2024 29th International Conference on Automation and Computing (ICAC) | 979-8-3503-6088-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICAC61394.2024.10718791

Enhancing Learning from Demonstration with DLS-IK and ProMPs

Shibao Yang
Dept. Computer Science
University of York
York, United Kingdom
shibao.yang@york.ac.uk

Pengcheng Liu
Dept. Computer Science
University of York
York, United Kingdom
pengcheng.liu@york.ac.uk

Nick Pears
Dept. Computer Science
University of York
York, United Kingdom
nick.pears@york.ac.uk

***Abstract*—Learning from Demonstration (LfD) techniques are invaluable for capturing complex human behaviors for robotic arm manipulations, yet they frequently encounter challenges such as avoiding singularities and respecting joint limits when directly applied to robotic systems. These challenges often lead to mechanical non-compliance, inaccuracies, and increased computational demands during control method adjustments. To address these issues, this study introduces a robust approach that integrates Damped Least Squares Inverse Kinematics (DLS-IK) with Probabilistic Movement Primitives (ProMPs). By leveraging DLS-IK to generate kinematically feasible trajectories, and embedding these within the ProMPs framework, our method not only ensures mechanical compliance but also capitalizes on the probabilistic modeling capabilities of ProMPs. This synergy addresses a significant gap in traditional LfD applications—aligning human demonstrations with the mechanical constraints of robotics, independent of the demonstrator's expertise. Our integrated approach refines the LfD process, enabling the generation of precise, reliable, and mechanically compliant movements in robotic arms, thereby reducing the typical inaccuracies and computational burdens associated with conventional LfD methods.**

***Index Terms*—Inverse Kinematics, DLS, LfD, ProMPs, Robotic arm**

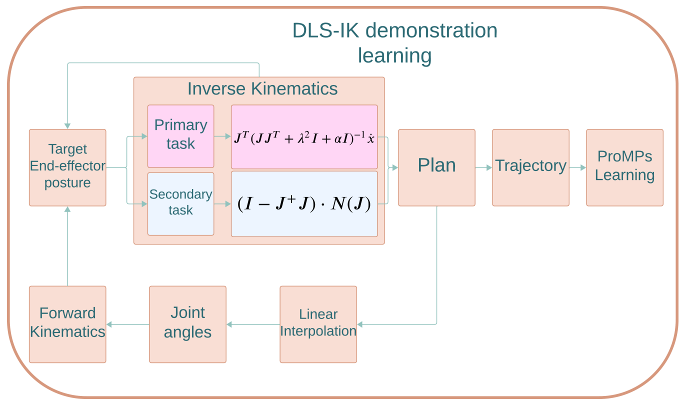

Fig. 1: The DLS-IK demonstration learning involves a structured process from the target end effector posture through forward and inverse kinematics for determining joint angles, ultimately culminating in the creation of a trajectory plan using ProMPs learning.

I. Introduction

The challenge of kinematic singularities is a prominent concern within the field of robotic manipulators [1], [2]. At such points, the robotic arm encounters configurations where it loses one or more degrees of freedom, complicating control over its movements. Traditionally, these issues are addressed using *Inverse Kinematics* (IK) solutions [3]–[5]. However, the implementation of *Learning from Demonstration* (LfD) techniques [6]–[8], where robots learn actions from human demonstrations, intensifies the difficulty of managing singularities. Human demonstrators often overlook the robotic arm's susceptibility to singularities and joint limits, which can lead to suboptimal guidance during demonstrations.

This study seeks to enhance LfD methods by addressing such inherent challenges when employing human demonstrations for training the robotic arm. Current LfD approaches may fail to consider the full mechanical constraints of robotics, resulting in increased computational complexity and suboptimal robotic performance. We introduce a novel integration of the *Damped Least Squares method for Inverse Kinematics* (DLS-IK, see Fig. 1) [9] with *Probabilistic Movement Primitives* (ProMPs) [10], [11], aimed at overcoming these limitations by ensuring both kinematically-feasible and mechanically-compliant trajectory learning.

The gap in traditional LfD methods lies in the alignment of human demonstrations with the mechanical and operational principles of robotic arms [12]. This misalignment often necessitates various ad-hoc adjustments and control methods, which compromise the accuracy of the outcomes and increase the system's computational burden. Our approach leverages the robust trajectory generation capabilities of DLS-IK to navigate the complexities of singularities and joint constraints effectively. Integrating these trajectories within the ProMPs framework utilizes the probabilistic modeling strengths of ProMPs, while ensuring that the learning process adheres to mechanical feasibility.

In summary, our contributions are: (1) An advanced DLS-IK method that not only controls velocity damping but also adheres to joint constraints through the null space principle of the robotic arm. (2) Incorporation of this refined DLS-IK approach into the ProMPs framework, which significantly enhances the robot's ability to learn and generate motion trajectories that are both mechanically compliant and efficient, effectively addressing singularity issues and joint limitations. (3) An investigation into the impact of the Jacobian condition number in the DLS-IK process on the stability and robustness of robotic arm movements, particularly near singularities. Our findings demonstrate improved stability and velocity control via local optimal parameter selection.

II. Related works

Inverse Kinematics (IK) is a fundamental technique in robotics and animation that involves determining the joint parameters required to position a robot's end effector, such as a gripper, at a desired location and orientation. The singularity problem, which arises when a robot's Jacobian matrix becomes ill-conditioned or singular, can be addressed using IK solutions [9], [13]. Among various IK methods, the DLS approach has proven effective in maintaining the stability of the robotic arm when operating near singularities. DLS introduces a damping factor that modifies the Jacobian matrix, enabling smooth and stable motion even in the presence of singularities.

Probabilistic Movement Primitives (ProMPs), an LfD framework, learns robot motion trajectories from human demonstrations. However, ProMPs heavily rely on the quality of the demonstrations [6], [14] and do not inherently address the singularity problem, which can lead to suboptimal or infeasible motions if the demonstrations fail to consider factors such as singularity avoidance and joint limits. Therefore, we propose integrating IK principles, specifically the DLS method, into the ProMPs framework to tackle the singularity issue and enhance the robustness and stability of the generated motions. By incorporating DLS-based IK solutions and considering joint limits during the learning and generation phases, our method seeks to improve the overall quality and feasibility of the generated robot motions.

III. Methods

A. Inverse Kinematics

Inverse kinematics is the inverse of forward kinematics, which ascertains the end effector's position from known joint parameters [15]. For robots with several degrees of freedom, it is common to employ the Jacobian matrix, which correlates the velocities of the robot's joints to the end effector's velocity. The IK challenge is often approached by calculating the Jacobian's inverse (or pseudo-inverse):

$$\dot{x} = J\dot{\theta} \tag{1}$$

where $\dot{\theta}$ denotes the vector of joint velocities, J is the Jacobian matrix, and $\dot{x}$ represents the Cartesian velocity of the end effector. This iterative method helps approximate joint angles for complex robotic arms.

B. Damped Least Squares

The Damped least Squares (DLS) method modifies the basic least squares solution by adding a damping term to improve stability and handle singular configurations [16]. The Jacobian matrix (J) of the robot, which relates changes in joint angles to changes in end effector position and orientation, plays a central role in this method.

Given a desired change in end effector position (Δx), the change in joint angles ($\Delta\theta$) can be approximated using the pseudo-inverse of the Jacobian:

$$\Delta\theta = J^{\dagger}\Delta x \tag{2}$$

However, directly using the pseudo-inverse can lead to large joint velocities in the vicinity of singularities. To mitigate this, the DLS method introduces a damping factor (λ), modifying the formula to:

$$\Delta\theta = (J^T J + \lambda^2 I)^{-1} J^T \Delta x \tag{3}$$

where J^T is the transpose of the Jacobian, I is the identity matrix, and λ is the damping factor, a small positive constant. This addition ensures that the solution is less sensitive to singularities and results in smoother motion.

Additionally, the relationship between the damping factor λ and the condition number of the Jacobian $\kappa(J)$ in the context of the DLS is critical for maintaining the stability of the robotic arm's movements, especially near singularities.

$$\kappa(J) = \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}} \tag{4}$$

where σ_{max} and σ_{min} are the maximum and minimum singular values of the Jacobian matrix J, respectively. High condition number of Jacobian ($\kappa(J)$ large) indicates that the Jacobian matrix is near singular or ill-conditioned. To reduce the influence of these instabilities, the larger λ provides more damping, which effectively regularizes the pseudo-inverse by reducing the contribution of directions associated with small singular values. This helps to prevent excessively large joint velocities and torques.

Low condition number of Jacobian ($\kappa(J)$ small) indicates that the Jacobian matrix is well-conditioned. A smaller λ allows the control to more precisely follow the desired end effector path or force application.

C. Controlling in the null space

Controlling in the null space refers to a technique used to achieve a specific task (the primary task), while simultaneously satisfying additional constraints or optimizing secondary objectives to achieve some secondary task [17]. The null space refers to the space of motions that don't move the end effector and hence don't affect the primary task.

In this approach, we first calculate the joint velocity ($\dot{q}_{primary}$) required for the primary task (e.g., reaching a target end effector position and orientation) using the DLS method. This involves modifying the pseudo-inverse calculation of the Jacobian to include a damping factor, which helps to prevent large joint velocities near singularities of the Jacobian. α is set to control the strength of the penalty for large joint velocities. A larger α value will result in smaller joint velocities and potentially lower energy consumption, but it might also slow down the convergence of the IK solution. The formula for $\dot{q}_{primary}$ using DLS is:

$$\dot{q}_{primary} = J^T(JJ^T + \lambda^2 I + \alpha I)^{-1}\dot{x} \tag{5}$$

The secondary task in IK minimizes the deviation of each joint's angle from the midpoint of its motion range, thereby calculating the joint velocity. To prevent this secondary task from affecting the primary task, the secondary joint velocity, $\dot{q}_{secondary}$, are projected onto the null space of the Jacobian related to the primary task. The null space comprises all joint velocities that do not influence the movement of the end effector [2]. Given the null space of the Jacobian ($N(J)$), the projection of $\dot{q}_{secondary}$ onto $N(J)$ is given by:

$$\dot{q}_{null} = (I - J^+J) \cdot \dot{q}_{secondary} \tag{6}$$

where I is the identity matrix. J^+J projects any vector onto the column space of J, and thus $I - J^+J$ projects onto the null space.

The final joint velocities $\dot{q}_{final}$ combine contributions from both tasks, prioritizing the primary task while incorporating the secondary task as much as possible without interference:

$$\dot{q}_{final} = \dot{q}_{primary} + \dot{q}_{null} \tag{7}$$

This ensures that the primary task's objectives are met first, and the secondary task's objectives are met within the degrees of freedom left by the primary task.

D. DLS-IK

The DLS-IK demonstration learning framework shown in Fig. 1 is a coherent structure that guides the robotic arm from an initial target end effector posture to the learned movements through ProMPs. It begins with the target end effector posture, which serves as the desired outcome of the demonstration learning. The forward kinematics module provides the necessary context by predicting the end effector's position from current joint angles.

Focusing on the inverse kinematics, the framework addresses two distinct tasks. Primary task (Equation (5)) employs the DLS approach to calculate joint velocities while maintaining stability and smoothness, especially near singularity.

Secondary task (Equation (6)) handles additional considerations that do not affect the primary goal, operating within the robot's null space. The secondary task ensures complementary movements that do not interfere with the primary task. These two tasks are then integrated into a comprehensive plan, as shown in Equation (7).

The resulting joint movements are interpolated linearly to generate a desired trajectory for the robot arm. Once the trajectory is defined, it serves as the foundation for ProMPs learning which encode the movement sequence into a probabilistic model, facilitating the robot's ability to learn smooth movements and adapt to new tasks and varying constraints. This integration allows the system to handle variability and uncertainty effectively. ProMPs represent movements as distributions over trajectories, enhancing adaptability and robustness in dynamic environments and shifting task objectives [18]. By utilizing the mean trajectory from the ProMPs, the robot can execute the learned movements with a balance of precision and flexibility, ensuring reliable performance across different scenarios.

The trajectory at any time t, $\mathbf{y}(t)$, is defined by:

$$\mathbf{y}(t) = \boldsymbol{\phi}(t)^T\mathbf{w} \tag{8}$$

where $\boldsymbol{\phi}(t)$ represents basis functions, such as Gaussian or polynomial, and $\mathbf{w}$ are weights encoding movement patterns. These weights are modeled as Gaussian distributed random variables for simplicity:

$$\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w) \tag{9}$$

This models the variability of movements, where $\boldsymbol{\mu}_w$ and $\boldsymbol{\Sigma}_w$ are the mean and covariance of the weights. The mean trajectory serves as the expected path that the robot will follow, ensuring reliable performance across different scenarios. Bayesian conditioning can further refine the movement by updating the weights based on new observations:

$$p(\mathbf{w}|\mathbf{y}_{\text{obs}}) \propto p(\mathbf{y}_{\text{obs}}|\mathbf{w})p(\mathbf{w}) \tag{10}$$

This conditioning process aids in synthesizing new movements by sampling from the $\mathbf{w}$ distribution and recalculating the trajectory with basis functions.

IV. Experiments

We present two experiments conducted to evaluate the efficacy of the proposed DLS-IK method in robotic arm manipulations. The first experiment, termed the singularity experiment, was designed around two distinct singularity scenarios to demonstrate how DLS-IK can effectively stabilize and smooth the manipulator's movements approaching singularity points. Additionally, this experiment provided insights into the relationship between the Jacobian condition number and the stability of the robot arm's movement, underlining the critical influence of mathematical conditions in robotic precision and control.

The second experiment focuses on integrating the ProMPs model to contrast the outcomes from DLS-IK demonstrations with those derived from human demonstrations. This comparison aims to identify and analyze the limitations inherent in using human demonstrations for robotic learning. The findings not only underscore the challenges of relying solely on human inputs but also validate the necessity of incorporating the DLS-IK method to enhance learning outcomes.

We employ the Robot Operating System (ROS) [19] framework and the MoveIt [20] library for planning and executing robotic arm movements. ROS is an open-source platform for building robot applications, while MoveIt provides a unified interface for performing motion tasks. We conduct the experiment on a computer equipped with an Intel i9-11900 processor and an NVIDIA 3050.

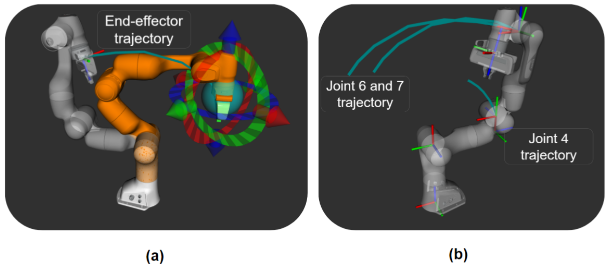

Fig. 2: (a) illustrates the operation sequence of a Franka robot arm, beginning with the initial stage (highlighted in orange) and progressing towards a state near to a singularity. (b) displays the motion trajectories of Joints 4, 6, and 7, which are influenced by the singularity.

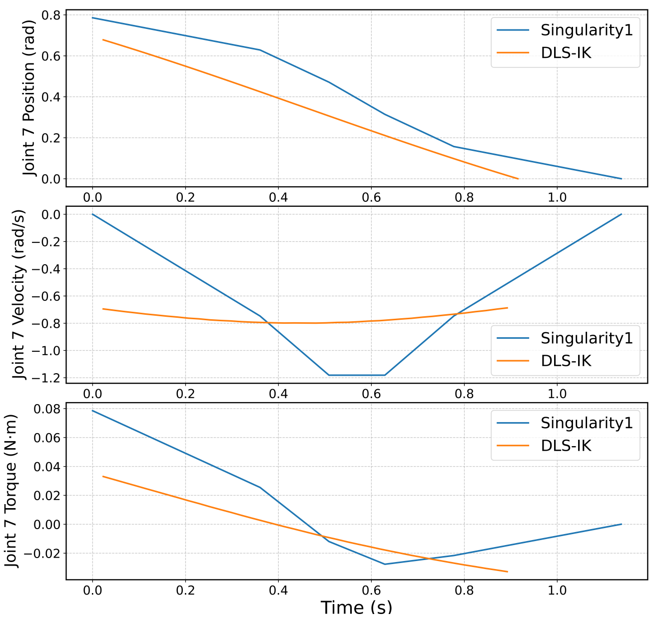

Fig. 3: The first singularity scenario shows the joint 7 position, velocity and torque.

A. Approaching a singularity

The experiment section evaluates the efficacy of the DLS-IK method in smoothing velocity profiles under singularity conditions by analyzing two specific scenarios. In the first scenario (Fig. 3), the joint velocity exhibit a sharp rise as the arm approaches singularity, with joint 7 notably reaching -1.0 rad/s. With DLS-IK, these velocities are effectively moderated to around -0.8 rad/s, demonstrating the method's capability for stable velocity control.

Torque profiles in the singularity context (Fig. 3) display small magnitudes with erratic fluctuations. In contrast, DLS-IK application results in more consistent torque trajectories, indicating improved control when near singularity.

The second scenario (Fig. 2 and 4) shows rapid velocity increases for joints 4, 6, and 7 without DLS-IK. The DLS-IK method effectively regulates the velocities of joints 6 and 7, which are observed to reach -0.6 rad/s and -1.4 rad/s,

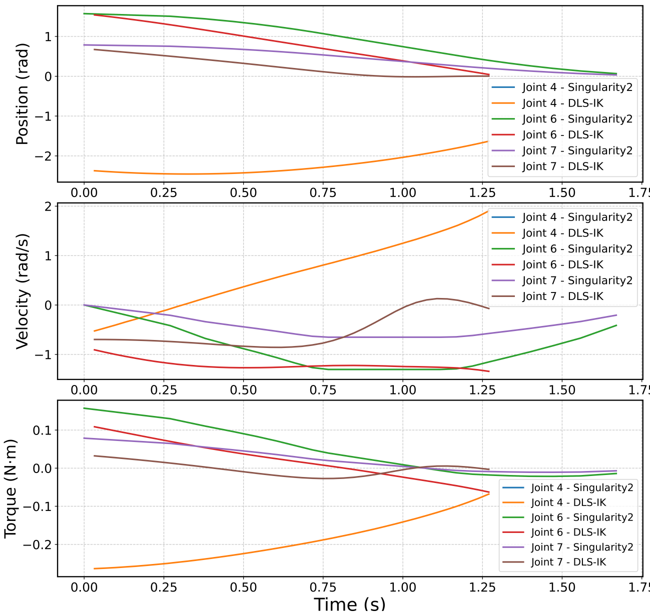

Fig. 4: The second singularity scenario shows the joint (4, 6 and 7) position, velocity and torque.

TABLE I: The first singularity scenario in terms of condition number of Jacobian.

Condition number of Jacobian	Less than 8 (%)	Less than 10 (%)
First singularity scenario	0%	100%
DLS-IK in first singularity scenario	20%	83%

respectively. Notably, even as joint 7's velocity increases past the 0.75-time mark, it exhibits a smooth ascension. Similarly, joint 4 maintains a steady rate of acceleration, contributing to the arm's overall movement stability.

Further analysis into the Jacobian condition number, shown in Tables I and II, suggests a correlation between lower condition numbers and more stable control. DLS-IK contributes to a higher incidence of condition numbers falling below critical thresholds, demonstrating its utility in achieving stable arm control near singularities.

B. Human demonstrations comparing with DLS-IK demonstrations

The figures above (Fig. 6, 7, and 8) illustrate the trajectories and velocity profiles for three distinct joints (Joint1, Joint3, and Joint5) which have significant variation we selected. These were derived from human demonstrations and DLS-IK demonstrations using ProMPs. Fig. 5 demonstrates the way of manually guiding the robotic arm from an initial to a final position to gather data for a human demonstration. DLS-IK is combined with the ProMPs model, following the methodology outlined in Fig. 1.

Figure 6 compares the outcomes of learning from human and DLS-IK demonstrations. For human demonstrations, the trajectory aligns well with the original, but the velocity profile shows marked discrepancies, particularly with reduced magni-

Fig. 5: An operator manually guiding the end effector of a Franka robot arm, indicated by the red arrow, to demonstrate the desired motion. This is essential for collecting human demonstrations for robotic training.

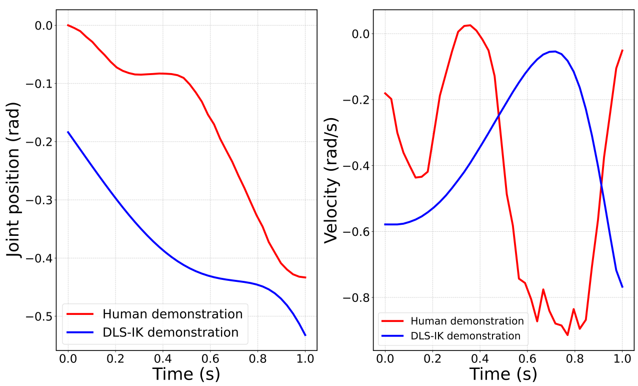

Fig. 6: Franka joint 1 position and velocity comparing human and DLS-IK demonstrations.

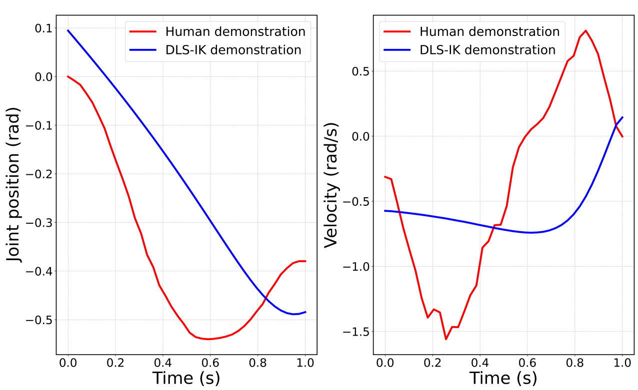

Fig. 7: Franka joint 3 position and velocity comparing human and DLS-IK demonstrations.

TABLE II: The second singularity scenario in terms of condition number of Jacobian.

Condition number of Jacobian	Less than 8 (%)	Less than 10 (%)
Second singularity scenario	0%	72%
DLS-IK in second singularity scenario	28%	83%

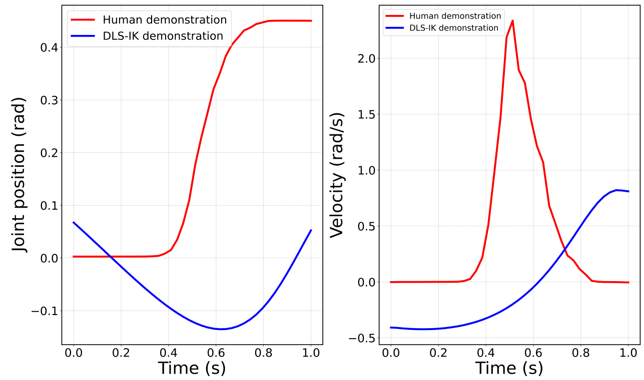

Fig. 8: Franka joint 5 position and velocity comparing human and DLS-IK demonstrations.

tudes and missing sharp peaks and valleys, indicating potential oversmoothing during the learning process. Conversely, the trajectory from the DLS-IK demonstration displays a smooth, continuous curve closely mirroring the input data, with a bell-shaped velocity profile that captures the dynamic characteristics of the motion, suggesting effective learning of fluid and natural-looking movements.

In Figure 7, the trajectory learned from human demonstrations is smoother than the original, possibly reflecting the generalization capabilities of the algorithm. However, the velocity profile from this model significantly flattens, failing to replicate the pronounced peaks and valleys observed in human motions, which suggests a loss of critical dynamic information. In contrast, the trajectory and velocity profile from the DLS-IK demonstration maintain a smooth, bell-shaped curve, affirming the ProMPs' efficiency in encapsulating the essential dynamics, thereby enabling the generation of seamless and graceful movements.

Figure 8 shows that while the trajectory from human demonstrations generally follows the shape of the demonstrated motion, the associated velocity profile diverges significantly, with peak velocities higher than those in the demonstration, hinting at an overestimation of the motion's speed. The velocity profile also demonstrates an oscillatory pattern absent in the original demonstration. In comparison, the trajectory and velocity profile from the DLS-IK demonstration are smooth and mostly symmetrical, although with slight asymmetry. Despite this, the overall fidelity of the velocity profile indicates successful learning of the motion dynamics by the ProMPs, enabling fluid and natural-looking movements.

C. Discussion

The implementation of DLS-IK has proven effective in managing the singularity challenges faced by robotic manipulators. As the manipulator nears a singularity, DLS-IK precisely moderates the joint velocities, ensuring they remain stable and controlled. This capability enhances the precision of the arm's movements and results in smoother velocity outputs, highlighting the direct link between the stability of the arm's movement and the condition number of the Jacobian—a lower condition number corresponds to increased stability and better control.

The ProMPs approach, while effectively learning and reproducing the overarching trajectories from human demonstrations, encounters difficulties with velocity profiles. The observed velocity inconsistencies point to the complexities of learning from the inherently variable and sometimes noisy human demonstration. Addressing these challenges may involve several strategies, such as preprocessing input data, diversifying the demonstration samples, fine-tuning the ProMP parameters, applying post-processing techniques, and working more closely with demonstrators for higher quality demonstrations. These actions underline the importance of careful and iterative refinement of the data to enhance the reliability of ProMP-based robotic learning.

Additionally, when ProMPs are informed by DLS-IK-generated demonstrations, they outperform those based on human demonstrations by better capturing the motion's smoothness and continuity. Smooth velocity profiles achieved across joints indicate that the learned motion primitives can emulate fluid and lifelike movements. These results shed light on the effectiveness of integrating ProMPs with DLS-IK, suggesting a method to replicate intricate robotic motions with enhanced fidelity and smoothness. Such smoothness in movement is vital for applications that require delicate and human-like interactions, thus demonstrating the dual advantage of DLS-IK: it not only improves robotic precision and stability but also empowers advanced learning methods like ProMPs to more accurately reflect the dynamics of human motion.

V. Conclusions

We have established a significant enhancement in LfD for robotic arm manipulations by integrating DLS-IK with ProMPs. This effectively overcomes the key challenges inherent in LfD methods, particularly those concerning singularities and joint limits.

Through the DLS-IK approach, we generate trajectories that were not only kinematically viable but also contributed to the smoothness and stability of robotic arm movements, especially near singular configurations. The incorporation of these trajectories into the ProMPs framework capitalizes on its probabilistic modeling strength, thereby reinforcing both the reliability and the mechanical soundness of the motion primitives derived from human demonstrations.

Looking ahead, tailoring the DLS-IK parameters for diverse robotic architectures and use cases could significantly extend the reach of this methodology. Further investigations might also refine the demonstration data collection from human, reducing noise and inconsistencies, where machine learning could play a role in streamlining the preprocessing of such data.

References

[1] D. Lee, W. Lee, J. Park, and W. K. Chung, "Task space control of articulated robot near kinematic singularity: forward dynamics approach," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 752–759, 2020.

[2] J. Manavalan and M. Howard, "Learning singularity avoidance," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6849–6854.

[3] S. Tittel, "Analytical solution for the inverse kinematics problem of the franka emika panda seven-dof light-weight robot arm," in *2021 20th International Conference on Advanced Robotics (ICAR)*. IEEE, 2021, pp. 1042–1047.

[4] C.-K. Ho and C.-T. King, "Automating the learning of inverse kinematics for robotic arms with redundant dofs," 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251040033

[5] X. Ma, S. Patidar, I. Haughton, and S. James, "Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 081–18 090.

[6] 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*, vol. 3, pp. 297–330, 2020.

[7] S. Liu and P. Liu, "A review of motion planning algorithms for robotic arm systems," in *RiTA 2020: Proceedings of the 8th International Conference on Robot Intelligence Technology and Applications*. Springer, 2021, pp. 56–66.

[8] S. Yang, P. Liu, and N. Pears, "Benchmarking of robot arm motion planning in cluttered environments," in *2023 28th International Conference on Automation and Computing (ICAC)*. IEEE, 2023, pp. 1–6.

[9] O. M. Omisore, S. Han, L. Ren, A. Elazab, L. Hui, T. Abdelhamid, N. A. Azeez, and L. Wang, "Deeply-learnt damped least-squares (dl-dls) method for inverse kinematics of snake-like robots," *Neural Networks*, vol. 107, pp. 34–47, 2018.

[10] G. Li, Z. Jin, M. Volpp, F. Otto, R. Lioutikov, and G. Neumann, "Prodmp: A unified perspective on dynamic and probabilistic movement primitives," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2325–2332, 2023.

[11] F. Frank, A. Paraschos, P. van der Smagt, and B. Cseke, "Constrained probabilistic movement primitives for robot trajectory adaptation," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2276–2294, 2021.

[12] A. D. Sosa-Ceron, H. G. Gonzalez-Hernandez, and J. A. Reyes-Avendaño, "Learning from demonstrations in human–robot collaborative scenarios: A survey," *Robotics*, vol. 11, no. 6, p. 126, 2022.

[13] X. Wang, X. Liu, L. Chen, and H. Hu, "Deep-learning damped least squares method for inverse kinematics of redundant robots," *Measurement*, vol. 171, p. 108821, 2021.

[14] R. A. Shyam, P. Lightbody, G. Das, P. Liu, S. Gomez-Gonzalez, and G. Neumann, "Improving local trajectory optimisation using probabilistic movement primitives," in *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2019, pp. 2666–2671.

[15] H. Simas and R. Di Gregorio, "A technique based on adaptive extended jacobians for improving the robustness of the inverse numerical kinematics of redundant robots," *Journal of Mechanisms and Robotics*, vol. 11, no. 2, p. 020913, 2019.

[16] S. Lloyd, R. A. Irani, and M. Ahmadi, "Fast and robust inverse kinematics of serial robots using halley's method," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2768–2780, 2022.

[17] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-space control of robot manipulators with null-space compliance," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 493–506, 2013.

[18] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Autonomous Robots*, vol. 42, pp. 529–551, 2018.

[19] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[20] I. A. Sucan and S. Chitta, "Moveit!" 2013.