Guiding reinforcement learning using constrained uncertainty-aware movement primitives

Abhishek Padalkar[1], **Freek Stulp**[1], **Gerhard Neumann**[2], **João Silvério**[1]
[1] German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchener Str. 20, 82234 Weßling, Germany.
[2] Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany.

Abstract:
Guided reinforcement learning (RL) presents an effective approach for robots to acquire skills efficiently, directly in the real world environments. Recent works indicate that incorporating hard constraints into RL can expedite the learning of manipulation tasks, enhance safety, and reduce the complexity of the reward function. In parallel, learning from demonstration (LfD) using movement primitives is a well-established method for initializing RL policies. In this paper, we propose a constrained, uncertainty-aware movement primitive representation that leverages both demonstrations and hard constraints to guide RL. By integrating hard constraints, our approach aims to facilitate safer and sample-efficient learning, as the robot is not required to violate these constraints during the learning process. At the same time, demonstrations are employed to offer a baseline policy that supports exploration. Our method enhances state-of-the-art techniques by introducing a projector that enables state-dependent noise derived from demonstrations while ensuring that the constraints are respected throughout training. Collectively, these elements contribute to safe and efficient learning alongside streamlined reward function design. We validate our framework through an insertion task involving a torque-controlled, 7-DoF robotic manipulator.

Keywords: Learning from Demonstrations, Reinforcement Learning, Constrained Learning, Guided Learning

1 Introduction

Learning from Demonstration (LfD) [1] has proven to be an effective method for motion generation, enabling a robot to imitate and adapt the demonstrated motions. Various architectures have been developed, including Dynamic Movement Primitives (DMPs) [2], Probabilistic Movement Primitives (ProMPs) [3], and Kernelized Movement Primitives (KMPs) [4], which effectively address common real-world challenges such as generalizing to new situations and avoiding obstacles. However, these methods often struggle in dynamic environments where demonstrations inadequately represent task dynamics, particularly during contact tasks. Collaborative robots aim to mitigate these challenges by employing impedance control to remain compliant while in contact, thus reacting to the inaccuracies caused by kinematics and dynamics. However, learning a robust LfD policy that can adapt to such uncertainties remains a significant challenge.

Reinforcement Learning (RL) addresses this challenge by training a reactive policy that considers the current state of both the robot and its environment. However, the necessity of a large number of trials, coupled with concerns about robot safety, presents a significant barrier for RL to be widely applicable in robotics. Transfer learning attempts to overcome this by learning a policy in a simulation and then applying it to the robot, yet it is still constrained by the sim-to-real gap [6].

Workshop on Safe and Robust Robot Learning for Operation in the Real World (SAFE-ROL) at CoRL 2024.

A framework proposed by Padalkar et al. [7] allows for guided RL, enabling tasks to be learned directly on the real robot. In this approach, available task knowledge is represented as constraints, facilitating effective policy search. Nonetheless, while this method is promising, the manual modelling of constraints can be challenging, particularly in complex tasks that involve contacts.

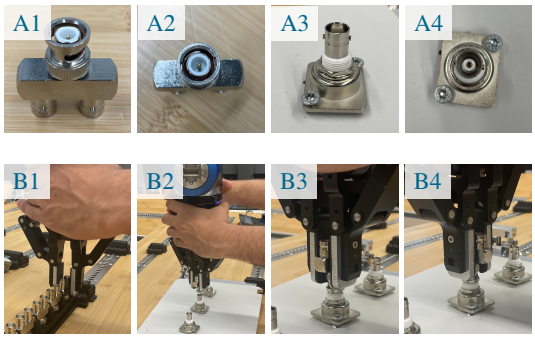

Figure 1: Figure shows the BNC connector assembly task from NIST assembly benchmark [5]. A1 to A4 show the male and female BNC connectors. B1 to B4 show human demonstrating the task by hand-guiding the robot. An LfD trajectory learned from the demonstrations was not able to solve the task as it does not model the contact dynamics and uncertainties in the kinematics.

To address the above-mentioned challenges, we propose to learn a nominal policy together with state-dependent exploration noise, from human demonstrations. Specifically, we introduce a novel movement primitive representation, Linearly Constrained Null-space Kernelized Movement Primitives (LC-NS-KMP), where a non-parametric imitation learning framework generates motion while adhering to the linear constraints on the state of the robot, simultaneously providing a *null-space* projector that allows the actions generated by the RL policy to modify the mean behavior of the imitation learning policy. The derived projector modifies the mean behavior of the LfD policy in accordance with the variance in the demonstrations. Consequently, the same null-space action will result in larger deviations in states where the variance in the demonstrations is higher. Projector learns the variance from the demonstrations, facilitating state-dependant eploration noise. We use this behavior for state-based exploration in RL while ensuring the safety of the robot by respecting the constraints on the state of the robot.

To fully demonstrate the capabilities of LC-NS-KMPs, we selected the BNC connector assembly task from the NIST assembly task board 1 [5], illustrated in Fig. 1. This task presents significant challenges, as it requires precise insertion of the connector while maintaining compliance to prevent damage to the components. Following the insertion, a complex series of translations and rotations are necessary to lock the connector in position. Our method is well-suited for such tasks because it 1) allows for the specification of constraints that ensure safe operation during state space exploration, and 2) guarantees uncertainty-aware, state-dependent exploration for reinforcement learning, which helps avoid unnecessary exploration in the low-variance regions of the motion.

2 Methodology

2.1 Background

Kernelized movement primitives (KMP). Huang et al. [4] presented an approach to learn probabilistic trajectories from demonstrations called Kernelized Movement Primitives (KMP). Consider M demonstrations $D = \{\{\boldsymbol{s}_{n,m}, \boldsymbol{\eta}_{n,m}\}_{n=1}^{N}\}_{m=1}^{M}$ where N is the length of a trajectory comprised of state $\boldsymbol{s}$ and corresponding output $\boldsymbol{\eta}$. A probabilistic policy can be learned from these demonstrations using GMM such that, $\mathcal{P}(s, \boldsymbol{\eta}) \sim \sum_{c=1}^{C} p_c \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$, where, p_c, $\boldsymbol{\mu}_c$,and $\boldsymbol{\Sigma}_c$ are the prior probability, mean and variance of the c^{th} Gaussian. We can employ GMR to obtain reference trajectory $T_r = \{\hat{\boldsymbol{\mu}}_n, \hat{\boldsymbol{\Sigma}}_n\}_{n=1}^{N}$ from above learned GMM. At the same time, a parametric trajectory can also be learned from the same demonstrations,

$$\boldsymbol{\eta}(\boldsymbol{s}) = \boldsymbol{\Theta}(\boldsymbol{s})^{\top}\mathbf{w}, \quad \boldsymbol{\Theta}(\boldsymbol{s}) = \begin{bmatrix} \boldsymbol{\varphi}(\boldsymbol{s}) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varphi}(\boldsymbol{s}) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \boldsymbol{\varphi}(\boldsymbol{s}) \end{bmatrix}, \tag{1}$$

where matrix $\boldsymbol{\Theta} \in \mathbb{R}^{\mathcal{B}\mathcal{O}\times\mathcal{O}}$, weight vector $\mathbf{w} \in \mathbb{R}^{\mathcal{B}\mathcal{O}}$, with $\boldsymbol{\varphi}(\boldsymbol{s})$ being a $\mathcal{B}$-dimensional basis function. Consider weights $\mathbf{w}$ drawn from $\mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)$, hence we can write $\boldsymbol{\eta}(\boldsymbol{s}) \sim \mathcal{N}(\boldsymbol{\Theta}(\boldsymbol{s})^\top \boldsymbol{\mu}_{\mathbf{w}}, \boldsymbol{\Theta}(\boldsymbol{s})^\top \boldsymbol{\Sigma}_{\mathbf{w}} \boldsymbol{\Theta}(\boldsymbol{s}))$. Huang et al. [4] proposed to minimize the KL-divergence between above-mentioned two Gaussian distributions, represented by T_r and $\mathcal{N}(\boldsymbol{\Theta}(\boldsymbol{s})^\top \boldsymbol{\mu}_{\mathbf{w}}, \boldsymbol{\Theta}(\boldsymbol{s})^\top \boldsymbol{\Sigma}_{\mathbf{w}} \boldsymbol{\Theta}(\boldsymbol{s}))$ leading to a *mean minimization subproblem* with cost function

$$\underset{\boldsymbol{\mu}_w}{\textbf{argmin}} \quad \sum_{n=1}^{N} \frac{1}{2} (\boldsymbol{\Theta}^\top(\boldsymbol{s}_n)\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_n)^\top \hat{\boldsymbol{\Sigma}}_n^{-1} (\boldsymbol{\Theta}^\top(\boldsymbol{s}_n)\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_n) + \frac{1}{2}\lambda \boldsymbol{\mu}_w^\top \boldsymbol{\mu}_w. \tag{2}$$

The prediction of a KMP is given by $\mathbb{E}(\boldsymbol{\eta}(\boldsymbol{s})) = \boldsymbol{k}^*(\boldsymbol{K} + \lambda\boldsymbol{\Sigma})^{-1}\boldsymbol{\mu}$, where $\boldsymbol{\mu} = [\hat{\boldsymbol{\mu}}_1^\top, \hat{\boldsymbol{\mu}}_2^\top, \ \dots, \hat{\boldsymbol{\mu}}_N^\top]$, $\boldsymbol{\Sigma} = \text{blockdiag}(\hat{\boldsymbol{\Sigma}}_1, \hat{\boldsymbol{\Sigma}}_2, \ \dots, \hat{\boldsymbol{\Sigma}}_N)$, and $\boldsymbol{k}^*$ and $\boldsymbol{K}$ are kernel matrices obtained after applying kernel treatment to the basis functions, which will be discussed in detail in Section 2.2. It should be noted that in this paper, we only focus on the mean minimization subproblem as our goal is to extract a policy that a robot can track.

Linearly-constrained KMP. Huang and Caldwell [8] formulated a linearly constrained imitation learning framework which incorporates linear inequality constraints on the state of the robot, and applied the same method to minimize the KL-divergence between two distributions as Huang et al. [4], to obtain a *constrained* mean minimization subproblem

$$\begin{aligned} \underset{\boldsymbol{\mu}_w}{\textbf{argmin}} \quad & \sum_{n=1}^{N} \frac{1}{2} (\boldsymbol{\Theta}^\top(\boldsymbol{s}_n)\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_n)^\top \hat{\boldsymbol{\Sigma}}_n^{-1} (\boldsymbol{\Theta}^\top(\boldsymbol{s}_n)\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_n) + \frac{1}{2}\lambda \boldsymbol{\mu}_w^\top \boldsymbol{\mu}_w \\ \textbf{s.t.} \quad & \boldsymbol{g}_{n,f}^\top \boldsymbol{\eta}(\boldsymbol{s}_n) \geq c_{n,f}, \forall f \in \{1, 2, \dots, F\}, \forall n \in \{1, 2, \dots, N\}, \end{aligned} \tag{3}$$

where F is the number of constraints imposed on a state. The mean prediction of LC-KMP is given by

$$\mathbb{E}(\boldsymbol{\eta}(\boldsymbol{s}^*)) = \boldsymbol{k}^*(\boldsymbol{K} + \lambda\boldsymbol{\Sigma})^{-1}\boldsymbol{\mu} + \boldsymbol{k}^*(\boldsymbol{K} + \lambda\boldsymbol{\Sigma})^{-1}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}\boldsymbol{\alpha}, \tag{4}$$

where,

$$\begin{aligned} \boldsymbol{G}_n &= [\boldsymbol{g}_{n,1}\ \boldsymbol{g}_{n,2}\ \boldsymbol{g}_{n,3}\ \dots\ \boldsymbol{g}_{n,F}], \forall n \in \{1, 2, 3, ..., N\}, \\ \bar{\boldsymbol{G}} &= \text{blockdiag}(\boldsymbol{G}_1,\ \boldsymbol{G}_2,\ \boldsymbol{G}_3,\ \dots\ \boldsymbol{G}_N), \\ \boldsymbol{\alpha} &= [\alpha_{1,1},\ \alpha_{1,2},\ \dots,\ \alpha_{1,F}\ \dots\ \dots\ \alpha_{N,1},\ \alpha_{N,2},\ \dots,\ \alpha_{N,F}]. \end{aligned}$$

The Lagrange multiplier vector $\boldsymbol{\alpha}$ is obtained by solving a convex optimization problem [8]. Prediction given by Eq. (4) respects the constraints defined in Eq. (3).

2.2 LC-NS-KMP formulation

In this paper, we derive an unified method which combines null-space modifier for KMPs proposed by Silvério and Huang [9] and linear constraints proposed by Huang and Caldwell [8]. Combining the desirable properties of these methods, our framework allows RL to modulate a mean trajectory predicted by KMPs adhering to linear constraints and variance in the demonstrations. It helps RL conduct an effective search by modulating the exploration noise in accordance with the variance and constraints.

We start from the same constrained mean optimization problem as Eq. (3), and introduce an additional cost term $\frac{1}{2}\beta(\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_w)^\top(\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_w)$ which results in a *soft* null space projector that modifies the mean trajectory (see [9] for details), to obtain,

$$\begin{aligned} \underset{\boldsymbol{\mu}_w}{\textbf{argmin}} \quad & \sum_{n=1}^{N} \frac{1}{2} (\boldsymbol{\Theta}^\top(\boldsymbol{s}_n)\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_n)^\top \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{\Theta}^\top(\boldsymbol{s}_n)\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_n) \\ & + \frac{1}{2}\lambda \boldsymbol{\mu}_w^\top \boldsymbol{\mu}_w + \frac{1}{2}\beta(\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_w)^\top(\boldsymbol{\mu}_w - \hat{\boldsymbol{\mu}}_w), \\ \textbf{s.t.} \quad & \boldsymbol{g}_{n,f}^\top \boldsymbol{\eta}(\boldsymbol{s}_n) \geq c_{n,f}, \forall f \in \{1, 2, \dots, F\}, \forall n \in \{1, 2, \dots, N\}. \end{aligned} \tag{5}$$

The term $\frac{1}{2}\lambda\boldsymbol{\mu}_w^\top\boldsymbol{\mu}_w$ regularizes the solution and the cost term $\frac{1}{2}\beta(\boldsymbol{\mu}_w-\hat{\boldsymbol{\mu}}_w)^\top(\boldsymbol{\mu}_w-\hat{\boldsymbol{\mu}}_w)$ inspired from [9] keeps the solution close to a desired one. Similarly to [8], we propose to solve Eq. (3) by introducing Lagrange multipliers $\alpha_{n,f}\geq 0$, with the Lagrange function

$$\begin{aligned}L(\boldsymbol{\mu}_w,\alpha)=&\sum_{n=1}^{N}\frac{1}{2}(\boldsymbol{\Theta}^\top(\boldsymbol{s}_n)\boldsymbol{\mu}_w-\hat{\boldsymbol{\mu}}_n)^\top\hat{\boldsymbol{\Sigma}}_n^{-1}(\boldsymbol{\Theta}^\top(\boldsymbol{s}_n)\boldsymbol{\mu}_w-\hat{\boldsymbol{\mu}}_n)+\frac{1}{2}\lambda\boldsymbol{\mu}_w^\top\boldsymbol{\mu}_w\\&+\frac{1}{2}\beta(\boldsymbol{\mu}_w-\hat{\boldsymbol{\mu}}_w)^\top(\boldsymbol{\mu}_w-\hat{\boldsymbol{\mu}}_w)-\sum_{n=1}^{N}\sum_{f=1}^{F}\alpha_{n,f}(\mathbf{g}_{n,f}^\top\boldsymbol{\Theta}(\boldsymbol{s}_n)^\top\boldsymbol{\mu}_w-c_{n,f}).\end{aligned}\tag{6}$$

For a desired output $\boldsymbol{\xi}=\hat{\boldsymbol{\Phi}}^\top\hat{\boldsymbol{\mu}}_w$, we can estimate the optimal weight vector $\hat{\boldsymbol{\mu}}_w$ given the target trajectory $\boldsymbol{\xi}$, using the right pseudo-inverse of $\hat{\boldsymbol{\Phi}}^\top$, hence, $\hat{\boldsymbol{\mu}}_w=\hat{\boldsymbol{\Phi}}(\hat{\boldsymbol{\Phi}}^\top\hat{\boldsymbol{\Phi}})^{-1}\boldsymbol{\xi}$.

With further simplification, we obtain,

$$\tilde{L}(\boldsymbol{\alpha})=\boldsymbol{\alpha}^\top\bar{\boldsymbol{G}}^\top\boldsymbol{\Sigma}\boldsymbol{A}\mathcal{A}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}\boldsymbol{\alpha}+(2\boldsymbol{\mu}^\top\boldsymbol{A}\mathcal{A}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}-\beta\boldsymbol{\xi}^\top(\hat{\boldsymbol{\Phi}}^\top\hat{\boldsymbol{\Phi}})^{-1}\hat{\boldsymbol{\Phi}}^\top\boldsymbol{\Phi}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}+\bar{\boldsymbol{C}}^\top)\boldsymbol{\alpha}+const,\tag{7}$$

and

$$\mathbb{E}(\boldsymbol{\eta}(\boldsymbol{s}^*))=\boldsymbol{\Theta}(\boldsymbol{s}^*)(\boldsymbol{\Phi}\boldsymbol{A}\boldsymbol{\mu}+\boldsymbol{\Phi}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}\boldsymbol{\alpha}+\frac{\beta}{\gamma}(\boldsymbol{I}-\boldsymbol{\Phi}\boldsymbol{A}\boldsymbol{\Phi}^\top)\hat{\boldsymbol{\Phi}}(\hat{\boldsymbol{\Phi}}^\top\hat{\boldsymbol{\Phi}})^{-1}\boldsymbol{\xi}),\tag{8}$$

respectively, where $\gamma=\lambda+\beta$, and $\bar{\boldsymbol{C}}=[\boldsymbol{C}_1^\top\ \boldsymbol{C}_2^\top\ \dots\ \boldsymbol{C}_N^\top]^\top$ with $\boldsymbol{C}_n=[c_{n,1}\ c_{n,2}\ \dots\ c_{n,F}]^\top,\forall n\in\{1,2,\dots,N\}$.

Huang et al. [4] proposed to kernelize the above equation using the kernel treatment, i.e. inner product of basis functions $\varphi(\boldsymbol{s}_i)$ and $\varphi(\boldsymbol{s}_j)$ is defined as $\varphi(\boldsymbol{s}_i)^\top\varphi(\boldsymbol{s}_j)=k(\boldsymbol{s}_i,\boldsymbol{s}_j)$, where $k(.,.)$ is a kernel function. With the kernel treatment, we can write

$$\tilde{L}(\boldsymbol{\alpha})=\boldsymbol{\alpha}^\top\bar{\boldsymbol{G}}^\top\boldsymbol{\Sigma}\boldsymbol{A}\mathcal{A}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}\boldsymbol{\alpha}+(2\boldsymbol{\mu}^\top\boldsymbol{A}\mathcal{A}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}-\beta\boldsymbol{\xi}^\top\underline{\boldsymbol{K}}^{-1}\hat{\boldsymbol{K}}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}+\bar{\boldsymbol{C}}^\top)\boldsymbol{\alpha}+const,\tag{9}$$

$$\mathbb{E}(\boldsymbol{\eta}(\boldsymbol{s}^*))=\boldsymbol{k}^*\boldsymbol{A}\boldsymbol{\mu}+\boldsymbol{k}^*\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}\boldsymbol{\alpha}+\frac{\beta}{\gamma}(\hat{\boldsymbol{k}}^*-\boldsymbol{k}^*\boldsymbol{A}\hat{\boldsymbol{K}})\underline{\boldsymbol{K}}^{-1}\boldsymbol{\xi},\tag{10}$$

with $\boldsymbol{A}=(\boldsymbol{K}+\lambda\boldsymbol{\Sigma})^{-1}$, and $\mathcal{A}=-\frac{1}{2}\boldsymbol{K}\boldsymbol{\Sigma}^{-1}\boldsymbol{K}-\frac{\gamma}{2}\boldsymbol{K}$, where,

$$\boldsymbol{k}^*=[\boldsymbol{k}(\boldsymbol{s}^*,\boldsymbol{s}_1),\dots,\boldsymbol{k}(\boldsymbol{s}^*,\boldsymbol{s}_N)],\tag{11}$$

$$\boldsymbol{K}=\begin{bmatrix}\boldsymbol{k}(\boldsymbol{s}_1,\boldsymbol{s}_1) & \dots & \boldsymbol{k}(\boldsymbol{s}_1,\boldsymbol{s}_N)\\ \vdots & \ddots & \vdots\\ \boldsymbol{k}(\boldsymbol{s}_N,\boldsymbol{s}_1) & \dots & \boldsymbol{k}(\boldsymbol{s}_N,\boldsymbol{s}_N))\end{bmatrix},\boldsymbol{k}(\boldsymbol{s}_i,\boldsymbol{s}_j)=k(\boldsymbol{s}_i,\boldsymbol{s}_j)\boldsymbol{I},\tag{12}$$

$$\underline{\boldsymbol{K}}=\hat{\boldsymbol{\Phi}}^\top\hat{\boldsymbol{\Phi}},\hat{\boldsymbol{K}}=\boldsymbol{\Phi}^\top\hat{\boldsymbol{\Phi}},\hat{\boldsymbol{k}}=\boldsymbol{\Phi}(\boldsymbol{s}^*)^\top\hat{\boldsymbol{\Phi}}.\tag{13}$$

We substitute $\mathcal{B}_1=\bar{\boldsymbol{G}}^\top\boldsymbol{\Sigma}\boldsymbol{A}\mathcal{A}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}$, and $\mathcal{B}_2=2\boldsymbol{\mu}^\top\boldsymbol{A}\mathcal{A}\boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}-\beta\boldsymbol{\xi}^\top\underline{\boldsymbol{K}}^{-1}\hat{\boldsymbol{K}}(-\boldsymbol{A})\boldsymbol{\Sigma}\bar{\boldsymbol{G}}+\bar{\boldsymbol{C}}^\top$, in Eq. (9) to obtain a quadratic function

$$\begin{aligned}&\tilde{L}(\boldsymbol{\alpha})=\boldsymbol{\alpha}^\top\mathcal{B}_1\boldsymbol{\alpha}+\mathcal{B}_2\boldsymbol{\alpha},\\&\textbf{s.t.}\quad\boldsymbol{\alpha}\geq 0.\end{aligned}\tag{14}$$

Here $\boldsymbol{A}\mathcal{A}\boldsymbol{A}=(\boldsymbol{A}\mathcal{A}\boldsymbol{A})^\top\preccurlyeq 0$ and $-\boldsymbol{A}=-\boldsymbol{A}^\top\preccurlyeq 0$, hence Eq. (14) presents a classical quadratic optimization problem with linear constraints. After computing the value of $\boldsymbol{\alpha}$ with quadratic programming, Eq. (10) can be used to make constrained predictions while taking into account modulations generated by $\boldsymbol{\xi}$.

2.3 Properties of LC-NS-KMP

We evaluated the properties of LC-NS-KMP using synthetically generated 2D time trajectories, shown in Fig. 2 (A1), alongside the learned Gaussian Mixture Model (GMM). We use GMM/GMR method for generating the reference trajectories and covariances in all of our experiments.

Fig. 2 (A2) illustrates the impact of various null-space actions $\boldsymbol{\xi}$ applied at t=3.2 on the trajectories. Despite the local modulation in the trajectory, smoothness is preserved while respecting the linear

inequality constraints defined in LC-NS-KMP, similar to the approach of Huang and Caldwell [8]. Finally, Fig. 2 (A3) shows trajectories generated using Eq. (8), where $\boldsymbol{\xi}$ is randomly sampled from a normal distribution at each time step. $\boldsymbol{\xi}$ modulates the trajectory in accordance with the variance in the demonstrations and hence it demonstrates the uncertainty aware exploration through null-space actions. The modulated trajectory also satisfies the constraints despite the noise amplitude which leads to the safe exploration in RL.

2.4 RL with LC-NS-KMP

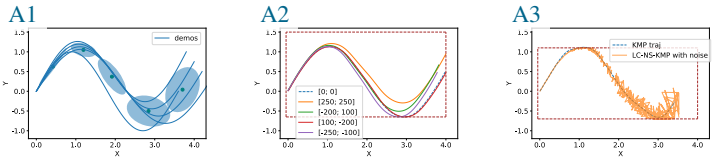

Figure 2: LC-NS-KMP properties: (A1) shows the demonstrations and the learned GMM; (A2) shows the modulations due to different $\boldsymbol{\xi}$ adhering to the linear constraints; (A3) shows the effect of randomly sampled $\boldsymbol{\xi}$.

In our RL framework, we use null-space actions $\boldsymbol{\xi}$ obtained from a RL policy $\pi(\boldsymbol{\xi}|\mathbf{s})$, to introduce modulations in the LfD trajectory learned from the demonstrations. LC-KMP [8] in Eq. (4) predicts a trajectory which respects the linear inequality constraints defined in Eq. (3). Our proposed method LC-NS-KMP in Eq. (10) allows modifications in the prediction using null-space action $\boldsymbol{\xi}$, whose magnitude depends on the variance in the demonstrations, while respecting the constraints in Eq. (3). This important property allows us to conduct efficient and safe RL search using null-space actions. Particularly, we obtain null-space actions from a RL policy $\pi(\boldsymbol{\xi}|\mathbf{s})$ modifying the prediction for further refinement as

$$\mathbb{E}(\boldsymbol{\eta}(\boldsymbol{s}^*)) = \boldsymbol{k}^* \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{k}^* \boldsymbol{A}\boldsymbol{\Sigma}\bar{\boldsymbol{G}}\boldsymbol{\alpha} + \frac{\beta}{\gamma}(\hat{\boldsymbol{k}} - \boldsymbol{k}\boldsymbol{A}\hat{\boldsymbol{k}})\pi(\boldsymbol{\xi}|\mathbf{s}). \tag{15}$$

3 Evaluation

3.1 Experiments in simulation

We evaluate the performance of our proposed framework against a baseline where an agent learns a residual RL policy to adapt the mean LfD trajectory. For this evaluation, we developed a simulation involving a robot that navigates a 2D environment with the primary objective of reaching a goal position while passing through a narrow passage and a secondary objective of avoiding an obstacle in its path. The setup for the simulation is depicted in Fig. 3.

We selected KMP, as described in Section 2.1, as the baseline LfD method, which produces the necessary time-based trajectory $p_t^{kmp} = \mathbb{E}(\boldsymbol{\eta}(t))$ to reach the goal, along with a residual RL policy $\pi_{res}(\Delta p_t|t)$ that modifies mean trajectory p_t for avoiding the obstacle. The robot follows the resultant 2D pose $p_t = p_t^{kmp} + \Delta p_t$ derived from the combined output of the policies.

We then compare the baseline residual RL solution to our LC-NS-KMP-RL algorithm outlined in Eq. (15), where an RL policy $\pi(\boldsymbol{\xi}|t)$ generates null space actions $\boldsymbol{\xi}$ that modify the trajectory using null-space projector in LC-NS-KMP. In both cases, the reward function for the robot is given by,

$$r_t = r_a + r_o + r_T, \quad r_a = -5\delta\boldsymbol{p}_t^\top \delta\boldsymbol{p}_t, \tag{16}$$

$$r_o = \begin{cases} -100(0.04 - \boldsymbol{d}_t), & \text{if } \boldsymbol{d}_t \leq 0.04 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

$$r_T = \begin{cases} 200 & \text{at terminal step } T \text{ if successful} \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

where $\boldsymbol{d}_t$ is the distance of the robot from the obstacle, the terminal reward r_T is given if the episode terminates successfully, r_o is obstacle avoidance cost, and r_a is the action cost.

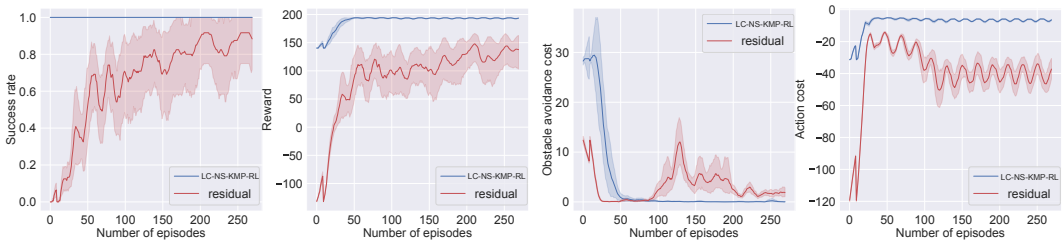

Figure 4: Comparison of the performance of residual RL with LfD to LC-NS-KMP-RL.

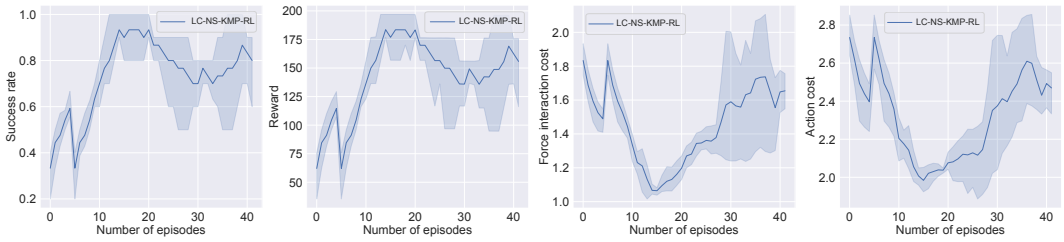

Figure 5: Performance of LC-NS-KMP-RL on BNC connector assembly task. The robot learns to solve the task in 15 episodes, simultaneously minimizing the interaction force.

The episode is considered successful if the robot reaches the goal within 400 time steps. Conversely, it is deemed unsuccessful if the robot becomes blocked in the narrow passage for 20 or more time steps, or if the maximum limit of 400 time steps is reached without achieving the goal. As illustrated in Fig. 3, trajectories were demonstrated to the robot. In each scenario, the mean trajectory consistently intersects with the obstacle. Consequently, the reinforcement learning policy must learn to navigate the environment by avoiding the obstacle while adhering to the hard constraints.

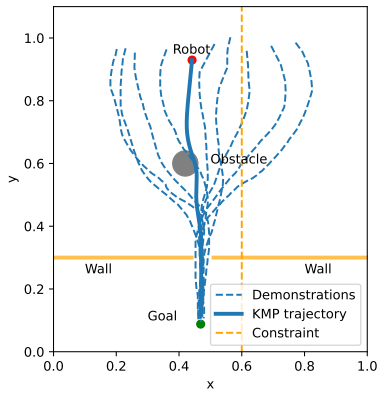

Figure 3: Simulated 2D environment where the robot navigates through a narrow passage to reach the goal. A trajectory for navigation can be learned from demonstration. Then an RL policy learns to avoid the obstacle in the path of the robot.

The performance comparison between residual RL with LfD and the LC-NS-KMP-RL framework is shown in Fig. 4. LC-NS-KMP-RL quickly achieves the primary objective while minimizing both obstacle avoidance and action costs. In contrast, residual learning takes much longer due to isotropic noise used for exploration, which often causes the robot to become stuck in the narrow passage. Conversely, LC-NS-KMP-RL modifies trajectories based on the variance in demonstrations, reducing unnecessary exploration in low-variance region near the narrow passage. Additionally, constraints in LC-NS-KMP-RL keep the robot within a safe zone, enhancing its overall performance.

3.2 Experiments on real robot

To evaluate our framework on a real robot, we selected a task from the NIST assembly benchmark 1 [5] involving the plugging of a BNC connector. This task is particularly challenging and requires multiple manipulation strategies for different phases: inserting, aligning, and locking the connector. The strategy learned from demonstration alone is insufficient to complete the task, and relying

purely on RL would necessitate an impractically large number of trials. Our approach utilizes state-dependent guided exploration, allowing the robot to explore the state-action space selectively, where necessary. Additionally, linear inequality constraints reduce the state space and ensure the robot's safety.

Fig. 1 illustrates the experimental setup. Images (A1) and (A2) present side and top views of the BNC male connector, while (A3) and (A4) show the respective views for the BNC female connector. Images (B1) to (B4) depict the stages of picking, aligning, inserting, and locking the BNC male connector, respectively, while human is demonstrating the task through hand-guided motion.

Demonstrations $D = \{\{t_{n,m}, \boldsymbol{p}_{n,m}\}_{n=1}^{400}\}_{m=1}^{5}$ were provided for the aligning, inserting, and locking phases. The 6D pose of the robot end-effector $\boldsymbol{p} = [x, y, z, a_z, a_y, a_x]^\top$ (where a_x, a_y, a_z represent the Euler angles) is measured in the target frame, which is the end-effector position when the connector is locked. For practical purposes, this target frame is assumed to be the last frame of each successful demonstration.

A learned KMP from these demonstrations was tested but found inadequate for task completion due to various factors, including kinematic and dynamic uncertainty and the KMP's inability to effectively capture the contact dynamics involved in the task.

We then formulated a LC-NS-KMP-RL problem and a RL policy $\pi(\boldsymbol{\xi}|\mathbf{s}_t)$ was learned to complete the task, where the state for RL policy $\mathbf{s}_t = [t;\ \boldsymbol{f}_t]$ with $\boldsymbol{f}_t$ being the 6D wrench measured at the center of compliance. We defined the linear inequality constraints in XY-plane so that RL exploration does not deviate too far from the alignment pose, and a constraint on the angular pose, such that the end-effector does not start rotation for locking before reaching a certain Z-position, which corresponds to the completion of the insertion phase,

$$\boldsymbol{g}_{n,1}^\top = [1,0,0,0,0,0], \quad c_{n,1} = 0.002, \quad \boldsymbol{g}_{n,2}^\top = [-1,0,0,0,0,0], \quad c_{n,2} = -0.002, \tag{19}$$

$$\boldsymbol{g}_{n,3}^\top = [0,1,0,0,0,0], \quad c_{n,3} = 0.002, \quad \boldsymbol{g}_{n,4}^\top = [0,-1,0,0,0,0], \quad c_{n,4} = -0.002, \tag{20}$$

$$\boldsymbol{g}_{n,5}^\top = [0,0,0,-1,0,0], \quad c_{n,5} = \begin{cases} -\pi/2 & \text{if} z < -0.002, \\ 0 & otherwise. \end{cases} \tag{21}$$

The reward function for the RL agent is given by,

$$r_t = -\frac{1}{2000}\delta\boldsymbol{\xi}_t^\top \delta\boldsymbol{\xi}_t - \frac{1}{600}\delta\boldsymbol{f}_t^\top \delta\boldsymbol{f}_t + r_T, \tag{22}$$

$$r_T = \begin{cases} 200 & \text{at terminal step } T \text{ if successful,} \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

Fig. 5 illustrates the overall performance of LC-NS-KMP-RL. The robot successfully learned to insert and lock the connector in under 15 episodes while significantly reducing force interactions with the environment—an important factor for ensuring the robot's long-term safe operation. Despite the sample efficient learning, learning agent shows a deteriorating behaviour over time. We plan to address this in our future work by introducing control framework with higher update rate and strict real time control loops.

4 Conclusion

In conclusion, our paper highlights the effectiveness of the LC-NS-KMP-RL framework in learning challenging manipulation tasks involving complex contacts directly on real robot. By integrating state-dependent guided exploration and linear inequality constraints, we were able to facilitate efficient learning and enhance the robot's operational safety. The ability of the robot to master the insertion and locking of the connector in fewer than 15 episodes underscores the potential of our approach to significantly reduce the time and effort required for complex manipulation tasks. Additionally, the reduction in force interactions with the environment indicates a pathway toward long-term reliability and safety in robotic operations. This work contributes to the ongoing development of adaptive robotic systems capable of performing intricate assembly tasks while prioritizing

safety and efficiency. Future work will focus on further refining this framework and exploring its applicability across a wider range of assembly challenges.

Acknowledgments

This work was partially funded by the DLR project "Factory of the Future Extended" and the European Union's Horizon Research and Innovation Programme under Grant 101136067 (INVERSE). We would also like to thank Antonin Raffin for the discussions on the off-policy RL algorithms and assisting with Stable Baselines 3 framework.

References

[1] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3: 297–330, 2020.

[2] S. Schaal. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.

[3] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann. Probabilistic movement primitives. *Advances in neural information processing systems*, 26, 2013.

[4] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell. Kernelized movement primitives. *International Journal of Robotics Research (IJRR)*, 38(7):833–852, 2019.

[5] K. Kimble, K. Van Wyk, J. Falco, E. Messina, Y. Sun, M. Shibata, W. Uemura, and Y. Yokokohji. Benchmarking protocols for evaluating small parts robotic assembly systems. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):883–889, 2020.

[6] E. Salvato, G. Fenu, E. Medvet, and F. A. Pellegrino. Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. *IEEE Access*, 9: 153171–153187, 2021.

[7] A. Padalkar, G. Quere, F. Steinmetz, A. Raffin, M. Nieuwenhuisen, J. Silvério, and F. Stulp. Guiding reinforcement learning with shared control templates. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 11531–11537. IEEE, 2023.

[8] Y. Huang and D. G. Caldwell. A linearly constrained nonparametric framework for imitation learning. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 4400–4406. IEEE, 2020.

[9] J. Silvério and Y. Huang. A non-parametric skill representation with soft null space projectors for fast generalization. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 2988–2994. IEEE, 2023.

[10] J. Luo, O. Sushkov, R. Pevceviciute, W. Lian, C. Su, M. Vecerik, N. Ye, S. Schaal, and J. Scholz. Robust multi-modal policies for industrial assembly via reinforcement learning and demonstrations: A large-scale study. *arXiv preprint arXiv:2103.11512*, 2021.

[11] Y. Narang, K. Storey, I. Akinola, M. Macklin, P. Reist, L. Wawrzyniak, Y. Guo, A. Moravanszky, G. State, M. Lu, et al. Factory: Fast contact for robotic assembly. *arXiv preprint arXiv:2205.03532*, 2022.

[12] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*, 2017.

[13] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

[14] J. Eßer, N. Bach, C. Jestel, O. Urbann, and S. Kerner. Guided reinforcement learning: A review and evaluation for efficient and effective real-world robotics [survey]. *IEEE Robotics & Automation Magazine (RAM)*, 30(2):67–85, 2023.

[15] W. Lian, T. Kelch, D. Holz, A. Norton, and S. Schaal. Benchmarking off-the-shelf solutions to robotic assembly tasks. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 1046–1053. IEEE, 2021.

[16] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems (RAS)*, 57(5):469–483, 2009.

[17] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *Proc. IEEE Intl Conf. on Robotics and Automation (ICRA)*, pages 6292–6299. IEEE, 2018.

[18] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, et al. Deep q-learning from demonstrations. In *Proc. AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[19] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[20] J. Ho and S. Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[21] J. Fu, K. Luo, and S. Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

[22] M. Alakuijala, G. Dulac-Arnold, J. Mairal, J. Ponce, and C. Schmid. Residual reinforcement learning from demonstrations. *arXiv preprint arXiv:2106.08050*, 2021.

[23] K. Pertsch, Y. Lee, Y. Wu, and J. J. Lim. Guided reinforcement learning with learned skills. *arXiv preprint arXiv:2107.10253*, 2021.