2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

A Multimodal Stochastic Planning Approach for Navigation and Multi-Robot Coordination

Mark Gonzales, Ethan Oh, Joseph Moore

***Abstract*— In this paper, we present a receding-horizon, sampling-based planner capable of reasoning over multimodal policy distributions. By using the cross-entropy method to optimize a multimodal policy under a common cost function, our approach increases robustness against local minima and promotes effective exploration of the solution space. We show that our approach naturally extends to multi-robot collision-free planning, enables agents to share diverse candidate policies to avoid deadlocks, and allows teams to minimize a global objective without incurring the computational complexity of centralized optimization. Numerical simulations demonstrate that employing multiple modes significantly improves success rates in trap environments and in multi-robot collision avoidance. Hardware experiments further validate the approach's real-time feasibility and practical performance.**

I. Introduction

Local minima pose a fundamental challenge for finite-horizon, gradient-based planning approaches. In multi-robot scenarios, local minima can arise not only from the environment but also from dynamic factors, such as the changing trajectories of teammates, which may inadvertently block or cut off routes that would otherwise be viable. These pitfalls often cause robots to become stuck, find suboptimal solutions, or fail to coordinate effectively in complex environments.

Sampling-based planners, such as Model Predictive Path Integral (MPPI) [1] and Cross-Entropy Method (CEM) [2], [3], attempt to improve the trajectory cost by stochastically sampling and evaluating trajectories in the cost landscape. In practice, these methods utilize hyperparameters, such as sampling variance, number of samples, and the horizon length, to adapt the exploration to the environment. However, both MPPI and CEM typically sample trajectories around the prior best policy, leading to a concentration of samples in a narrow region of the solution space. This localized search impedes the planner's ability to effectively navigate around traps or escape from local minima once they occur, especially in environments with challenging topology. As a result, the planner can become stuck in suboptimal regions, regardless of the variance or adaptation strategy.

In multi-robot systems, the difficulty is exacerbated by the need for robots to coordinate planned trajectories. Centralized control approaches [4], [5], [6], [7] can, in principle, achieve globally optimal coordination; however, they suffer from scalability issues and high computational costs as the team size increases. Distributed methods, while scalable, often require robots to individually select their optimal trajectory, subsequently negotiating with teammates to reach a feasible consensus. When each robot contributes only a single candidate trajectory, the team risks deadlock or persistent local minima, as a lack of trajectory diversity reduces the likelihood of discovering collision-free, cooperative maneuvers, especially when teammates dynamically update their plans or block each other's routes in real-time.

Johns Hopkins University
{MGonza60, EOh18, JLMoore}@jh.edu

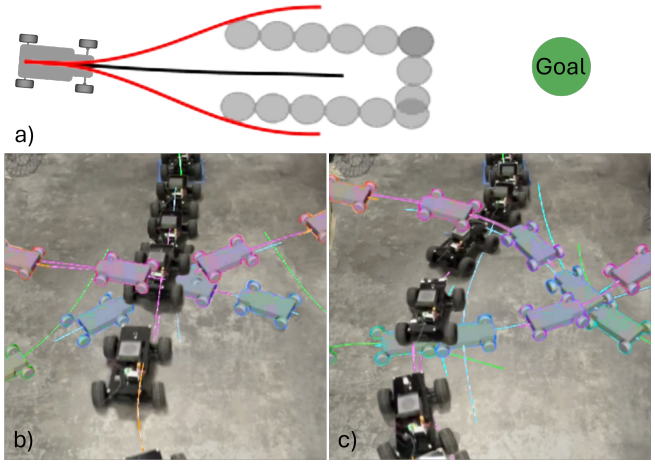

Fig. 1: a) A visualization of a multimodal policy with three modes. b) Timelapse of rally car failing to avoid other virtual robots using one mode. c) Timelapse of rally car successfully avoiding other virtual robots using two modes.

To overcome these limitations, we introduce a multimodal sampling and clustering framework that maintains multiple policy candidates for each robot, thereby increasing diversity and robustness against local minima in both environmental and collaborative planning contexts. Our contributions are:

- A cross-entropy planning approach capable of preserving multiple policy modes for increased planning robustness.
- A multi-robot coordination framework that enables reasoning about sets of candidate policies to avoid local minima and deadlocks more reliably.

II. Related Work

This section reviews recent advances in multimodal policy optimization and multi-robot planning, which jointly motivate the approach proposed in this work.

A. Multi-Modal Policy Optimization

Numerous approaches have been proposed to address the multimodality inherent in trajectory optimization. In reinforcement learning, multi-policy and ensemble strategies are explored, such as maintaining a buffer of diverse candidate policies while pruning redundant, low-performing policies

to improve exploration and diversity [8], or leveraging latent representations to discover richer policy landscapes [9].

Diffusion models have recently gained traction for their ability to generate diverse trajectory distributions and systematically explore the trajectory space, thereby avoiding local minima, particularly in cases of cul-de-sacs [10]. Alternative methods handle multimodal path planning by explicitly sampling both trajectories and cost functions [11].

In settings with uncertain future behaviors of dynamic agents, belief estimation and event likelihoods are exploited to optimize over multiple candidate policies, often by generating samples under differing cost functions that reflect possible environment predictions [12], [13]. Parallel, sample-based policy optimization has also been implemented, where each instance targets a distinct goal or environmental hypothesis [14], [15], [16].

Other random sampling-based approaches attempt to map the cost landscape by identifying and retaining promising modes, or, in some cases, facilitating more rapid collapse toward a single optimal solution [17], [18], [19]. Other techniques, including the application of variational autoencoders, have been explored to capture the manifold of local minima in the planning landscape and to guide sampling in latent spaces [20]. Other approaches sample global plans by partitioning the environment to find all the homotopy classes [21]

B. Real-Time Planning for Multi-Robot Coordination

Over the years, many approaches for multi-robot coordination and control have been proposed [22]. Recently, nonlinear model predictive control (NMPC) has become a popular framework for trajectory optimization and planning in multi-robot systems [23], [24]. Extensions include NMPC combined with conflict-based search to efficiently identify collision-free solutions [25], as well as priority-based schemes where planning times are offset to allow sequential path negotiation [26]. Distributed NMPC approaches have also emerged, either by sampling and sharing predicted states from other robots' planned trajectories [27]; by integrating on-demand collision avoidance mechanisms within the NMPC optimization loop [28]; or by prioritizing collision avoidance based on more urgent predicted collisions [29]. [30] presents a distributed tube-based NMPC approach for aerial vehicle swarms, while [31] combines a stochastic NMPC approach, PAC-NMPC [32], with an objective function inspired by gyroscopic obstacle avoidance to address challenges with deadlock in distributed NMPC.

In this paper, we present an approach for real-time stochastic multimodal policy optimization to improve both single-robot navigation and multi-robot collision avoidance. To our knowledge, this is the first approach to use multimodal stochastic NMPC for coordinated multi-robot maneuvers.

III. Problem Formulation

Here, we present a formulation for the single and multi-robot stochastic optimization problems.

A. Single Robot Planning

Consider a robot whose stochastic dynamics are defined by the probability density function $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$, where $\mathbf{x}_t \in \mathbb{R}^{N_x}$ is a vector of state values and $\mathbf{u}_t \in \mathbb{R}^{N_u}$ is a vector of control inputs. Given a policy $\mathbf{u}_t = \boldsymbol{\pi}(\mathbf{x}_t, \boldsymbol{\xi})$ with policy parameters $\boldsymbol{\xi}$, we can write a trajectory distribution

$$p(\boldsymbol{\tau}|\boldsymbol{\xi}) \;=\; p(\mathbf{x}_0)\prod_{t=0}^{T} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) \tag{1}$$

where $\boldsymbol{\tau}$ is a discrete-time trajectory sequence $\{\mathbf{x}_0, \mathbf{u}_0, \mathbf{x}_1, \mathbf{u}_1 ..., \mathbf{u}_{N_T}, \mathbf{x}_{N_T+1}\}$ and N_T is the number of timesteps. Our goal is to minimize the cost $J(\boldsymbol{\tau}) = \sum_{t=0}^{N_T-1} q(\mathbf{x}_t, \mathbf{u}_t) + q_f(\mathbf{x}_{N_T})$ such that constraint $C(\boldsymbol{\tau}) \leq 0$. Here, $q(\mathbf{x}_t, \mathbf{u}_t)$ is the running cost at time t and $q_f(\mathbf{x}_{N_T})$ is the cost at the final time.

To ensure collision avoidance and adhere to state bounds, we formulate the constraints as

$$g_b(\mathbf{x}_t) = (\mathbf{x}_t - \mathbf{x}_l) < 0 \vee (\mathbf{x}_u - \mathbf{x}_t) < 0 \tag{2}$$

$$g_o(\mathbf{x}_t) = \{\text{dist}(\mathbf{x}_t, \mathbf{p}^{o^m}) - r < 0\} \;\forall\; m \tag{3}$$

$$c(\mathbf{x}_t) = g_b(\mathbf{x}_t) \leq 0 \vee g_o(\mathbf{x}_t) \leq 0 \tag{4}$$

$$C(\boldsymbol{\tau}) = c(\mathbf{x}_0) \vee c(\mathbf{x}_1) \cdots \vee c(\mathbf{x}_{N_T}) \tag{5}$$

where $\mathbf{x}_l$ and $\mathbf{x}_u$ are the lower and upper state bounds respectively, $\mathbf{p}^{o^m}$ is the m^{th} obstacle position, and r is the collision radius. We then seek to enforce the probability of constraint violation such that $\mathbb{E}[C(\boldsymbol{\tau})] \leq P$ where $P \in [0, 1]$.

B. Multi-Robot Planning

For the multi-robot planning problem, we include additional inter-robot constraints. Let $\mathbf{x}_t^i$ be the state of the i^{th} robot at time t. We then define g_a as a constraint to prevent collisions between dynamic agents and given as:

$$g_a(\mathbf{x}_t^i) = \{dist(\mathbf{x}_t^i, \mathbf{x}_t^n) - L < 0\} \;\forall\; n : i \neq n, \tag{6}$$

where $dist()$ is the euclidean distance between the robots, L is the collision radius, and $\mathbf{x}_t^n$ is n^{th} robot on the team. We can then modify equation 4 for the i^{th} robot as follows:

$$c(\mathbf{x}_t^i) = g_b(\mathbf{x}_t^i) \leq 0 \vee g_o(\mathbf{x}_t^i) \leq 0 \vee g_a(\mathbf{x}_t^i) \leq 0. \tag{7}$$

IV. Background

Here we review the Cross-Entropy Method [2], a sample-based stochastic optimization algorithm, and its application to trajectory optimization under complex cost and constraint landscapes [3].

A. Cross-Entropy Optimization

Consider a cost function $J(\mathbf{z})$, where parameter $\mathbf{z}$ is an instance of the random variable $\mathbf{Z}$ described by probability density function $p(\mathbf{z}, \bar{\boldsymbol{\nu}})$ with hyperparameters $\bar{\boldsymbol{\nu}}$. CEM attempts to estimate the rare event probability

$$\ell = \mathbb{P}_{\bar{\boldsymbol{\nu}}}(J(\mathbf{Z}) \leq \gamma) = \mathbb{E}_{\bar{\boldsymbol{\nu}}}[\mathbb{I}_{J(\mathbf{Z})\leq\gamma}], \tag{8}$$

where γ is a small positive constant and $\mathbb{I}$ is the indicator function. Instead of sampling using true hyperparameters $\bar{\boldsymbol{\nu}}$, we can use importance sampling to write

$$\ell = \mathbb{E}_{\boldsymbol{\nu}}\left[\mathbb{I}_{J(\mathbf{Z})\leq\gamma}\frac{p(\mathbf{Z}, \bar{\boldsymbol{\nu}})}{p(\mathbf{Z}, \boldsymbol{\nu})}\right]. \tag{9}$$

The optimal distribution, with hyperparameters $\boldsymbol{\nu}^*$, that minimizes the estimator variance, is given as $p(\mathbf{z}, \boldsymbol{\nu}^*) = \frac{\mathbb{I}_{J(\mathbf{z})\leq\gamma} p(\mathbf{z}, \bar{\boldsymbol{\nu}})}{\ell}$. Since ℓ is unknown, CEM seeks a tractable proposal distribution $p(\mathbf{z}, \boldsymbol{\nu})$, that approximates $p(\mathbf{z}, \boldsymbol{\nu}^*)$ by minimizing the Kullback-Leiber (KL) divergence

$$\boldsymbol{\nu}^* = \arg\min_{\boldsymbol{\nu}} KL(p(\mathbf{z}, \boldsymbol{\nu}^*), p(\mathbf{z}, \boldsymbol{\nu})). \tag{10}$$

By applying the definition of the KL divergence and substituting in for $p(\mathbf{z}, \boldsymbol{\nu}^*)$, we have

$$\boldsymbol{\nu}^* = \arg\max_{\boldsymbol{\nu}} \mathbb{E}_{\bar{\boldsymbol{\nu}}} \left[\mathbb{I}_{J(\mathbf{Z})\leq\gamma} \log p(\mathbf{Z}, \boldsymbol{\nu}) \right], \tag{11}$$

which can be approximated via sampling by

$$\hat{\boldsymbol{\nu}}^* = \arg\max_{v} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{J(\mathbf{Z}_i)\leq\gamma} \log p(\mathbf{Z}_i, \boldsymbol{\nu}). \tag{12}$$

The multi-level CEM algorithm adaptively updates the threshold γ using elite samples as follows:

1) Sample $\mathbf{Z}_1, \ldots, \mathbf{Z}_N \sim p(\mathbf{z}, \hat{\boldsymbol{\nu}}_{j-1})$ and set γ to the ρ-quantile of $J(\mathbf{Z})$.
2) Update parameters such that

$$\hat{\boldsymbol{\nu}}_j = \arg\max_{\boldsymbol{\nu}} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{J(\mathbf{Z}_i)\leq\gamma} \log p(\mathbf{Z}_i, \boldsymbol{\nu}). \tag{13}$$

B. Cross-Entropy Trajectory Optimization

As described in [3], CEM can be used for stochastic trajectory optimization. We present this approach in the context of the problem formulation in Section III.

Consider the stochastic dynamics $p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)$ as defined in Section III. Let us parameterize an open-loop control policy $\mathbf{u}_t = \boldsymbol{\pi}(\boldsymbol{\xi}) = \boldsymbol{\zeta}_t$, where $\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\zeta}_0^T & \boldsymbol{\zeta}_1^T & \ldots & \boldsymbol{\zeta}_{N_T}^T \end{bmatrix}^T$ and $\boldsymbol{\zeta}_t \in \mathbb{R}^{N_u}$. Let us also parameterize a distribution $p(\boldsymbol{\xi}|\boldsymbol{\nu})$ as a multivariate Gaussian over the discrete-time control sequence so that $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For computational tractability, we will also assume a diagonal covariance matrix $\boldsymbol{\Sigma}$ so the distribution parameters are given as $\boldsymbol{\nu} \triangleq \left[\boldsymbol{\mu}^T, diag(\boldsymbol{\Sigma})^T\right]^T$ where $diag(\boldsymbol{\Sigma}) = \begin{bmatrix} \boldsymbol{\eta}_0^T & \boldsymbol{\eta}_1^T & \ldots & \boldsymbol{\eta}_{N_T}^T \end{bmatrix}^T$ and $\boldsymbol{\eta}_t \in \mathbb{R}^{N_u}$. We can then write a joint distribution $p(\boldsymbol{\tau}, \boldsymbol{\xi}|\boldsymbol{\nu}) = p(\boldsymbol{\tau}|\boldsymbol{\xi}) p(\boldsymbol{\xi}|\boldsymbol{\nu})$, where $p(\boldsymbol{\tau}|\boldsymbol{\xi})$ is given by equation 1.

Given a cost function $J(\boldsymbol{\tau})$, where $\boldsymbol{\tau} \sim p(\boldsymbol{\tau}, \boldsymbol{\xi}|\boldsymbol{\nu})$, we can use the CEM approach described in IV-A to optimize the distribution by letting $\boldsymbol{\tau} = \mathbf{z}$. Algorithm 1 outlines the adaptation of CEM for sample-based trajectory optimization.

Algorithm 1 CEM for Trajectory Optimization

```
Initialize distribution parameters ν
while termination condition not met do
    for j = 1 to M do
        Sample control sequence ξ ~ N(ξ|ν)
        Simulate trajectory τ ~ p(τ, ξ|ν)
        Evaluate cost J_j = J(τ)
        Evaluate constraints C_j = C(τ)
    Select elite set: E = top ρ% samples
    Update (ν) using elite E
Output: Best trajectory τ*
```

V. Multimodal Cross-Entropy Planning

In this section, we present a modified CEM algorithm for real-time planning in environments that give rise to multimodal policy distributions.

A. Multimodal Policy Parameterization

To parameterize our multimodal policy, $\boldsymbol{\pi}(\boldsymbol{\xi})$, we choose a Gaussian Mixture Model (GMM) such that

$$p(\boldsymbol{\xi}|\boldsymbol{\nu}) = \sum_{k=1}^{K} \phi_k \mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{14}$$

where $\boldsymbol{\nu} \triangleq \left[\boldsymbol{\mu}_1^T, diag(\boldsymbol{\Sigma}_1)^T, \ldots \boldsymbol{\mu}_K^T, diag(\boldsymbol{\Sigma}_K)^T\right]^T$, ϕ_k are scalar weights, and K is the number of modes. Similar to Section IV-B, each mode is represented by a $N_u N_T$-dimensional multivariate Gaussian, where $\boldsymbol{\mu}_k \in \mathbb{R}^{N_u N_T}$ and $diag(\boldsymbol{\Sigma}_k) \in \mathbb{R}^{N_u N_T}$.

Given $p(\boldsymbol{\tau}|\boldsymbol{\xi})$, sampling trajectory sequences $\boldsymbol{\tau}$ using this multimodal policy distribution is then straightforward, since $p(\boldsymbol{\tau}, \boldsymbol{\xi}|\boldsymbol{\nu}) = p(\boldsymbol{\tau}|\boldsymbol{\xi}) p(\boldsymbol{\xi}|\boldsymbol{\nu})$. However, if we wish to preserve multimodal policies during planning, Algorithm 1 often cannot be directly applied.

B. Challenges with Multimodal CEM Planning

One challenge associated with planning multimodal policies using CEM trajectory optimization is that the standard CEM algorithm can lead to mode collapse, even in a single step of the algorithm, when two feasible policies result in significantly different costs. Consider the case where D represents a finite set of feasible trajectory samples, and where $D = D1 \cup D2$ and $D1 \cap D2 = \emptyset$. Let $D1$ be contained in mode $M1$, such that $D1 \subset M1$. Let $D2$ be contained in $M2$, such that $D2 \subset M2$. We also assume $M1 \cap M2 = \emptyset$. Now, let the set of costs for $D1$ be denoted as $J(D1)$ and the set of costs for $D2$ be denoted as $J(D2)$. If, for a given environment, $\sup J(D1) < \inf J(D2)$ and $|D1| > |S_{\mathcal{E}}|$, where $S_{\mathcal{E}}$ is the set of elite samples, it follows from Algorithm 1, that $S_{\mathcal{E}} \subset D1$ and the resulting policy will collapse into $M1$.

For this reason, we modify the CEM algorithm to preserve higher-cost modes, as these modes may become lower-cost modes under finite-horizon planning as the horizon recedes.

C. Trajectory Clustering and Feasibility Sampling

Consider the CEM parameter update equation 13, now modified for the CEM trajectory optimization:

$$\hat{\boldsymbol{\nu}}_j = \arg\max_{\boldsymbol{\nu}} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}_{J(\boldsymbol{\tau}_i)\leq\gamma} \log p(\boldsymbol{\tau}_i|\boldsymbol{\nu}). \tag{15}$$

While closed-form solutions exist for the maximum-likelihood estimation of unimodal normal distribution parameters, such solutions do not exist for a GMM, and heuristic solutions have been proposed [33]. Therefore, to preserve the multimodal nature of our policy distribution, we propose sampling from the set of feasible (constraint-free) trajectories and applying equation 15 to distinct modes.

1) Feasibility Sampling: For our receding-horizon navigation and obstacle avoidance tasks, we are primarily concerned with multimodal policies that yield constraint-violation-free trajectories. To this end, we sample from our policy and only preserve constraint-violation-free samples. In the case where no constraint-free trajectories exist, the constraints are relaxed, and the total number of violations is incorporated as an additive penalty to the cost. All trajectories are then used for clustering.

2) Trajectory Clustering: We identify distinct solution modes in these feasible samples via the K-means clustering algorithm [34]. K-means is an unsupervised learning method that partitions data into K clusters, where K is a tunable hyperparameter, by iteratively assigning each sample to its nearest centroid (chosen randomly from the data at initialization), then updating centroids to be the mean of each cluster until convergence. In particular, we cluster trajectories based on state sequences, which promotes the discovery of distinct homotopy classes or task-relevant policy modes. Let $\boldsymbol{\chi}_n = \begin{bmatrix}\mathbf{x}_{n,0}^T & \mathbf{x}_{n,1}^T & \dots & \mathbf{x}_{n,N_T+1}^T\end{bmatrix}^T$, where $\mathbf{x}_{n,k}$ represents the k^{th} time sample of the state n^{th} feasible trajectory sample. K-means clustering seeks K cluster centroids $\{\mathbf{y}_1, \dots, \mathbf{y}_K\}$ by iteratively performing $z_n = \arg\min_k \|\boldsymbol{\chi}_n - \mathbf{y}_k\|_2^2 \quad \forall n$ and $\mathbf{y}_k = \frac{1}{|S_k|}\sum_{n\in S_k}\boldsymbol{\chi}_n$ where $S_k = \{n : z_n = k\}$ is the set of trajectories assigned to cluster k, and $\|\cdot\|_2$ denotes the Euclidean norm.

3) Multimodal Policy Update: Given the policy modes, we apply the CEM update step to each cluster. Within each mode, we select the top ρ-quantile elite samples by cost and use Maximum Likelihood Estimation to fit new Gaussian parameters to their control policies. Equation 15 becomes

$$\hat{\boldsymbol{\nu}}_j = \arg\max_{\boldsymbol{\nu}} \frac{1}{|S_k|}\sum_{i\in S_k} \mathbb{I}_{J_k(\boldsymbol{\tau}_i)\le\gamma} \log p(\boldsymbol{\tau}_i|\boldsymbol{\nu}). \tag{16}$$

The complete algorithm for Multimodal CEM Trajectory Optimization can be seen in Algorithm 2.

Algorithm 2 Multimodal Cross Entropy

Inputs: $\hat{\nu}_0$
while termination conditions not met **do**
  for $j = 1$ to M **do**
    Sample control sequence $\boldsymbol{\xi} \sim p(\boldsymbol{\xi}|\boldsymbol{\nu})$
    Simulate trajectory $\boldsymbol{\tau} \sim p(\boldsymbol{\tau}|\boldsymbol{\xi})$,
    Evaluate cost $J_j = J(\boldsymbol{\tau})$
    Evaluate constraints $C_j = C(\boldsymbol{\tau})$
  Filter feasible set $\mathcal{T}_{\text{free}} = \{\boldsymbol{\tau}_j : C_j = 0\}$
  Cluster feasible trajectories using K-means$(\mathcal{T}_{\text{free}}, K)$
  for each cluster k **do**
    Select elite set: $\mathcal{E}_k = \text{Top}_{\rho\%}(\mathcal{S}_k)$
    Update $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ from $\mathcal{E}_k$

D. Warm Starts for Secondary Modes

To accelerate convergence and improve sample efficiency in iterative replanning, we employ warm-start initialization strategies for both primary and secondary policy modes. For the primary mode, we warm start the new policy by shifting the prior policy forward by the number of executed steps. For each secondary mode, naively shifting controls can result in poor initialization, since the robot is not following those control inputs. To preserve these modes, a time-varying linear quadratic regulator (TVLQR) policy on the prior nominal trajectory is used to calculate gains $\boldsymbol{\kappa}_t$ for each time step. Given the current state $\mathbf{x}_0$, and mode k, the warm-started control at step t is $\mathbf{u}_t^* = \mathbf{u}_t^k + \boldsymbol{\kappa}_t^k(\mathbf{x}_t - \mathbf{x}_t^k)$.

E. Receding-Horizon Planning

Given the multimodal CEM trajectory optimization approach described above, we can now construct a receding-horizon planning algorithm, as described in Algorithm 3.

Algorithm 3 Receding-Horizon Multimodal CEM Planning

Input: $\boldsymbol{\nu}_0^*$
$\mathbf{x}_0 \leftarrow$ GetCurrentState()
while objective not completed **do**
  $\hat{\boldsymbol{\nu}}^* \leftarrow$ Optimize$(\boldsymbol{\nu}_0^*, \mathbf{x}_0)$ ▷ See Alg. 2
  $\mathbf{u}^d \leftarrow$ MaximumLikelihoodEstimate$(\boldsymbol{\nu}^*)$
  for $t = 0$ to N_T **do**
    $\mathbf{x}_{t+1}^d = \mathbf{x}_t^d + f(\mathbf{x}_t^d, \mathbf{u}_t^d)\Delta t$
  $\boldsymbol{\tau}^d \leftarrow \{\mathbf{x}_0^d, \mathbf{u}_0^d, \mathbf{x}_1^d, \mathbf{u}_1^d, \dots, \mathbf{u}_{N_T}^d, \mathbf{x}_{N_T+1}^d\}$
  Execute$(\boldsymbol{\tau}^d)$
  $\mathbf{x}_0 \leftarrow$ GetCurrentState()
  $\boldsymbol{\nu}_0^* \leftarrow$ InitializePrior$(\boldsymbol{\nu}^*, \boldsymbol{\tau}^d, \mathbf{x}_0)$
return

VI. Multi-robot Cross-Entropy Planning

While planning multimodal policies can be particularly beneficial for navigating through complex static obstacle fields, it is also relevant for for multi-robot coordination. Here, we aim to leverage multimodal policies to achieve computationally tractable multi-robot planning by decoupling discrete high-level mode coordination from continuous low-level policy optimization.

A. Multimodal Policy Sharing

For multi-robot planning, we assume communication among robot team members. At the end of each receding-horizon planning cycle, all robots synchronously share their current state and their multimodal policy distributions. In this way, a robot can generate predicted trajectories for its team members and evaluate inter-robot constraints during the next cycle of trajectory optimization. While this trajectory-sharing approach does not ensure joint synchronous optimization of robot control policies, it does reduce the computational requirements compared to a centralized policy optimization approach.

B. Inter-robot Collision Constraints

To estimate the probability of collision between robots, we check for collisions between a robot's planned trajectories and the predicted trajectories of its neighbors, across all modes. Given the i^{th} sampled trajectory from robot j, $\boldsymbol{\tau}^i$, let $\mathbf{x}_t^{n,m,k}$ be the state of the n^{th} robot team member's m^{th}

trajectory from the k^{th} mode at time t, where $j \neq n$. We can estimate the probability of collision of trajectory $\boldsymbol{\tau}^i$ as

$$g_A^{n,k}(\boldsymbol{\tau}^i) = \frac{1}{M} \sum_{m=1}^{M} g_A^{n,m,k}(\boldsymbol{\tau}^i), \tag{17}$$

where

$$g_a^{n,m,k}(\mathbf{x}_t^i) = \mathbb{I}\Big(\text{dist}\big(\mathbf{x}_t^i, \mathbf{x}_t^{n,m,k}\big) < L\Big), \tag{18}$$

$$g_A^{n,m,k}(\boldsymbol{\tau}^i) = \bigvee_{t=0}^{N_T} g_a^{n,m,k}(\mathbf{x}_t^i), \tag{19}$$

and $\bigvee$ is logical OR. We can then define a chance constraint as

$$C_A^{n,k}(\boldsymbol{\tau}^i) = \mathbb{I}\left(g_A^{n,k}(\boldsymbol{\tau}^i) \geq P\right) \tag{20}$$

where probability $P \in (0, 1)$. A candidate trajectory $\boldsymbol{\tau}^i$ is considered unsafe with respect to robot n only if it violates the chance constraint in all modes of that robot, such that

$$C_A^n(\boldsymbol{\tau}^i) = \bigwedge_{k=1}^{K} C_A^{n,k}(\boldsymbol{\tau}^i), \tag{21}$$

where $\bigwedge$ denotes logical AND.

The overall multi-robot collision constraint is then

$$C_A(\boldsymbol{\tau}^i) = \bigvee_{n \neq j} C_A^n(\boldsymbol{\tau}^i). \tag{22}$$

Finally, the total constraint for $\boldsymbol{\tau}^i$ is updated as

$$C(\boldsymbol{\tau}^i) = c(\mathbf{x}_0^i) \vee c(\mathbf{x}_1^i) \vee \cdots \vee c(\mathbf{x}_{N_T}^i) \vee C_A(\boldsymbol{\tau}^i). \tag{23}$$

C. Multimodal Policy Coordination

Once each robot $i \in \{1, \cdots, N\}$ has optimized its candidate policies using the shared information from the *prior* planning phase, a final coordination phase is conducted using the *current* multimodal policies of all robots to maximize joint feasibility with respect to inter-robot collisions.

Since individual robot policies are optimized using robot team information from the last planning phase, a robot cannot greedily select among its individual receding-horizon policies and ensure the chosen nominal trajectories will avoid inter-robot collisions. Therefore, at the conclusion of the current planning cycle, a centralized coordination algorithm finds the optimal set of policy modes.

Let $c = \{k_1, \cdots, k_N\}$ denote a selected set of policies, one for each robot, such that the nominal trajectory for robot j is given by $\boldsymbol{\tau}^j$. For each selection, we want to minimize the global cost while avoiding collisions:

$$c^* = \arg\min_c \sum_{j=1}^{N} J(\boldsymbol{\tau}^j) \tag{24}$$

$$\text{s.t.} \quad g_a(\mathbf{x}_t^j) > 0; \ \forall\, t, j. \tag{25}$$

In the event that no candidate selection c is feasible, we select c^* to minimize the total number of constraint violations

$$c^* = \arg\min_c \sum_t \mathbb{I}(g_a(\mathbf{x}_t^j) \leq 0) \tag{26}$$

and break ties by using total cost $J_{\text{total}} = \sum_{i=1}^{N} J(\boldsymbol{\tau}^i)$. Here, we solve this via exhaustive search, but note that more computationally tractable approaches may be needed in larger settings, as the search space grows exponentially with the number of agents and polynomial with the number of modes.

Algorithm 4 Coordination of Joint Trajectory Set

Inputs: $\{\mathcal{P}_i\}$: candidate policy sets for all robots
$\mathcal{C} \leftarrow$ Generate all possible combinations of policies
Initialize best_cost $\leftarrow \infty$; best_combo $\leftarrow$ None
for each joint combination $\mathbf{c} \in \mathcal{C}$ **do**
    total_cost $\leftarrow 0$; total_violations $\leftarrow 0$
    for each robot i in $\mathbf{c}$ **do**
        total_cost += $J(\tau_i)$
    for all (i, j) robot pairs **do**
        for each timestep t **do**
            Compute inter-robot distance $d_{ij}(t)$
            if $d_{ij}(t) < r_i + r_j$ **then**
                total_violations += 1
    cost_with_penalty $\leftarrow$ total_cost + $\lambda \cdot$ total_violations
    if cost_with_penalty $<$ best_cost **then**
        best_cost $\leftarrow$ cost_with_penalty
        best_combo $\leftarrow \mathbf{c}$
Return: best_combo

VII. Simulation Experiments

We conducted extensive Monte Carlo simulations to evaluate the effectiveness of our proposed multimodal cross-entropy planner in two scenarios: navigation in trap environments with many local minima and distributed multi-robot collision avoidance.

A. Dynamics Model

Each agent is modeled as a stochastic bicycle with acceleration and steering rate inputs. The agent's state vector is defined as $\mathbf{x}_t = [p_x, p_y, \theta, v, \delta_s]^T$, comprising position, heading, velocity, and steering angle. The control vector is $\mathbf{u}_t = [\dot{v}, \dot{\delta}_s]^T$, with wheelbase $l = 0.33\,\text{m}$. Robot dynamics are given by $\mathbf{x}_{t+1} \sim p(\cdot|\mathbf{x}_t, \mathbf{u}_t) = \mathbf{x}_t + [f(\mathbf{x}_t, \mathbf{u}_t) + \boldsymbol{\omega}]\Delta t$ where $f(\mathbf{x}_t, \mathbf{u}_t) = [v\cos(\theta),\ v\sin(\theta),\ v\tan(\delta_s)/l,\ \dot{v},\ \dot{\delta}_s]^T$, $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma})$, and $\boldsymbol{\Gamma} = \text{diag}([0.001,\ 0.001,\ 0.012,\ 0.1,\ 0.006])$. Here $\boldsymbol{\omega}$ is Gaussian noise and $\boldsymbol{\Gamma}$ is an estimation of the process covariance. The acceleration is limited to $\dot{v} \in [-1, 1]\ \text{m/s}^2$, the steering rate is limited to $\dot{\delta}_s \in [-1, 1]\ \text{rad/s}$, the velocity is limited to $v \in [-0.5, 2]\ \text{m/s}$, and the steering angle is limited to $\delta_s \in [-0.4, 0.4]$ radians.

The optimization and environment hyperparameters are summarized in Table I.

B. Trap Environment

To evaluate planner robustness against local minima, we conducted a total of 2,700 "trap environment" trials. These trials spanned 54 distinct U-shaped obstacle configurations, systematically varying trap width and depth. For each obstacle type, we ran 50 randomized trials with unique initial and goal state pairs. In every trial, initial and goal positions were randomly sampled within the workspace bounds, $(p_x, p_y) \in$

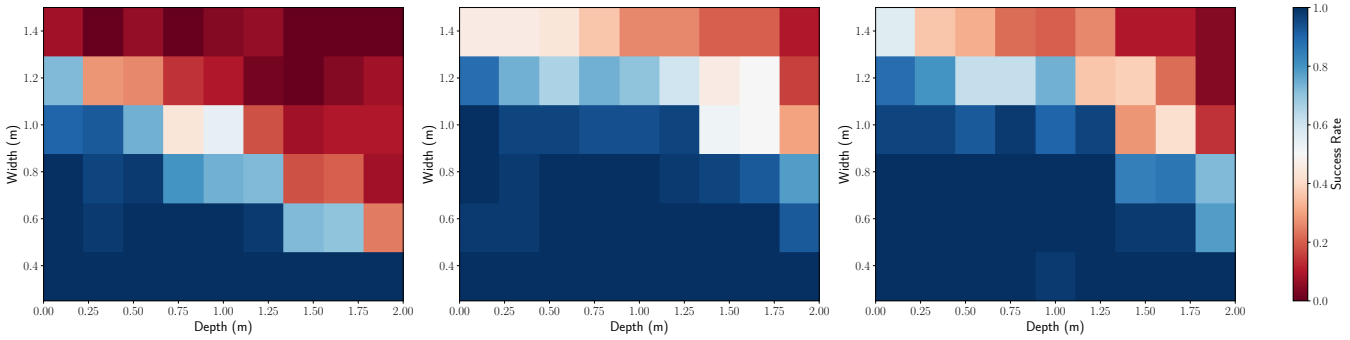

Fig. 2: Success rate heatmaps across different trap parameters. From left to right: single-mode CE, two-mode CE, and two-mode CE without TVLQR warm-starts. Success is defined as reaching the goal within 10 seconds without colliding with obstacles. The two-mode CE approach outperformed the single-mode CE in environments with moderately deep and wide traps. Furthermore, incorporating TVLQR-based warm starts for the secondary modes led to additional performance gains in the deepest and widest trap configurations.

TABLE I: Planner and Cost Function Parameters

Parameter	**Value**
Number of samples (M)	1024
Elite fraction ($\rho\%$)	10%
Planning horizon (T)	40 time steps (2 seconds)
Time step (Δt)	0.05 s
Stage cost weight (Q)	diag([1, 1, 0, 0, 0])
Terminal cost weight (Q_f)	diag([40, 40, 0, 0, 0])
Control input cost (R)	diag([0.1, 0.1])
TVLQR Weights	
Stage state cost (Q)	diag([10, 10, 1, 1, 1])
Control input cost (R)	diag([1, 1])
Terminal state cost (Q_f)	diag([100, 100, 10, 10, 10])

$[(-1,-6),(11,6)]$ meters, with resampling to enforce a minimum Euclidean distance of 7 meters between them. The agent's initial heading was oriented toward the goal, while its initial velocity and steering angle were sampled uniformly from $v \in [-0.5, 2.0]\,\mathrm{m/s}$ and $\delta_s \in [-0.1, 0.1]\,\mathrm{radians}$, respectively.

Each trap environment contained a single U-shaped obstacle, created by overlapping circles of radius 0.25 meters and positioned two-thirds of the distance from the start to the goal configurations, with the cavity facing the initial robot position. Throughout our study, trap widths and depths were varied in increments of 0.25 meters, spanning 0.25 to 1.5 meters for width and 0.0 to 2.0 meters for depth, resulting in six discrete widths, nine discrete depths, and a total of 54 distinct trap geometries for evaluation. Success was defined as the agent reaching the goal within 10 seconds without colliding with the trap obstacle.

As shown in Figure 2, the two-mode planner outperformed the one-mode planner in avoiding deeper and wider traps. The two-mode planner also performed similarly regardless of whether TVLQR warm-starts were used in thin and shallow traps. However, TVLQR warm-starts provided a distinct advantage when navigating deeper and wider traps.

To further compare single-mode and multimodal policies, we generated a diverse set of trap environments. In each, the agent's start was uniformly sampled from $[(-1,-6),(0,6)]$ meters and the goal from $[(10,-6),(11,6)]$ meters, with heading set toward the goal and zero initial velocity/steering. Twelve U-shaped obstacles (composed of 0.25 meter-radius circles) were randomly placed with their cavities facing

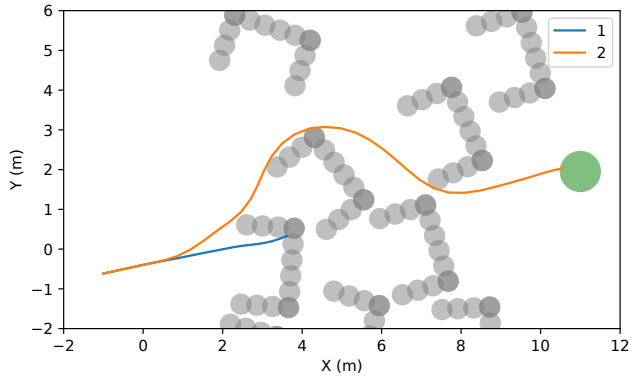

Fig. 3: In the trap environment, single-mode cross-entropy optimization fails due to local minima, but employing a two-mode policy successfully navigates to the goal.

the start, each oriented within $\pm\pi/8$ radians. Out of 650 generated environments, 342 with feasible solutions were retained. For each environment, six trials were conducted: the robot generated 1024 samples for optimizing $K = 1$ and $K = 2$ modes, and 2048 samples for optimizing $K = 1$, 2, 3, and 4 modes. Figure 3 illustrates that multiple modes can help avoid local minima where single-mode policies get trapped. As shown in Table II, performance improved as K increased up to three modes, but declined with four modes—likely because samples became too thinly distributed across clusters, reducing effective exploration.

TABLE II: Performance Across Different Numbers of Modes

# Samples	1 Mode	2 Modes	3 Modes	4 Modes
1024	35.7%	73.4%	X	X
2048	52.9%	75.4%	78.7%	74.7%

C. Multi-Agent Collision Avoidance Trials

Inter-agent collision experiments were conducted for sets of two to eight agents, with 50 trials each for each set. Agents were initialized at equal intervals along the circumference of a circle with a radius of 3 meters, each oriented radially inward. Their goals were positioned at antipodal points on the circle. This scenario tested the system's robustness in identifying distinct modes and employing policy coordination to select a collision-free trajectory set as the difficulty

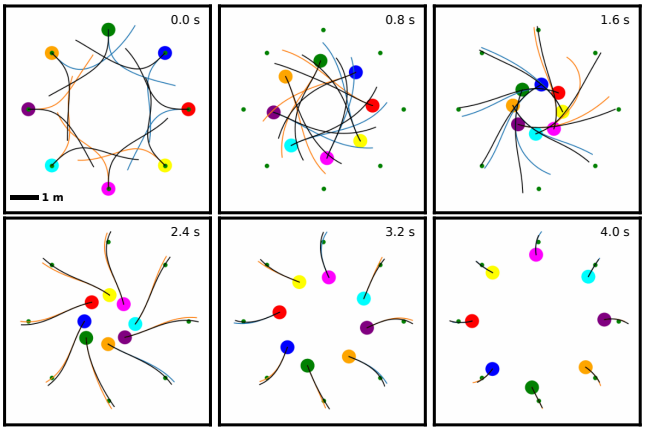

Fig. 4: Trajectories of eight robots in a 3-meter radius antipodal simulation, optimizing a two-mode policy. Blue and orange denote the order in which candidate policies were provided to the centralized controller; black indicates the final policy selected for execution.

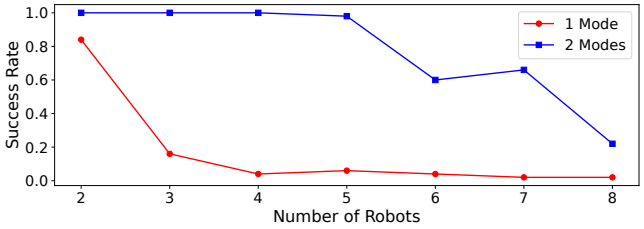

Fig. 5: Results from multi-robot antipodal experiments, where robots were initialized uniformly on a circle of radius three meters. The two-mode case achieved perfect performance with four agents and also had a higher success rate than the single-mode case for all numbers of robots.

increased with the addition of more agents. An example with eight agents in a two-modes trial is shown in Figure 4. As shown in Figure 5, the single-mode approach experiences a sharp decline in the number of collision-free paths found as the number of agents increases. In contrast, the two-mode approach maintains a high success rate up to five agents before its performance begins to drop. Notably, even with eight agents, the two-mode approach outperforms the single-mode case with only three agents. Beyond five agents, we observed that K-means started to struggle to find distinct modes.

VIII. Hardware Experiments

To validate our approach in real-world conditions, we deployed the multimodal CE planner on a physical robotic platform. We used a $1/10^{\text{th}}$ scale Traxxas Maxx car whose dynamics were modeled as a stochastic bicycle system with a wheelbase of $l = 0.354$ meters. The velocity was constrained to $v \in [-0.5, 1.0]$ m/s, and the steering angle limited to $\delta_s \in [-0.3, 0.3]$ radians.

State estimates were obtained via motion capture, with control commands generated from planned trajectories and executed in real-time using TVLQR. We validated our approach in both a single-obstacle trap environment and a three-agent antipodal navigation scenario with two virtual agents. Planning and policy computations were performed online at 5 Hz on separate computers with an Nvidia 4070

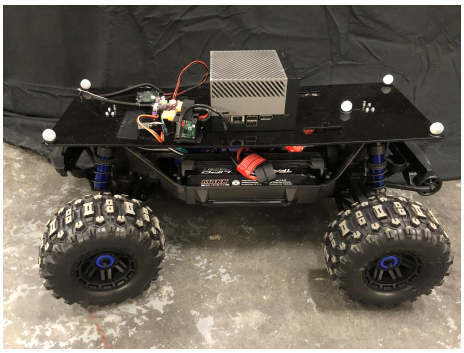

Fig. 6: Photo of $1/10^{\text{th}}$ scale Traxxas Maxx car

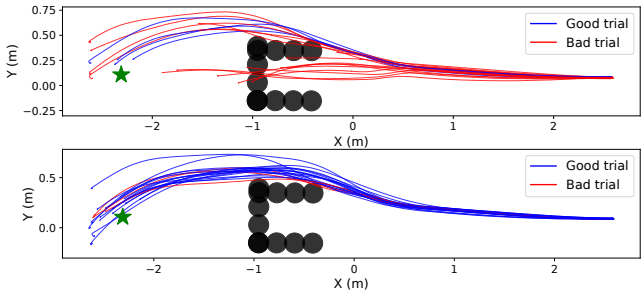

Fig. 7: Trap Environment Hardware Trials, (Top) 1 mode: 3 trials made it to the goal collision-free as seen in blue, and nine trials got caught in the local minima. (Bottom) 2 modes: 17 trials made it to the goal collision-free and all trials avoided the local minima.

GPU, while a Jetson Orin AGX on board the vehicle executed the TVLQR controller. A photo of the car is shown in Figure 6.

We tested the hardware using a virtual trap environment with a width of 0.5 meters and a depth of 0.6 meters. Figure 7 shows the trajectories from the single mode experiment, where only 20% of the trials made it to the goal while being collision-free. We can see that nine trajectories become trapped in the local minima, whereas in the two-mode experiment, no trajectories are caught in the trap, and they succeed with an 85% success rate. We further evaluated the antipodal scenario by testing a real robot in conjunction with two virtual robots on a circle with a radius of 1.7 meters, keeping all other parameters constant. In this setting, the two-mode approach achieved collision-free trajectories in eight out of ten trials, whereas the single-mode approach succeeded in only one trial. Table III shows the performance of the hardware trials. The reported average and median minimum distances reflect the closest approach between agents across all trials.

TABLE III: Antipodal Hardware Performance

Number of Modes	Success Rate	Average Minimum Distance (m)	Median Minimum Distance (m)
1	10%	0.184	0.136
2	80%	0.474	0.504

IX. Discussion

In this paper, we introduced a distributed receding-horizon stochastic planning framework capable of navigating agents

through trap environments and resolving multi-agent trajectory conflicts. By applying K-means clustering to sampled trajectories, we identified up to four distinct policies that enabled agents to escape local minima. We further demonstrated that TVLQR can be utilized to warm-start secondary policies, thereby preserving policy diversity and enhancing optimization efficiency. Our multi-modal trajectory optimization approach enabled robust deconfliction of trajectories among multiple agents, demonstrating high success rates and real-time performance on hardware. Future work includes developing improved mode identification and selection strategies, applying to more complex dynamics, as well as extending the framework to incorporate sensor fusion and planning under environmental uncertainty.

ACKNOWLEDGMENT

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-25-2-0033. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1433–1440.

[2] Z. Botev and D. Kroese, "Global likelihood optimization via the cross-entropy method with an application to mixture models," in *Proceedings of the 2004 Winter Simulation Conference, 2004.*, vol. 1, 2004, p. 535.

[3] M. Kobilarov, "Cross-entropy motion planning," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 855–871, 2012.

[4] S. Kumar and S. Chakravorty, "Multi-agent generalized probabilistic roadmaps: Magprm," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3747–3753.

[5] Y. Wu, B. Jiang, and H. Xu, "Formation control strategy of multi-agent system with improved probabilistic roadmap method applied in restricted environment," in *2021 6th International Conference on Computational Intelligence and Applications (ICCIA)*, 2021.

[6] B. R. Hikmet Beyoglu, Stephan Weiss, "Multi-agent path planning and trajectory generation for confined environments," *International Conference on Unmanned Aircraft Systems*, 2022.

[7] X. Liu, S. S. Ge, and C.-H. Goh, "Formation potential field for trajectory tracking control of multi-agents in constrained space," *International Journal of Control*, vol. 90, no. 10, pp. 2137–2151, 2017.

[8] L. Pan, Q. Cai, and L. Huang, "Exploration in policy optimization through multiple paths," *Autonomous Agents and Multi-Agent Systems*, vol. 35, no. 2, Jun 2021.

[9] Z. Huang, L. Liang, Z. Ling, X. Li, C. Gan, and H. Su, "Reparameterized policy learning for multimodal trajectory optimization," 2023. [Online]. Available: https://arxiv.org/abs/2307.10710

[10] C. Knuth, C. Dimmig, and B. Bittner, "Generative planning with fast collision checks for high speed navigation," 2024. [Online]. Available: https://arxiv.org/abs/2405.04498

[11] J. Carvalho, A. Le, P. Kicki, D. Koert, and J. Peters, "Motion planning diffusion: Learning and adapting robot motion planning with diffusion models," 2025. [Online]. Available: https://arxiv.org/abs/2412.19948

[12] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson, "Mpdm: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1670–1677.

[13] B. Zhou, W. Schwarting, D. Rus, and J. Alonso-Mora, "Joint multi-policy behavior estimation and receding-horizon trajectory planning for automated urban driving," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2388–2394.

[14] Y. Zhang, C. Pezzato, E. Trevisan, C. Salmi, C. H. Corbato, and J. Alonso-Mora, "Multi-modal mppi and active inference for reactive task and motion planning," *IEEE Robotics and Automation Letters*, vol. 9, no. 9, p. 7461–7468, Sep. 2024.

[15] L. Zhou, Z. Li, Y. Li, and S. Bai, "Parallel mppi with gradient-velocity modulated sdf cost for high-performance real-time dynamic obstacle avoidance by robot manipulators," *IEEE Transactions on Robotics*, 2025.

[16] E. Trevisan and J. Alonso-Mora, "Biased-mppi: Informing sampling-based model predictive control by fusing ancillary controllers," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5871–5878, 2024.

[17] T. Osa, "Multimodal trajectory optimization for motion planning," *The International Journal of Robotics Research*, vol. 39, no. 8, p. 983–1001, Jun. 2020.

[18] K. Honda, N. Akai, K. Suzuki, M. Aoki, H. Hosogaya, H. Okuda, and T. Suzuki, "Stein variational guided model predictive path integral control: Proposal and experiments with fast maneuvering vehicles," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 7020–7026.

[19] M. Okada and T. Taniguchi, "Variational inference mpc for bayesian model-based reinforcement learning," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 258–272.

[20] T. Osa, "Motion planning by learning the solution manifold in trajectory optimization," *The International Journal of Robotics Research*, vol. 41, no. 3, pp. 281–311, 2022.

[21] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robotics and Autonomous Systems*, vol. 88, pp. 142–153, 2017.

[22] J. K. Verma and V. Ranga, "Multi-robot coordination analysis, taxonomy, challenges and future scope," *Journal of intelligent & robotic systems*, vol. 102, no. 1, p. 10, 2021.

[23] V. Roldão, R. Cunha, D. Cabecinhas, C. Silvestre, and P. Oliveira, "A leader-following trajectory generator with application to quadrotor formation flight," *Robotics and Autonomous Systems*, 2014.

[24] H. Xiao, Z. Li, and C. L. P. Chen, "Formation control of leader–follower mobile robots' systems using model predictive control based on neural-dynamic optimization," *IEEE Transactions on Industrial Electronics*, 2016.

[25] A. Tajbakhsh, L. T. Biegler, and A. M. Johnson, "Conflict-based model predictive control for scalable multi-robot motion planning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 562–14 568.

[26] J. Tordesillas and J. P. How, "MADER: trajectory planner in multi-agent and dynamic environments," *CoRR*, vol. abs/2010.11061, 2020. [Online]. Available: https://arxiv.org/abs/2010.11061

[27] S. Satir, Y. F. Aktaş, S. Atasoy, M. M. Ankaralı, and E. Sahin, "Distributed model predictive formation control of robots with sampled trajectory sharing in cluttered environments," *IEEE/RJS International Conference on Intelligent RObots and Systems*, 2023.

[28] C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Robotics and Automation Letters*, 2020.

[29] B. Lindqvist, P. Sopasakis, and G. Nikolakopoulos, "A scalable distributed collision avoidance scheme for multi-agent uav systems," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 9212–9218.

[30] V. Madabushi, Y. Kopel, A. Polevoy, and J. Moore, "Dense fixed-wing swarming using receding-horizon nmpc," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[31] M. Gonzales, A. Polevoy, M. Kobilarov, and J. Moore, "Multi-agent feedback motion planning using probably approximately correct nonlinear model predictive control," in *2025 IEEE 21st International Conference on Automation Science and Engineering(CASE)*, 2025.

[32] A. Polevoy, M. Kobilarov, and J. Moore, "Probably approximately correct nonlinear model predictive control (pac-nmpc)," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7226–7233, 2023.

[33] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[34] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1. University of California Press, 1967, pp. 281–297.