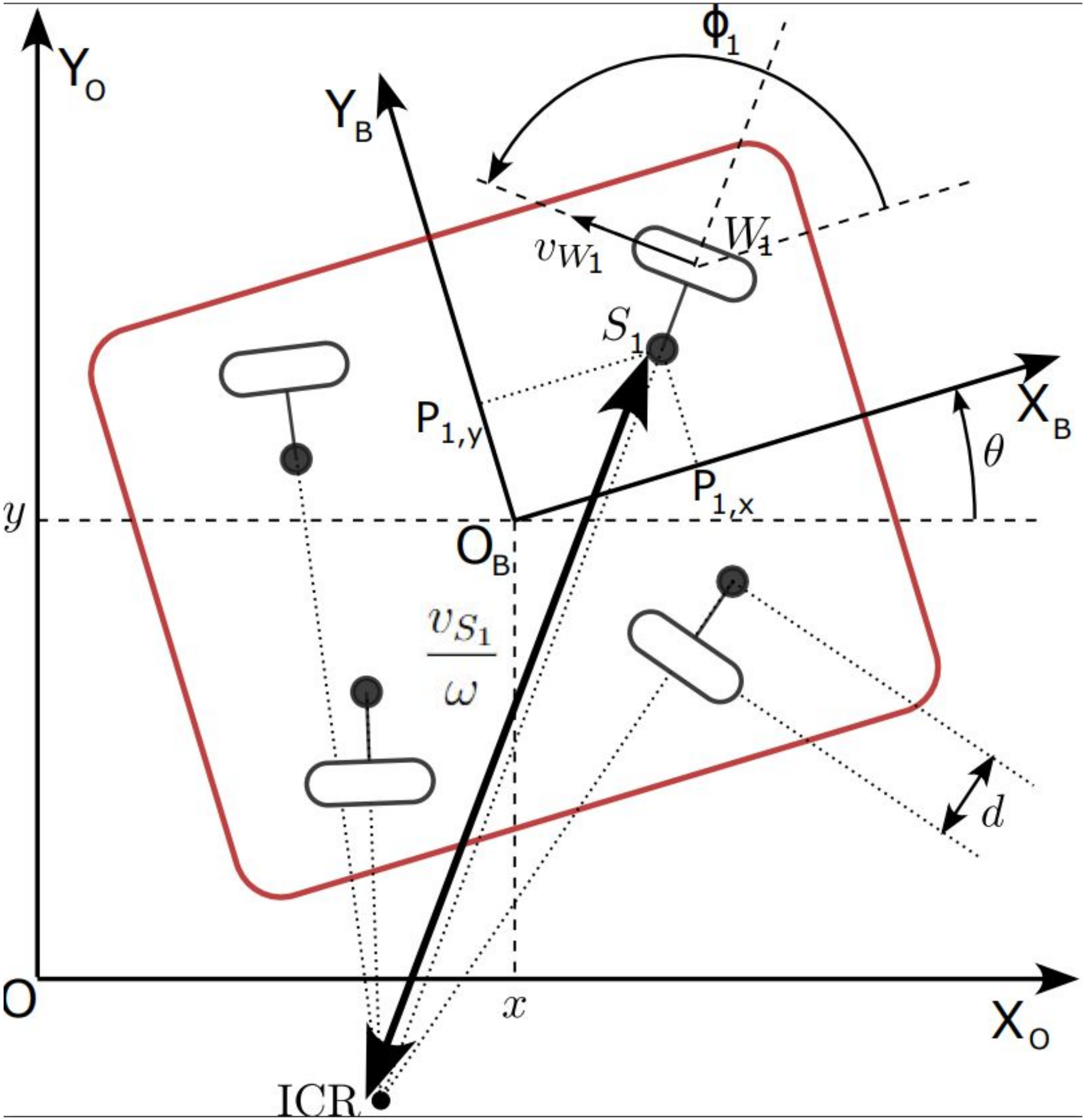
Explicit general kinematic model

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta + \phi_1) \\ \sin(\theta + \phi_1) \\ 0 \\ 0 \end{bmatrix} v_{S_1} + \begin{bmatrix} P_{1,x}\sin(\theta) + P_{1,y}\cos(\theta) \\ -P_{1,x}\cos(\theta) + P_{1,y}\sin(\theta) \\ 1 \\ 0 \end{bmatrix} \omega + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_{\phi_1}$$

- v_{S_1} velocity of the point S_1 in the wheel frame
- ω angular velocity of the platform
- v_{ϕ_1} steering velocity of the wheel W_1

The driving velocity of W_1 can be evaluated as:

$$v_{W_i} = v_{S_i} + d(\dot{\theta} + \dot{\phi}_i)$$



ICR on the W_1 axes with distance v_{S_1}/ω from the joint S_1

Coordinating functions (1)

$$\dot{x} = v_{S_1}\cos(\theta+\phi_1) + \omega(P_{1,x}\sin\theta + P_{1,y}\cos\theta)$$
$$\dot{y} = v_{S_1}\sin(\theta+\phi_1) + \omega(-P_{1,x}\cos\theta + P_{1,y}\sin\theta)$$
$$\dot{\theta} = \omega$$
$$\dot{\phi}_1 = v_{\phi_1}$$

$$\Downarrow$$

$$\phi_i = h_i(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}}) = \arctan\left(\frac{-\sin\theta\dot{x} + \cos\theta\dot{y} + P_{i,x}\dot{\theta}}{\cos\theta\dot{x} + \sin\theta\dot{y} - P_{i,y}\dot{\theta}}\right)$$

$$=$$

$$\phi_i = h_i(v_{S_1}, \omega, \phi_1) = \arctan\left(\frac{v_{S_1}\sin\phi_1 + \omega(P_{i,x} - P_{1,x})}{v_{S_1}\cos\phi_1 + \omega(P_{1,y} - P_{i,y})}\right)$$

Coordinating function (2)

$$\phi_i = h_i(v_{S_1}, \omega, \phi_1) = \arctan\left(\frac{v_{S_1}\sin\phi_1 + \omega(P_{i,x} - P_{1,x})}{v_{S_1}\cos\phi_1 + \omega(P_{1,y} - P_{i,y})}\right)$$

- This functions return two solutions. We need to pick that closer to the previous ϕ_i
- Structural singularities when ICR is located on i-th joint, and the ϕ_i can have arbitrary orientation.

Imposing velocity of S_i in the wheel frame W_i different from zero during the path in order to prevent the joints to be ICR

$$v_{S_i} = \begin{bmatrix}\cos(\theta+\phi_i)\\ \sin(\theta+\phi_i)\end{bmatrix}^T \dot{\mathbf{o}}_{S_i} \neq 0$$

Dynamic extension of general kinematic model

$$\dot{v}_{S_1} = a_{S_1},$$
$$\dot{\omega} = a_{\omega}$$

$$\Rightarrow$$

$$\dot{x} = v_{S1}\cos(\theta + \phi_1) + \omega(P_{1,x}\sin\theta + P_{1,y}\cos\theta),$$
$$\dot{y} = v_{S1}\sin(\theta + \phi_1) + \omega(-P_{1,x}\cos\theta + P_{1,y}\sin\theta),$$
$$\dot{\theta} = \omega,$$
$$\dot{\phi}_1 = v_{\phi_1},$$
$$\dot{v}_{S_1} = a_{S_1},$$
$$\dot{\omega} = a_{\omega}$$

$$\boldsymbol{u} = \left[v_{\phi_1}, a_{S_1}, a_{\omega}\right]^T$$

$$\boldsymbol{q} = \left[x, y, \theta, \phi_1, v_{S_1}, \omega\right]^T$$

Previous model controlled directly velocity of S_1 and angular velocity of the platform, with no free evolution of the state.

Why this extension

1. It's easier to calculate the derivative of coordinating functions (necessary for driving velocity of each wheel) having in the model directly derivative of v_{S_1} and ω
2. As we will see later, the linearization of this dynamic model is controllable, that is a necessary condition for KRRT*.

Driving velocity of wheel W_i :

$$v_{W_i} = v_{S_i} + d(\dot{\theta} + \dot{\phi}_i)$$

where

$$\phi_i = h_i(v_{S_1}, \omega, \phi_1) = \arctan\left(\frac{v_{S_1}\sin\phi_1 + \omega(P_{i,x} - P_{1,x})}{v_{S_1}\cos\phi_1 + \omega(P_{1,y} - P_{i,y})}\right)$$

KRRT*

Motion Planning

Find a collision-free motion that takes the robot from an initial to a final configuration

KRRT*

- Probabilistic method
- Robot with linear dynamics and with non-holonomic constraints
- Asymptotic optimality

RRT (Rapidly-exploring Random Tree)

- Probabilistic method
  - utilize random sampling to construct a roadmap of the configuration space as a Tree T structure where the root is the initial configuration
- Probabilistically complete
- Single query
- Exploration biased towards larger Voronoi regions

RRT*

- RRT extension
- Use of a cost function
- Update the costs in the tree at each iteration
- Asymptotically optimal

1. Select the closest nodes $\mathbf{Q}_{\text{neighbors}}$ to $\mathbf{q}_{\text{new}}$ in the Tree $\mathcal{T}$, using a fixed radius r.
2. Choose the parent $\mathbf{q}_{\text{parent}}$ of $\mathbf{q}_{\text{new}}$ between $\mathbf{Q}_{\text{neighbors}}$ as the node with lower cost= $C^*(\mathbf{q}_{\text{parent}}) + c(\mathbf{q}_{\text{parent}}, \mathbf{q}_{\text{new}})$ and such that the segment between $\mathbf{q}_{\text{parent}}$ and $\mathbf{q}_{\text{new}}$ is free of collisions.
3. For each node $\mathbf{q}_{\text{near}}$ in $\mathbf{Q}_{\text{neighbors}}$ if its current cost is higher than the cost of reaching it passing from the newly added node (and the segment between the two is free of collision) update its parent and its cost.
4. Add $\mathbf{q}_{\text{new}}$ to $\mathcal{T}$.

KRRT*

- RRT* extension for more complex dynamics
- Builds a tree of trajectories in the free configuration space, rooted in the initial configuration
- Optimall connection of any pair of states, with a fixed-final-state free-final-time controller
- Probabilistic complete
- Asymptotically optimal

Algorithm for optimally connecting two states

Applied to controllable linear system of this type:

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{c}$$

Definition of trajectory of the system

$$\boldsymbol{\pi} = (\boldsymbol{q}(), \boldsymbol{u}(), \tau, c)$$

τ is the total time of the trajectory

$\boldsymbol{q} : [0, \tau] \rightarrow \boldsymbol{Q}$ where $\boldsymbol{Q} = \mathbb{R}^n$

$\boldsymbol{u} : [0, \tau] \rightarrow \boldsymbol{U}$ where $\boldsymbol{U} = \mathbb{R}^m$

Cost of trajectory:

$$c(\boldsymbol{\pi}) = \int_0^{\tau} (1 + \boldsymbol{u}(t)^T \boldsymbol{R} \boldsymbol{u}(t))\, dt \qquad \boldsymbol{R} \in \mathbb{R}^{m \times m}$$

Objective:

Given q_0 and q_1 and a controllable linear system, Finding the best trajectory connetting q_0 and q_1 that minimize the cost function c* both in terms of control efforts and time τ^* that are in a trade off relationship

For arbitrary τ the best control policy is:

$$\boldsymbol{u}(t) = \boldsymbol{R}^{-1}\boldsymbol{B}^T \exp\left(\boldsymbol{A}^T(\tau - t)\right)\boldsymbol{d}(\tau)$$

where

$$\boldsymbol{d}(\tau) = \boldsymbol{G}(\tau)^{-1}(\boldsymbol{q_1} - \bar{\boldsymbol{q}}(\tau))$$

Differential equations

G(t) (called weighted controllability matrix) is the solution to this Lyapunov equation:

$$\dot{\boldsymbol{G}}(t) = \boldsymbol{A}\boldsymbol{G}(t) + \boldsymbol{G}(t)\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T, \quad \boldsymbol{G}(0) = \boldsymbol{0}.$$

If the system is controllable, then G(t) is always positive-definite and then invertible

$\bar{q}(t)$ (called free evolution of the state) is the solution to this differential equation:

$$\dot{\bar{\boldsymbol{q}}}(t) = \boldsymbol{A}\bar{\boldsymbol{q}}(t) + \boldsymbol{c}, \quad \bar{\boldsymbol{q}}(0) = \boldsymbol{q}_0.$$

How to find τ^*

$$\boldsymbol{u}(t) = \boldsymbol{R}^{-1}\boldsymbol{B}^T \exp\left(\boldsymbol{A}^T(\tau - t)\right)\boldsymbol{d}(\tau)$$

$$\Downarrow$$

$$c(\boldsymbol{\pi}) = \int_0^\tau \left(1 + \boldsymbol{u}(t)^T \boldsymbol{R}\boldsymbol{u}(t)\right) dt$$

$$=$$

$$c(\tau) = \tau + (\boldsymbol{q_1} - \bar{\boldsymbol{q}}(\tau))^T \boldsymbol{d}(\tau)$$

$$\boldsymbol{d}(\tau) = \boldsymbol{G}(\tau)^{-1}(\boldsymbol{q_1} - \bar{\boldsymbol{q}}(\tau))$$

Properties of $c(\tau)$

$$c(\tau) = \tau + (\boldsymbol{q_1} - \bar{\boldsymbol{q}}(\tau))^T \boldsymbol{d}(\tau)$$

$$\boldsymbol{d}(\tau) = \boldsymbol{G}(\tau)^{-1}(\boldsymbol{q_1} - \bar{\boldsymbol{q}}(\tau))$$

- $c(\tau)$ can have multiple local minima
- $c(\tau)$> τ for all τ>0 being G(t) positive-definite

$c(\tau)$= τ + positive num.

We can study this function to find τ^* that minimizes cost function

Global minima of $c(\tau)$

To find the value of τ that globally minimizes the cost function, we can utilize the secondary properties of $c(\tau)$ to monitor the value of c* during the forward integration loop of $\dot{\boldsymbol{G}}(t)$ and $\dot{\bar{\boldsymbol{q}}}(t)$ using the 4th-order Runge-Kutta method. The value c* is guaranteed to be a global minimum when the iterative τ exceeds c* and then can no longer decrease.

Algorithm 1 Forward Integration of $\dot{\boldsymbol{G}}(t)$ and $\dot{\bar{\boldsymbol{q}}}(t)$

$\tau, k = 0$
$\boldsymbol{G}[0] = \boldsymbol{0}$
$\bar{\boldsymbol{q}}[0] = \boldsymbol{q_0}$
$c^* = \infty$
$\Delta t = 0.1$
$\mathrm{h}=\Delta t$
while $\tau < c^*$ **do**
  $\tau \leftarrow \tau + \Delta t$
  $\mathrm{k} \leftarrow \mathrm{k} + 1$
  $\bar{\boldsymbol{q}}[\mathrm{k}] \leftarrow \bar{\boldsymbol{q}}[\mathrm{k}\text{-}1] + \mathrm{RK4}(\dot{\bar{\boldsymbol{q}}}, \bar{\boldsymbol{q}}[\mathrm{k}\text{-}1], \mathrm{h})$
  $\boldsymbol{G}[\mathrm{k}] \leftarrow \boldsymbol{G}[\mathrm{k}\text{-}1] + \mathrm{RK4}(\dot{\boldsymbol{G}}, \boldsymbol{G}[\mathrm{k}\text{-}1], \mathrm{h})$
  $c[\mathrm{k}] \leftarrow \tau + (\boldsymbol{q_1} - \bar{\boldsymbol{q}}[\mathrm{k}])^T \boldsymbol{G}[\mathrm{k}]^{-1} (\boldsymbol{q_1} - \bar{\boldsymbol{q}}[\mathrm{k}])$
  if $c[\mathrm{k}] < c^*$ **then**
    $c^* \leftarrow c[\mathrm{k}]$
    $\tau^* \leftarrow \tau$
    $k^* \leftarrow k$
  end if
end while
$\boldsymbol{d}(\tau^*) = \boldsymbol{G}[k^*]^{-1}(\boldsymbol{q_1} - \bar{\boldsymbol{q}}[k^*])$

- Estimation of G(t) $\bar{\boldsymbol{q}}(t)$ and c(t) every $k\Delta t$ seconds
- Monitoring step-by-step value of c(t) and c*
- h is the interval of time used in 4-th order Runge kutta method
- Global minimal when τ>c* found until now

$$\boldsymbol{u}(t) = \boldsymbol{R}^{-1}\boldsymbol{B}^T \underbrace{\exp\left(\boldsymbol{A}^T(\tau^* - t)\right)\boldsymbol{d}(\tau^*)}_{y(t)}$$

y(t) is the solution to this differential equation:

$$\dot{\boldsymbol{y}}(t) = -\boldsymbol{A}^T\boldsymbol{y}(t), \quad \boldsymbol{y}(\tau^*) = \boldsymbol{d}(\tau^*)$$

Substituting the optimal control policy in the linear system:

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}\boldsymbol{q}(t) + \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{y}(t) + \boldsymbol{c}, \quad \boldsymbol{q}(\tau^*) = \boldsymbol{q}_1$$

Composite differential equation

$$\begin{bmatrix} \dot{\boldsymbol{q}}(t) \\ \dot{\boldsymbol{y}}(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T \\ \boldsymbol{0} & -\boldsymbol{A}^T \end{bmatrix} \begin{bmatrix} \boldsymbol{q}(t) \\ \boldsymbol{y}(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}(\tau^*) \\ \boldsymbol{y}(\tau^*) \end{bmatrix} = \begin{bmatrix} \boldsymbol{q_1} \\ \boldsymbol{d}(\tau^*) \end{bmatrix}$$

Integration backward in time through 4-th order Runge kutta method to obtain q(t) and y(t) estimated every $k\Delta t$ seconds

$$\boldsymbol{u}(t) = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{y}(t)$$

Wrap up of the algorithm

Input:

- Linear Controllable system
- q_0 and q_1

Output:

Optimal trajectory in terms of time and control effort

$$\boldsymbol{\pi}^* = (\boldsymbol{q}(t), \boldsymbol{u}(t), \tau^*, c^*)$$

where

$$\boldsymbol{q}(0) = \boldsymbol{q_0} \text{ and } \boldsymbol{q}(\tau^*) = \boldsymbol{q_1}.$$

KRRT* algorithm

Algorithm 2 Kinodynamic RRT*

```
$\boxed{\mathcal{T} \leftarrow \{\mathbf{q}_{\text{start}}\}}$
$\mathbf{q}_{\text{start}} \in \mathbf{Q}_{\text{free}}, \mathbf{q}_{\text{goal}} \in \mathbf{Q}_{\text{free}}$
for $i = 1, \infty$ do
    Randomly sample $\mathbf{q}_{\mathbf{i}} \in \mathbf{Q}_{\text{free}}$
    $\mathbf{q} \leftarrow \arg\min_{\{\mathbf{q} \in \mathcal{T} | c^*[\mathbf{q}, \mathbf{q}_{\mathbf{i}}] < r \wedge \text{COLLISIONFREE}[\pi^*[\mathbf{q}, \mathbf{q}_{\mathbf{i}}]]\}} (\text{cost}[\mathbf{q}] + c^*[\mathbf{q}, \mathbf{q}_{\mathbf{i}}])$
    parent$[\mathbf{q}_{\mathbf{i}}] \leftarrow \mathbf{q}$
    cost$[\mathbf{q}_{\mathbf{i}}] \leftarrow$ cost$[\mathbf{q}] + c^*[\mathbf{q}, \mathbf{q}_{\mathbf{i}}]$
    for all $\{\mathbf{q} \in \mathcal{T} \cup \{\mathbf{q}_{\text{goal}}\}\} | c^*[\mathbf{q}_{\mathbf{i}}, \mathbf{q}] < r \wedge \text{cost}[\mathbf{q}_{\mathbf{i}}] + c^*[\mathbf{q}_{\mathbf{i}}, \mathbf{q}] < \text{cost}[\mathbf{q}] \wedge$
  COLLISIONFREE$[\pi^*[\mathbf{q}_{\mathbf{i}}, \mathbf{q}]]$ do
        cost$[\mathbf{q}] \leftarrow$ cost$[\mathbf{q}_{\mathbf{i}}] + c^*[\mathbf{q}_{\mathbf{i}}, \mathbf{q}]$
        parent$[\mathbf{q}] \leftarrow \mathbf{q}_{\mathbf{i}}$
    end for
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{q}_{\mathbf{i}}\}$
end for
```

KRRT* algorithm

Algorithm 2 Kinodynamic RRT*

$\mathcal{T} \leftarrow \{\mathbf{q}_{\mathrm{start}}\}$

$\mathbf{q}_{\mathrm{start}} \in \mathbf{Q}_{\mathrm{free}}, \mathbf{q}_{\mathrm{goal}} \in \mathbf{Q}_{\mathrm{free}}$

for $i = 1, \infty$ **do**

RANDOM SAMPLING

  $\boxed{\text{Randomly sample } \mathbf{q_i} \in \mathbf{Q}_{\mathrm{free}}}$

  $\mathbf{q} \leftarrow \arg\min_{\{\mathbf{q} \in \mathcal{T} | c^*[\mathbf{q}, \mathbf{q_i}] < r \wedge \mathrm{COLLISIONFREE}[\pi^*[\mathbf{q}, \mathbf{q_i}]]\}} (\mathrm{cost}[\mathbf{q}] + c^*[\mathbf{q}, \mathbf{q_i}])$

  $\mathrm{parent}[\mathbf{q_i}] \leftarrow \mathbf{q}$

  $\mathrm{cost}[\mathbf{q_i}] \leftarrow \mathrm{cost}[\mathbf{q}] + c^*[\mathbf{q}, \mathbf{q_i}]$

  for all $\{\mathbf{q} \in \mathcal{T} \cup \{\mathbf{q}_{\mathrm{goal}}\}\} | c^*[\mathbf{q_i}, \mathbf{q}] < r \wedge \mathrm{cost}[\mathbf{q_i}] + c^*[\mathbf{q_i}, \mathbf{q}] < \mathrm{cost}[\mathbf{q}] \wedge$ $\mathrm{COLLISIONFREE}[\pi^*[\mathbf{q_i}, \mathbf{q}]]$ **do**

    $\mathrm{cost}[\mathbf{q}] \leftarrow \mathrm{cost}[\mathbf{q_i}] + c^*[\mathbf{q_i}, \mathbf{q}]$

    $\mathrm{parent}[\mathbf{q}] \leftarrow \mathbf{q_i}$

  end for

  $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{q_i}\}$

end for

KRRT* algorithm

Algorithm 2 Kinodynamic RRT*

```
T ← {q_start}
q_start ∈ Q_free, q_goal ∈ Q_free
for i = 1, ∞ do
    Randomly sample q_i ∈ Q_free
```

PARENT SELECTION

$\mathbf{q} \leftarrow \arg\min_{\{\mathbf{q} \in \mathcal{T} \mid c^*[\mathbf{q}, \mathbf{q_i}] < r \wedge \text{COLLISIONFREE}[\pi^*[\mathbf{q}, \mathbf{q_i}]]\}} (\text{cost}[\mathbf{q}] + c^*[\mathbf{q}, \mathbf{q_i}])$

$\text{parent}[\mathbf{q_i}] \leftarrow \mathbf{q}$

$\text{cost}[\mathbf{q_i}] \leftarrow \text{cost}[\mathbf{q}] + c^*[\mathbf{q}, \mathbf{q_i}]$

for all $\{\mathbf{q} \in \mathcal{T} \cup \{\mathbf{q_{goal}}\}\} \mid c^*[\mathbf{q_i}, \mathbf{q}] < r \wedge \text{cost}[\mathbf{q_i}] + c^*[\mathbf{q_i}, \mathbf{q}] < \text{cost}[\mathbf{q}] \wedge \text{COLLISIONFREE}[\pi^*[\mathbf{q_i}, \mathbf{q}]]$ **do**

$\quad \text{cost}[\mathbf{q}] \leftarrow \text{cost}[\mathbf{q_i}] + c^*[\mathbf{q_i}, \mathbf{q}]$

$\quad \text{parent}[\mathbf{q}] \leftarrow \mathbf{q_i}$

end for

$\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{q_i}\}$

end for

KRRT* algorithm

Algorithm 2 Kinodynamic RRT*

$\mathcal{T} \leftarrow \{\mathbf{q}_{\text{start}}\}$

$\mathbf{q}_{\text{start}} \in \mathbf{Q}_{\text{free}}, \mathbf{q}_{\text{goal}} \in \mathbf{Q}_{\text{free}}$

for $i = 1, \infty$ **do**

  Randomly sample $\mathbf{q_i} \in \mathbf{Q}_{\text{free}}$

  $\mathbf{q} \leftarrow \arg\min_{\{\mathbf{q} \in \mathcal{T} | c^*[\mathbf{q}, \mathbf{q_i}] < r \wedge \text{COLLISIONFREE}[\pi^*[\mathbf{q}, \mathbf{q_i}]]\}} (\text{cost}[\mathbf{q}] + c^*[\mathbf{q}, \mathbf{q_i}])$

  $\text{parent}[\mathbf{q_i}] \leftarrow \mathbf{q}$

  $\text{cost}[\mathbf{q_i}] \leftarrow \text{cost}[\mathbf{q}] + c^*[\mathbf{q}, \mathbf{q_i}]$

REWIRING

  for all $\{\mathbf{q} \in \mathcal{T} \cup \{\mathbf{q}_{\text{goal}}\}\} | c^*[\mathbf{q_i}, \mathbf{q}] < r \wedge \text{cost}[\mathbf{q_i}] + c^*[\mathbf{q_i}, \mathbf{q}] < \text{cost}[\mathbf{q}] \wedge$ $\text{COLLISIONFREE}[\pi^*[\mathbf{q_i}, \mathbf{q}]]$ **do**

    $\text{cost}[\mathbf{q}] \leftarrow \text{cost}[\mathbf{q_i}] + c^*[\mathbf{q_i}, \mathbf{q}]$

    $\text{parent}[\mathbf{q}] \leftarrow \mathbf{q_i}$

  end for

  $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{q_i}\}$

end for

KRRT* algorithm

Algorithm 2 Kinodynamic RRT*

$\mathcal{T} \leftarrow \{\mathbf{q}_{\text{start}}\}$

$\mathbf{q}_{\text{start}} \in \mathbf{Q}_{\text{free}}, \mathbf{q}_{\text{goal}} \in \mathbf{Q}_{\text{free}}$

for $i = 1, \infty$ **do**

  Randomly sample $\mathbf{q_i} \in \mathbf{Q}_{\text{free}}$

  $\mathbf{q} \leftarrow \arg\min_{\{\mathbf{q} \in \mathcal{T} | c^*[\mathbf{q}, \mathbf{q_i}] < r \wedge \text{COLLISIONFREE}[\pi^*[\mathbf{q}, \mathbf{q_i}]]\}} (\text{cost}[\mathbf{q}] + c^*[\mathbf{q}, \mathbf{q_i}])$

  $\text{parent}[\mathbf{q_i}] \leftarrow \mathbf{q}$

  $\text{cost}[\mathbf{q_i}] \leftarrow \text{cost}[\mathbf{q}] + c^*[\mathbf{q}, \mathbf{q_i}]$

  for all $\{\mathbf{q} \in \mathcal{T} \cup \{\mathbf{q}_{\text{goal}}\}\} | c^*[\mathbf{q_i}, \mathbf{q}] < r \wedge \text{cost}[\mathbf{q_i}] + c^*[\mathbf{q_i}, \mathbf{q}] < \text{cost}[\mathbf{q}] \wedge$ $\text{COLLISIONFREE}[\pi^*[\mathbf{q_i}, \mathbf{q}]]$ **do**

    $\text{cost}[\mathbf{q}] \leftarrow \text{cost}[\mathbf{q_i}] + c^*[\mathbf{q_i}, \mathbf{q}]$

    $\text{parent}[\mathbf{q}] \leftarrow \mathbf{q_i}$

  end for

  $\boxed{\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathbf{q_i}\}}$

end for

KRRT* implementation

- Nearest neighbours: full tree
- Exactly arrives at the final configuration
- Collision check:
  - Footprint of the robot not collides with any obstacle
  - Inputs in the admissible ranges
  - The constraints on v_{si} are not violated

Linearization

KRRT* for non-linear systems

Linearization of the original system with first order taylor expansion:

$$\boldsymbol{A} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}(\hat{\boldsymbol{q}}, \boldsymbol{0}), \quad \boldsymbol{B} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}(\hat{\boldsymbol{q}}, \boldsymbol{0}), \quad \boldsymbol{c} = \boldsymbol{f}(\hat{\boldsymbol{q}}, \boldsymbol{0}) - \boldsymbol{A}\hat{\boldsymbol{q}}.$$

where

$$\boldsymbol{f} = \begin{aligned} \dot{x} &= v_{S1}\cos(\theta+\phi_1) + \omega(P_{1,x}\sin\theta + P_{1,y}\cos\theta), \\ \dot{y} &= v_{S1}\sin(\theta+\phi_1) + \omega(-P_{1,x}\cos\theta + P_{1,y}\sin\theta), \\ \dot{\theta} &= \omega, \\ \dot{\phi}_1 &= v_{\phi_1}, \\ \dot{v}_{S_1} &= a_{S_1}, \\ \dot{\omega} &= a_\omega \end{aligned}$$

$$\boldsymbol{u} = \left[v_{\phi_1}, a_{S_1}, a_\omega\right]^T$$

$$\boldsymbol{q} = \left[x, y, \theta, \phi_1, v_{S_1}, \omega\right]^T$$

At the beginning of each iteration re-linearization of the system in $\hat{\boldsymbol{q}}=\mathrm{q_i}$ that is the randomly sampled configuration

Linearization for a generic point

$$
\boldsymbol{A} = \begin{bmatrix}
0 & 0 & -\hat{v}_{S_1} s(\hat{\theta}+\hat{\phi}_1) + \hat{\omega}(P_{1,x} c\hat{\theta} - P_{1,y} s\hat{\theta}) & -\hat{v}_{S_1} s(\hat{\theta}+\hat{\phi}_1) & c(\hat{\theta}+\hat{\phi}_1) & P_{1,y} c\hat{\theta} + P_{1,x} s\hat{\theta} \\
0 & 0 & \hat{v}_{S_1} c(\hat{\theta}+\hat{\phi}_1) + \hat{\omega}(P_{1,y} c\hat{\theta} + P_{1,x} s\hat{\theta}) & \hat{v}_{S_1} c(\hat{\theta}+\hat{\phi}_1) & s(\hat{\theta}+\hat{\phi}_1) & -P_{1,x} c\hat{\theta} + P_{1,y} s\hat{\theta} \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

$$
\boldsymbol{B} = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\quad
\boldsymbol{c} = \begin{bmatrix}
\hat{v}_{S_1}(\hat{\theta}+\hat{\phi}_1) s(\hat{\theta}+\hat{\phi}_1) - \hat{\omega}\hat{\theta}(P_{1,x} c\hat{\theta} - P_{1,y} s\hat{\theta}) \\
-\hat{v}_{S_1}(\hat{\theta}+\hat{\phi}_1) c(\hat{\theta}+\hat{\phi}_1) - \hat{\omega}\hat{\theta}(P_{1,y} c\hat{\theta} + P_{1,x} s\hat{\theta}) \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
$$

Controllability of the resulting linearized system

The controllability matrix is:

$$\mathcal{C} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \boldsymbol{A}^2\boldsymbol{B} & \cdots & \boldsymbol{A}^5\boldsymbol{B} \end{bmatrix}$$

the rank of submatrix $\begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} \end{bmatrix}$ is always 6 if $v_{S_1} \neq 0$

This condition is always satisfied because we already discard the trajectories that do not meet it in COLLISIONFREE function

Therefore, the linear system is controllable as $\mathcal{C}$ consistently maintains full rank

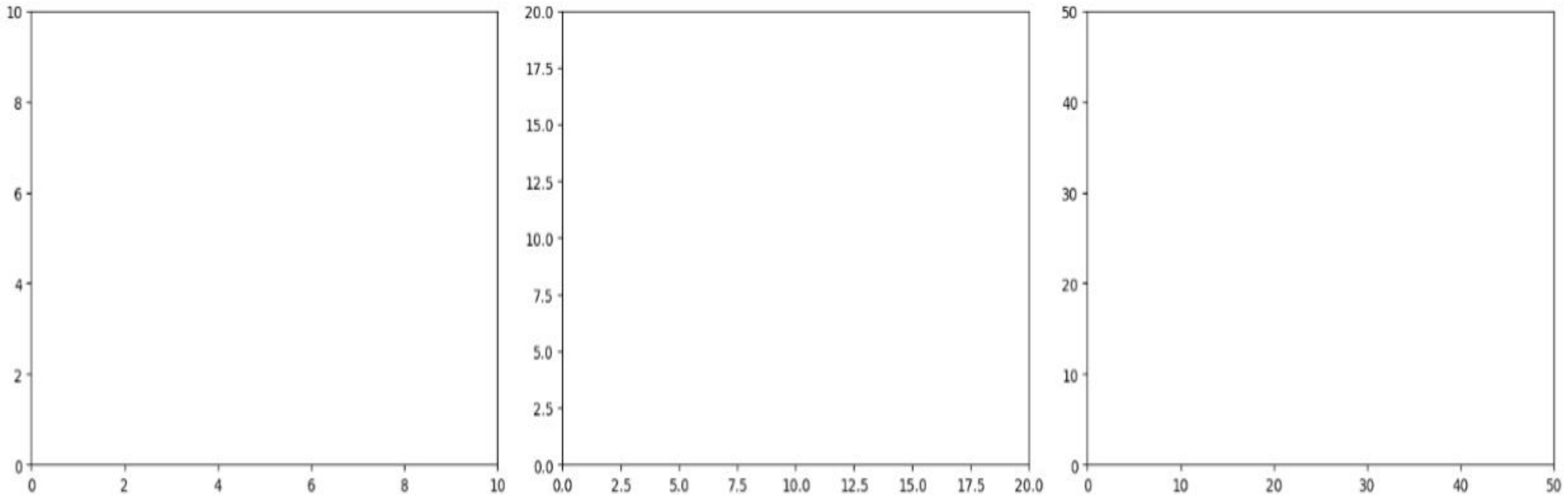
SIMULATIONS

Maps Parameters

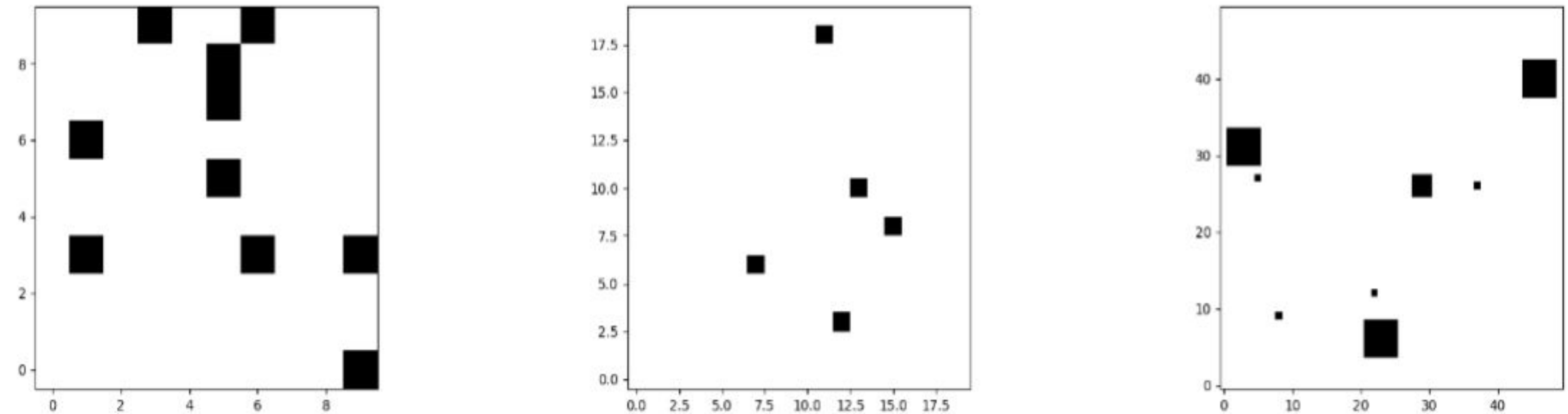
Map	Dimension	Resolution	Obstacle-probability
Small-empty	10 x 10 (m)	1 (m)	0.0
Small-few	10 x 10 (m)	1 (m)	0.12
Small-many	13 x 13 (m)	1 (m)	custom
Medium-empty	20 x 20 (m)	1 (m)	0.0
Medium-few	20 x 20 (m)	1 (m)	0.01
Medium-many	20 x 20 (m)	1 (m)	0.04
Big-empty	50 x 50 (m)	1 (m)	0.0
Big-few	50 x 50 (m)	1 (m)	0.005
Big-many	50 x 50 (m)	1 (m)	0.008

Maps

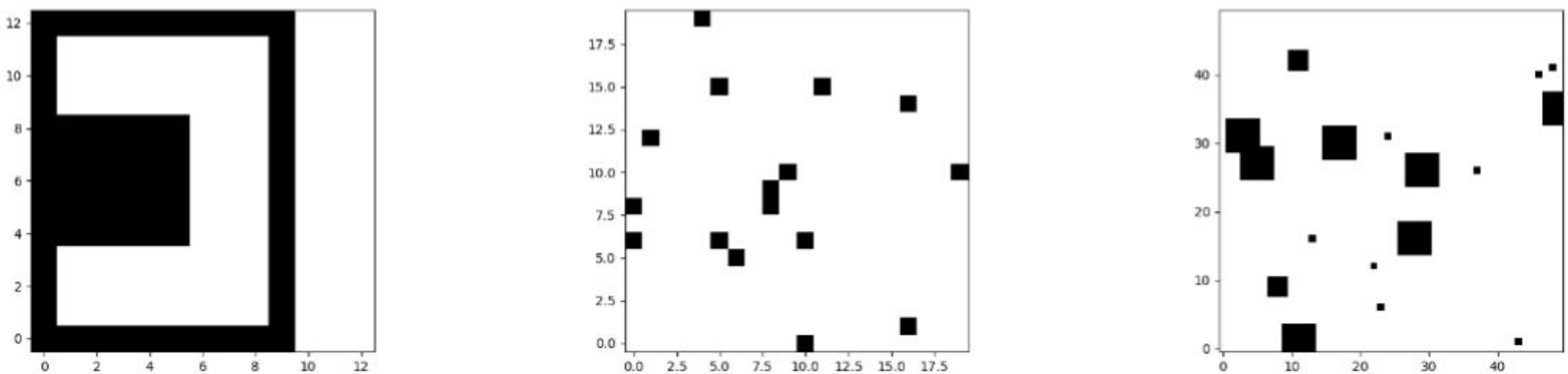
Empty maps



Few Obstacles



Many Obstacles



Hyperparameters

- We have tuned the matrix **R** in the following way:

$$\boldsymbol{R} = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix}$$

where $w_1 = w_2 = 1$ and $w_3 = 10$

- We have constrained v_{s_1} to be always positive:

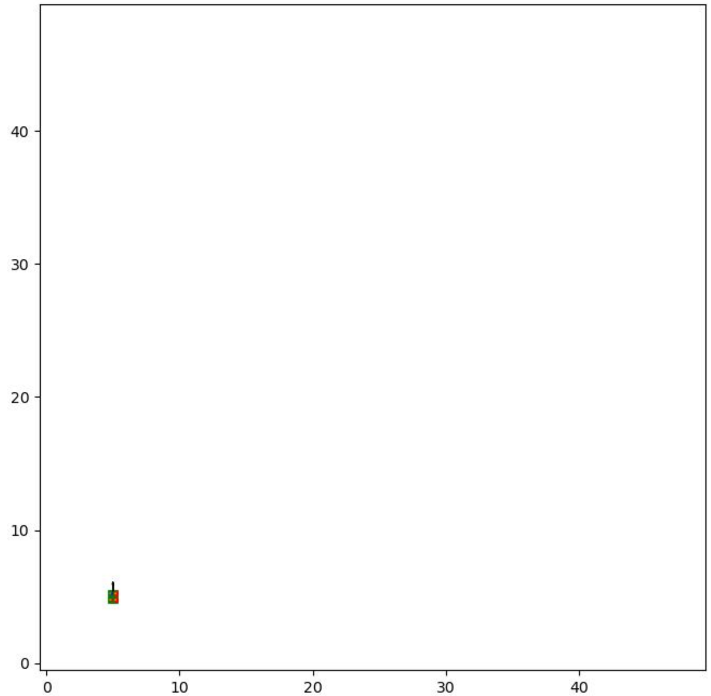
$$v_{s_1} > 0 \, \frac{m}{s}$$

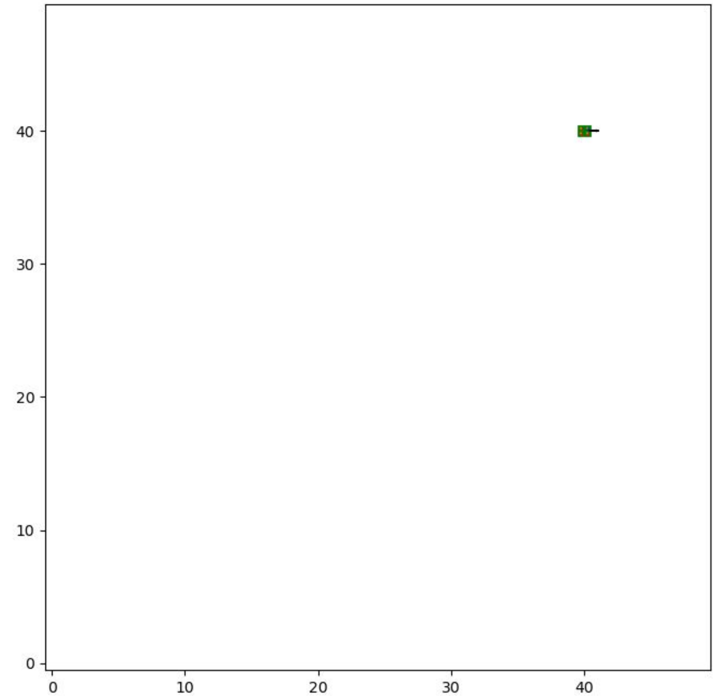
Simple - Hard trajectories

Empty maps

Big-empty Map - Simple Trajectory

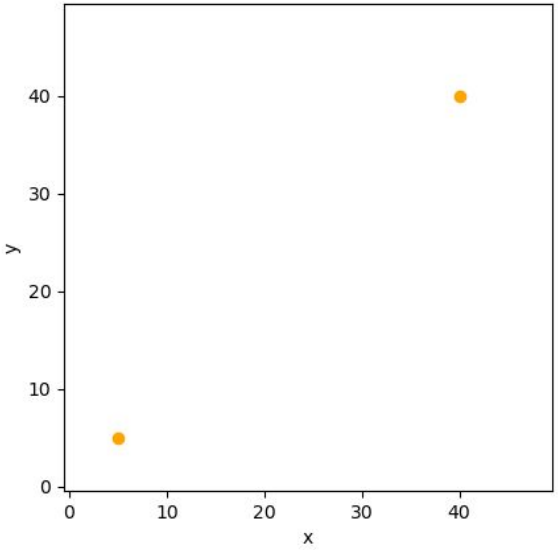
$$\text{Start state} = [5,\ 5,\ \tfrac{\pi}{2},\ 0,\ 1,\ 0] \quad \text{Goal state} = [40,\ 40,\ 0,\ 0,\ 1,\ 0]$$

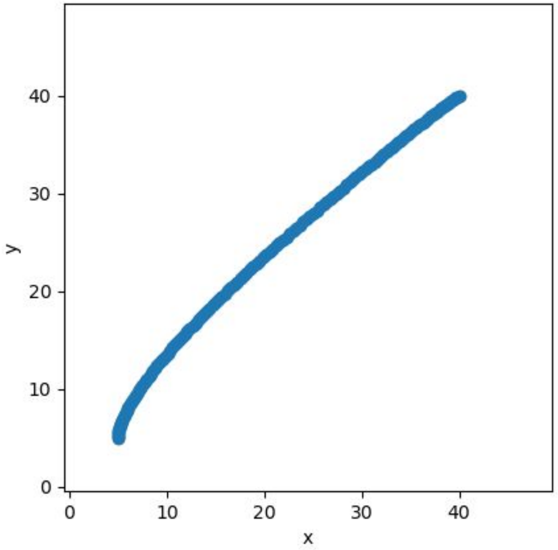




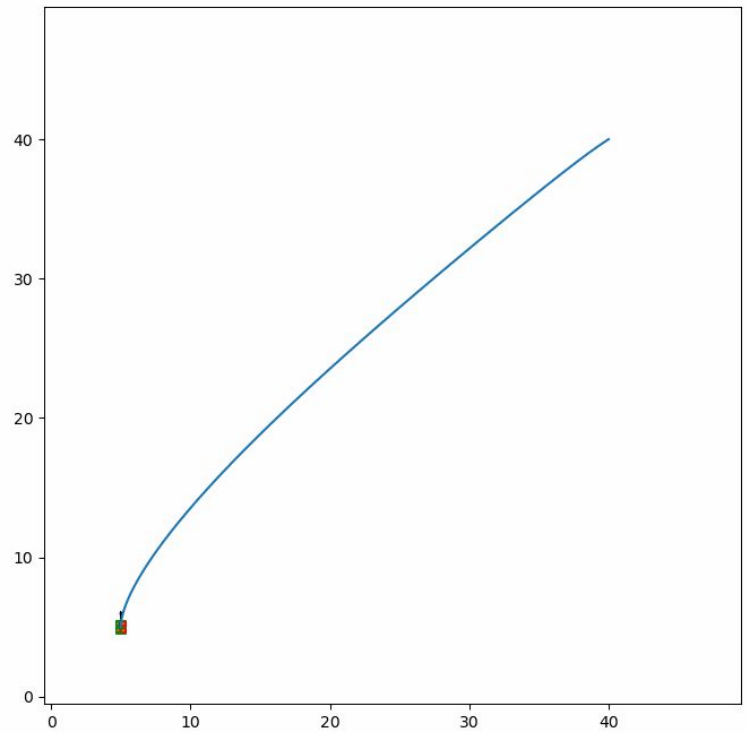
Big-empty Map - Simple Trajectory

$N = 1$



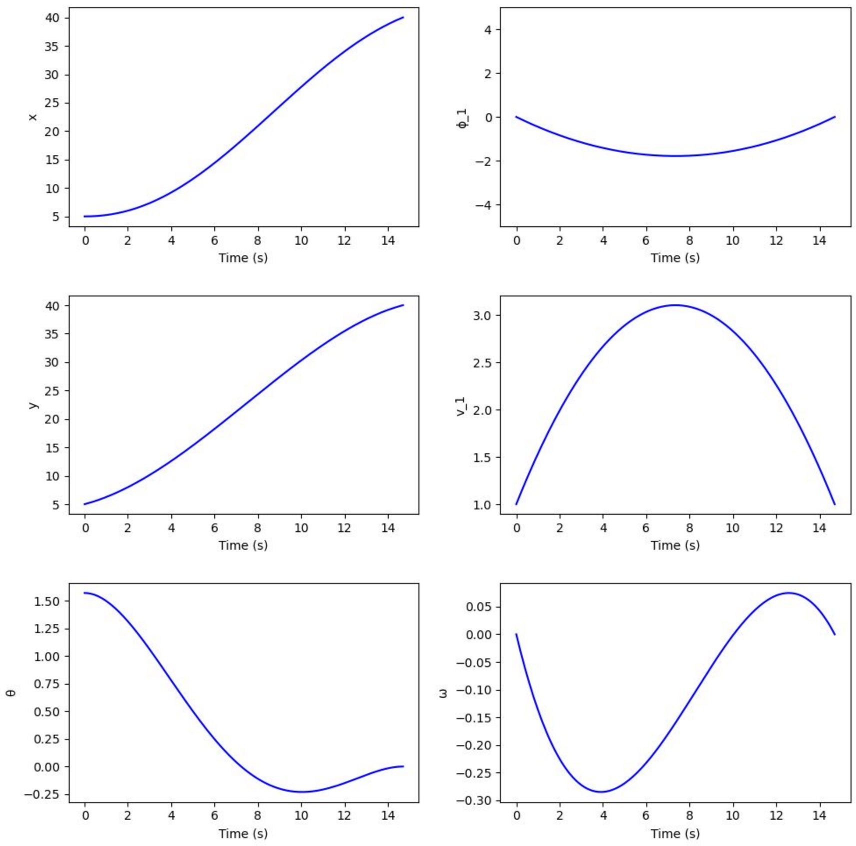


Big-empty Map - Simple Trajectory

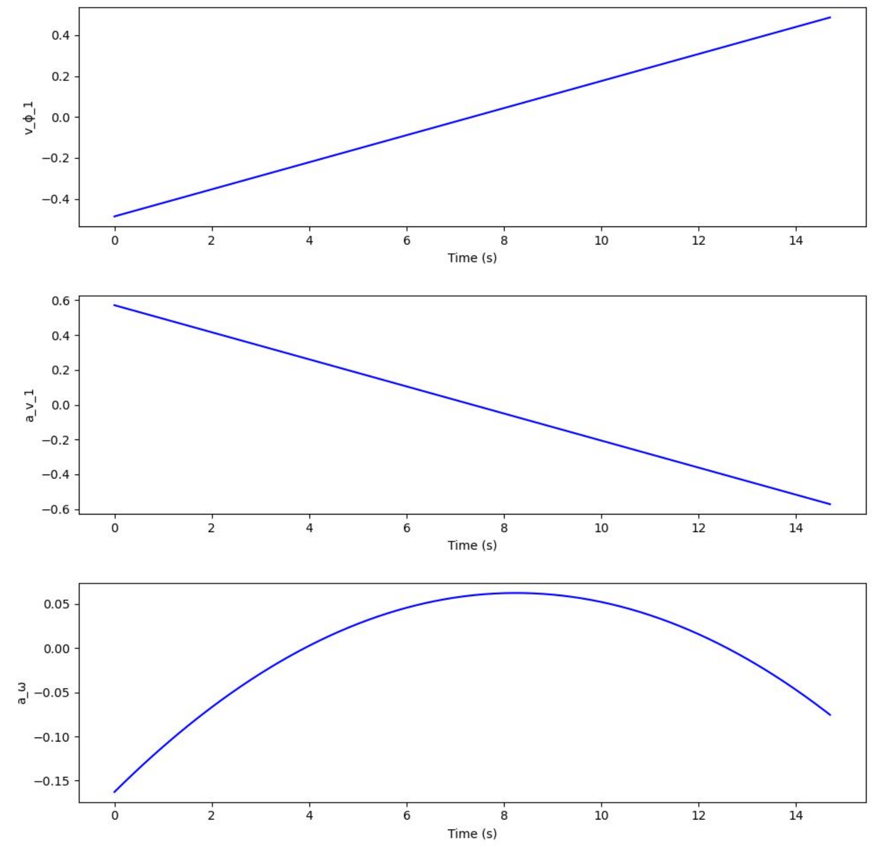


Big-empty Map - Simple Trajectory

States

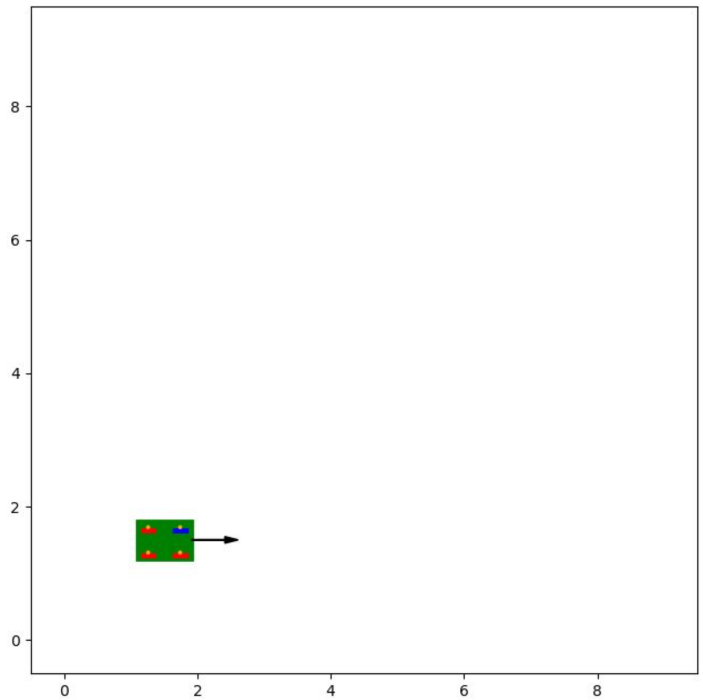


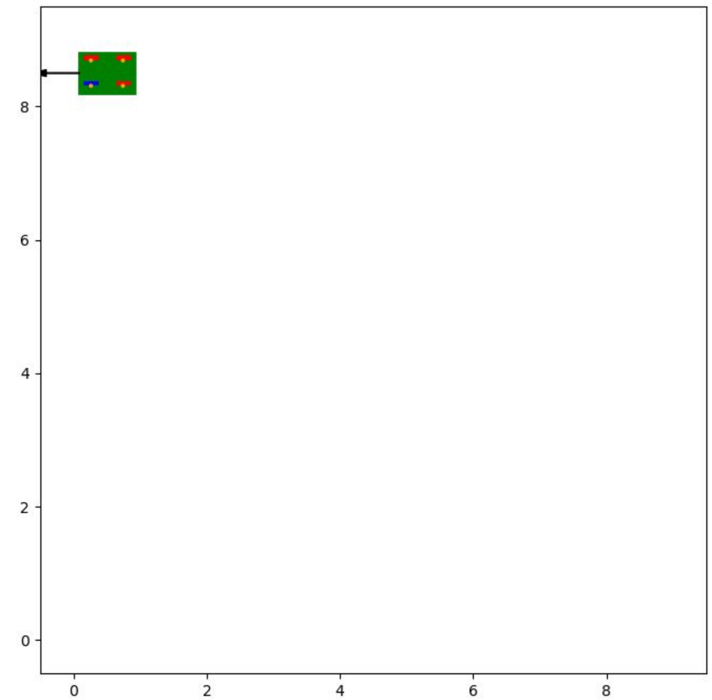
Inputs



Small-empty Map - Hard Trajectory

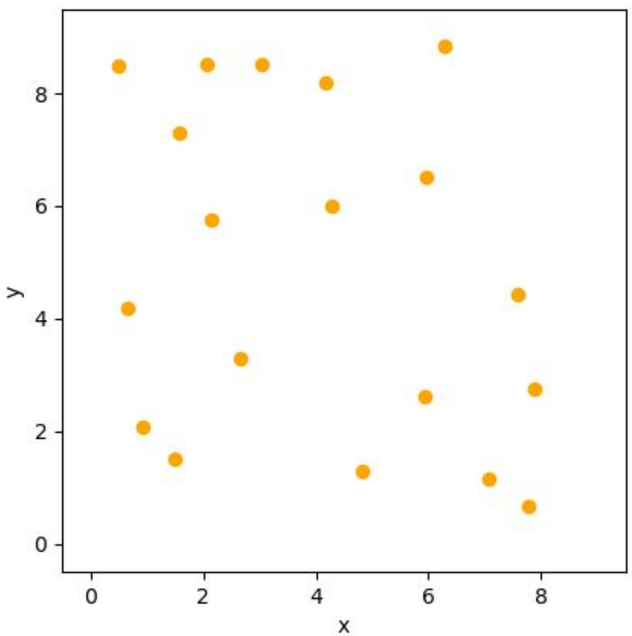
Start state $= [1.5, 1.5, 0, 0, 0.01, 0]$ Goal state $= [0.5, 8.5, \pi, 0, 0.01, 0]$

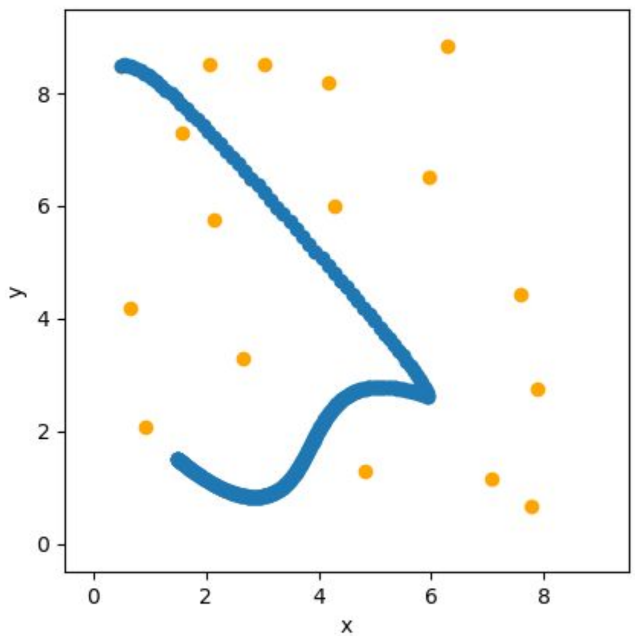




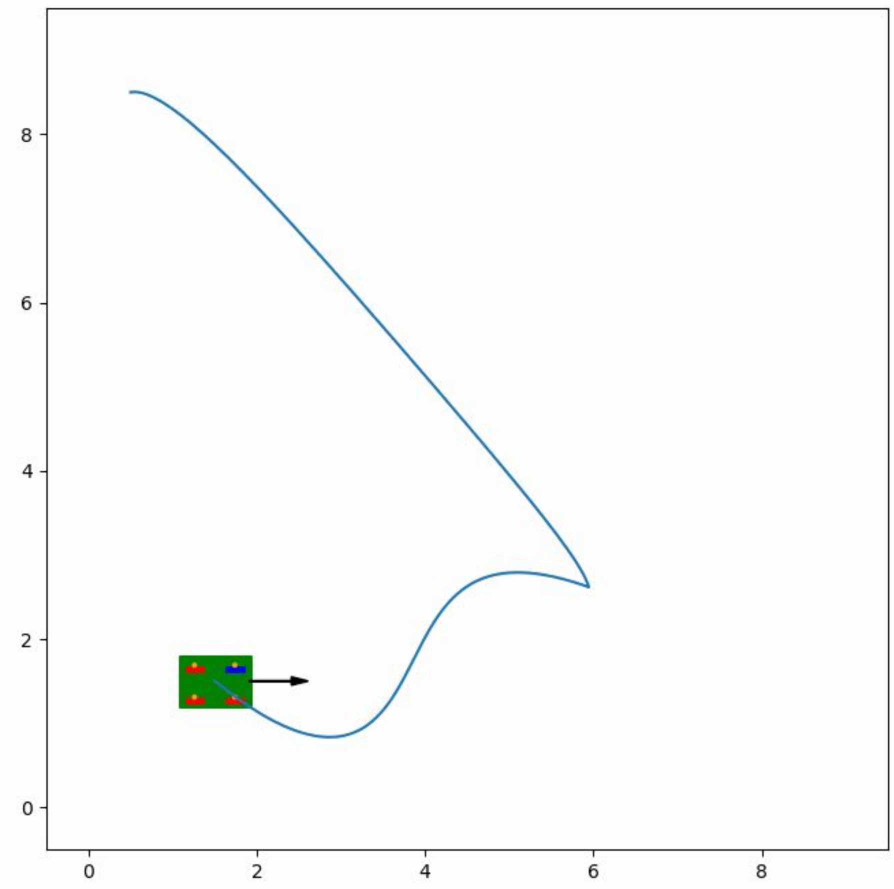
Small-empty Map - Hard Trajectory

N = 100



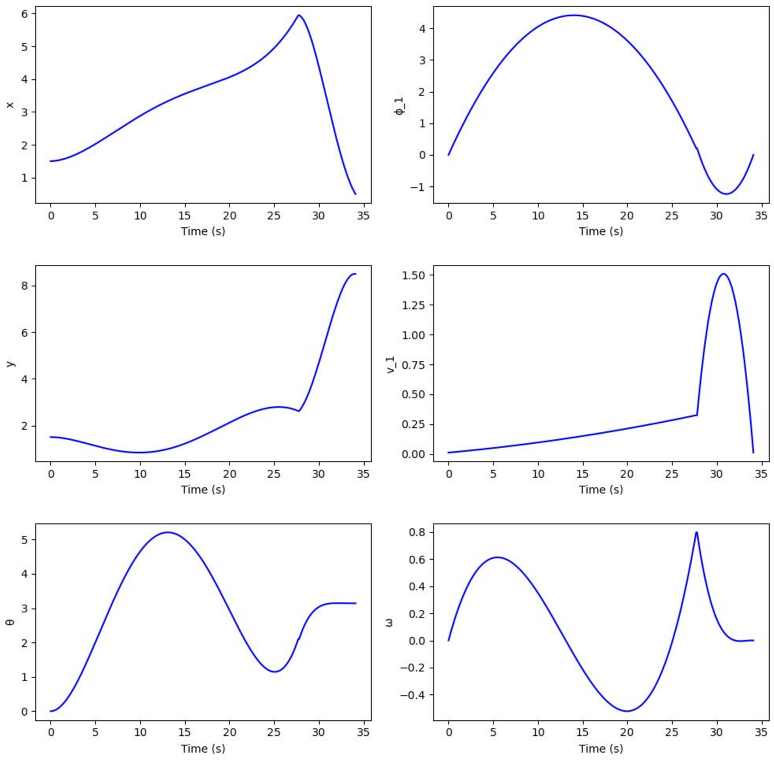


Small-empty Map - Hard Trajectory

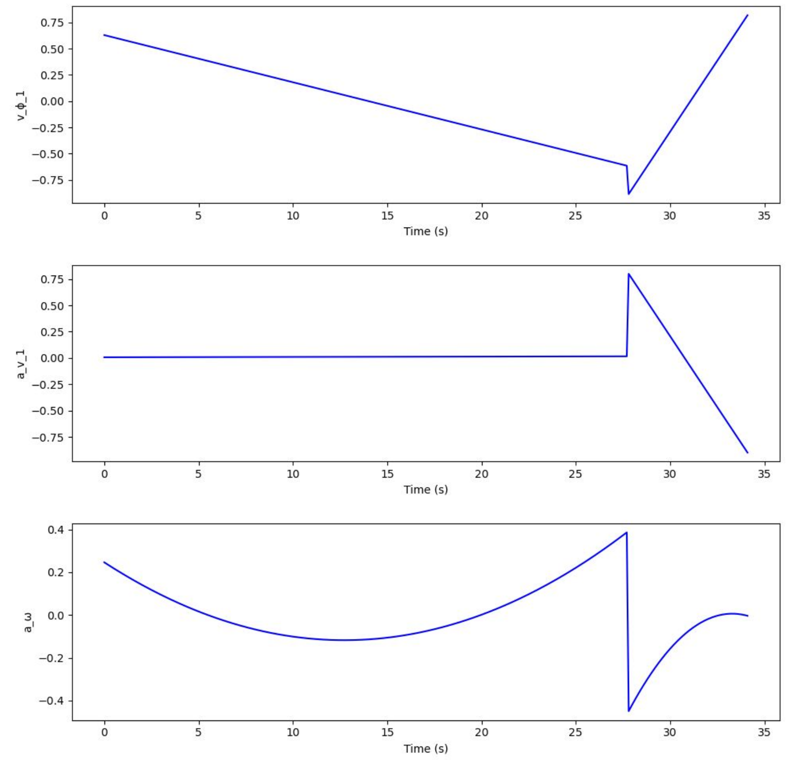


Small-empty Map - Hard Trajectory

States



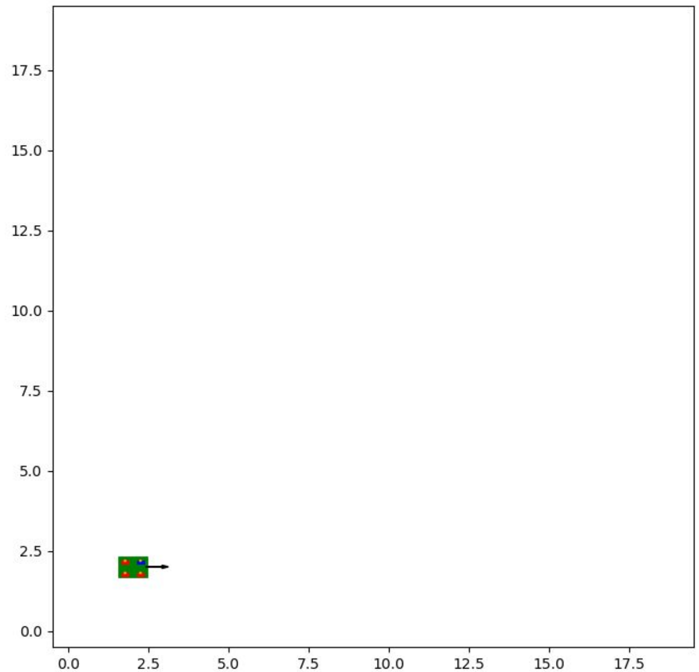
Inputs

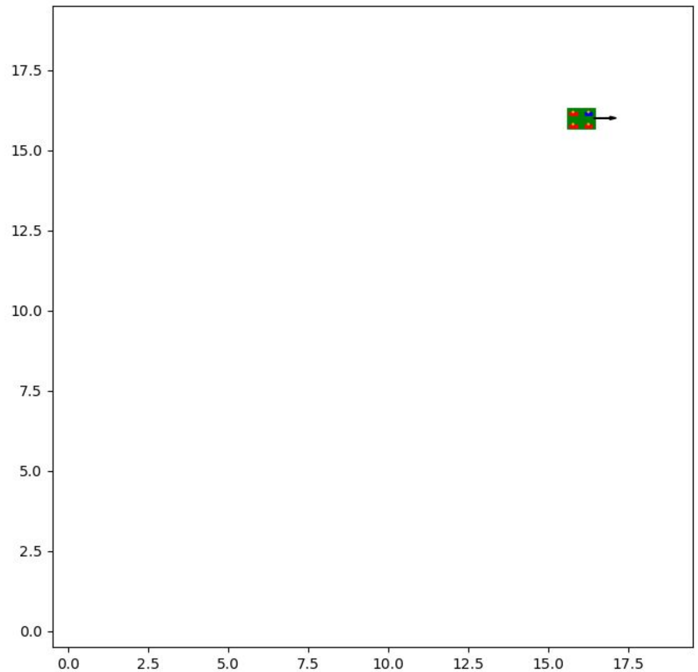


Different R Matrices

Medium-empty Map

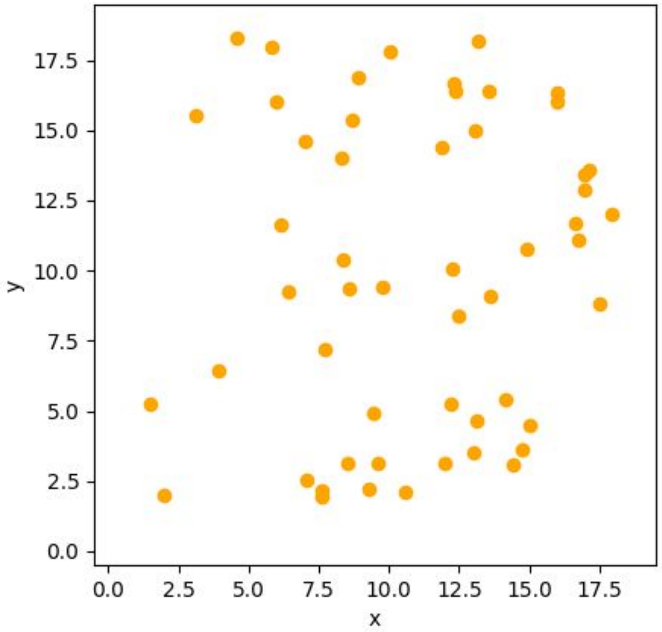
Start state $= [2, 2, 0, 0, 0.1, 0]$ Goal state $= [16, 16, 0, 0, 0.1, 0]$

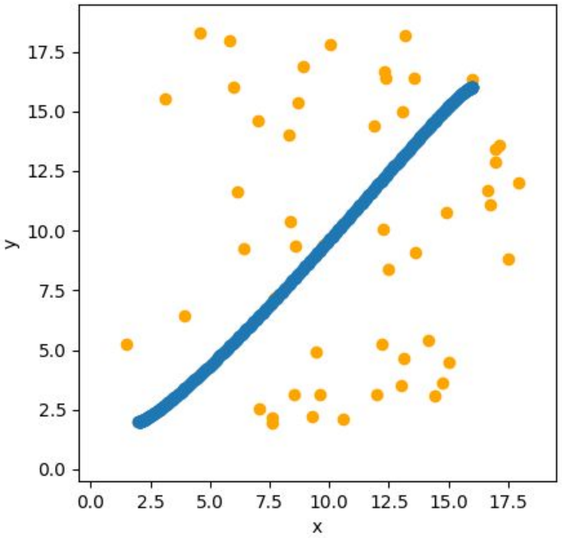




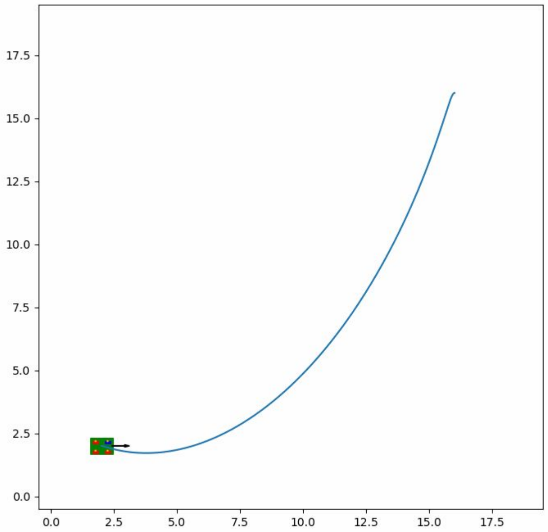
Medium-empty Map

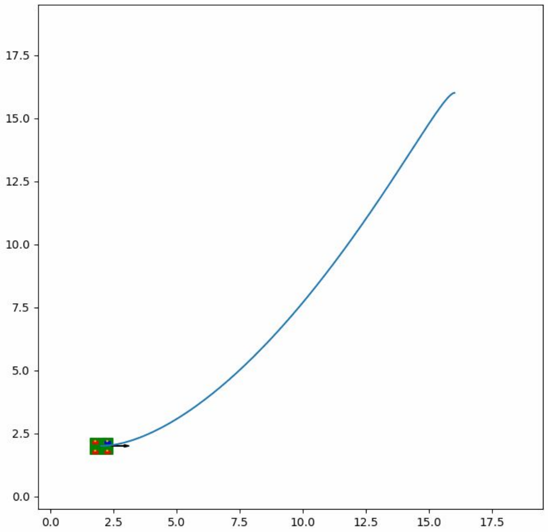
N = 100

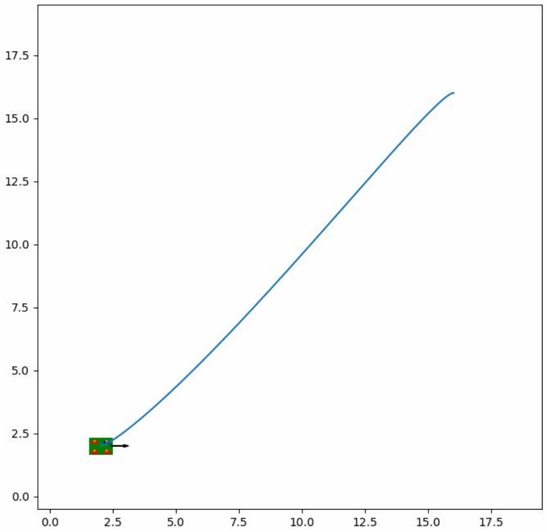




Medium-empty Map







$$w_1, w_2, w_3 = 1$$

$$w_1, w_2 = 1 \; and \; w_3 = 10$$

$$w_1, w_2 = 1 \; and \; w_3 = 100$$

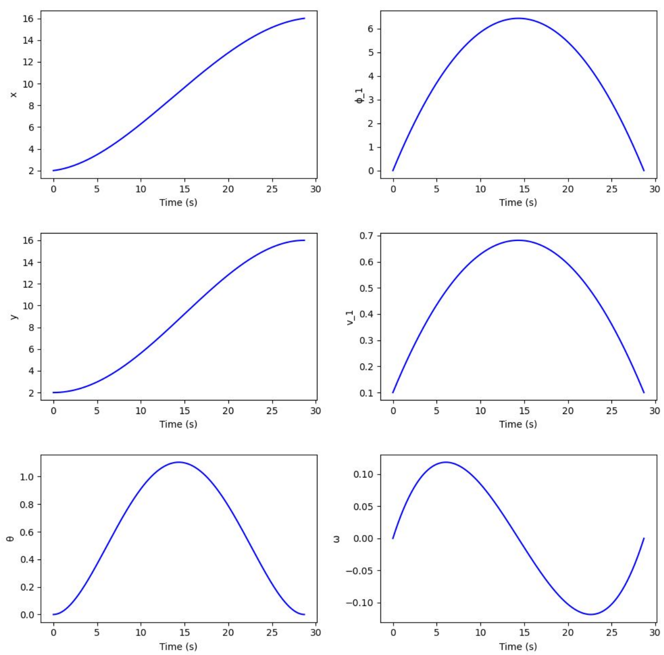
cost = 24

cost = 32

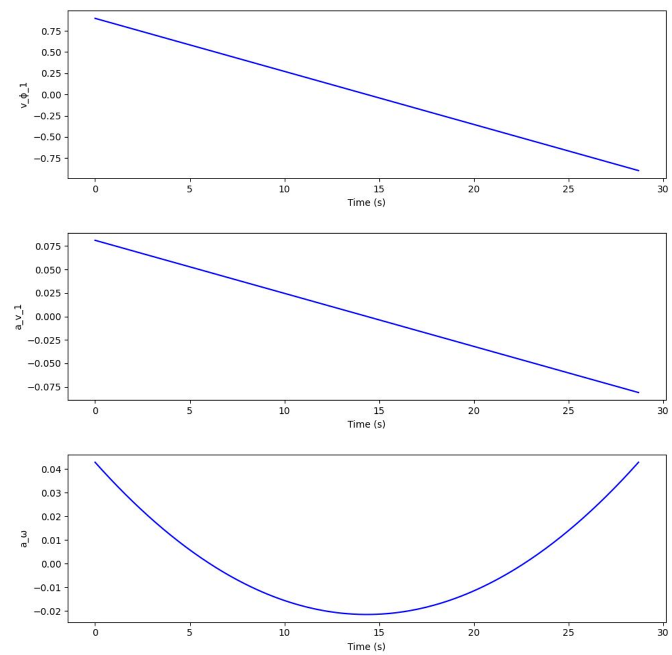
cost = 37

Medium-empty Map

States



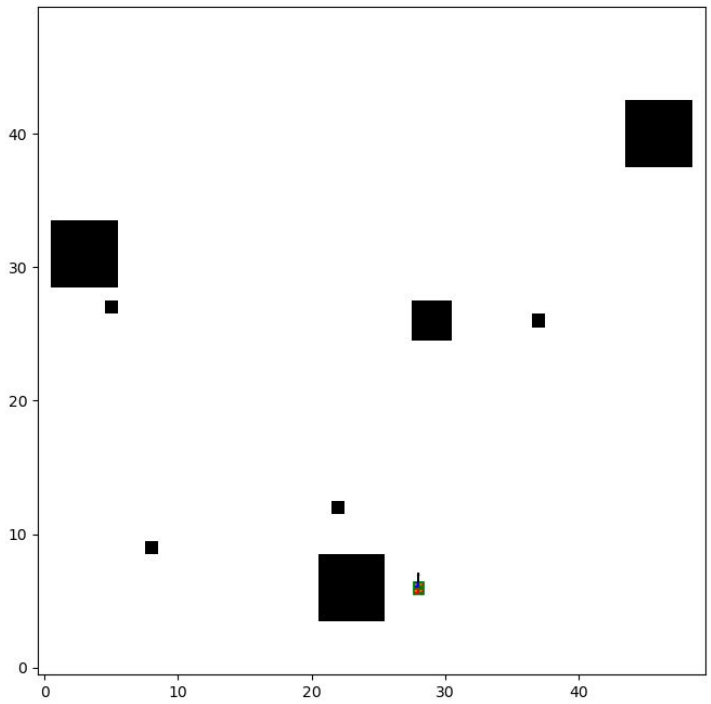
Inputs

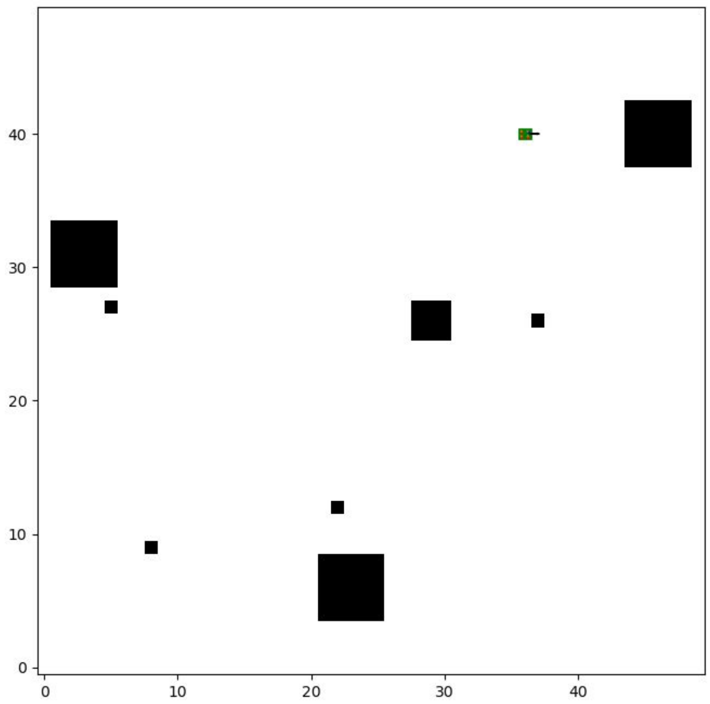


Forcing Nodes

Big-few Map

$\text{Start state} = [28,\ 6,\ \frac{\pi}{2},\ 0,\ 1,\ 0]$ $\text{Goal state} = [36,\ 40,\ 0,\ 0,\ 1,\ 0]$

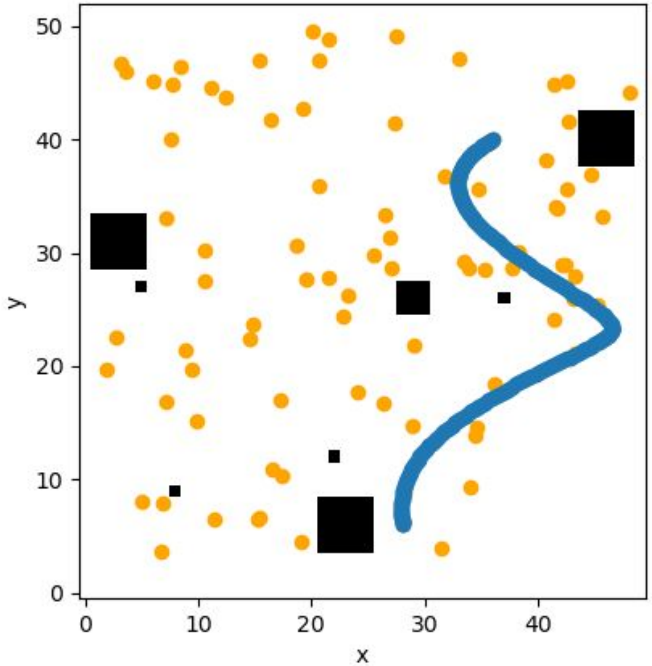




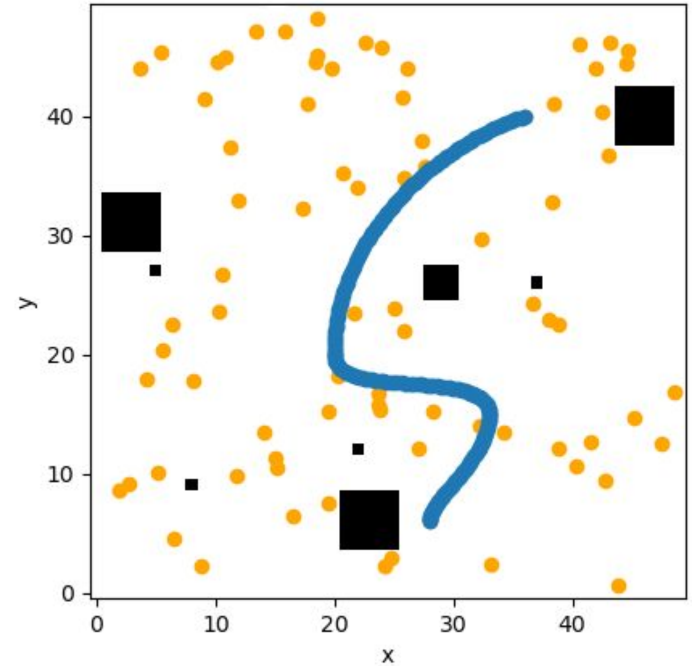
Big-few Map

N = 100

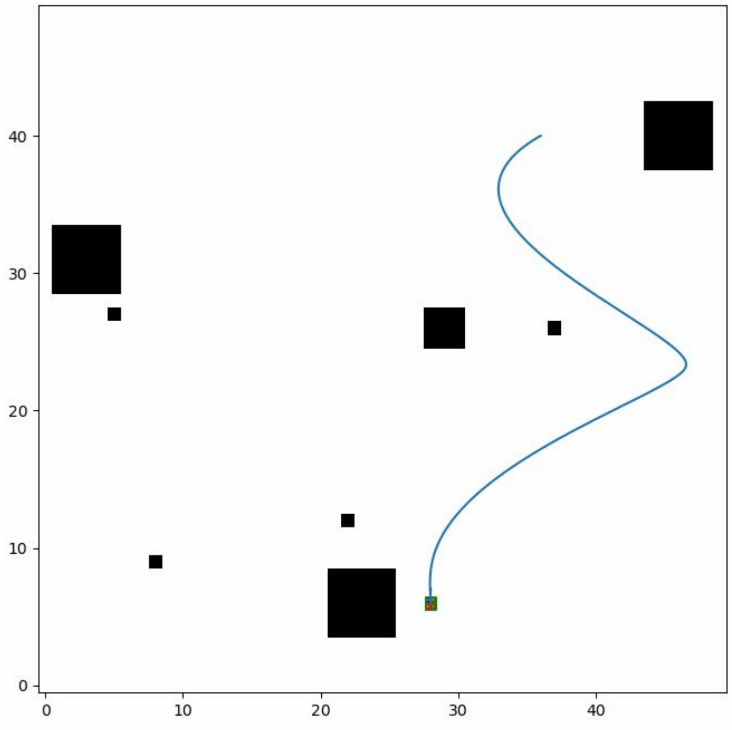
Without
Forced Nodes

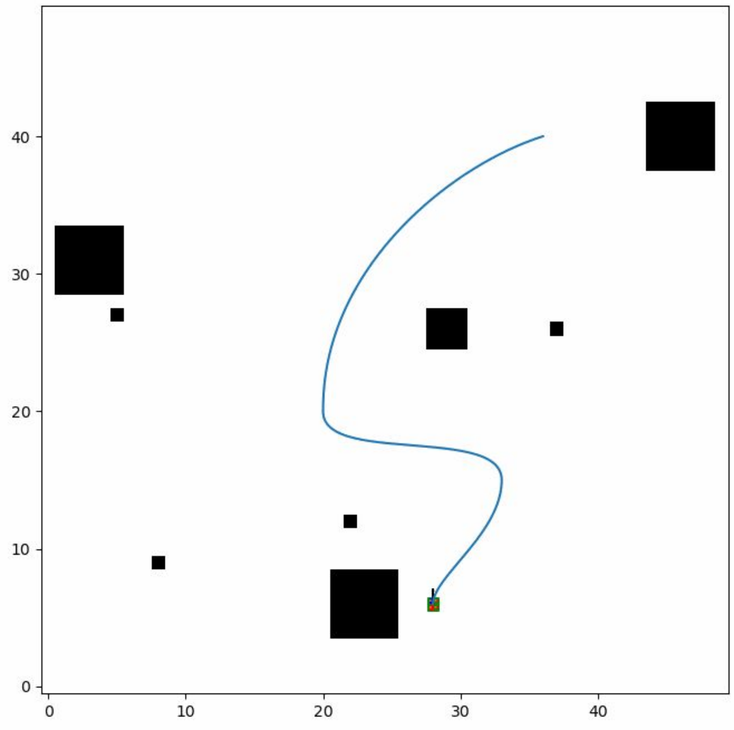


With
Forced Nodes



Big-few Map





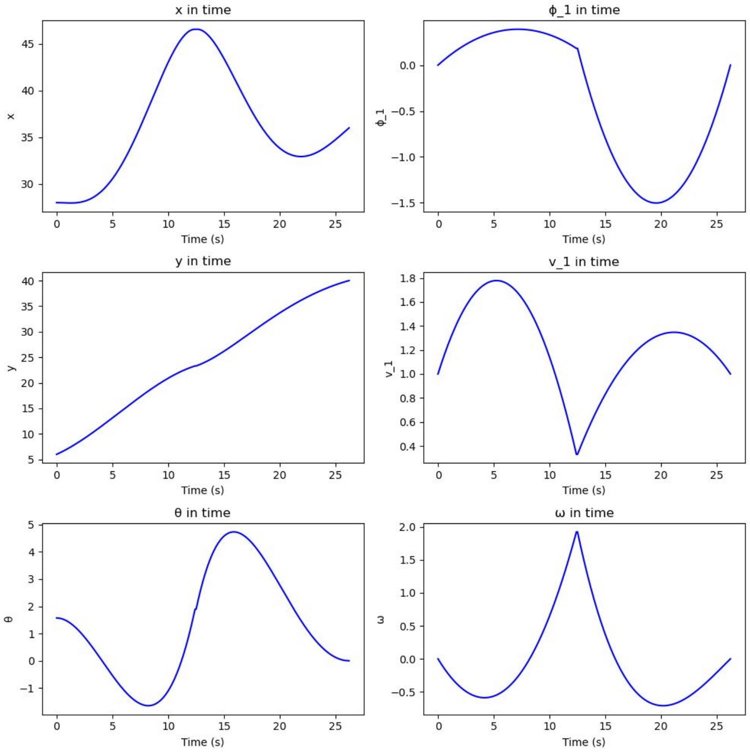
cost = 77

cost = 61

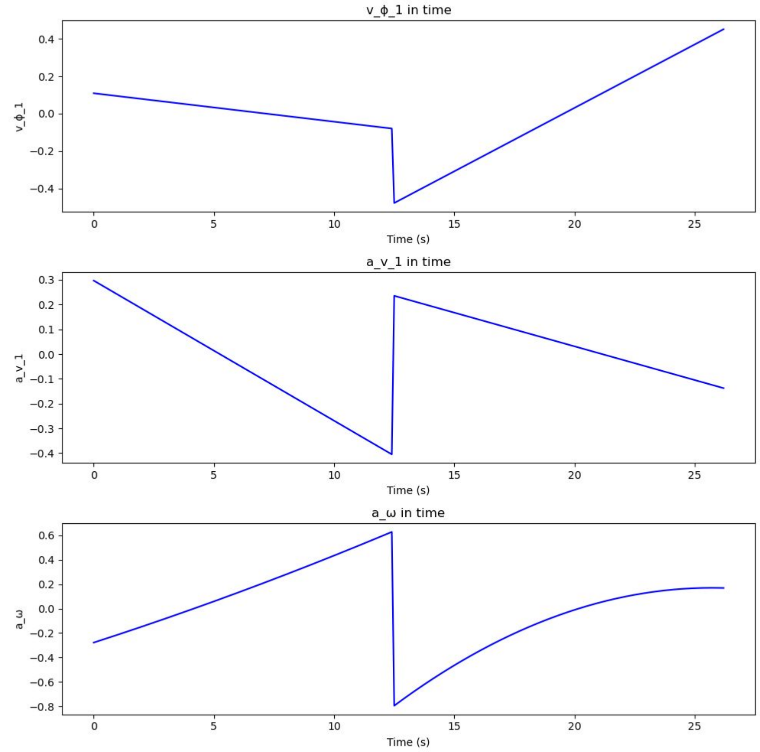
Big-few Map

Without Forced Nodes

States



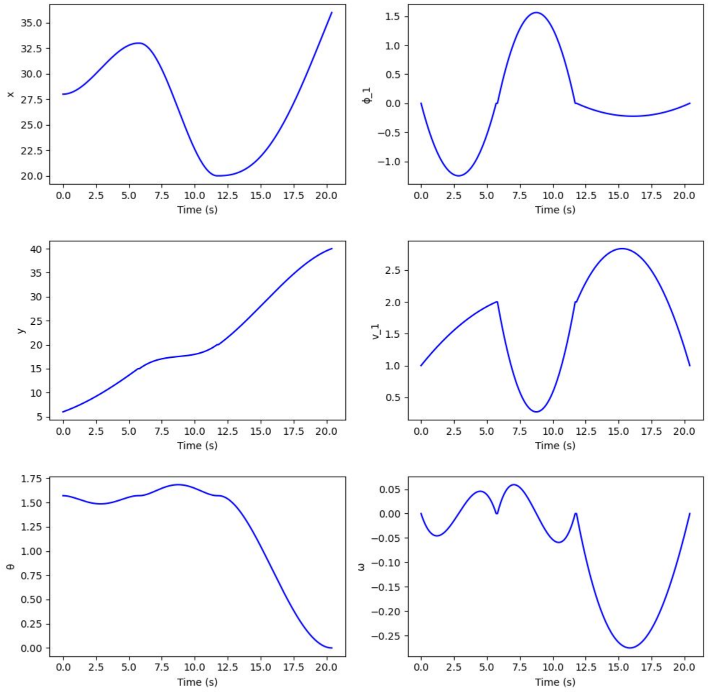
Inputs



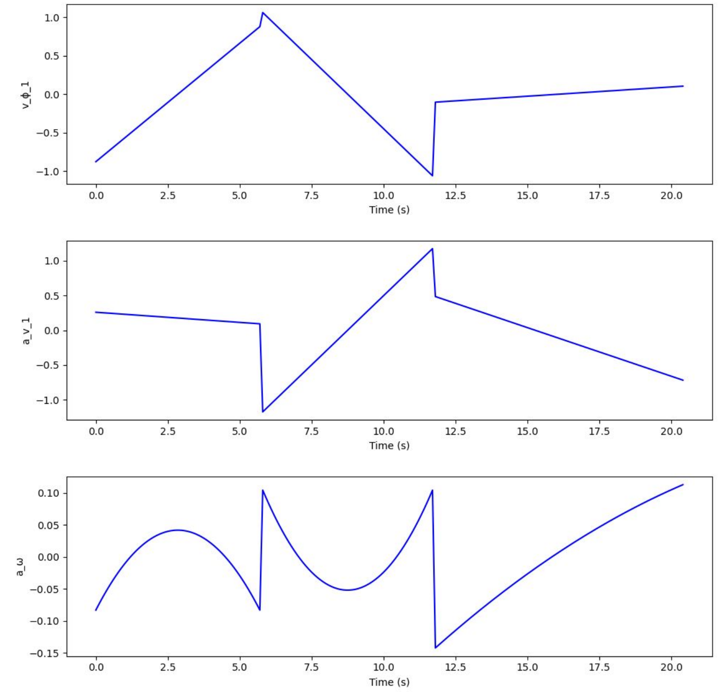
Big-few Map

With Forced Nodes

States

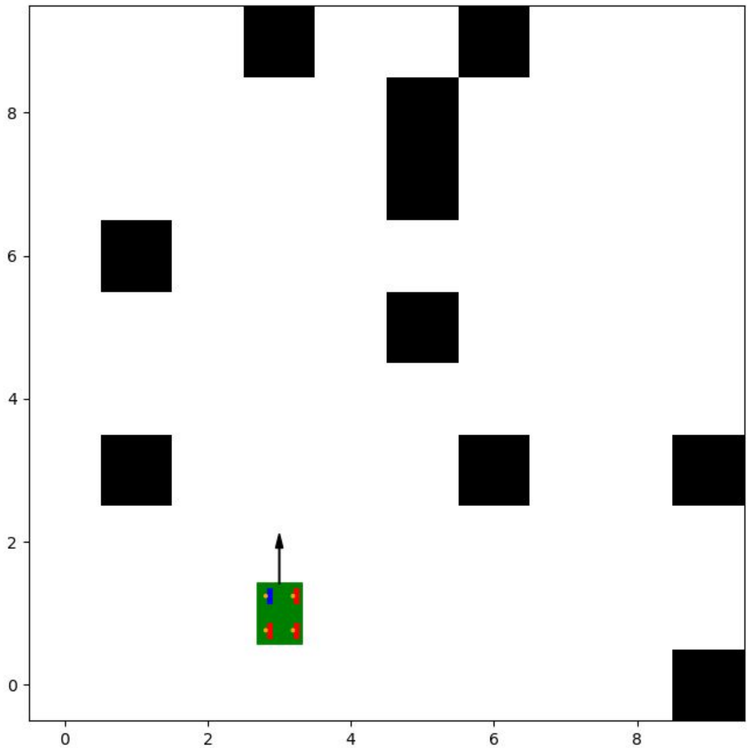


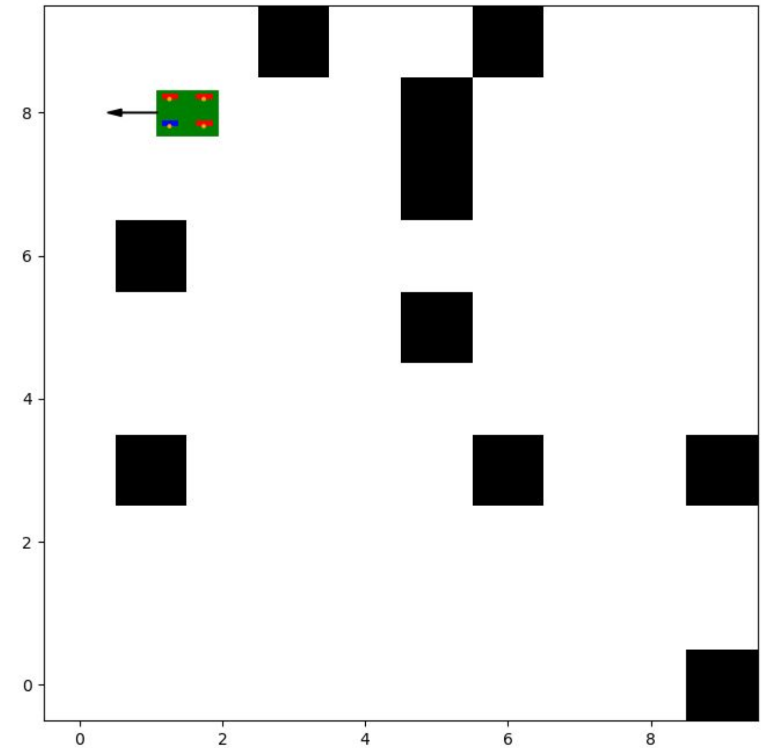
Inputs



Small-few Map

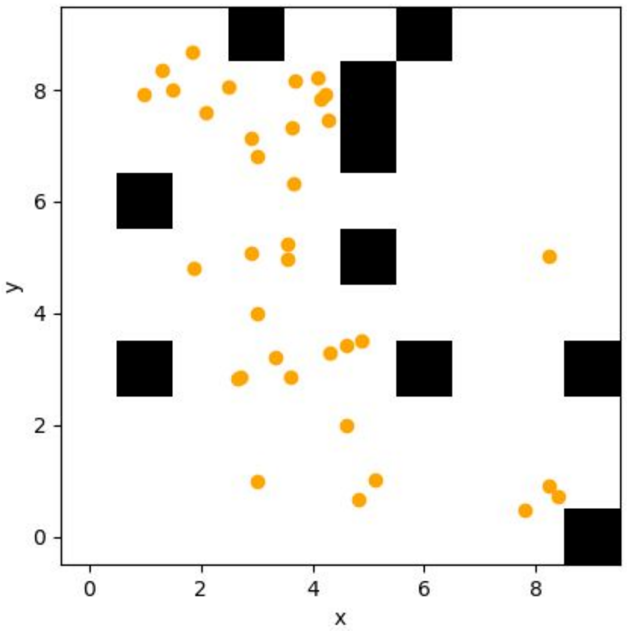
Start state $= [3,\ 1,\ \frac{\pi}{2},\ 0,\ 0.01,\ 0]$ Goal state $= [1.5,\ 8,\ \pi,\ 0,\ 1,\ 0]$

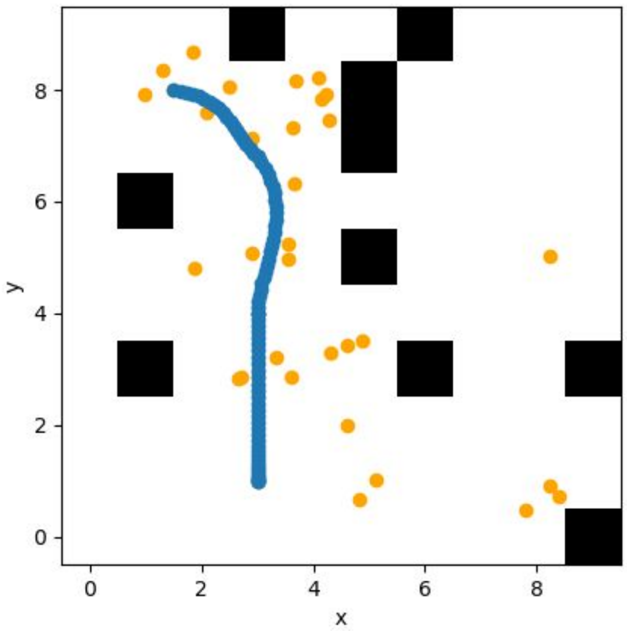




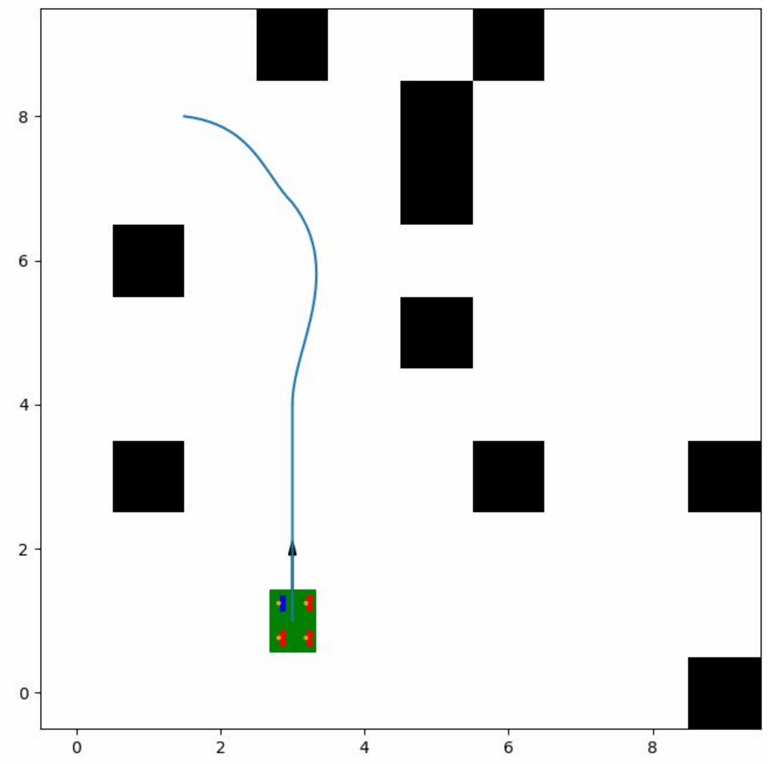
Small-few Map

N = 200



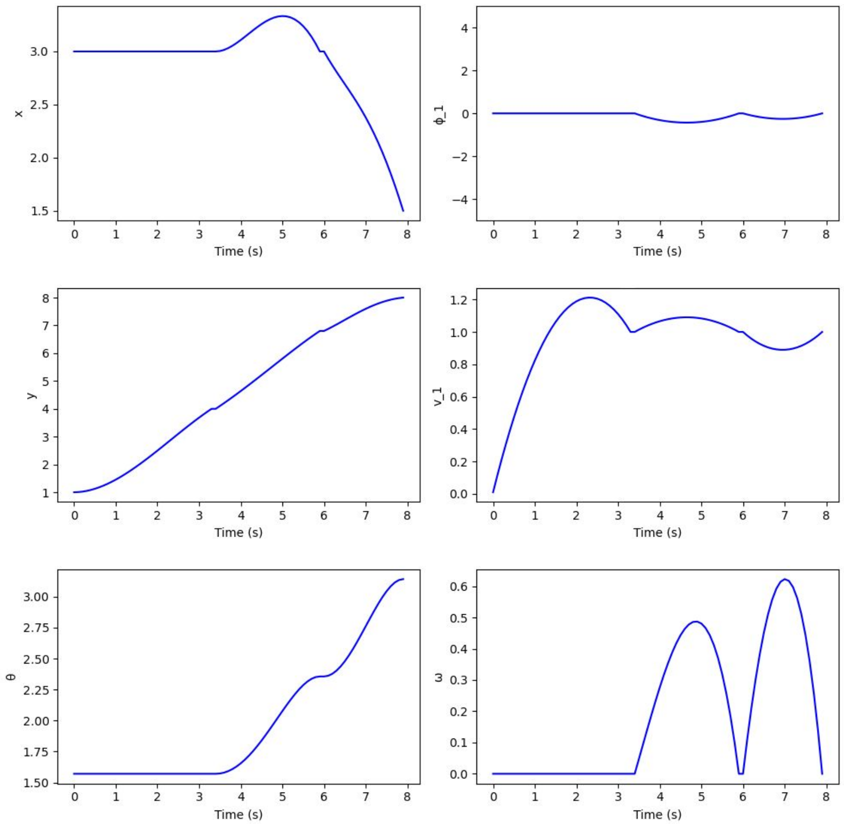


Small-few Map

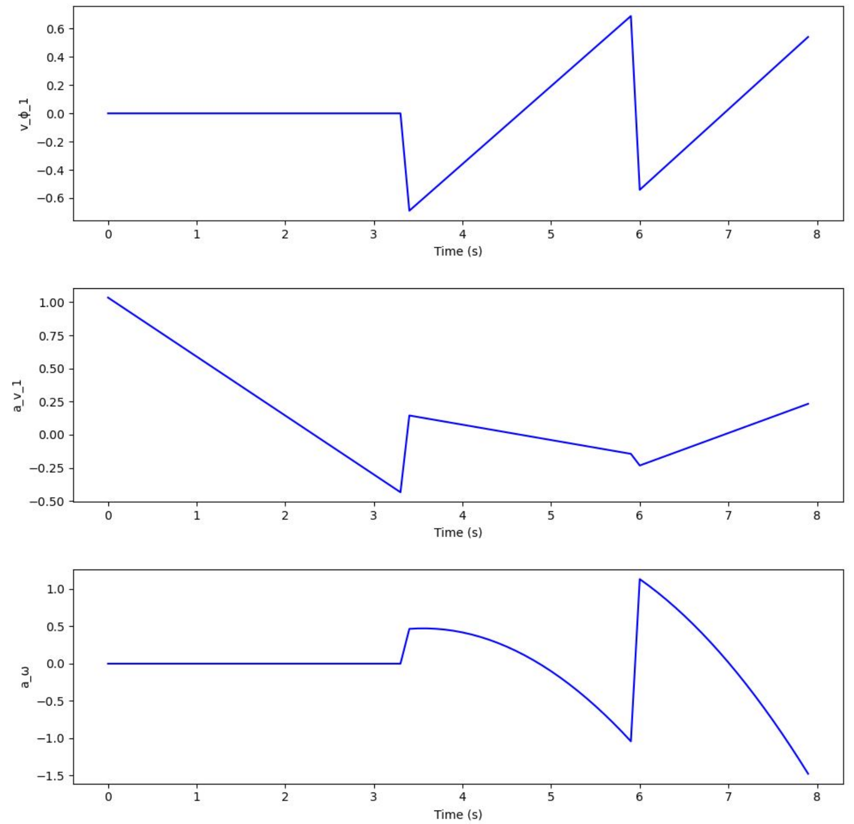


Small-few Map

States



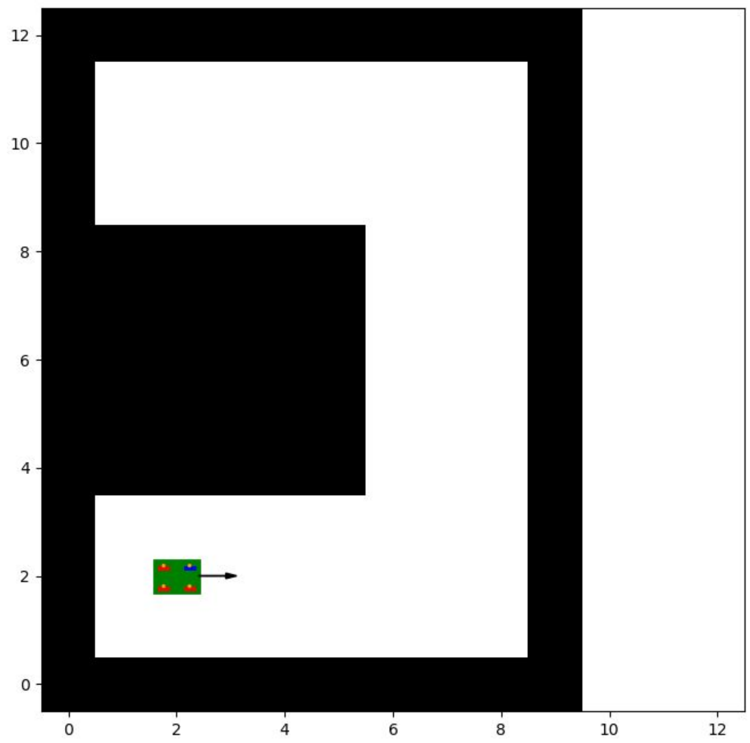
Inputs



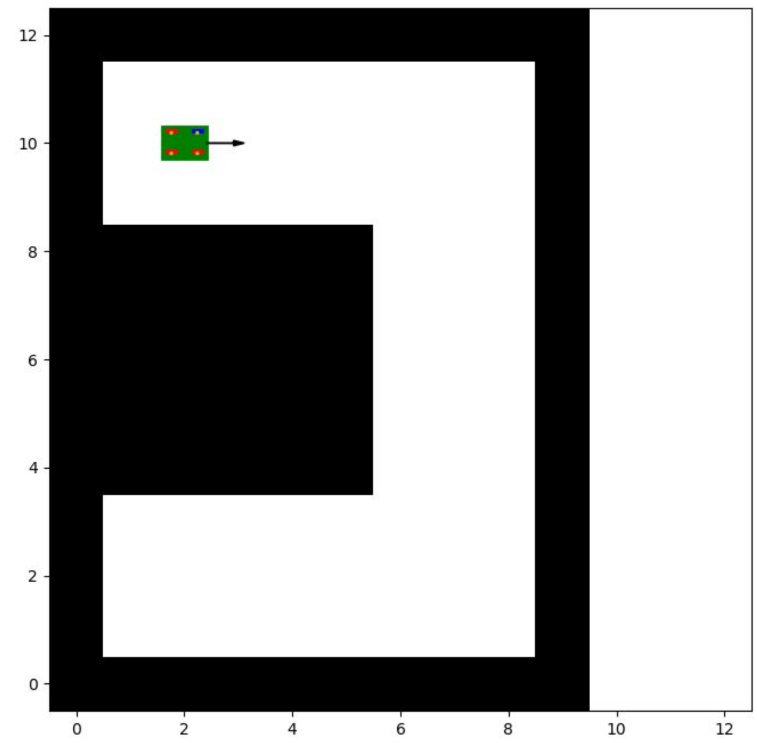
Custom Map

Small-many Map

Start state $= [2, 2, 0, 0, 0.01, 0]$

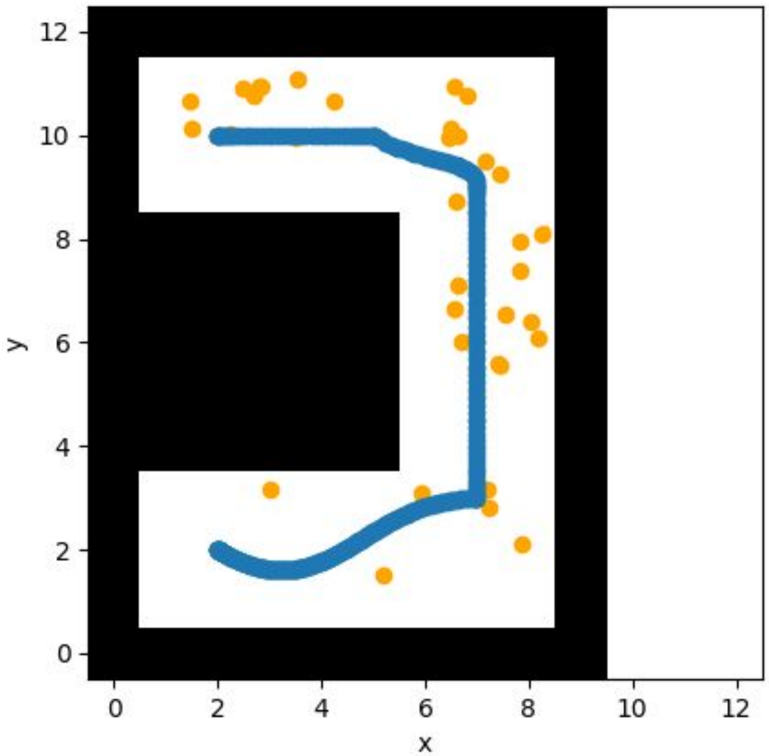


Goal state $= [2, 10, 0, 0, 0.01, 0]$

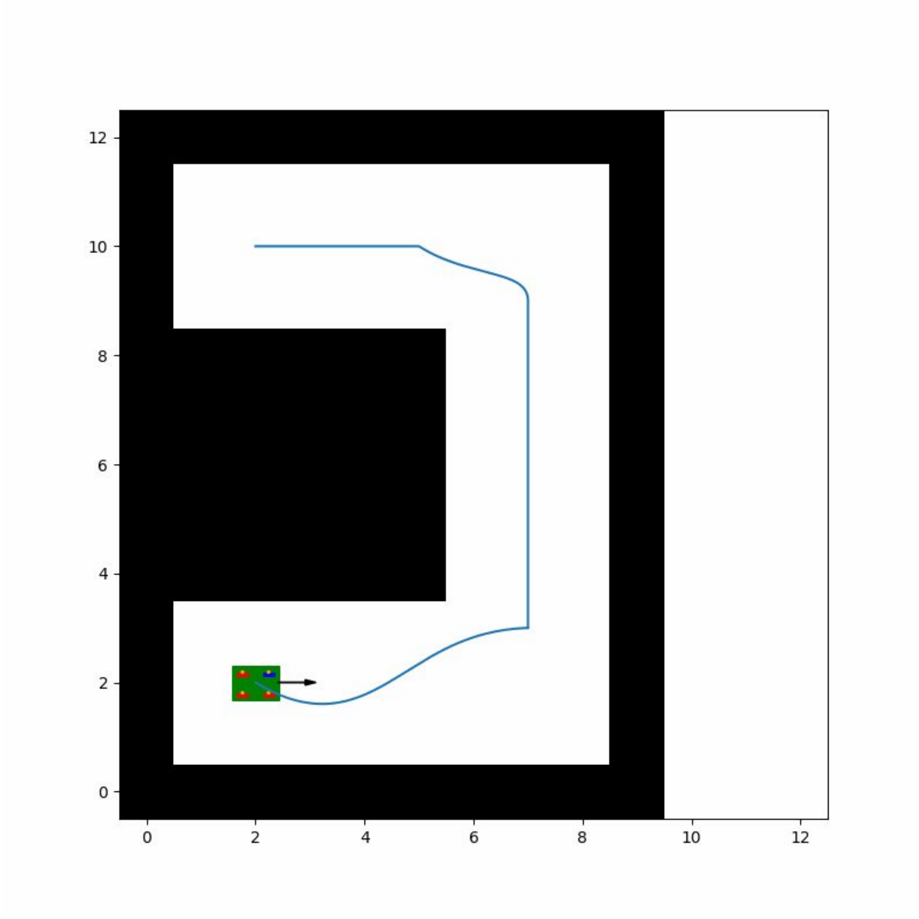


Small-many Map

N = 500

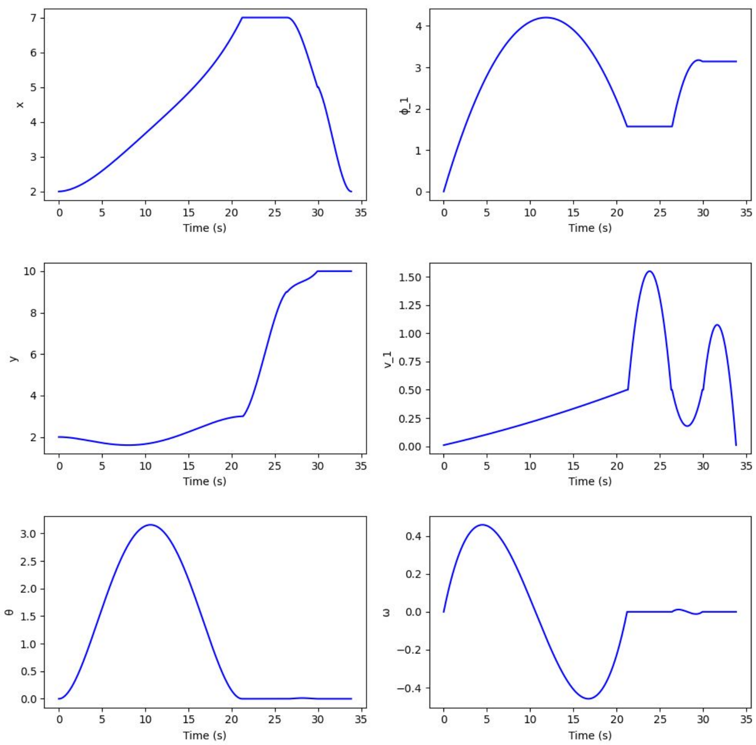


Small-many Map

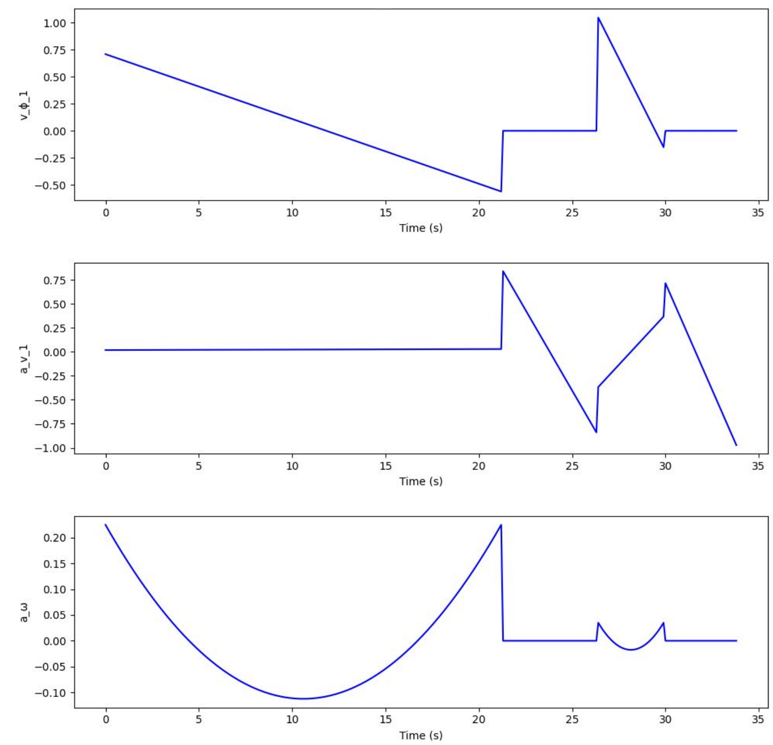


Small-many Map

States



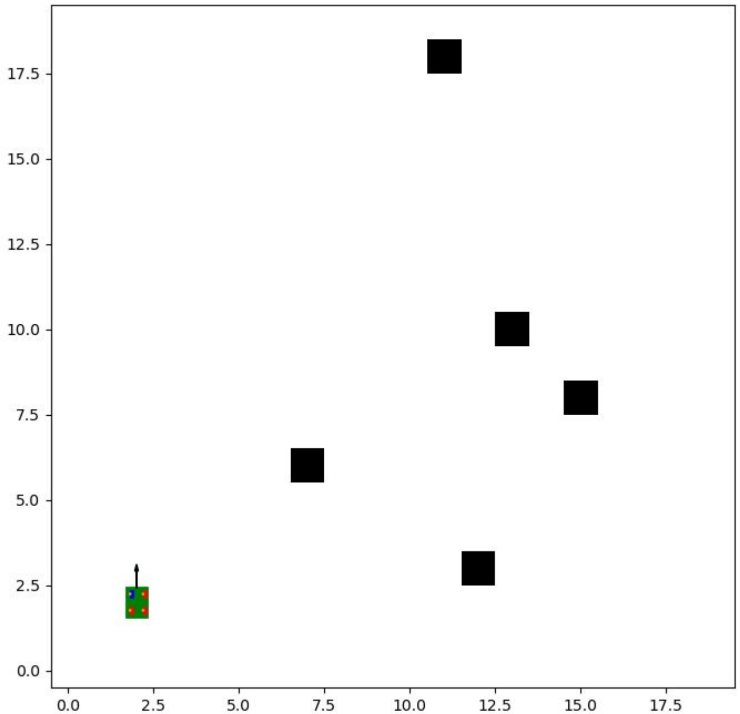
Inputs



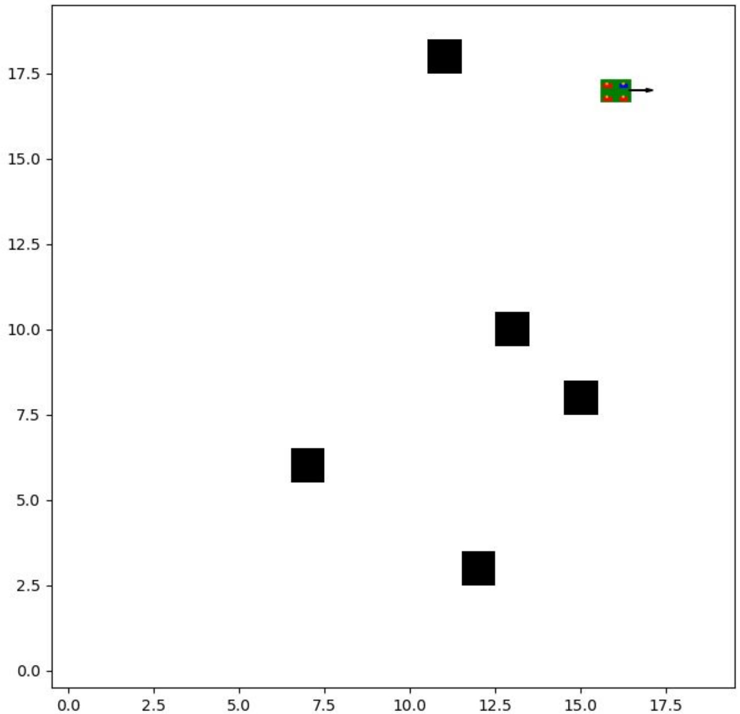
Other Simulations

Medium-few Map

Start state $= [2,\ 2,\ \frac{\pi}{2},\ 0,\ 1,\ 0]$

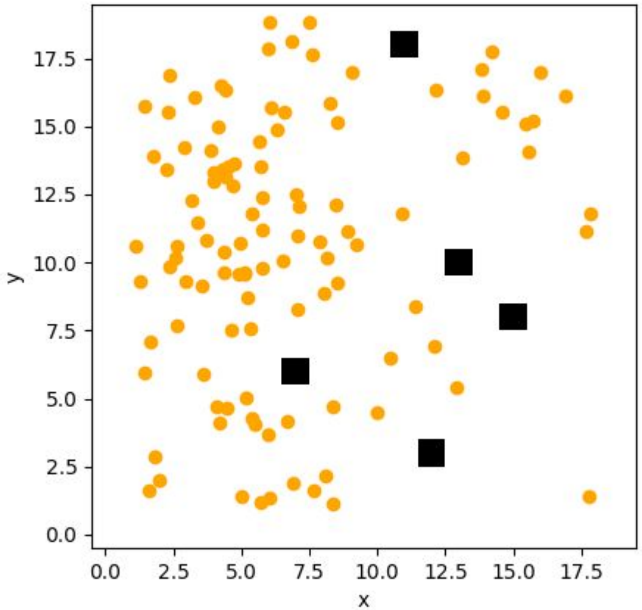


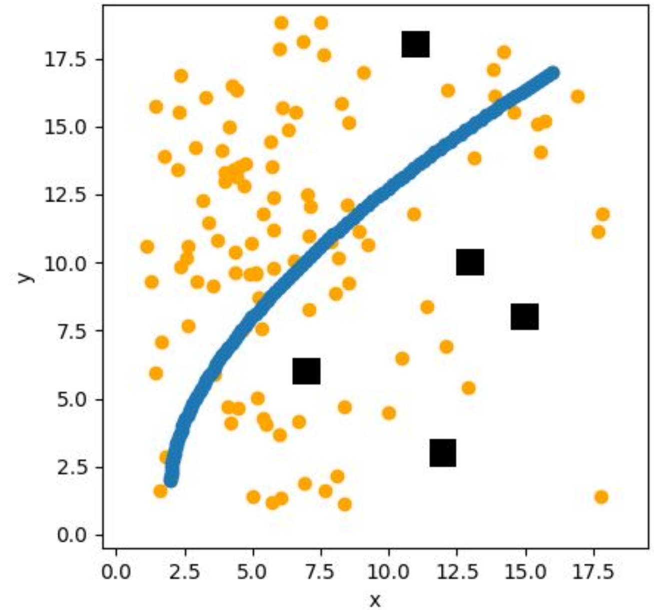
Goal state $= [16,\ 17,\ 0,\ 0,\ 1,\ 0]$



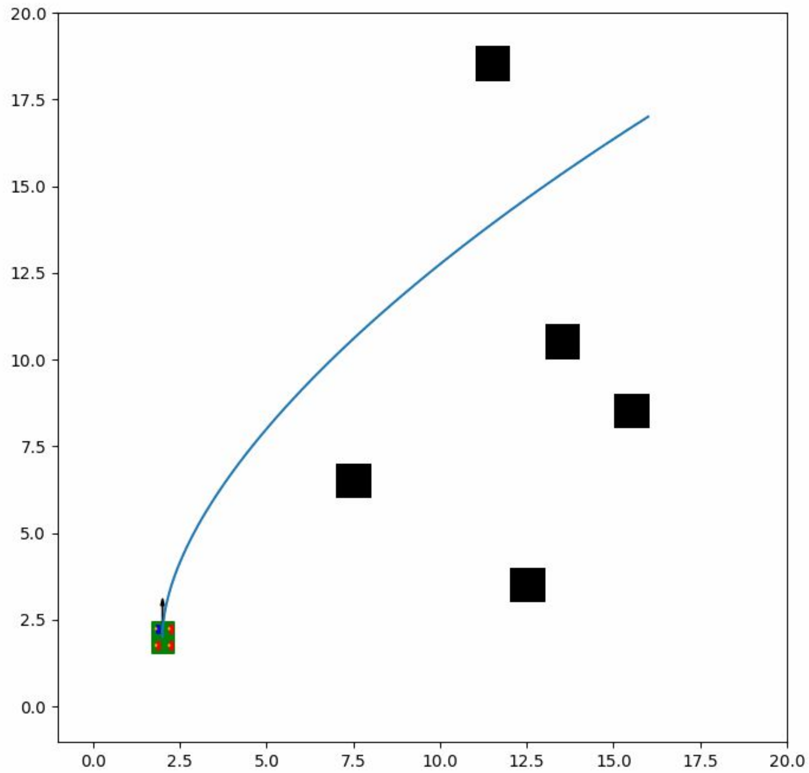
Medium-few Map

N = 500



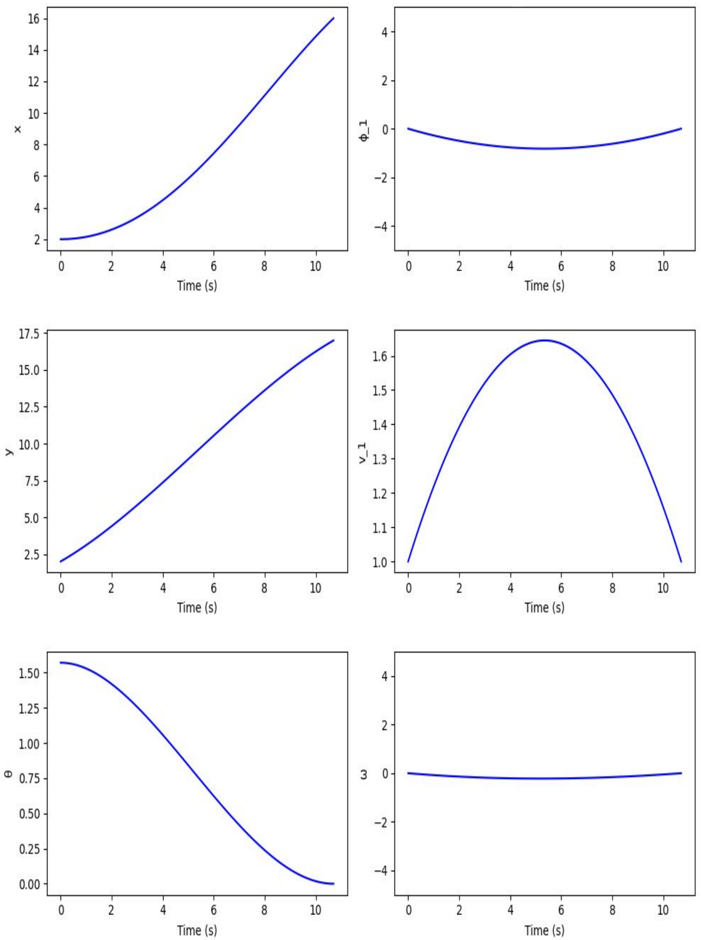


Medium-few Map

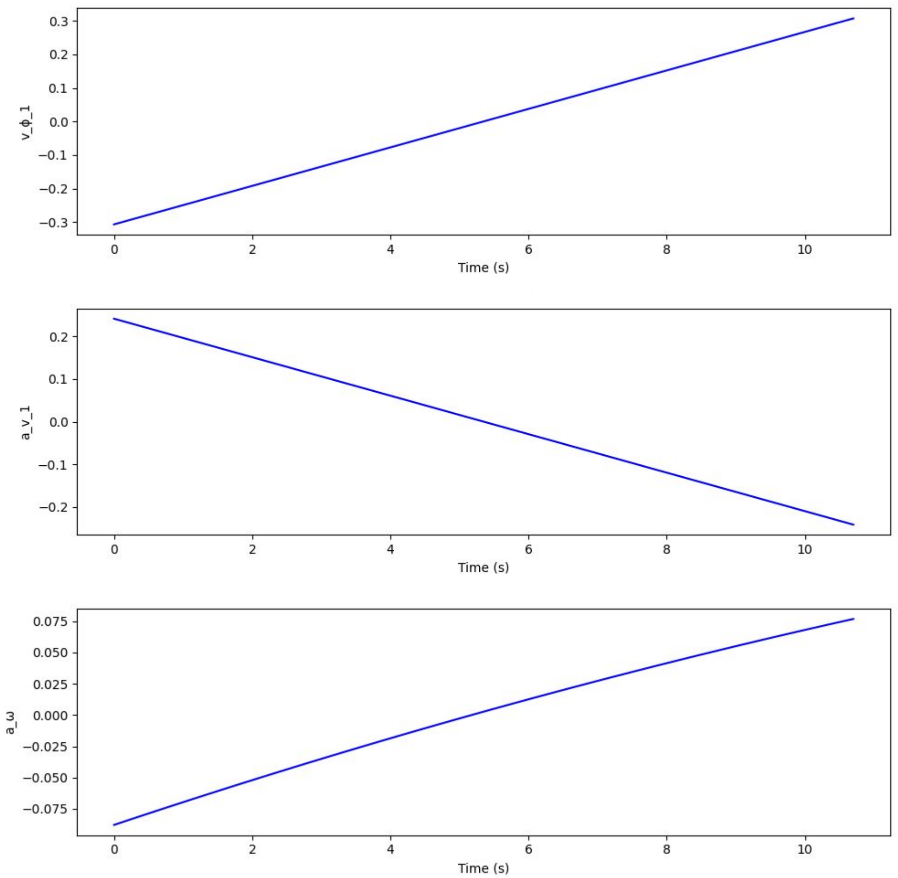


Medium-few Map

States



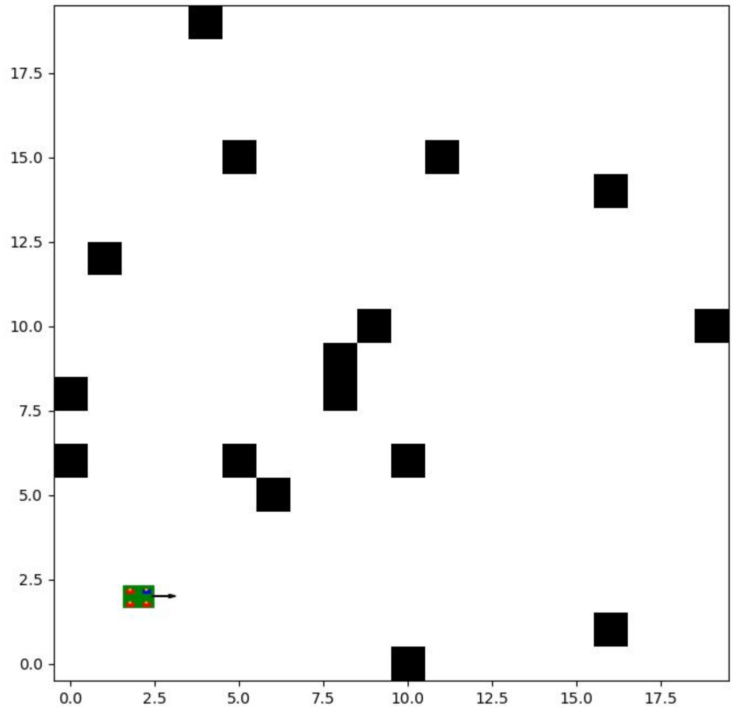
Inputs

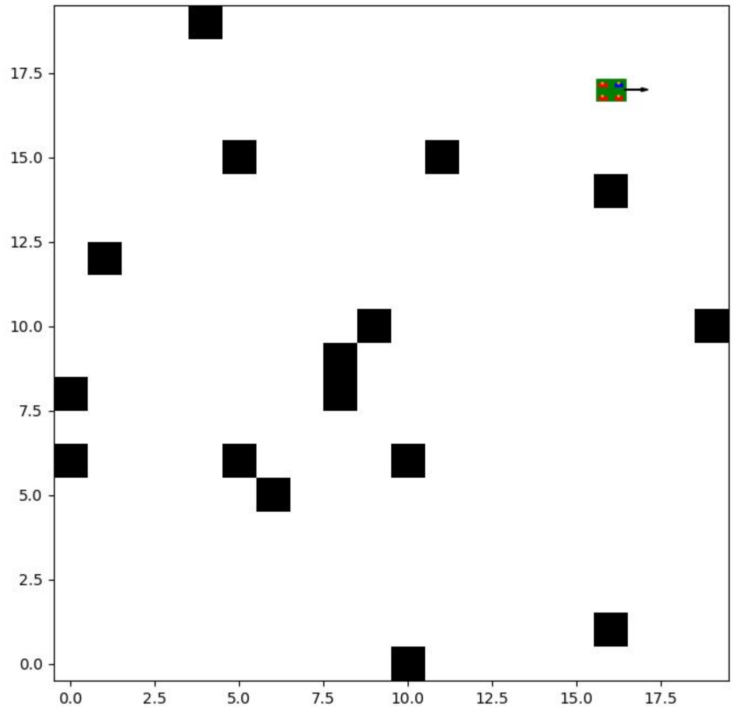


Medium-many Map

$$\text{Start state} = [2, 2, 0, 0, 0.1, 0]$$

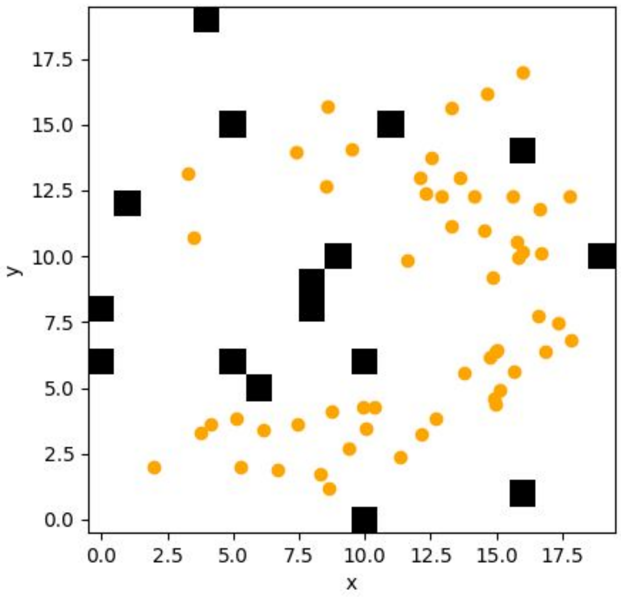
$$\text{Goal state} = [16, 17, 0, 0, 0.1, 0]$$

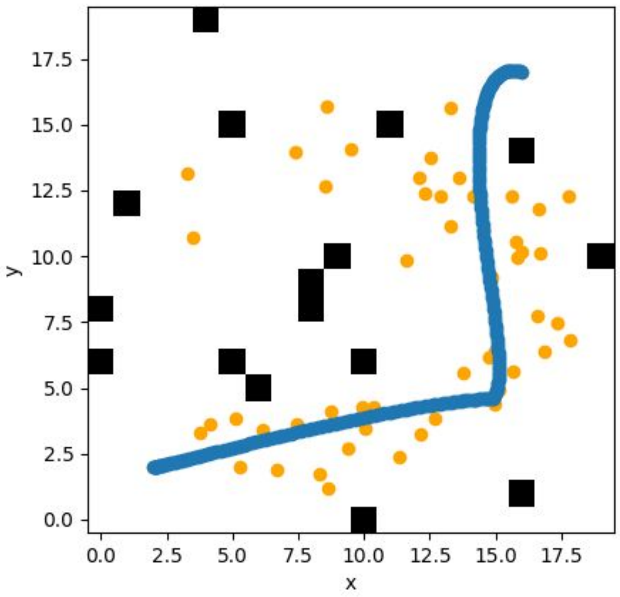




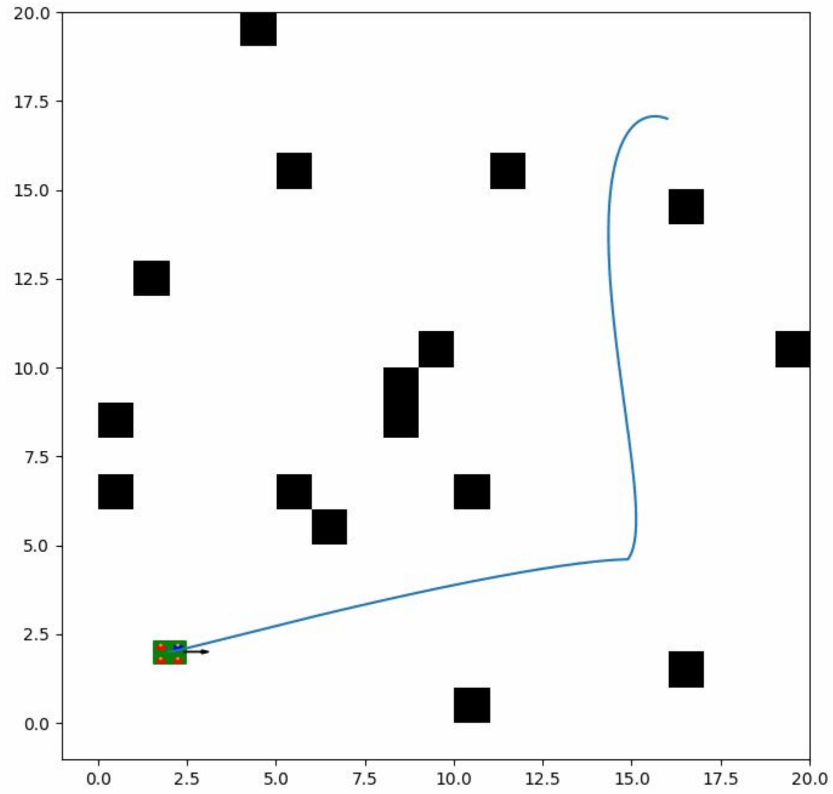
Medium-many Map

N = 500



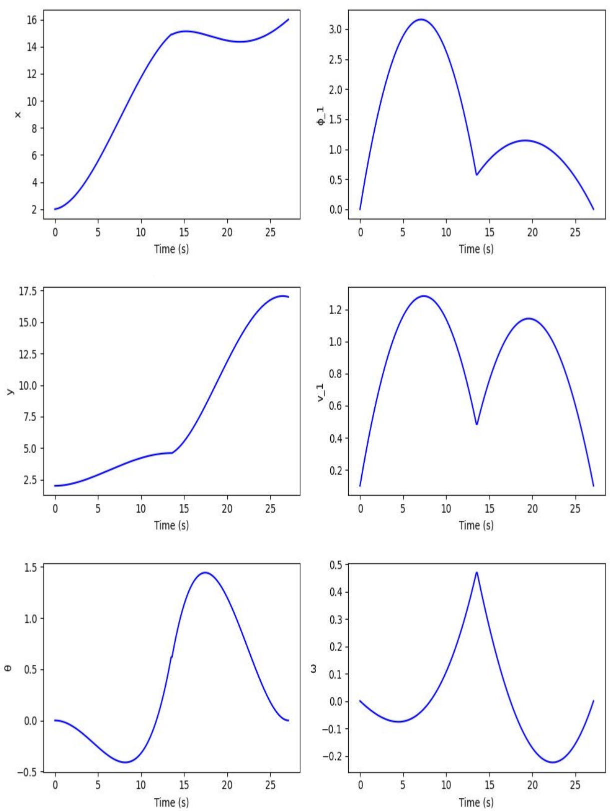


Medium-many Map

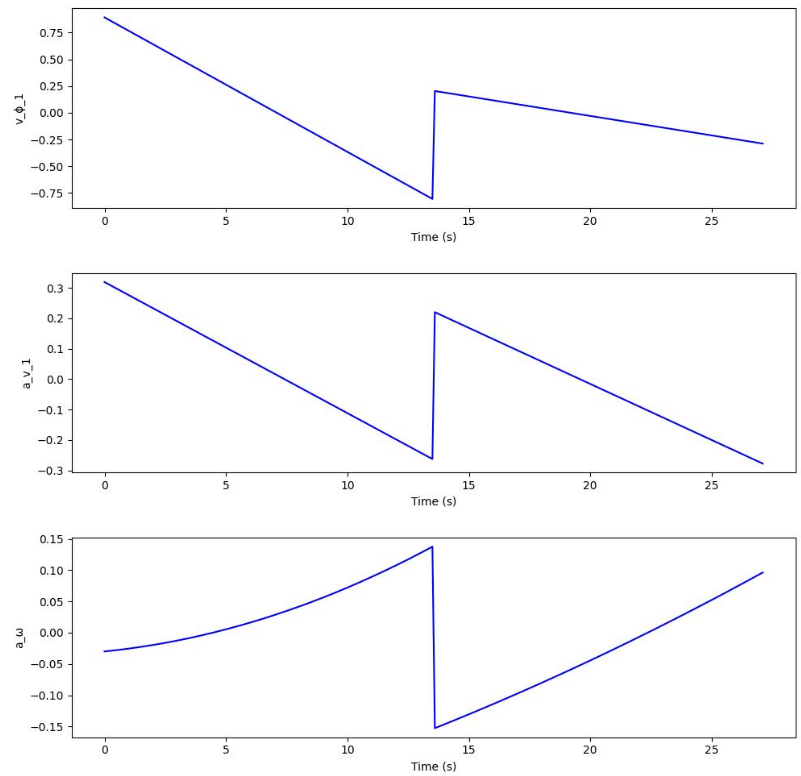


Medium-many Map

States

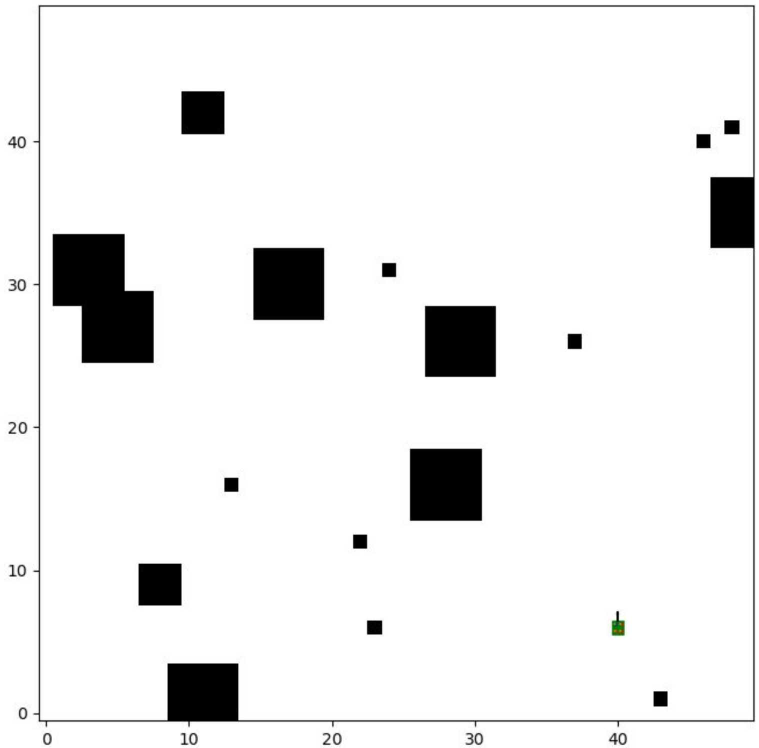


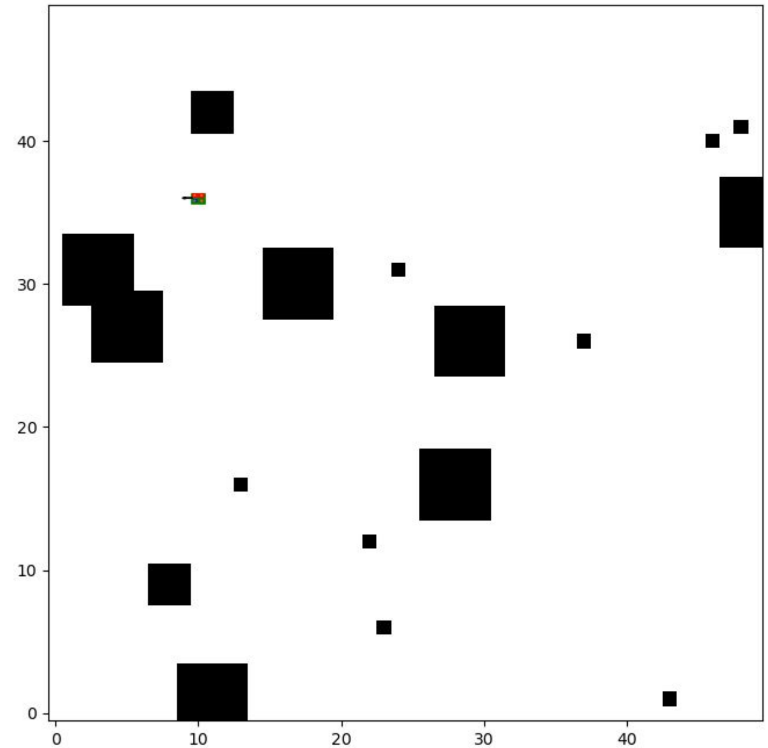
Inputs



Big-many Map

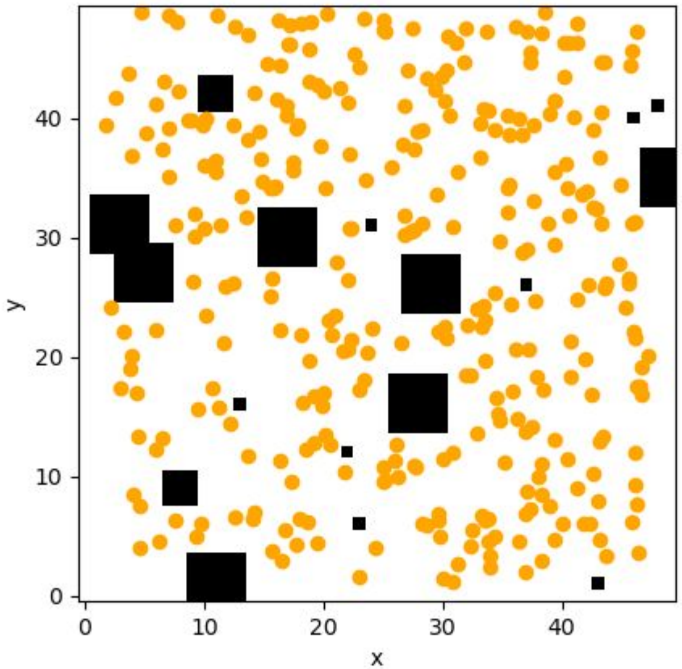
$$\text{Start state} = [40,\ 6,\ \tfrac{\pi}{2},\ 0,\ 1,\ 0] \quad \text{Goal state} = [10,\ 36,\ \pi,\ 0,\ 1,\ 0]$$

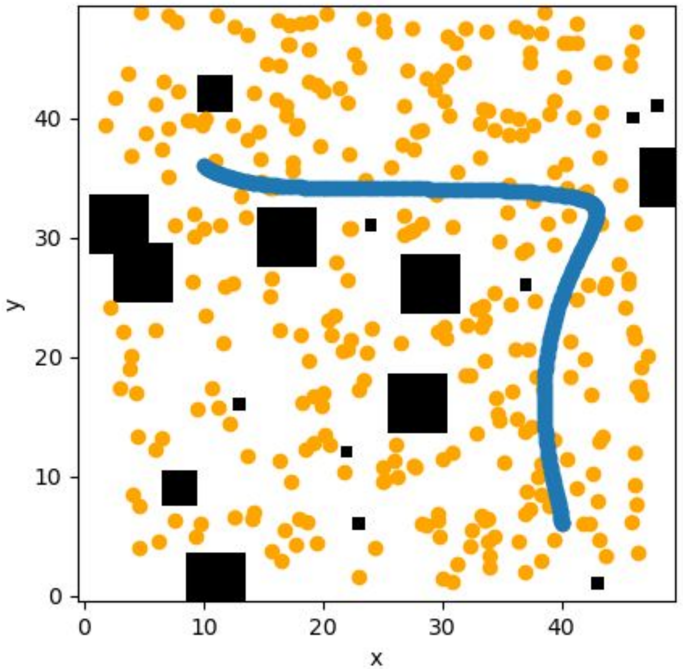




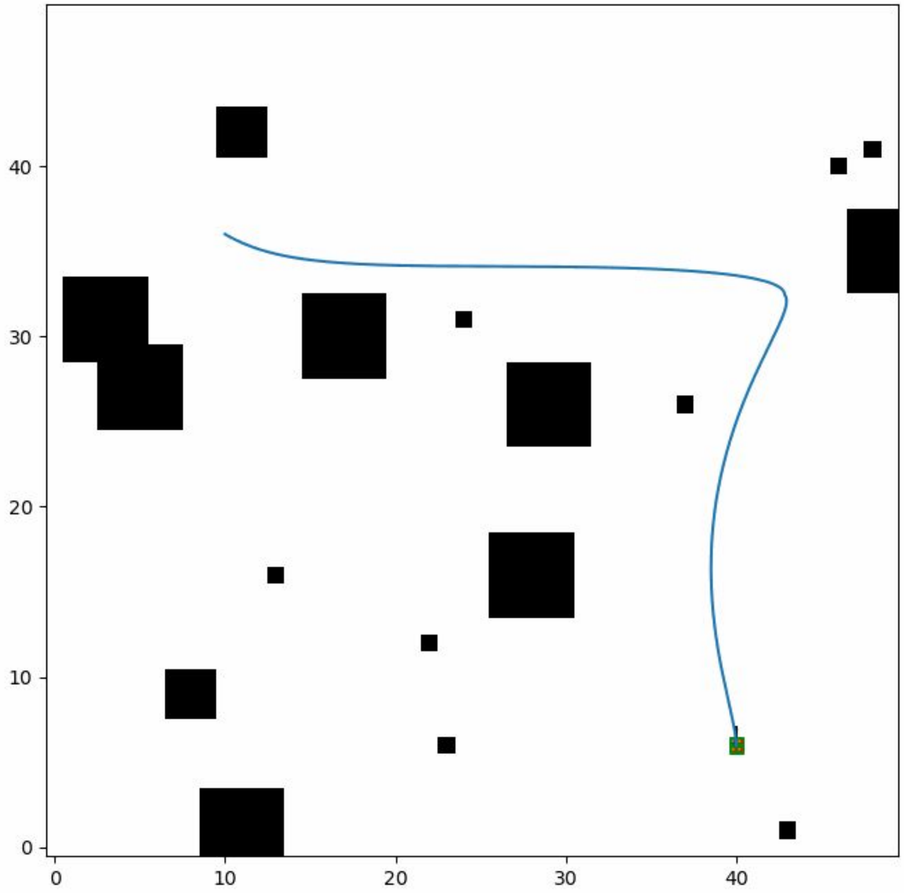
Big-many Map

N = 500



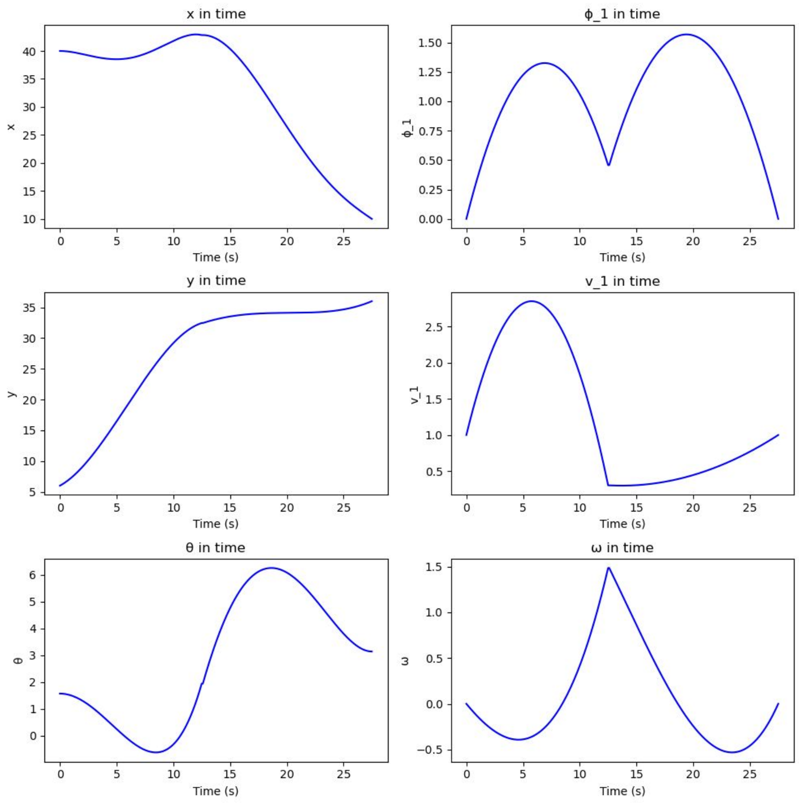


Big-many Map

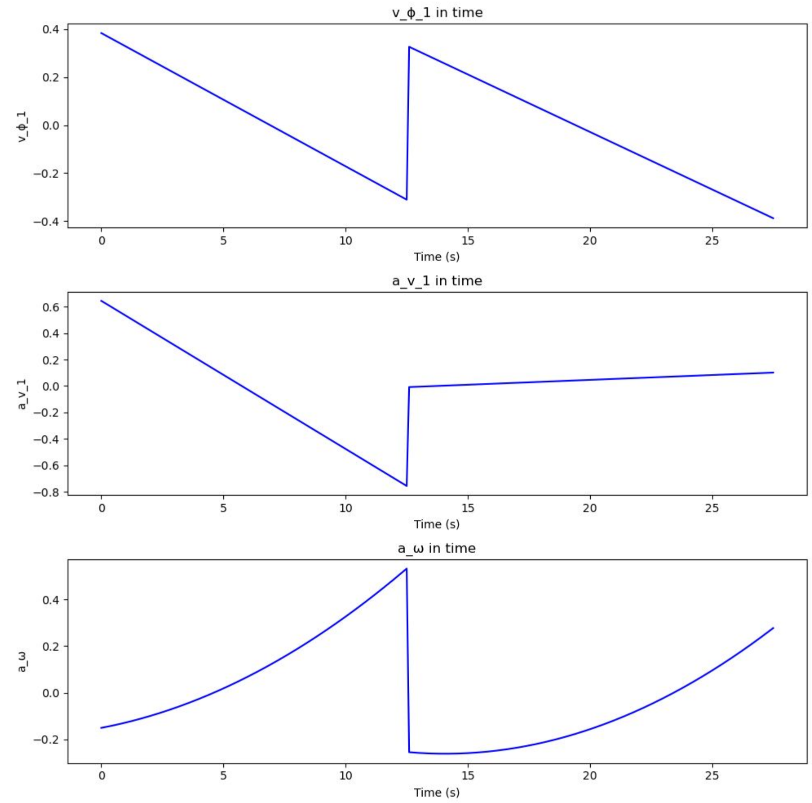


Big-many Map

States



Inputs



Results

Map	Cost	Iterations	KRRT* Time	Traj. Time	Nodes
Small-empty	84.59	100	42 (s)	33 (s)	19
Small-few	26.8	200	301 (s)	8 (s)	37
Small-many	249.69	500	1362 (s)	34 (s)	55
Medium-empty	37.51	100	367 (s)	28 (s)	53
Medium-few	13.66	500	2659 (s)	12 (s)	109
Medium-many	66.79	500	1491 (s)	28 (s)	56
Big-empty	18	1	0.12 (s)	15 (s)	2
Big-few	61.82	100	426 (s)	21 (s)	84
Big-many	64.691	500	9349 (s)	27 (s)	324

Conclusions

- Kinematic Model for SWMR
- KRRT*
- Different R matrices
- Number of iterations
- Obstacle avoidance

Future works

- K-d tree for efficient Neighbors search
- Controller for trajectory tracking

THANK YOU FOR YOUR ATTENTION