Parking Algorithm based on Multiple Paths (pending to update)

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

Contents

1 · Introduction

The implementation and simulations presented in this work are available in the project repository: https://github.com/aldajo92/DockingAlgorithmEafit.

1.1 · Problem Definition

This work develops a probabilistic algorithm to address the autonomous parking problem. The vehicle is modeled 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, this model is discretized and formulated 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 this 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, numerical simulations are conducted 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. This uncertainty is modeled 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.

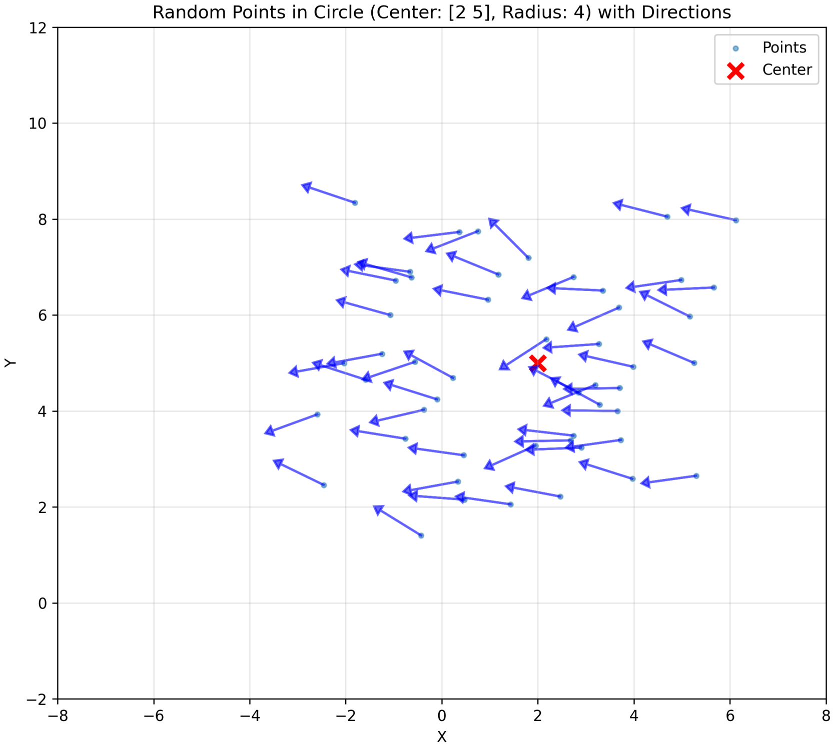

Figure 1: Distribution of initial vehicle configurations. The arrows indicate orientation, demonstrating the Gaussian spread in both position and heading.

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 $X_{\text{target}} = (x_{\text{target}}, y_{\text{target}}, \theta_{\text{target}})^T$, smooth trajectories are generated using cubic spline interpolation. To ensure that the generated paths respect the initial and final orientations, tangent boundary conditions are enforced at both endpoints:

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

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

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.

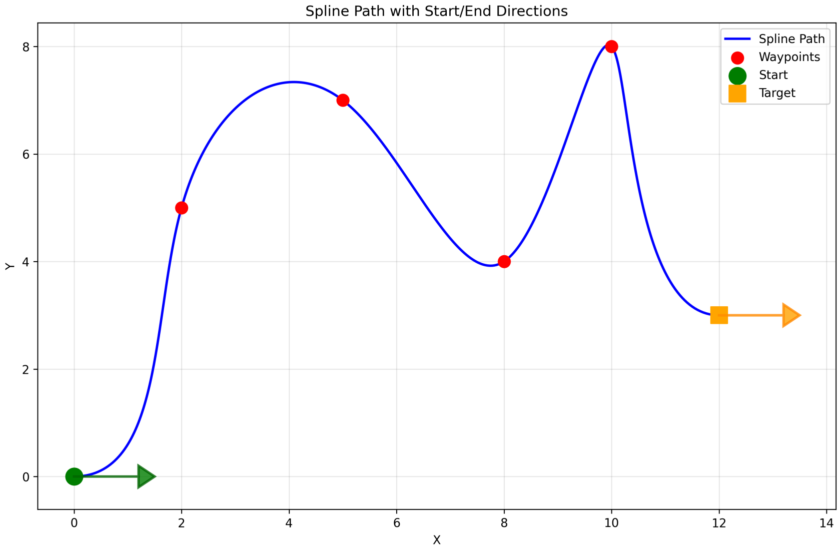

Figure 2: Example of a cubic spline path from start (green) to target (orange) through multiple waypoints (red). The arrows indicate the enforced tangent directions at the boundaries.

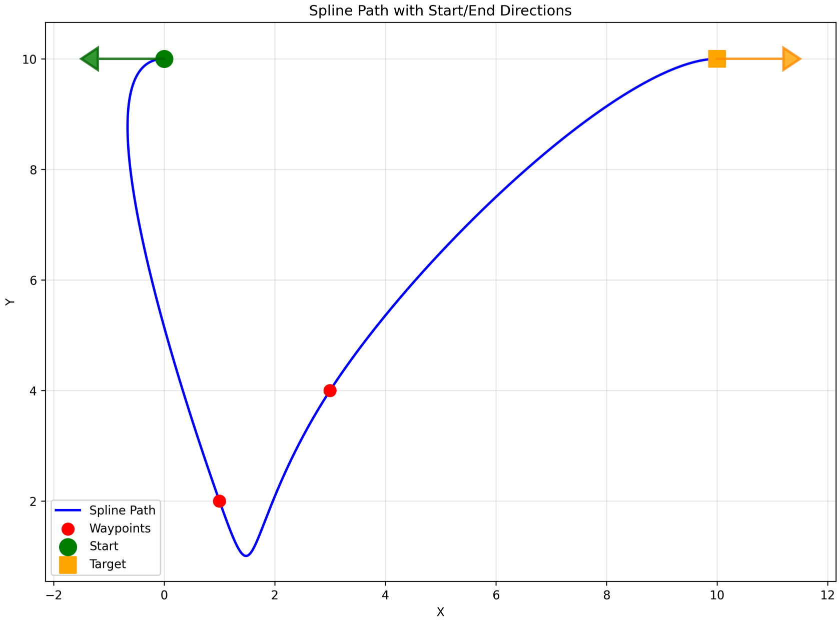

Figure 3: Another example showing how boundary conditions ensure smooth entry and exit angles aligned with the specified orientations.

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, a collection of trajectories is generated 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.

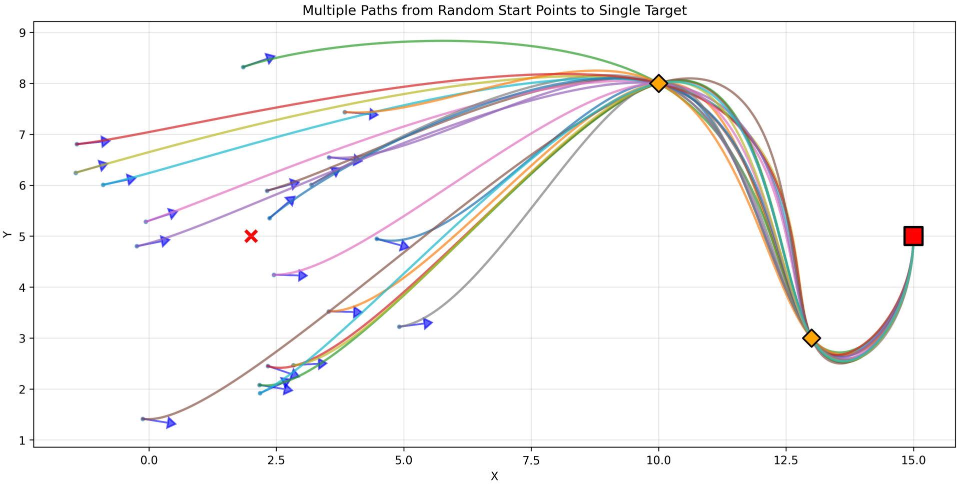

Figure 4: Multiple trajectories generated from 20 random initial configurations. Each path smoothly connects a start pose (shown with arrow) to the target docking pose through intermediate waypoints.

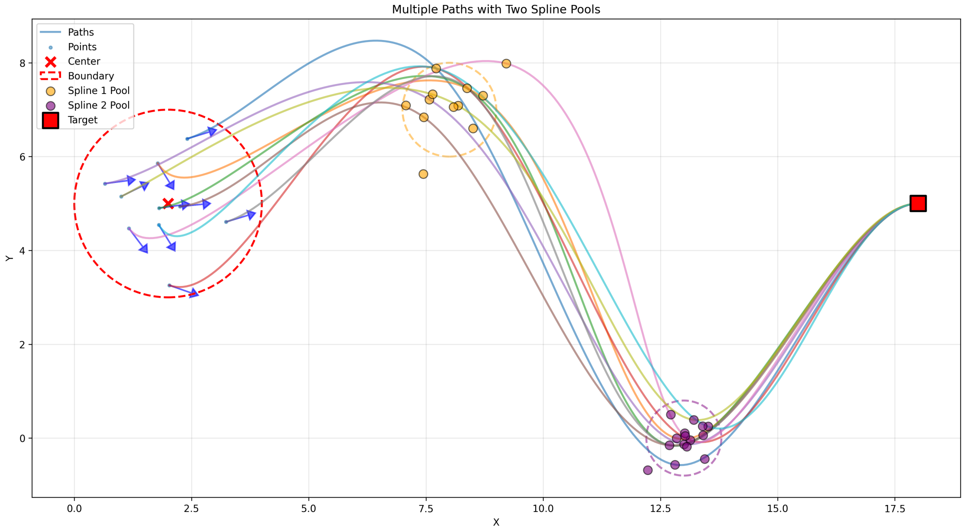

Figure 5: Illustration of the path generation strategy using two waypoint pools. Each trajectory randomly selects one waypoint from the first pool (orange) and one from the second pool (purple), ensuring geometric diversity while converging to the shared target.

1.2.4 · Modified Band Depth Analysis

To identify the most representative trajectory among the generated paths, the **Modified Band Depth (MBD)** criterion is employed, a statistical measure from functional data analysis. The MBD quantifies how “central” a trajectory is relative to the entire ensemble. A trajectory with high MBD is more representative and typical of the collection, whereas one with low MBD is more peripheral or atypical.

1.2.4.1 · Trajectory Discretization

Since trajectories may be represented by different numbers of points or sampled at non-uniform intervals, they are first standardized through interpolation. Each trajectory $\gamma_i : [0, L_i] \to \mathbb{R}^2$ is resampled to a fixed number of points M:

$$\tilde{\gamma}_i = \{\boldsymbol{p}_i^1, \boldsymbol{p}_i^2, ..., \boldsymbol{p}_i^M\}$$

where each $\boldsymbol{p}_i^j = \left(x_i^j, y_i^j\right)^T$ corresponds to a uniformly spaced point along the arc-length parameterization of the trajectory. This ensures that all trajectories are represented in a common functional space.

1.2.4.2 · Band Depth Between Trajectories

For three discretized trajectories $\tilde{\gamma}_i$, $\tilde{\gamma}_j$, and $\tilde{\gamma}_k$, the **band** formed by $\tilde{\gamma}_j$ and $\tilde{\gamma}_k$ at each discrete parameter value $t \in \{1, ..., M\}$ is defined as the rectangular region:

$$B_t(\tilde{\gamma}_j, \tilde{\gamma}_k) = \left[\min\left(x_j^t, x_k^t\right), \max\left(x_j^t, x_k^t\right)\right] \times \left[\min\left(y_j^t, y_k^t\right), \max\left(y_j^t, y_k^t\right)\right]$$

A point $\boldsymbol{p}_i^t$ of trajectory $\tilde{\gamma}_i$ lies inside the band if:

$$\boldsymbol{p}_i^t \in B_t(\tilde{\gamma}_j, \tilde{\gamma}_k)$$

The **band depth** of $\tilde{\gamma}_i$ with respect to the pair $(\tilde{\gamma}_j, \tilde{\gamma}_k)$ is then:

$$\mathrm{BD}(\tilde{\gamma}_i \mid \tilde{\gamma}_j, \tilde{\gamma}_k) = \frac{1}{M} \sum_{t=1}^{M} \mathbb{1}_{\boldsymbol{p}_i^t \in B_t}$$

where $\mathbb{1}_{.}$ is the indicator function. This proportion measures how often $\tilde{\gamma}_i$ remains within the envelope defined by $\tilde{\gamma}_j$ and $\tilde{\gamma}_k$.

1.2.4.3 · Modified Band Depth

Given a set of N trajectories, the Modified Band Depth of trajectory $\tilde{\gamma}_i$ is defined as the average band depth over all pairs of distinct trajectories:

$$\mathrm{MBD}(\tilde{\gamma}_i) = \left(\binom{N}{2}\right)^{-1} \sum_{j<k, j,k \neq i} \mathrm{BD}(\tilde{\gamma}_i \mid \tilde{\gamma}_j, \tilde{\gamma}_k)$$

A higher MBD value indicates that $\tilde{\gamma}_i$ tends to lie within the envelope formed by most pairs of other trajectories, making it more “central” to the distribution. Conversely, a lower MBD indicates a more peripheral or outlier trajectory.

1.2.4.4 · Computational Complexity

For N trajectories and M discretization points, the computational complexity is $\mathcal{O}(N^3 M)$, since the algorithm evaluates N trajectories against $\binom{N}{2} \approx N^2/2$ pairs, and each band depth computation requires $\mathcal{O}(M)$ operations. For large trajectory sets, random sampling of trajectory pairs can be employed to reduce the computational burden to $\mathcal{O}(NKM)$, where $K \ll \binom{N}{2}$ is the number of sampled pairs.

1.2.4.5 · Results

The MBD analysis is applied to two scenarios: trajectories generated from random initial configurations with fixed waypoints, and trajectories with additional variability introduced through randomized waypoint selection from predefined pools.

In the first scenario, with 30 trajectories from Gaussian-distributed initial poses, the MBD analysis identifies the most central path (green) that best represents the typical behavior of the ensemble. The peripheral trajectories (red) correspond to initial configurations at the extremes of the distribution.

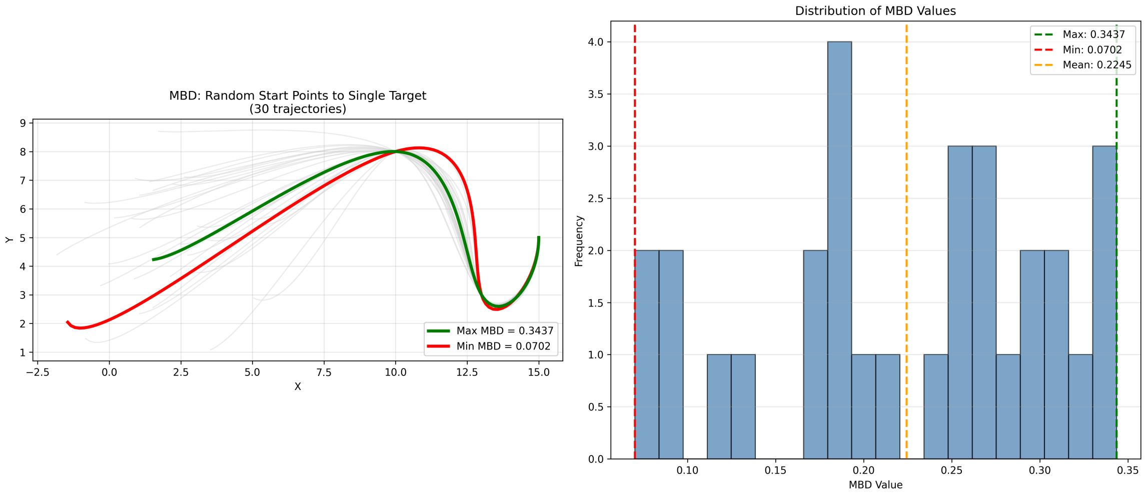

Figure 6: MBD analysis for 30 trajectories from random start points. Left: All trajectories (gray) with the most central (green, MBD = 0.3437) and most peripheral (red, MBD = 0.0702) highlighted. Right: Distribution of MBD values showing mean = 0.2245.

In the second scenario, 25 trajectories are generated with randomized waypoint selection from two spatial pools. This introduces geometric diversity not only in the initial conditions but also in the intermediate path structure. The MBD analysis successfully identifies trajectories that navigate through typical waypoint selections versus those that follow more unusual routes.

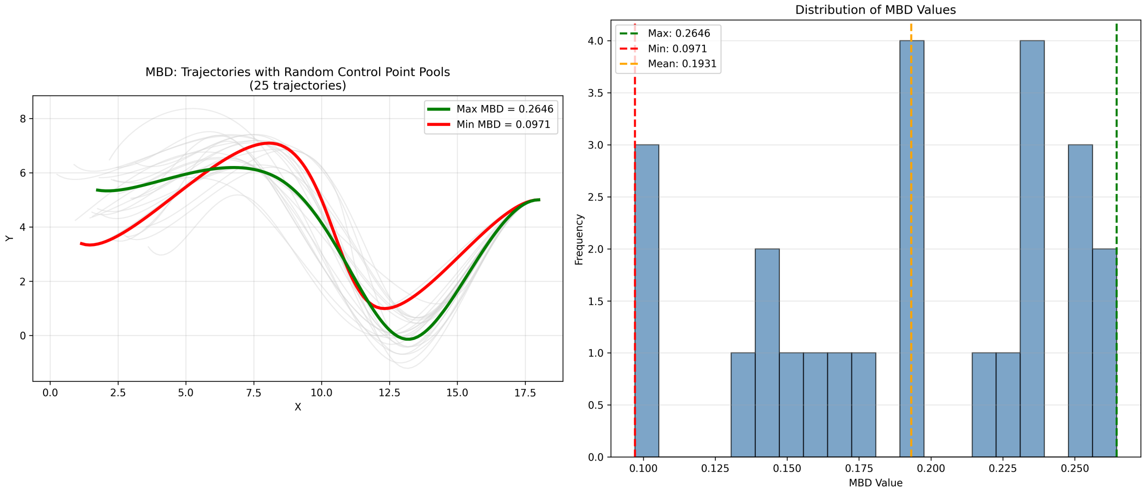

Figure 7: MBD analysis for 25 trajectories with random control point pools. The most central trajectory (green, MBD = 0.2646) represents a typical waypoint selection strategy, while the peripheral trajectory (red, MBD = 0.0971) follows a less common route. Mean MBD = 0.1931.

To further analyze the path centrality, all trajectories are ranked by their MBD values. The top-ranked trajectories represent the most typical or "consensus" paths, while the bottom-ranked ones correspond to outliers or edge cases that may be suboptimal or risky for autonomous navigation.

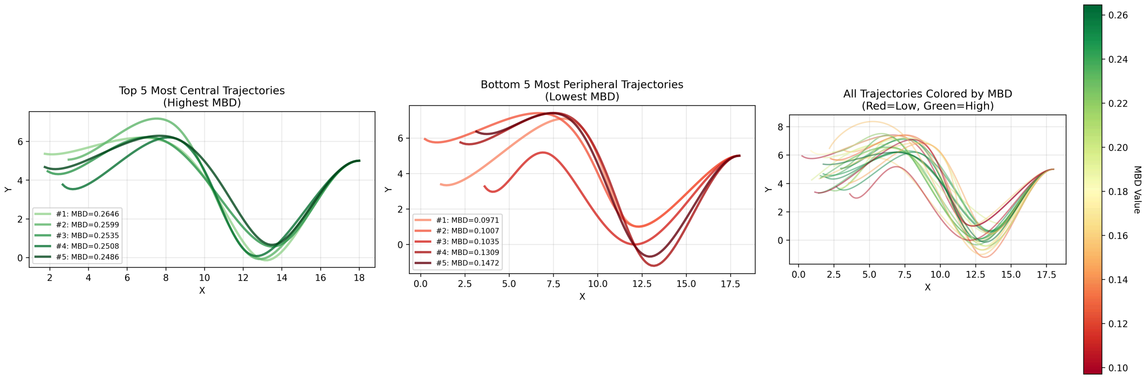

Figure 8: MBD ranking visualization. Left: Top 5 most central trajectories (green shades). Center: Bottom 5 most peripheral trajectories (red shades). Right: All trajectories color-coded by MBD value (red = low, green = high), providing a global view of path centrality.

The ranking analysis enables the selection of robust reference paths for motion planning. High-MBD trajectories are more likely to be feasible under uncertainty, as they represent typical behaviors of the system. This information can guide the choice of nominal trajectories for trajectory tracking controllers or serve as heuristic priors for optimization-based planners.