Parking Algorithm based on Multiple Paths (pending to update)

Alejandro D.J Gomez Florez, November 26, 2025
Chair of Ms Applied Mathematics | Universidad EAFIT

UNIVERSIDAD EAFIT

Contents

1 · Introduction 1
1.1 · Problem Definition 1
1.2 · Simulations 2
2 · Related Work 4

1 · Introduction

1.1 · Problem Definition

In this work, we develop a probabilistic algorithm to address the autonomous parking problem. We model the vehicle as a point mass in a 2D plane with position and orientation, following the unicycle kinematic model. The vehicle's state is represented by the vector $X = (x, y, \theta)^T$, where (x, y) denotes the position and θ represents the orientation angle. The continuous-time dynamics are governed by:

$$\dot{X} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \end{pmatrix}$$

where v is the linear velocity and ω is the angular velocity, which constitute the control inputs $u = (v, \omega)^T$ to the vehicle. For implementation purposes, we discretize this model and formulate it as a discrete-time state-space system:

$$X_{k+1} = f(X_k, u_k) + w_k$$

where $f(\cdot)$ denotes the discretized unicycle dynamics, and $w_k \sim \mathcal{N}(0, Q)$ is the process noise capturing model uncertainties such as actuator imperfections, wheel slip, and unmodeled dynamics.

In the autonomous parking scenario, the vehicle must reach a target configuration known as the **docking pose**. However, neither its position nor its orientation is perfectly known due to sensor noise, odometry drift, and environmental disturbances. To properly account for these uncertainties, the vehicle's state is modeled not as a deterministic vector, but as a probability distribution over X. This distribution, commonly referred to as the *belief state*, represents the vehicle's knowledge about its own configuration. A standard choice is to approximate this belief as a Gaussian distribution:

$$b(X) = \mathcal{N}(X \mid \mu, P)$$

where μ is the mean state estimate and P is the covariance matrix quantifying the uncertainty.

In our setting, the orientation θ is obtained from an absolute heading sensor (a magnetometer), while (x, y) are estimated independently. This implies that the initial uncertainties in position and orientation can be treated as statistically independent, leading to an initial covariance matrix of the form

$$P_0 = \begin{pmatrix} \sigma_x^2 & \operatorname{cov}(x, y) & 0 \\ \operatorname{cov}(y, x) & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{pmatrix}$$

Although the initial correlations $\operatorname{cov}(x, \theta)$ and $\operatorname{cov}(y, \theta)$ are zero, this independence does not persist during motion. Due to the structure of the unicycle model, even small errors in orientation induce lateral deviations in (x, y) after one prediction step. As a result, the covariance propagation

$$P_{k+1}^- = F_k P_k F_k^T + Q$$

naturally introduces correlations between position and orientation, where F_k is the Jacobian of the motion model. This effect reflects a fundamental property of differential-drive vehicles: uncertainty in orientation inevitably affects uncertainty in position during movement.

1.2 · Simulations

To evaluate the proposed approach, we conduct numerical simulations in a 2D planar environment. The simulations focus on generating feasible trajectories from uncertain initial configurations to a target docking pose.

1.2.1 · Initial Configuration Generation

The vehicle's initial pose is not precisely known. We model this uncertainty by sampling initial configurations from a Gaussian distribution centered around an expected region. Specifically, initial positions (x, y) are drawn from a bivariate normal distribution with center $\mu_0 = (x_0, y_0)$ and standard deviation $\sigma_{\text{pos}} = \frac{r}{2}$, where r characterizes the spatial uncertainty. Similarly, the initial orientation θ follows a normal distribution with mean θ_0 and standard deviation σ_θ.

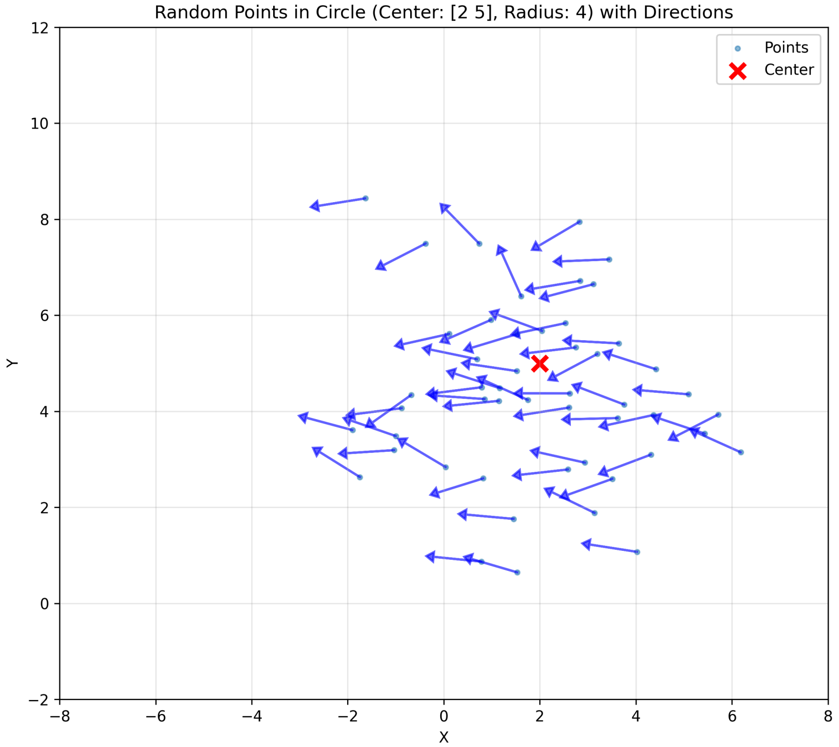

Figure 1: Distribution of initial vehicle configurations. The arrows indicate orientation, demonstrating the Gaussian spread in both position and heading.

1.2.2 · Spline-Based Path Generation

Given a start pose $X_{\text{start}} = (x_{\text{start}}, y_{\text{start}}, \theta_{\text{start}})^T$ and a target pose $X_{\text{target}} = \left(x_{\text{target}}, y_{\text{target}}, \theta_{\text{target}}\right)^T$, we generate smooth trajectories using cubic spline interpolation. To ensure that the generated paths respect the initial and final orientations, we enforce tangent boundary conditions at both endpoints:

$$(\dot{x}(0), \dot{y}(0)) = \lambda(\cos(\theta_{\text{start}}), \sin(\theta_{\text{start}}))$$

$$\left(\dot{x}(t_f), \dot{y}(t_f)\right) = \lambda\left(\cos\left(\theta_{\text{target}}\right), \sin\left(\theta_{\text{target}}\right)\right)$$

where $\lambda > 0$ is a derivative magnitude parameter that controls how strongly the path adheres to the boundary orientations. Intermediate waypoints are selected to shape the trajectory, but do not carry orientation constraints.

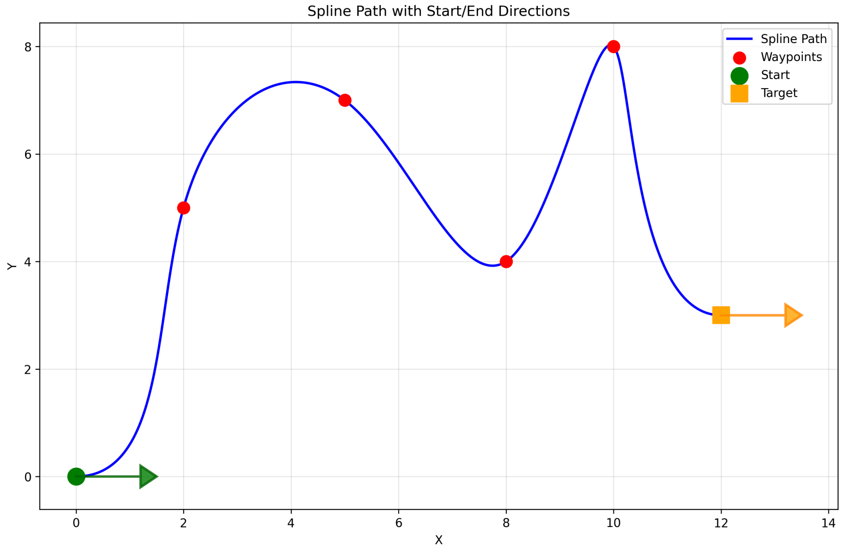

Figure 2: Example of a cubic spline path from start (green) to target (orange) through multiple waypoints (red). The arrows indicate the enforced tangent directions at the boundaries.

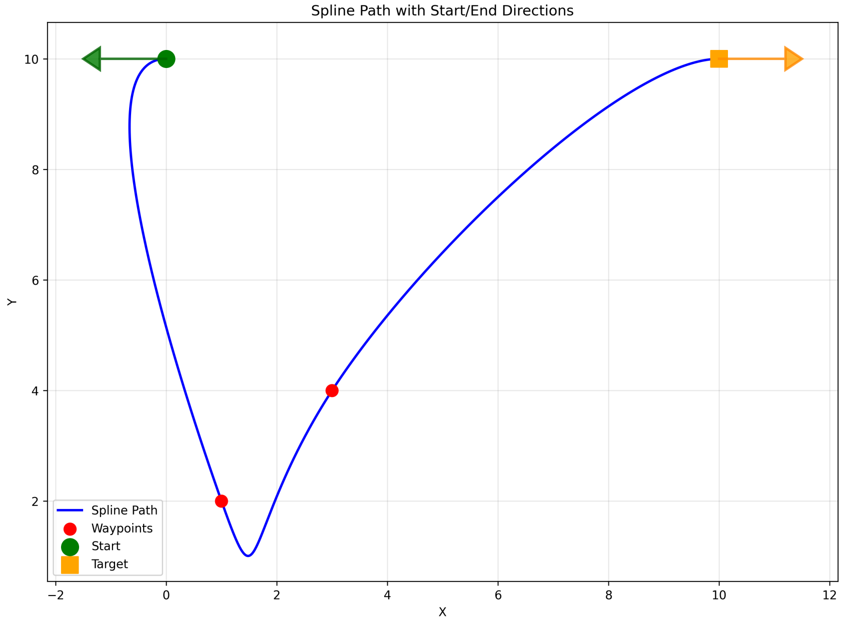

Figure 3: Another example showing how boundary conditions ensure smooth entry and exit angles aligned with the specified orientations.

1.2.3 · Multiple Paths from Uncertain Initial States

In a realistic parking scenario, multiple initial configurations exist simultaneously, and each must be routed toward the common docking pose. To address this, we generate a collection of trajectories by randomly sampling waypoints from predefined waypoint pools. This approach introduces diversity in the path geometry while maintaining smoothness and orientation constraints.

The simulation generates N starting points from the uncertain initial distribution. For each starting pose, a path is constructed by selecting one waypoint from each of two waypoint pools, followed by the target. This results in N distinct trajectories, each tailored to a different initial condition.

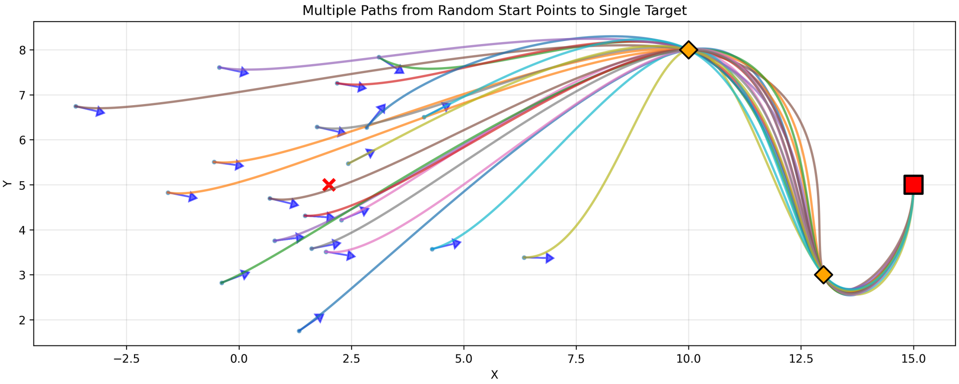

Figure 4: Multiple trajectories generated from 20 random initial configurations. Each path smoothly connects a start pose (shown with arrow) to the target docking pose through intermediate waypoints.

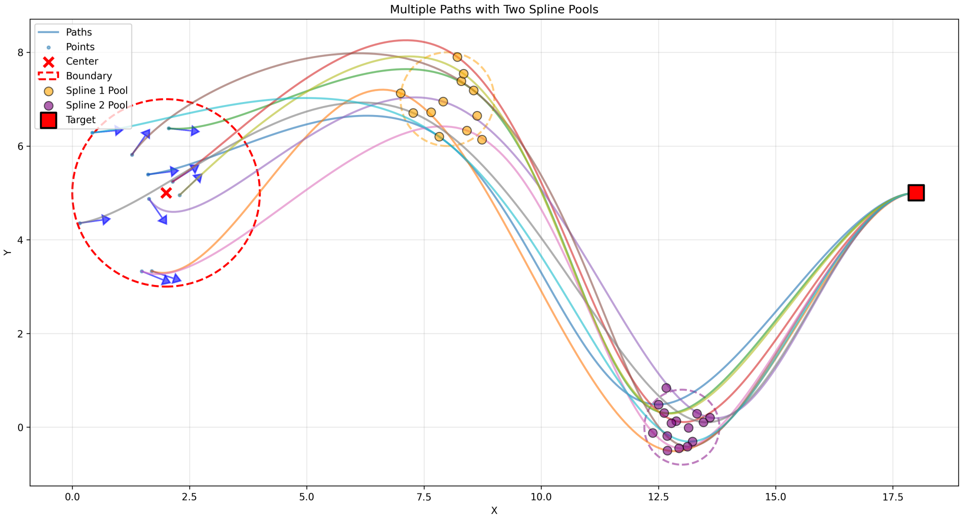

Figure 5: Illustration of the path generation strategy using two waypoint pools. Each trajectory randomly selects one waypoint from the first pool (orange) and one from the second pool (purple), ensuring geometric diversity while converging to the shared target.

These simulations demonstrate the feasibility of generating smooth, orientation-consistent paths from a distribution of uncertain initial states to a precise docking configuration.

1.2.4 · Really Small Stuff

2 · Related Work