Robot Path Planning via Flow Matching with Safety and Adaptivity through Predictive Control

Philipp Holzmann[1], Maik Pfefferkorn[1], João Carvalho[2], Ali Younes[3], An T. Le[2], Georgia Chalvatzaki[3,5], Jan Peters[2,4,5], Rolf Findeisen[1]

***Abstract*— Learning-based path planners based on diffusion and flow matching can generate diverse trajectories from demonstrations but classically lack guarantees on safety and constraint satisfaction during deployment. We propose a framework that integrates flow-matching-based path planning, trained on demonstrations, with model predictive path-following control to combine data-driven path diversity with real-time safe execution. The flow-matching model efficiently captures multimodal path distributions, while predictive controller adapts the motion online, ensuring satisfaction of state/input constraints and obstacle avoidance. We introduce an event-triggered re-planning scheme that biases new path generation using solutions from the predictive controller, enabling safety even in environments with previously unseen obstacles.**

I. Introduction

Recently generative models based on diffusion [1] and flow matching [2], [3] have been explored for robot path planning and control, as they excel at capturing multimodal distributions in imitation learning [4], [5] and motion planning [6], [7]. These models generate diverse motions from demonstrations but classically lack safety guarantees during deployment. For instance, a learned policy executed in open loop may fail when facing unseen obstacles or dynamic constraints. In this work, we adopt the shortcut model [8], a flow-matching variant known for fast sampling [9], and propose a framework to enforce safety during execution.

To address the lack of guarantees, we integrate the generative planner with a model predictive path-following controller (MPPFC) [10], [11], [12]. Unlike standard tracking MPC [13], [14], MPPFC treats path progression and velocity as optimization variables. This allows the controller to adapt the motion profile online—slowing down or deviating locally—to ensure constraint satisfaction and collision avoidance while maintaining the geometric structure of the expert path.

Combining generative models with predictive control is an emerging area of research. Recent works have utilized diffusion to warm-start optimizations [15], parameterize MPC feedback [16], or guide sampling-based controllers like MPPIC [17], [18], [19]. However, these approaches often focus on simplified or unimodal settings. Our method differs by employing MPPFC not just as a predictive safety filter [20], [21], but as the primary controller that actively shapes path execution. This combination enables the safe, adaptive deployment of data-driven plans in complex, cluttered environments.

II. Problem Formulation

We consider a dynamical system:

$$\dot{x}(t) = f(x(t), u(t)), \quad y(t) = h(x(t))$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, and $y \in \mathbb{R}^{n_y}$ represent state, input, and output respectively. The objective is to navigate from a start to a goal state while avoiding obstacles, mimicking expert behavior observed in a dataset of trajectories $\mathcal{D}_\tau$.

III. From Expert Demonstrations to Path Data

Given N expert trajectories $\boldsymbol{\tau}^{(i)} = \{(t_j, y(t_j))\}_{j=0}^{n}$, we aim to model behavior as geometric curves rather than time-dependent trajectories. This decouples the model from velocity profiles and trajectory lengths. Using a B-spline parametrization, we convert the raw trajectory data into a dataset of coefficients $\mathcal{D}_\xi = \{\boldsymbol{\xi}^{(i)}\}_{i=1}^{N}$ by solving the linear least squares problem:

$$\boldsymbol{\xi}^{(i)} = \arg\min_{\boldsymbol{\xi}} \sum_{j=0}^{n_i} |y(t_j) - p(\theta_j)|^2, \qquad (1)$$

where θ_j represents normalized time. The solution is computed in closed-form [22].

A. Modeling and Learning Expert Behaviors

To capture the multimodal distribution of expert paths, we train a shortcut model [8]—a fast-sampling variant of flow matching [2]—on $\mathcal{D}_\xi$. The model learns a flow ϕ that maps an initial noise distribution q_0 to the target data distribution q_1 via the differential equation:

$$\frac{\mathrm{d}\phi(\lambda, \boldsymbol{\xi}_0)}{\mathrm{d}\lambda} = v(\lambda, \phi(\lambda, \boldsymbol{\xi}_0)), \quad \phi(0, \boldsymbol{\xi}_0) = \boldsymbol{\xi}_0 \sim q_0, \qquad (2)$$

where v is a learned velocity field conditioned on the start and goal positions. Assuming a linear interpolation path $\phi(\lambda, \boldsymbol{\xi}_0) = (1-\lambda)\boldsymbol{\xi}_0 + \lambda\boldsymbol{\xi}_1$, the velocity field is learned by regressing to the target direction $\boldsymbol{\xi}_1 - \boldsymbol{\xi}_0$:

$$L_v = \mathbb{E}\boldsymbol{\xi}_0, \boldsymbol{\xi}_1, \lambda \left[|v(\lambda, \boldsymbol{\xi}\lambda) - (\boldsymbol{\xi}_1 - \boldsymbol{\xi}_0)|^2 \right]. \qquad (3)$$

To enable efficient sampling with large integration steps δ, the shortcut model enforces self-consistency. It minimizes

[1]P. Holzmann, M. Pfefferkorn and R. Findeisen are with the Control and Cyber-Physical Systems Laboratory, Technical University of Darmstadt, Germany {givenname.surname}@iat.tu-darmstadt.de

[2]J. Carvalho, A. T. Le and J. Peters are with the Intelligent Autonomous Systems Laboratory, Technical University of Darmstadt, Germany {givenname.surname}@tu-darmstadt.de

[3]A. Younes and G. Chalvatzaki are with the Interactive Robot Perception and Learning Laboratory, Technical University of Darmstadt, Germany {givenname.surname}@tu-darmstadt.de

[4]German Research Center for AI (DFKI), Research Department SAIROL

[5]Hessian Centre for Artificial Intelligence

the discrepancy between a single step of size δ and two steps of size 0.5δ using the augmented loss [8]:

$$L_v = \mathbb{E}\left[|v(\lambda, \boldsymbol{\xi}_\lambda, 0) - (\boldsymbol{\xi}_1 - \boldsymbol{\xi}_0)|^2 + |v(\lambda, \boldsymbol{\xi}_\lambda, \delta) - \bar{v}|^2\right], \tag{4}$$

where $\bar{v}$ represents the averaged velocity of the two smaller steps. This ensures accurate generation of expert-like coefficients with minimal computational cost.

IV. Model Predictive Path-Following Control

We employ a model predictive path-following controller (MPPFC) [10], [11], [12] to track the generated reference path and to ensure safety. The optimization problem is formulated as:

$$\begin{aligned}
\min_{\bar{u}, \bar{w}} \quad & J\big(\bar{e}, \bar{u}, \theta, \bar{w}\big) & \text{(5a)}\\
\text{s.t.} \quad & \forall s \in [t_k, t_k + T]: \\
& \dot{\bar{x}}(s) = f(\bar{x}(s), \bar{u}(s)), \quad \bar{x}(t_k) = x_k, & \text{(5b)}\\
& \dot{\theta}(s) = \rho(\theta(s), \bar{w}(s)), \quad \theta(t_k) = \theta_k, & \text{(5c)}\\
& \bar{y}(s) = h(\bar{x}(s)), & \text{(5d)}\\
& \bar{e}(s) = p(\theta(s)) - \bar{y}(s), & \text{(5e)}\\
& \theta(s) \in \mathcal{C}_\theta, \ \bar{w}(s) \in \mathcal{W}, & \text{(5f)}\\
& \bar{x}(s) \in \mathcal{X}_{\text{safe}} \subseteq \mathcal{X}, \ \bar{u}(s) \in \mathcal{U}, & \text{(5g)}\\
& \big[\bar{x}^\top(t_k + T) \quad \theta^\top(t_k + T)\big]^\top \in \Omega. & \text{(5h)}
\end{aligned}$$

At time t_k, we predict the system behavior over a horizon T using (5b)–(5d) to optimize the control input $\bar{u}$. Crucially, we employ virtual dynamics (5c) for the path parameter θ, governed by the virtual input $\bar{w}$. By treating θ as a degree of freedom, the MPPFC actively optimizes path progression, integrating planning and control. This allows for adaptive motion profiles to satisfy input and safety constraints (5g), such as obstacle avoidance. The cost function (5a) is defined as:

$$\begin{aligned} J\big(\bar{e}, \bar{u}, \theta, \bar{w}\big) = \int_{t_k}^{t_k+T} & L\big(\bar{e}(s), \bar{u}(s), \theta(s), \bar{w}(s)\big) ds \\ & + E\big(\bar{x}(t_k + T), \theta(t_k + T)\big), \end{aligned} \tag{6}$$

The stage cost L penalizes contouring errors (5e), control effort, and lack of path progress. The terminal cost E and constraint (5h) are designed to ensure convergence to the path [11].

V. Safety Through Model Predictive Path-Following Control

While generated paths can be safe regarding known static obstacles, they often lack formal guarantees, particularly when environmental changes occur. Re-evaluating the global generative model for every local change is computationally prohibitive [7]. Instead, we employ MPPFC to locally re-plan the system behavior, ensuring safety against new obstacles detected by onboard sensors (e.g., LiDAR). At initialization, we select a collision-free path from the generative batch to define the terminal constraints of the MPPFC. As established in [11], if the MPPFC finds a solution converging to this safe reference at the end of the prediction horizon, the execution is safe. However, unforeseen obstacles may render the optimal control problem infeasible if the system cannot realign with the path within the prediction horizon T. To address this, we employ an iterative horizon extension strategy. If the MPPFC fails to find a solution, we temporarily increase T until feasibility is recovered, allowing the solver to plan a detour around the obstacle (see Fig. 1). Once the obstacle is cleared, the horizon is shrunk back to its nominal length. While formal verification of this switching logic is challenging, this heuristic ensures local safety and task completion provided a physical path exists. If the necessary deviation is significant, we trigger an event-based re-planning of the global reference.

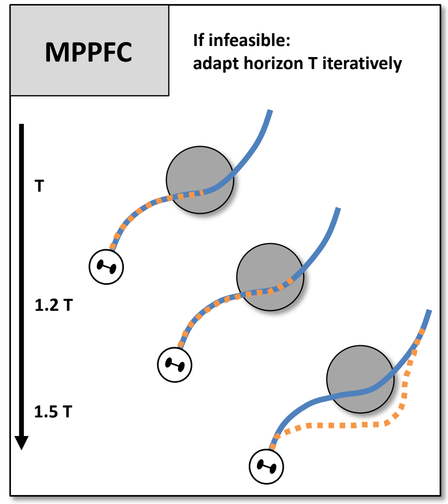

Fig. 1. Adaptive prediction horizon for safe navigation.

A. Biasing the Planner Toward the MPPFC Solution

When the MPPFC performs a large evasive maneuver, the resulting trajectory, while safe, may diverge significantly from the expert distribution. In such cases, it is advantageous to resample the generative planner. To avoid the computational cost of sampling from pure noise, we propose initializing the process from an intermediate state. We utilize the latest MPPFC predicted trajectory as a clean approximation $\hat{\boldsymbol{\xi}}_1$ of the desired path. We fit B-spline coefficients to this prediction using (1) and generate a partially noisy sample $\boldsymbol{\xi}_{\bar{\lambda}}$ at an intermediate step $\bar{\lambda} \in (0, 1)$. By initializing the flow matching model with $(\boldsymbol{\xi}_{\bar{\lambda}}, \bar{\lambda})$, we efficiently generate a new batch of expert-like paths that align with the currently feasible, safe local plan.

VI. Conclusion

We propose a framework combining flow-matching planners with Model Predictive Path-Following Control (MPPFC) to ensure safe execution of diverse, data-driven trajectories. MPPFC dynamically adapts motion to satisfy constraints, while an event-triggered re-planning scheme—biased by the controller's prediction—maintains consistency during evasive maneuvers. This synergy bridges expressive generative modeling with rigorous safety, enabling robust navigation in cluttered environments. Future work will address perception integration and dynamic obstacles.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, 2020.

[2] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *arXiv:2210.02747*, 2023, Version 2.

[3] Y. Lipman, M. Havasi, P. Holderrieth, N. Shaul, M. Le, B. Karrer, R. T. Q. Chen, D. Lopez-Paz, H. Ben-Hamu, and I. Gat, "Flow matching guide and code," *arXiv:2412.06264*, 2024, Version 1.

[4] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *R:SS XIX*, 2023.

[5] M. Reuss, M. Li, X. Jia, and R. Lioutikov, "Goal-conditioned imitation learning using score-based diffusion policies," in *R:SS XIX*, 2023.

[6] J. Carvalho, M. Baierl, J. Urain, and J. Peters, "Conditioned score-based models for learning collision-free trajectory generation," in *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

[7] J. Carvalho, A. T. Le, P. Kicki, D. Koert, and J. Peters, "Motion planning diffusion: Learning and adapting robot motion planning with diffusion models," *IEEE Trans. Robot.*, 2025.

[8] K. Frans, D. Hafner, S. Levine, and P. Abbeel, "One step diffusion via shortcut models," *arXiv:2410.12557*, 2024, Version 3.

[9] J. Schusterbauer, M. Gui, F. Fundel, and B. Ommer, "Diff2Flow: Training flow matching models via diffusion model alignement," *arXiv:2506.02221*, 2025, Version 1.

[10] J. Matschek, T. Bäthge, T. Faulwasser, and R. Findeisen, "Nonlinear predictive control for trajectory tracking and path following: An introduction and perspective," in *Handbook of Model Predictive Control*, Springer, 2019, pp. 169–198.

[11] T. Faulwasser and R. Findeisen, "Nonlinear model predictive control for constrained output path following," *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 1026–1039, 2015.

[12] T. Faulwasser, B. Kern, and R. Findeisen, "Model predictive path-following for constrained nonlinear systems," in *CDC*, 2009, pp. 8642–8647.

[13] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Madison, Wisconsin: Nob Hill Publishing, 2019.

[14] S. Lucia, M. Kögel, P. Zometa, D. E. Quevedo, and R. Findeisen, "Predictive control, embedded cyberphysical systems and systems of systems–a perspective," *Annu. Rev. Control*, vol. 41, pp. 193–207, 2016.

[15] G. Zhou, S. Swaminathan, R. V. Raju, J. S. Guntupalli, W. Lehrach, J. Ortiz, A. Dedieu, M. Lázaro-Gredilla, and K. Murphy, "Diffusion model predictive control," *arXiv:2410.05364*, 2025, Version 2.

[16] P. Marquez Julbe, J. Nubert, H. Hose, S. Trimpe, and K. J. Kuchenbecker, "Diffusion-based approximate MPC: Fast and consistent imitation of multi-modal action distributions," *arXiv:2504.04603*, 2025, Version 2.

[17] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-order sampling-based for torque-level locomotion control via diffusion-style annealing," *arXiv:2409.15610*, 2024, Version 1.

[18] V. Kurtz and J. W. Burdick, "Generative predictive control: Flow matching policies for dynamic and difficult-to-demonstrate tasks," *arXiv:2502.13406*, 2025.

[19] Z. Huang, Y. Lin, F. Yang, and D. Berenson, "Subgoal diffuser: Coarse-to-fine subgoal generation to guide model predictive control for robot manipulation," *arXiv:2403.13085*, 2024, Version 1.

[20] E. Milios, K. P. Wabersich, F. Berkel, and L. Schwenkel, "Stability mechanisms for predictive safety filters," in *CDC*, 2023, pp. 2409–2416.

[21] K. P. Wabersich and M. N. Zeilinger, "A predictive safety filter for learning-based control of constrained nonlinear dynamical systems," *Automatica*, vol. 129, 2021, 109597.

[22] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4.