# COMP 550 Project 4 part a

Shrikant Ravindra Malagi, Nikhil Davanam Shivakumar
Net ID: sm315, nd66

October 28th, 2025

## Overview

This checkpoint implements two control-based motion planning algorithms—**RRT (Rapidly-exploring Random Tree)** and **KPIECE1 (Kinodynamic Planning by Interior-Exterior Cell Exploration)**—for two dynamic systems: a torque-controlled pendulum and a non-holonomic car navigating through obstacles.

## Implementation

### System 1: Pendulum Swing-Up

**Problem:** Swing an underactuated pendulum from hanging down ($\theta = 0°$) to upright ($\theta = 180°$) using limited torque control.
**State Space:** $(\theta, \omega)$ representing angle and angular velocity.
**Dynamics:**

$$\dot{\theta} = \omega, \qquad \dot{\omega} = -g \cdot \cos(\theta) + \tau$$

**Control:** Torque $\tau \in [-\text{torque\_limit}, +\text{torque\_limit}]$.
**KPIECE1 Configuration:** 2D projection on $(\theta, \omega)$ for discretization-based exploration.

### System 2: Car Navigation with Obstacles

**Problem:** Navigate a $5 \times 5$ square car through a constrained environment with 8 rectangular obstacles from start $(-40, -40)$ to goal $(40, 40)$.
**State Space:** $SE(2) \times \mathbb{R}$ representing $(x, y, \theta, v)$ — position, heading, and velocity.
**Dynamics (Non-holonomic):**

$$\dot{x} = v\cos(\theta), \qquad \dot{y} = v\sin(\theta), \qquad \dot{\theta} = \omega, \qquad \dot{v} = a$$

**Control:** $(\omega, a)$ — angular velocity $\omega \in [-2, 2]$ rad/s and acceleration $a \in [-3, 3]$ m/s$^2$.
**Collision Checking:** A custom `CarValidityChecker` class checks all four corners of the $5 \times 5$ square car, rotated by heading angle $\theta$, against obstacle boundaries using line-segment intersection tests.
**KPIECE1 Configuration:** 3D projection on $(x, y, v)$ to account for both spatial and dynamic constraints.

# Results

### Pendulum Performance

| Planner | Torque Limit | Time (s) | States Created | Outcome |
|---|---|---|---|---|
| RRT | 3 | 0.10 | ∼100 | Success |
| RRT | 5 | 3.90 | ∼400 | Success |
| KPIECE1 | 3 | 30.04 | ∼3000 | Success |

**Key Finding:** Lower torque limits do not necessarily increase difficulty—RRT with torque = 3 found solutions faster than higher torque values, likely due to the more constrained control space reducing exploration overhead.

### Car Navigation Performance

| Planner | Solve Time (s) | States Created | Final Position | Distance to Goal |
|---|---|---|---|---|
| RRT | 5.06 | 2,256 | (39.89, 39.36) | 0.65 units |
| KPIECE1 | 1.14 | 2,175 | ∼(40, 40) | $< 2.0$ units |

**Key Finding:** KPIECE1 was 4.4× faster than RRT for the car problem, demonstrating the advantage of discretization-based exploration for systems with complex kinodynamic constraints. The 3D projection $(x, y, v)$ helped KPIECE1 effectively balance spatial and velocity exploration.

# Technical Challenges

- **Square Collision Checking:** The initial implementation treated the car as a point robot, allowing invalid paths through obstacles. The solution involved implementing `isValidSquare()` to transform all four corners of the $5 \times 5$ car based on position and heading, then check edge-obstacle intersections.
- **Non-holonomic Constraints:** The car can only move along its heading direction (forward/backward), never sideways, making planning significantly harder than for holonomic systems.

# Visualizations

Comprehensive visualizations were generated, including:

- Trajectory plots showing the $5 \times 5$ square car at multiple timesteps with heading indicators.
- Workspace views with color-coded obstacles and start/goal positions.
- State evolution plots $(x(t), y(t), \theta(t), v(t))$ demonstrating constraint satisfaction.
- Pendulum phase portraits showing angular position and velocity evolution.

# 1 Environment

## Car

The Environment we have here is a $45 \times 45$ grid, several obstacle that arrange themselves to have several narrow passages. Start position is at $(-40, -40)$ and goal position is $(40, 40)$

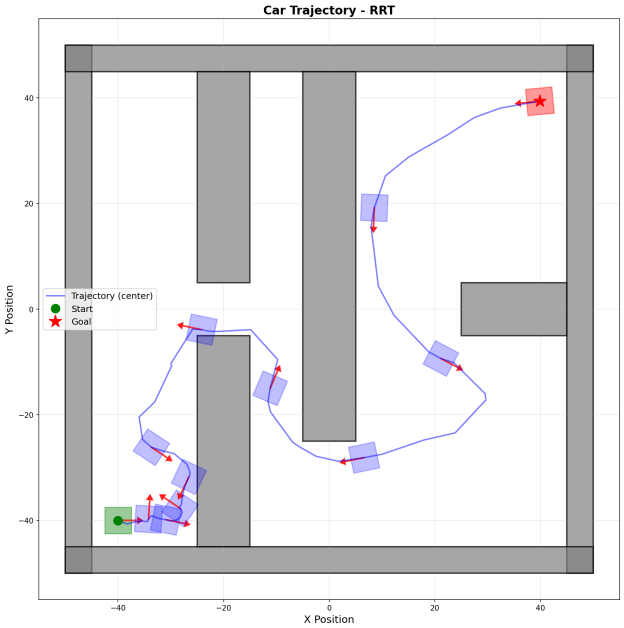


**Figure 1:** Car trajectory for rrt

In figure 1 the obstacles form narrow corridors, forcing the planner to generate longer, curved paths that respect the car's non-holonomic dynamics. The plotted heading arrows along the path indicate the vehicle's orientation at intermediate states, confirming dynamically feasible motion without sideways translation.

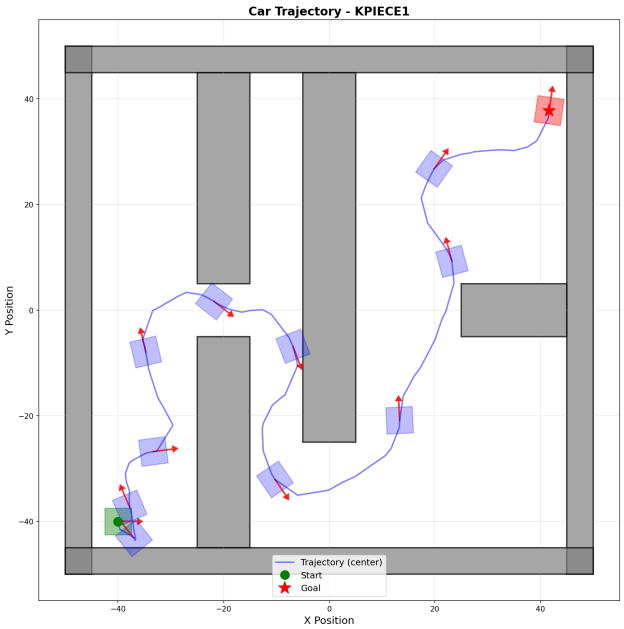


**Figure 2:** Car trajectory for KPIECE

In figure 2 the KPIECE planner produces a smoother, more directed path through the same environment. Its cell-based exploration in the (x, y, v) projection allows faster convergence by emphasizing regions of progress toward the goal. The resulting trajectory reaches the goal more efficiently and with fewer oscillations in heading compared to RRT.

## Pendulum

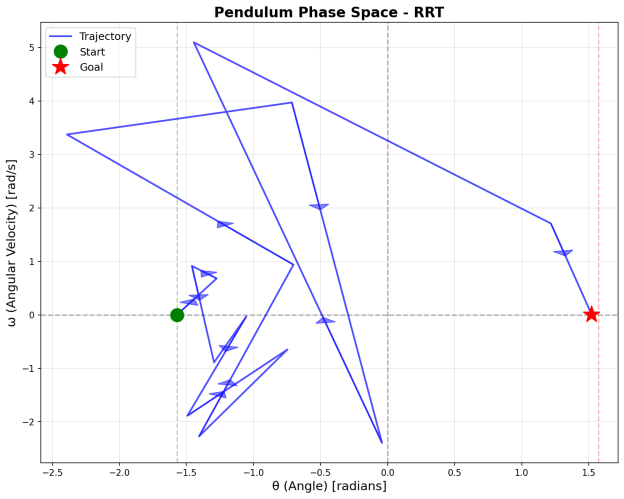


**Figure 3:** Phase space trajectory for RRT with 3 torque

In figure 3 the phase-space plot ( vs ) shows the pendulum swinging up to the upright position using limited torque. The repeated back and forth path demonstrates the accumulation of energy through successive swings.

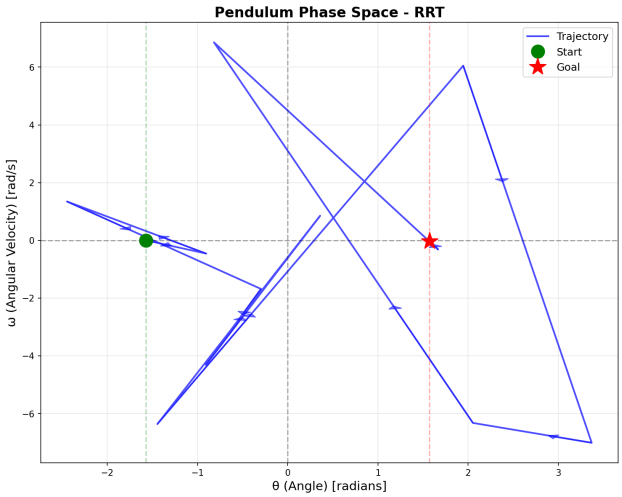


**Figure 4:** Phase space trajectory for RRT with 5 torque

In figure 4 with a higher torque limit, the pendulum reaches the upright state more directly, requiring fewer oscillations. The phase trajectory expands outward more quickly, indicating stronger actuation.

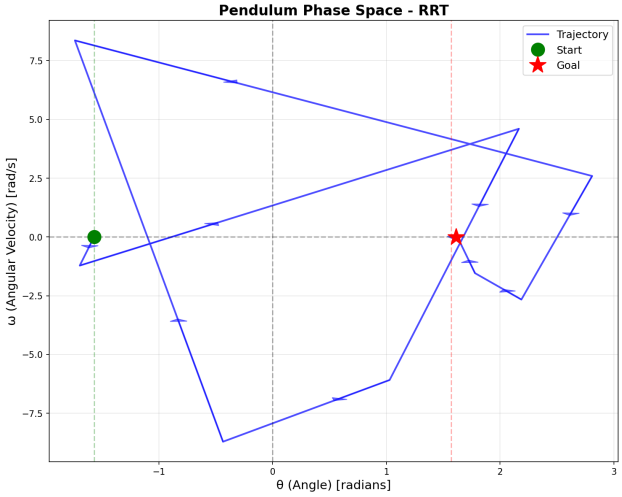


**Figure 5:** Phase space trajectory for RRT with 10 torque

In figure 5 nearly geometric behavior is visible here: the pendulum moves rapidly to the goal with minimal swinging, as the high torque overcomes gravitational potential almost instantaneously.

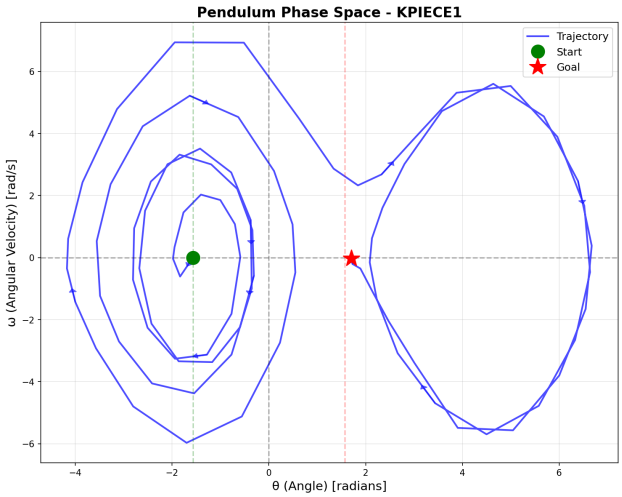


**Figure 6:** Phase space trajectory for KPIECE with 3 torque

In figure 6 KPIECE efficiently explores the phase space, showing consistent energy accumulation toward the target region. The discretized projection on (, ) yields stable coverage of feasible states.

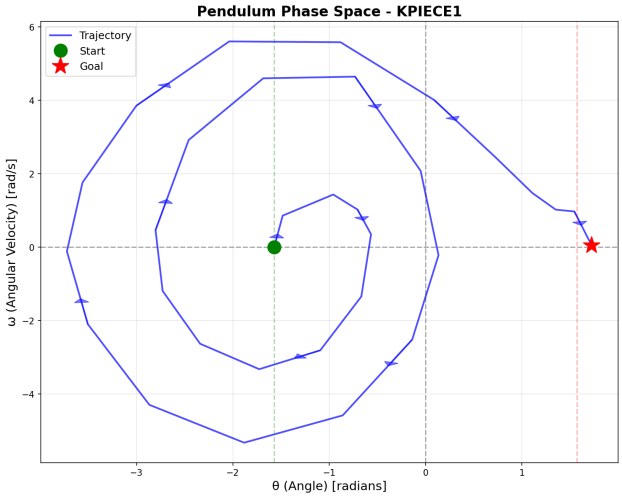


**Figure 7:** Phase space trajectory for KPIECE with 5 torque

In figure 7 the trajectory converges faster to the upright position than with $= 3$, confirming better control authority. The exploration grid highlights KPIECE's systematic search pattern.

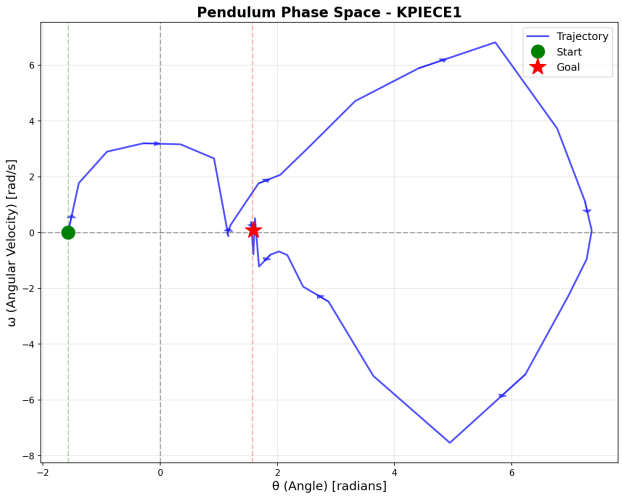


**Figure 8:** Phase space trajectory for KPIECE with 10 torque

In figure 8high torque again produces a near-straight path to the goal. The figure illustrates how larger control bounds simplify the planning landscape, with fewer required intermediate swings.

All visualizations confirm that solutions respect system dynamics and collision constraints.

## Conclusion

Both RRT and KPIECE1 successfully solved control-based planning problems for underactuated and non-holonomic systems. KPIECE1's discretization-based approach with appropriate projections provides significant performance advantages for kinodynamically constrained problems like car navigation. The implementation correctly handles complex collision checking for non-point robots and respects all dynamic constraints.

# 2 Difficulty and contribution

- Exercise 1 : difficulty 6/10 took us around 10 hours, figuring out how to check valid states took us some time.
- Exercise 2: difficulty 6/10 took us around 12 hours.

Shrikant worked on setting up environment for car and planning for car and Nikhil worked on Pendulum and visualization. We also sort of started working on benchmarking