

SafeDMPs: Integrating Formal Safety with DMPs for Adaptive HRI

Pranav Tiwari*, Soumyodipta Nath*, Ravi Prakash

Abstract—Robots operating in human-centric environments must be both robust to disturbances and provably safe from collisions. Achieving these properties simultaneously and efficiently remains a central challenge. While Dynamic Movement Primitives (DMPs) offer inherent stability and generalization from single demonstrations, they lack formal safety guarantees. Conversely, formal methods like Control Barrier Functions (CBFs) provide provable safety but often rely on computationally expensive, real-time optimization, hindering their use in high-frequency control. This paper introduces SafeDMPs, a novel framework that resolves this trade-off. We integrate the closed-form efficiency and dynamic robustness of DMPs with a provably safe, non-optimization-based control law derived from Spatio-Temporal Tubes (STTs). This synergy allows us to generate motions that are not only robust to perturbations and adaptable to new goals, but also guaranteed to avoid static and dynamic obstacles. Our approach achieves a closed-form solution for a problem that traditionally requires online optimization. Experimental results on a 7-DOF robot manipulator demonstrate that SafeDMPs is orders of magnitude faster and more accurate than optimization-based baselines, making it an ideal solution for real-time, safe, and collaborative robotics.

I. INTRODUCTION

For robots to seamlessly integrate into human-centric environments, they must operate with a high degree of robustness and safety. Robustness is the capacity to maintain stable, goal-directed behavior despite external disturbances, while safety is the guarantee of avoiding harm to people and property. While these two properties are paramount for real-world deployment, achieving them simultaneously in a computationally efficient manner remains a central challenge in robotic control.

One major line of research addresses robustness and generalization through Learning from Demonstration (LfD) using movement primitives. Among these, Dynamic Movement Primitives (DMPs) [1] are particularly effective. DMPs encode trajectories as attractor-based dynamical systems, which provides inherent stability and robustness to perturbations from a single demonstration. This offers a distinct advantage over methods like Gaussian Mixture Models (GMMs) [2] or Probabilistic Movement Primitives (ProMPs) [3], which often require numerous demonstrations and can be computationally complex. Other approaches, such as Kernelized Movement Primitives (KMPs) [4] or task-space deformation techniques [5], can adapt motions but require explicit, user-defined constraints that are often difficult to specify. Despite their strengths in robust trajectory encoding, DMPs fundamentally lack a principled mechanism for safety. They

are typically paired with reactive, heuristic methods like Artificial Potential Fields (APFs) [6], which are notoriously susceptible to local minima and unstable oscillations, providing no formal safety guarantees.

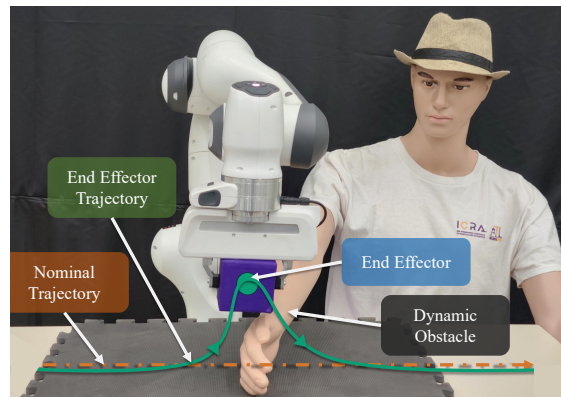


Fig. 1: Hardware setup featuring the Franka Emika Panda robot equipped for human-robot interaction tasks. Illustration of SafeDMPs for adaptive Human Robot Interaction (HRI) in response to dynamic interference.

With stability in mind, research in this field has evolved to provide provable guarantees using formal methods. For stability, while DMPs provide it inherently, other sophisticated learning frameworks like Stable-BC [7], Lyapunov Density Models [8], and Euclideanizing Flows [9], [10] have been developed. However, these methods typically require extensive training over multiple demonstrations. For safety, the use of Control Barrier Functions (CBFs) has become a cornerstone for guaranteeing forward invariance of a safe set. Frameworks combining Dynamical Systems [11], [12] with Control Lyapunov Functions (CLFs) and CBFs, such as NODE-CLF-CBF [13], can enforce provable safety and stability. The critical drawback of this class of methods is their reliance on solving a constrained quadratic program at every control step. This online optimization imposes a significant computational burden, leading to sluggish, non-smooth behavior unsuitable for high-frequency control. While alternative closed-form safety methods like the Rotational Obstacle Avoidance Method (ROAM) [14] exist, they offer less explicit control over the safety margin, leading to less predictable path shapes compared to CBFs.

The literature presents a clear trade-off: efficient and robust trajectory generation methods, like DMPs, that lack formal safety guarantees, versus formally safe methods that are computationally expensive due to online optimization. This dichotomy highlights a core challenge in robotics,

*denotes equal contribution

All authors are with the Cyber Physical Systems, Indian Institute of Science, Bengaluru, 560012, India. (e-mail: {pranavtiwari, soumyodiptan, raviivr}@iisc.ac.in)

i.e., balancing performance with provable safety. This paper presents SafeDMP, a novel framework that resolves this trade-off by marrying the robustness and efficiency of DMPs with the provable safety of formal methods, without the computational overhead of online optimization.

Our core contribution is the integration of Spatio-Temporal Tubes (STTs), which define a time-varying safe envelope around the nominal DMP trajectory. The STT provides a closed-form, non-optimization-based control law that provably ensures the robot’s motion remains within a collision-free region. As seen in Figure 1, the robot deviates from its nominal trajectory in the presence of human interference while maintaining a minimum safety margin, and subsequently converges back to the nominal path once the interference is removed. This synergy results in a system that is not only inherently robust to perturbations but also guaranteed to adapt safely in the presence of static and dynamic obstacles. Extensive experiments demonstrate that SafeDMP achieves orders-of-magnitude faster execution times and more accurate trajectory reproduction than optimization-based baselines, making it an ideal solution for real-time deployment in dynamic, human-centric environments.

II. PRELIMINARIES

This section introduces the fundamental mathematical frameworks that underpin our approach to safe and adaptive human-robot interaction.

A. Dynamic Movement Primitives (DMPs) [1]

DMPs provide a unified framework for learning, representing, and generalizing robot motions from demonstrations. They model trajectories as a second-order dynamical system with a nonlinear forcing term that encodes the shape of the motion. The dynamics are governed by:

$$\tau^2 \ddot{x} = \alpha_z (\beta_z (g - x) - \tau \dot{x}) + f(z), \quad (1)$$

where $x(t) \in \mathbb{R}^d$ is the trajectory in d -dimensional space, $g \in \mathbb{R}^d$ is the goal state, $\tau > 0$ is a temporal scaling factor controlling duration, $\alpha_z, \beta_z > 0$ are gains (with $\beta_z = \alpha_z/4$ for critical damping), and $f(z)$ is the nonlinear forcing function that shapes the trajectory governed by a canonical monotonically decreasing phase variable

$$\tau \dot{z} = -\alpha_z z, \quad z(0) = 1, \quad (2)$$

From a demonstration, the target forcing function is obtained by rearranging (1) and is parameterized using a weighted combination of Gaussian basis functions.

This formulation enables flexible adaptation through multiple scaling mechanisms:

a) Stable convergence: Attractor dynamics guarantees goal reaching even under perturbations.

b) Spatial Scaling: Trajectories can be spatially transformed by modifying the initial position y_0 and goal g where the forcing function automatically scales the trajectory shape to the new spatial extent.

c) Temporal Scaling: The duration and speed of the motion are controlled by the scaling parameter $\tau > 0$, where $\tau > 1$ slows execution and $0 < \tau < 1$ speeds it up.

d) Obstacle Avoidance: DMPs can be extended with coupling terms for real-time obstacle avoidance:

$$\tau^2 \ddot{x} = \alpha_z (\beta_z (g - x) - \tau \dot{x}) + f(z) + f_{\text{coupling}}(x, \dot{x}, \text{obstacles}), \quad (3)$$

where f_{coupling} generates repulsive forces from detected obstacles.

B. Spatio-Temporal Tubes (STT) [15]

STTs enforce state-dependent safety by constraining each state dimension $x_i(t)$ within time-varying bounds:

$$\rho_{i,L}(t) < x_i(t) < \rho_{i,U}(t), \quad \forall t \geq 0, \quad (4)$$

without requiring online optimization by using a normalized error representation. Define the sum and difference of bounds:

$$\rho_{i,s}(t) = \rho_{i,U}(t) + \rho_{i,L}(t), \quad (5)$$

$$\rho_{i,d}(t) = \rho_{i,U}(t) - \rho_{i,L}(t), \quad (6)$$

The normalized error for dimension i is:

$$e_i(x_i, t) = 2\rho_{i,d}^{-1}(t) \left(x_i(t) - \frac{1}{2}\rho_{i,s}(t) \right), \quad (7)$$

which maps the tube interior to $(-1, 1)$ and centers the error at zero when $x_i(t)$ is at the tube center.

The control function that enforces tube constraints is:

$$\xi_i(x_i, t) = 4\rho_{i,d}^{-1}(t) (1 - e_i(x_i, t)^2), \quad (8)$$

which provides maximum control authority at the tube boundaries and minimum at the center.

The STT control law ensures finite-time convergence to the target set while maintaining tube constraints:

$$u_i(x_i, t) = -k \xi_i(x_i, t) e_i(x_i, t), \quad k > 0, \quad (9)$$

$$\epsilon_i(x_i, t) = \ln \left(\frac{1 + e_i(x_i, t)}{1 - e_i(x_i, t)} \right), \quad (10)$$

This logarithmic error term ϵ_i ensures:

- $\epsilon_i \rightarrow -\infty$ as $e_i \rightarrow -1$ (approaching lower bound)
- $\epsilon_i = 0$ when $e_i = 0$ (at tube center)
- $\epsilon_i \rightarrow +\infty$ as $e_i \rightarrow +1$ (approaching upper bound)

This property makes STTs particularly effective for human-robot interaction, where responsiveness is critical and online optimization may be infeasible.

III. METHODOLOGY

We propose a two-stage framework for generating robust, adaptive, and safe robot motion in dynamic environments. The framework consists of (i) learning nominal motion plans from demonstrations and (ii) enforcing robustness and safety during execution. Our method operates in the end-effector space, directly encoding Cartesian trajectories to improve generalization and maintain explicit safety margins.

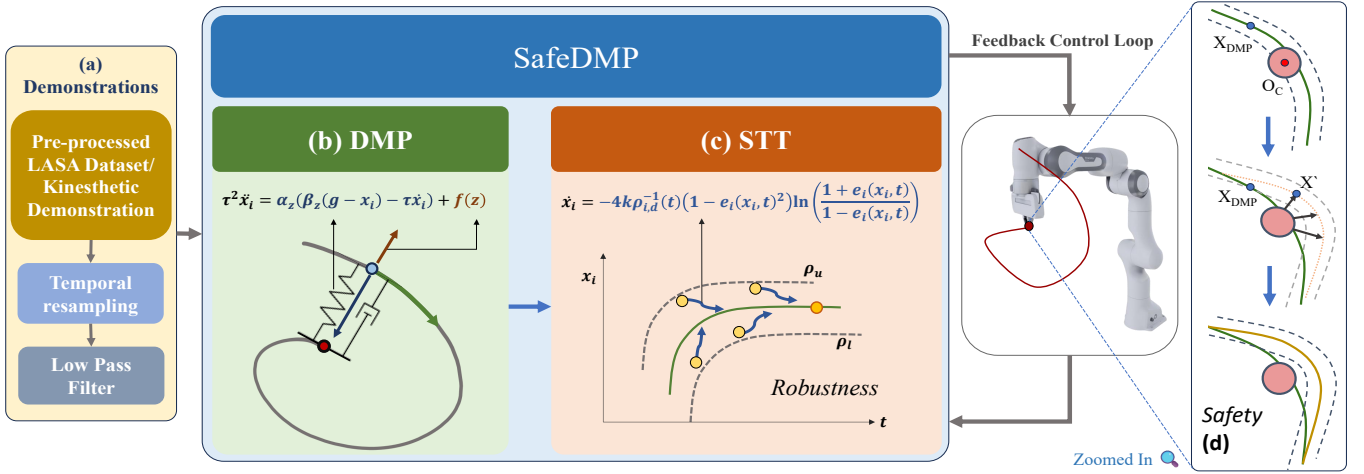


Fig. 2: Proposed framework pipeline. (a) Demonstrations are recorded in end-effector space. (b) DMPs encode the nominal motion plan. (c) Spatio-Temporal Tubes (STTs) define a safe envelope around the DMP. (d) During execution, trajectory deformation is triggered when obstacles are encountered, and the system safely reroutes motion within the STT while ensuring convergence.

A. DMPs as Nominal Motion Plans

Learning from demonstration (LfD) allows robots to acquire complex motor behaviors from kinesthetic teaching. We adopt Dynamic Movement Primitive (DMPs) [1] as our nominal representation because they provide: stable convergence guarantees, temporal flexibility (through the parameter τ), and spatial generalization (see Sec. II-A). In addition, we can easily embed an internal feedback term that enables smooth recovery from deviations without requiring an external correction layer. This adaptability is particularly valuable when integrated with Spatio-Temporal Tubes (STTs), because the deviations caused by safety constraints can be absorbed into the primitive’s dynamics.

B. SafeDMPs: Enforcing Robustness and Safety via Spatio-Temporal Tubes

While DMPs ensure convergence and adaptability, they do not guarantee safety in dynamic environments. To address this, we integrate STTs [15] into the DMP execution layer. STTs define time-varying safety envelopes for each state dimension around the nominal trajectory (see Sec. II-B, Eq. (7)–(10)), and enforce constraint satisfaction through a closed-form control law (Eq. (9)) without requiring online optimization. Figure 2 illustrates this integration.

The closed-loop system additively integrates the expressiveness of DMPs with the safety guarantees of STTs to yield a closed-form, optimization-free dynamical system (Fig. 2(d)). This structure adaptively regulates motion while preserving the distinct advantages of both frameworks. The DMP provides a critically damped, second-order attractor for robust goal convergence, while the STT introduces an obstacle-aware modulation term, f_{STT} , to handle environmental perturbations:

$$\tau^2 \ddot{x} = \alpha_z (\beta_z (g - x) - \tau \dot{x}) + f(z) + f_{\text{STT}}(z, x, \mathcal{O}). \quad (11)$$

The STT ensures forward invariance within a safe envelope ($\rho_{i,L}(t) < x_i(t) < \rho_{i,U}(t)$) using a diverging logarithmic

error term that dominates near boundaries to strictly prevent constraint violations. When dynamic obstacles \mathcal{O} arise, the STT safely deforms the nominal DMP trajectory (\mathbf{x}_{DMP}) based on the Euclidean distance d to the obstacle center \mathbf{o}_c :

$$\mathbf{x}' = \begin{cases} \mathbf{x}_{\text{DMP}}, & \text{if } d(\mathbf{x}_{\text{DMP}}, \mathbf{o}_c) > r_{\text{safe}}, \\ \mathbf{o}_c + \frac{\mathbf{x}_{\text{DMP}} - \mathbf{o}_c}{d(\mathbf{x}_{\text{DMP}}, \mathbf{o}_c)} \cdot r_{\text{clearance}}, & \text{otherwise.} \end{cases} \quad (12)$$

This rerouting strategy enforces a minimum clearance ($r_{\text{clearance}}$) upon breaching a buffer zone (r_{safe}), securely guiding the end-effector while the underlying DMP ensures goal convergence. While formulated for spherical constraints, replacing the point-to-center distance $d(\cdot)$ with a *signed distance function* naturally extends this approach to arbitrary obstacle geometries.

C. Adaptive Timing for Recovery

To handle large deviations from the nominal path, we further adapt the temporal scaling parameter τ :

$$\dot{e} = \alpha_e (\mathbf{x}_{\text{actual}} - \mathbf{x}_{\text{DMP}} - \mathbf{e}), \quad (13)$$

$$\tau = \tau_{\text{nominal}} + k_c \|\mathbf{e}\|^2. \quad (14)$$

If deviations grow large, τ increases, slowing down execution and allowing more time for recovery; when deviations are small, τ decreases, speeding up execution. This ensures recovery remains smooth and bounded rather than abrupt (Fig. 2(d)).

IV. EXPERIMENTAL EVALUATION

1) *Robot and Environment*: All simulations were conducted using a model of the 7-DOF Franka Emika Panda manipulator in Isaac Sim within a cubic workspace measuring $1.1 \times 1.1 \times 1.1$ m (Fig. 3). The controller ran at a frequency of 200 Hz ($\Delta t = 0.005$ s)

Algorithm 1 Integrated DMP-STT Motion Planning Framework with Embedded STT Safety Control

Require: Reference trajectory $\mathbf{X}_{ref} \in \mathbb{R}^{N \times d}$, DMP parameters α , $n_{features}$, STT parameters K , δ_γ , obstacle (\mathbf{o}_c, r_o) , time step Δt

Ensure: Executed safe trajectory

- 1: **Phase 1: DMP Learning**
- 2: Initialize: $\beta \leftarrow \alpha/4$, $\alpha_z \leftarrow \alpha/6$, $\alpha_e \leftarrow \alpha/10$, $k_c \leftarrow 2\alpha$
- 3: $\mathbf{x}_0 \leftarrow \mathbf{X}_{ref}[0]$, $\mathbf{g} \leftarrow \mathbf{X}_{ref}[-1]$
- 4: Learn Weights: $\mathbf{w} \leftarrow DMP_{train}(\mathbf{X}_{ref}, \beta, \alpha, \alpha_z)$
- 5: **Phase 2: Real-time Execution**
- 6: Initialize: $z \leftarrow 1$, $\mathbf{X} \leftarrow \mathbf{x}_0$, $\dot{\mathbf{X}} \leftarrow \mathbf{0}$, $\mathbf{e} \leftarrow \mathbf{0}$
- 7: **while** $\|\mathbf{X} - \mathbf{g}\| > \epsilon_{thresh}$ **do**
- 8: **DMP Step:**
- 9: $\psi \leftarrow \exp(-\mathbf{h}_z \cdot (z - \mathbf{c}_z)^2)$
- 10: $\mathbf{f}_{ext} \leftarrow (\mathbf{g} - \mathbf{x}_0) \cdot z \cdot ((\psi / \sum \psi)^T \mathbf{w})$
- 11: $\ddot{\mathbf{X}} \leftarrow \frac{1}{\tau^2} (\alpha(\beta(\mathbf{g} - \mathbf{X}) - \tau \dot{\mathbf{X}}) + \mathbf{f}_{ext})$
- 12: **Embedded STT Safety Control:**
- 13: $\mathbf{X}_{target} \leftarrow \mathbf{X} + \dot{\mathbf{X}} \cdot \Delta t$
- 14: $\mathbf{x}_{safe} \leftarrow \mathbf{X}_{target}$
- 15: **if** obstacle present **then**
- 16: $\mathbf{v}_{ot} \leftarrow \mathbf{X}_{target} - \mathbf{o}_c$, $d_{ot} \leftarrow \|\mathbf{v}_{ot}\|$
- 17: $clearance \leftarrow r_o + 0.5\delta_\gamma$
- 18: **if** $d_{ot} \leq clearance$ **then**
- 19: $\mathbf{x}_{safe} \leftarrow \mathbf{o}_c + \frac{\mathbf{v}_{ot}}{d_{ot}} \cdot clearance$
- 20: **end if**
- 21: **end if**
- 22: Error: $\epsilon \leftarrow \frac{\mathbf{X}_{current} - \mathbf{x}_{safe}}{0.5\delta_\gamma}$
- 23: Clip: $\epsilon \leftarrow \text{clip}(\epsilon, -0.99, 0.99)$
- 24: Transform: $\epsilon_t \leftarrow \log\left(\frac{1+\epsilon}{1-\epsilon}\right)$
- 25: Control: $\mathbf{u}_{stt} \leftarrow -K \cdot \frac{4}{\delta_\gamma(1-\epsilon^2)} \cdot \epsilon_t$
- 26: $\mathbf{X}_{desired} \leftarrow \mathbf{X}_{target} + \mathbf{u}_{stt} \cdot \Delta t$
- 27: **Robot Control:**
- 28: Move to $\mathbf{X}_{desired}$
- 29: $\mathbf{X}_{current} \leftarrow \text{get_current_ee_position}()$
- 30: **Update DMP State:**
- 31: $\mathbf{e} \leftarrow \mathbf{e} + \alpha_e(\mathbf{X}_{current} - \mathbf{X})\Delta t$
- 32: $\tau \leftarrow \tau_{base} + k_c\|\mathbf{e}\|^2$
- 33: $z \leftarrow z - \frac{\alpha_z z \Delta t}{\tau}$
- 34: $\mathbf{X} \leftarrow \mathbf{X} + \dot{\mathbf{X}}\Delta t + 0.5\ddot{\mathbf{X}}\Delta t^2$
- 35: $\dot{\mathbf{X}} \leftarrow \dot{\mathbf{X}} + \ddot{\mathbf{X}}\Delta t$
- 36: **end while**

2) *Motion Data and Nominal Plan:* We utilized the LASA handwriting dataset [11] as the source of example motions. The original 2D demonstrations were lifted into 3D space (with a fixed z -height), randomly rotated, temporally resampled, and smoothed with a low-pass filter. These processed trajectories were encoded as DMPs using 25 basis functions to serve as the nominal, obstacle-free motion plan for all subsequent simulation experiments.

All experiments were conducted on an Intel Core i7-12700K workstation with 16 GB RAM. The complete set of hyperparameters and implementation details for both hardware & software are provided in the repository at [GitHub](#).

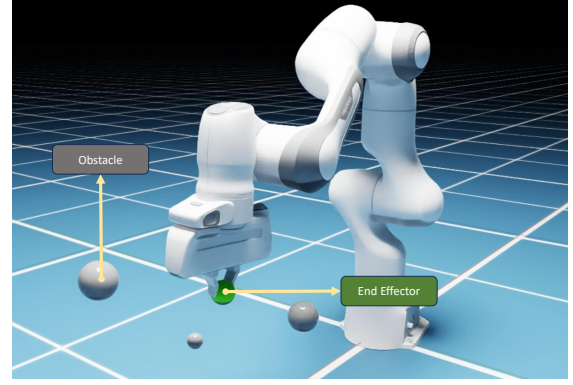


Fig. 3: Simulation setup featuring the Franka robot performing tasks within a workspace containing obstacles.

A. Baselines for Comparison

We benchmark the performance of our proposed SafeDMPs against two widely used methods:

- **NODE-CLF-CBF** [16]: learning-based dynamics combined with control barrier functions, offering formal safety guarantees via online optimization.
- **DMP-APF** [6]: nominal motion encoding with artificial potential fields for reactive avoidance.

B. Experimental Tasks and Evaluation Metrics

To comprehensively evaluate each algorithm, we designed the following experimental tasks, complemented by quantitative measures that capture both efficiency and reliability.

1) *Nominal Trajectory Reproduction:* Execution of the learned motion in an obstacle-free environment establishes a baseline for performance. The evaluation includes: i) **Execution Time (s):** The average time required for computing control commands per control loop to reflect computational efficiency. ii) **Memory Usage (MB):** The average memory footprint measured during execution. iii) **Mean Absolute Error (Nominal):** The deviation (in m) between the executed trajectory and the reference demonstration, quantifying imitation fidelity under nominal conditions.

2) *Robustness:* Two sharp impulse perturbations are applied to the end-effector at identical instants across all algorithms. Evaluation focuses on: i) **Mean Absolute Error (Perturbation):** The deviation (in m) from the reference trajectory under disturbances, capturing resilience to perturbations. ii) **Convergence Time (Perturbation):** The average elapsed time for the perturbed trajectory to re-converge with the nominal path.

3) *Safety:* Safety is assessed in the presence of environmental constraints: i) **Static Obstacle Avoidance:** Obstacles are placed at random positions intersecting the nominal trajectory. The evaluation measures whether collisions are avoided and the smoothness of trajectory deformation. ii) **Dynamic Obstacle Avoidance:** Obstacle moves with constant velocity on a path intersecting the nominal trajectory. Algorithms are evaluated on their ability to predict obstacle motion and generate proactive collision-free trajectories. iii)

TABLE I: Qualitative comparison and Quantitative evaluation of motion generation methods.

Method	Qualitative Properties				Quantitative Metrics					
	Formal Safety	Robust Recovery	No Oscill. / Minima	Real-Time Efficiency	Exec. Time (s)	Mem. (MB)	MAE Nominal	MAE Perturb.	Conv. Time (OA)	Conv. Time (Pert.)
NODE-CLF-CBF [16]	✓	✗	✓	✗	0.3207	600	0.2244	0.3921	0.411	1.294
NODE-CLF-CBF (JIT) [16]	✓	✗	✓	✗	1.8e-3	0.311	0.2244	0.3921	0.411	1.294
DMP-APF [6]	✗	✗	✗	✓	6.7e-4	0.0208	0.02304	0.0228		0.411
DMP-APF (JIT) [6]	✗	✗	✗	✓	2.3e-4	0.0208	0.02304	0.0228		0.411
SafeDMPs	✓	✓	✓	✓	1.03e-4	0.109	0.0113	0.0531	0.6325	0.025
SafeDMPs (JIT)	✓	✓	✓	✓	3.7e-5	0.161	0.0113	0.0531	0.6325	0.025

Qualitative metrics: ✓ Formal Safety: proven forward invariance; ✓ Robust Recovery: bounded error convergence post-perturbation; ✓ No Oscill./Minima: zero instances across 100 trials. **JIT implementation:** Uses numba (SafeDMP, DMP-APF, CLF-CBF) and PyTorch JIT (NODE), excluding initial compilation overhead to capture steady-state control.

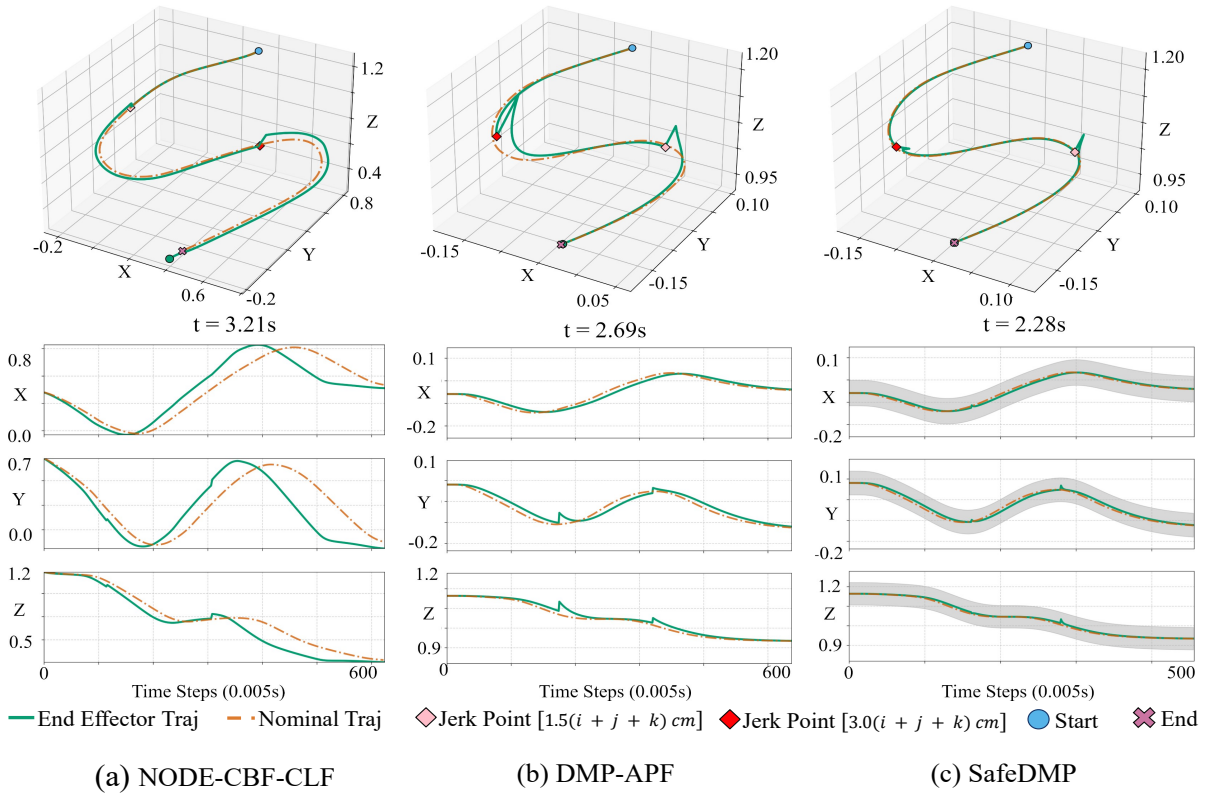


Fig. 4: Perturbation response across methods. The plots illustrate the evolution of the end-effector trajectory over time (top) and the corresponding X , Y , and Z coordinates (bottom) under external disturbances.

Convergence Time (Obstacle Avoidance): The mean additional time required for the trajectory to return to the goal after executing obstacle avoidance maneuvers per obstacle.

For clarity in visualization, the results presented in Figures 4–6 depict the Cartesian path of the end-effector highlighted in Figure 3.

C. Results and Analysis

1) **Robustness to Perturbations:** Figure 4 illustrates the response of each method to external perturbations of varying magnitudes. The NODE-CLF-CBF method (Fig. 4(a)) reacts strongly to small perturbations, accelerating away from the nominal trajectory and taking the longest time to settle; under larger perturbations, it deviates further, overshoots the goal,

and produces large trajectory errors. The DMP-APF method (Fig. 4(b)) exhibits the sharpest deviation along perturbation, but recovers faster than NODE-CLF-CBF. In contrast, the proposed SafeDMPs method (Fig. 4(c)) demonstrates smooth and bounded recovery, maintaining its trajectory within a spatio-temporal tube and converging back to the nominal path most quickly.

2) **Safety from Static and Dynamic Obstacles:** The obstacle avoidance capabilities are shown in Figures 5 (static) and 6 (dynamic). NODE-CLF-CBF (Figs. 5(a), 6(a)) successfully avoids collisions but often with minimal clearance, leaving little tolerance for execution errors or modeling uncertainties. Increasing the clearance parameter improves mar-

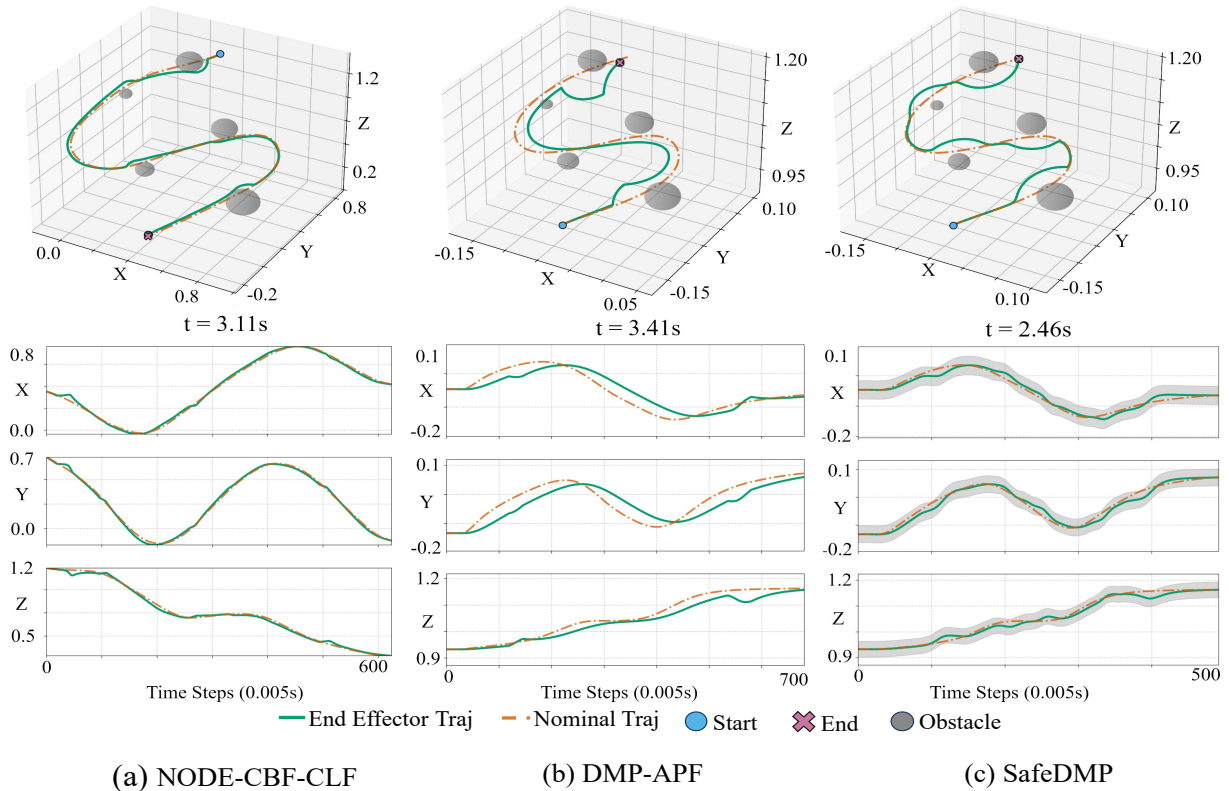


Fig. 5: Obstacle avoidance across methods with static obstacles. The plots show the evolution of the end-effector trajectory when static obstacles are introduced during execution (top) and the corresponding X , Y , and Z coordinates (bottom).

gins but results in abrupt, jerky motion, reflecting a trade-off between smoothness and safety. DMP-APF (Fig. 5(a), 6(b)) suffers from local minima, frequently resulting in collisions or erratic, oscillatory paths. SafeDMPs (Figs. 5(c), 6(c)) consistently generates smooth, safe detours around both static and dynamic obstacles while maintaining a well-defined safety margin and stable convergence to the goal.

A summary of the comprehensive qualitative and quantitative comparison is presented in Table I. The proposed SafeDMPs method demonstrates superior performance across all metrics, successfully combining formal safety guarantees with robust, efficient, and stable motion generation. Notably, the obstacle avoidance convergence time for the DMP+APF baseline is omitted due to its high collision rate, which prevented a meaningful measurement.

D. Hardware Validation Results

To validate SafeDMP’s real-world applicability, we deployed it on a Franka Emika Panda robot operating at 1 kHz. A mannequin emulated human-robot interaction scenarios (Fig. 1). Task training data, collected via kinesthetic demonstrations, was temporally resampled and low-pass filtered prior to DMP learning. For real-time intervention detection, an Intel RealSense camera combined with Mediapipe pose detection tracked the mannequin’s hand. To account for vision-based sensing latency and tracking noise, conservative STT safety margins were incorporated to maintain tube invariance against bounded estimation errors. Ultimately,

the high-frequency control loop and closed-form modulation rapidly compensate for perception jitter, ensuring collision-free behavior under realistic uncertainties.

1) *Demonstrated Tasks and Results:* We evaluated the proposed framework through three distinct human-robot interaction (HRI) scenarios, each involving intentional human interventions to test safety and reactivity:

- **Pick-and-Place with Dynamic Obstruction:** The robot executed a standard pick-and-place operation while a moving human hand randomly continued to obstruct its nominal path. SafeDMPs generated smooth trajectory deviations to reactively avoid unsafe contact while ensuring successful task completion (Fig. 7(a)).
- **Assistive Wiping with Intermittent Obstruction:** The robot performed a wiping motion on a surface while a health worker randomly blocked its path intermittently. The framework reliably deformed the trajectory to maintain a safe clearance and seamlessly resumed contact-based wiping once the obstruction was removed (Fig. 7(b)).
- **Whiteboard Cleaning with Cluttered Obstacles:** The robot performed a whiteboard cleaning task in the presence of multiple static obstacles intentionally placed close to its nominal path. SafeDMPs adapted the end-effector’s trajectory to maneuver around the obstructions while respecting predefined safety margins and maintaining task progress (Fig. 7(c)).

Across all hardware demonstrations, SafeDMPs exhibited the

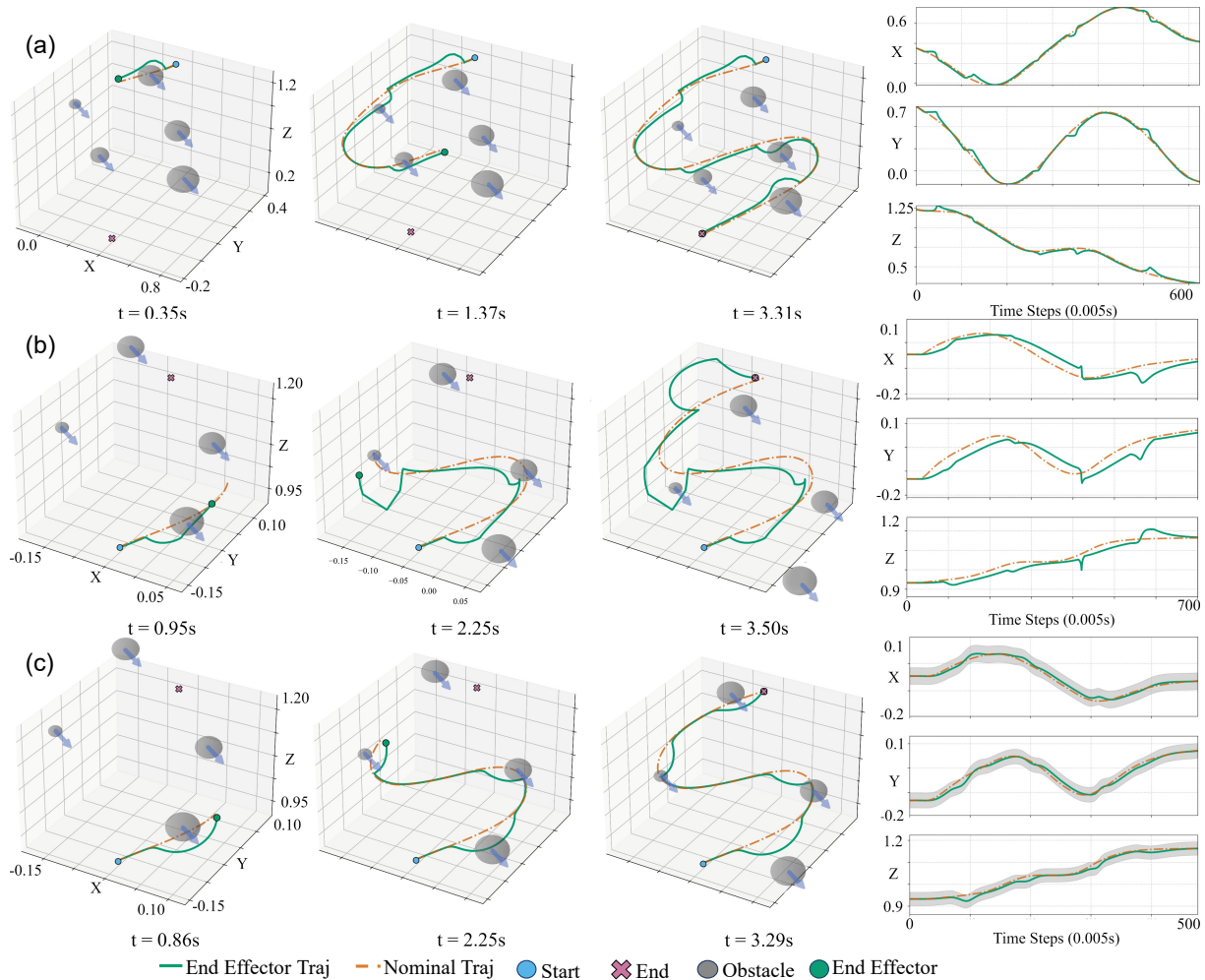


Fig. 6: Obstacle avoidance across methods with moving obstacles. The plots show three 3D frames (left) illustrating the evolution of the end-effector trajectory and the corresponding X , Y , and Z coordinates (right).

same safe interactions and reliably converged back to its nominal path, confirming the framework’s robustness in real-world, human-interactive environments. (Figs. 7(a)–7(c))

V. CONCLUSION

Our modular framework, **SafeDMPs**, generates safe and adaptive motion by integrating Dynamic Movement Primitives with Spatio-Temporal Tubes. This approach avoids online constrained optimization and leverages closed-form safety guarantees. SafeDMPs achieves real-time execution at high control frequencies while ensuring provable collision avoidance.

A comprehensive evaluation in simulation and on a physical Franka Emika Panda robot demonstrated that SafeDMPs consistently outperforms both optimization-based and heuristic baselines. Compared to NODE-CLF-CBF, SafeDMPs reduced execution time by up to **99.97%**, lowered memory footprint by **65%**, and decreased nominal trajectory error by **95%** and perturbation error by **86%**. Furthermore, SafeDMPs achieved up to a **94%** faster convergence under perturbations and **35%** faster convergence during obstacle

avoidance. Unlike DMP-APF, SafeDMPs avoided oscillations and local minima, ensuring smooth and safe trajectories even in cluttered or dynamic environments.

The combination of formal safety guarantees, computational efficiency, and robustness makes SafeDMPs well-suited for deployment in collaborative, assistive, and industrial robotics, where reliable real-time adaptation in human-centered environments is essential.

VI. LIMITATIONS AND FUTURE WORK

While the proposed framework demonstrates strong performance in real-time, adaptive, and safe motion generation, several limitations remain. The current implementation relies on manual parameter tuning, which may limit scalability and generalization. Additionally, the control behavior near the boundaries of the STT produces increasingly strong corrective actions as the state approaches the tube limits, theoretically tending toward infinite control to enforce invariance, which cannot be realized due to actuator constraints.

Future work will investigate adaptive parameter selection strategies that automatically adjust tube parameters and control gains based on task context, perception uncertainty, and

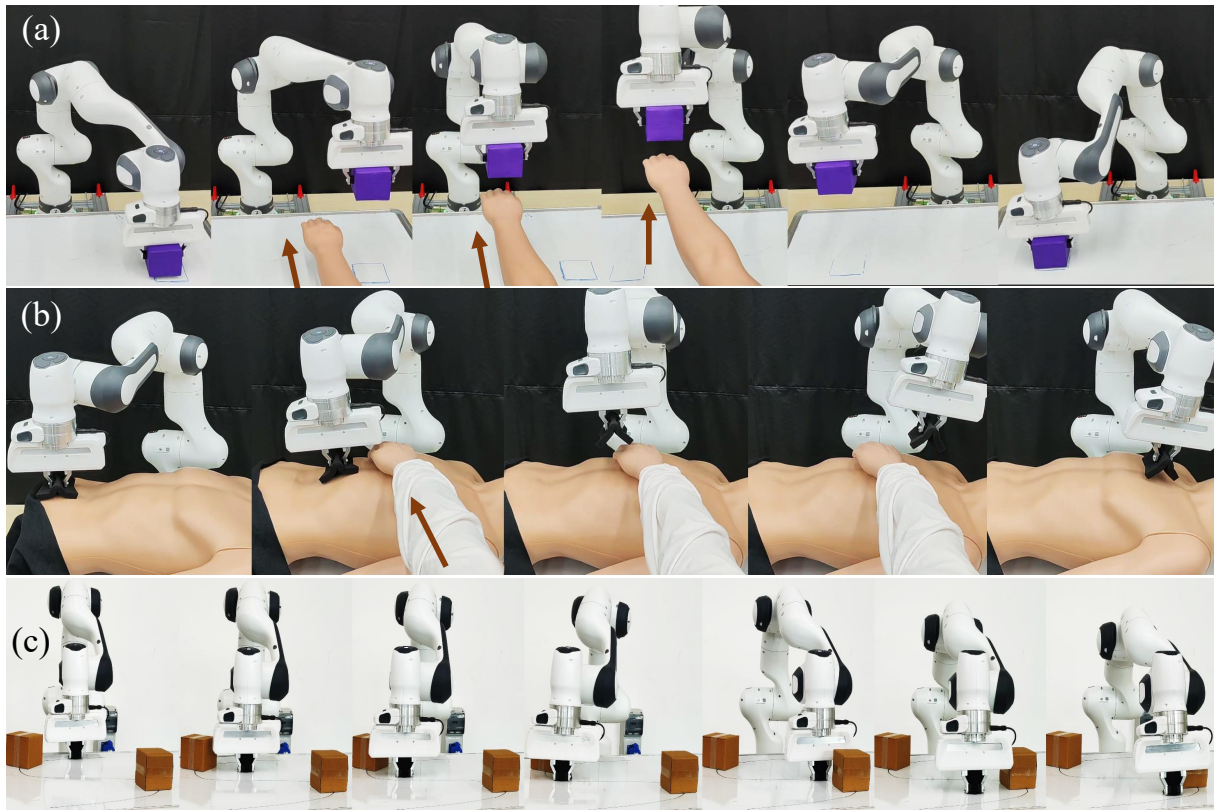


Fig. 7: Hardware validation of the SafeDMP framework in human-robot interaction (HRI) scenarios. (a) Pick-and-Place: The robot reactively deviates from its nominal trajectory to avoid a dynamically moving human hand. (b) Assistive Wiping: The robot adjusts its wiping path on a mannequin to maintain a safe distance from an intruding hand within the workspace. (c) Cluttered Cleaning: The robot successfully navigates around multiple static obstacles during a whiteboard cleaning task.

observed disturbances. We also plan to develop bounded-control formulations and tube redirection strategies using a world-model-informed planner to generate new feasible tubes when large disturbances occur. Furthermore, we aim to extend the framework to incorporate whole-body safety and integrate high-level spatio-temporal logic planners for handling complex task specifications and bi-arm coordination.

REFERENCES

- [1] A. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, pp. 328–373, 02 2013.
- [2] N. Jaquier, D. Ginsbourger, and S. Calinon, "Learning from demonstration with model-based gaussian process," *CoRR*, vol. abs/1910.05005, 2019.
- [3] A. Paraschos, G. Neumann, and J. Peters, "A probabilistic approach to robot trajectory generation," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, pp. 477–483.
- [4] Y. Huang, L. D. Rozo, J. Silvério, and D. G. Caldwell, "Kernelized movement primitives," *CoRR*, vol. abs/1708.08638, 2017.
- [5] T. Li and N. Figueroa, "Task generalization with stability guarantees via elastic dynamical system motion policies," in *7th Annual Conference on Robot Learning*, 2023.
- [6] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 2587–2592.
- [7] S. A. Mehta, Y. U. Ciftci, B. Ramachandran, S. Bansal, and D. P. Losey, "Stable-bc: Controlling covariate shift with stable behavior cloning," *IEEE Robotics and Automation Letters*, 2025.
- [8] K. Kang, P. Gradu, J. J. Choi, M. Janner, C. Tomlin, and S. Levine, "Lyapunov density models: Constraining distribution shift in learning-based control," in *International Conference on Machine Learning*. PMLR, 2022, pp. 10 708–10 733.
- [9] M. A. Rana, A. Li, D. Fox, B. Boots, F. Ramos, and N. Ratliff, "Euclideanizing flows: Diffeomorphic reduction for learning stable dynamical systems," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 630–639.
- [10] R. Pérez-Dattari and J. Kober, "Stable motion primitives via imitation and contrastive learning," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3909–3928, 2023.
- [11] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.
- [12] N. Figueroa and A. Billard, "A physically-consistent bayesian non-parametric mixture model for dynamical system learning," in *CoRL*, 2018, pp. 927–946.
- [13] F. Nawaz, T. Li, N. Matni, and N. Figueroa, "Learning complex motion plans using neural odes with safety and stability guarantees," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 17 216–17 222.
- [14] L. Huber, J.-J. Slotine, and A. Billard, "Avoidance of concave obstacles through rotation of nonlinear dynamics," *IEEE Transactions on Robotics*, vol. 40, pp. 1983–2002, 2023.
- [15] R. Das, A. Basu, and P. Jagtap, "Spatiotemporal tubes for temporal reach-avoid-stay tasks in unknown systems," *IEEE Transactions on Automatic Control*, pp. 1–8, 2025.
- [16] F. Nawaz, S. Peng, and et. al., "Reactive temporal logic-based planning and control for interactive robotic tasks," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 108–12 115.