

Geometric and Model Priors in Motion Primitives

Maximilian Mühlbauer^{1,2}, Arne Sachtler^{1,2}, Alin Albu-Schäffer^{2,1}, João Silvério²

Abstract—When learning probabilistic policies from human demonstrations, data-efficient learning is a key requirement. Often, only few demonstrations or even only probabilistic via points are available for movement modeling. Probabilistic machine learning techniques such as Kernelized Movement Primitives (KMPs), Linear Quadratic Tracking (LQT) or Nadaraya-Watson kernel regression allow for modeling a rich set of motions with specific priors using scarce data. Traditionally, these methods are however only defined for Euclidean data. We show an extension to manifolds commonly used in robotics, allowing us to model full poses or other Riemannian manifolds. Each method induces distinct priors on the modeled primitives, resulting in different characteristics of the generated motions as seen in the evaluation.

I. INTRODUCTION

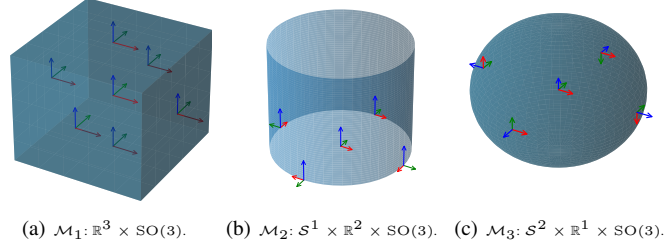
Probabilistic representations of movement primitives are key in applications with requirements in robustness and safety guarantees, as for example in space robotics [1], underwater robotics [2] or healthcare assistance [3]. Especially when executing tasks in Shared Control, it is often desirable to provide guidance when only via points or few demonstrations are available. Key for the applicability of such methods for robotics problems in task space is the ability to model full poses on $SE(3)$, including the orientation on $SO(3)$. Often, this curved manifold is linearized at a single tangent space (STS), leading to distortions [4]. Our goal is thus to find formulations respecting the geometry of the manifold. Furthermore, different charts on $SE(3)$ [5] (Fig. 1) can help to also incorporate geometric priors into the model.

We consider three different models in this work. Nadaraya-Watson kernel regression [6], [7] allows to model a movement primitive from demonstrations in a non-parametric manner, using kernels as weighting functions. Lopez et al. [8] extend this method to manifolds and provide a method for estimating the aleatoric uncertainty. In Section III-A, we show how the method can be formulated utilizing the Riemannian Gaussian distribution [9], [10]. This method is particularly useful for large motions on curved manifolds between data points, as no *linearization points* are required for computing the mean. It is best suited when relatively *dense* demonstration data is available, e.g., from trajectory demonstrations.

Next, we consider Kernelized Movement Primitives (KMPs) which provide estimates for both aleatoric as well as epistemic uncertainties – contrarily to Nadaraya-Watson regression. This allows for detecting out of distribution scenarios. While a linearized version on a STS has already been shown [11], we provide a Riemannian derivation based on formulating KMPs as a Gaussian conditioning (Section III-C).

¹ Department of Computer Engineering, Technical University of Munich, Friedrich-Ludwig-Bauer-Str. 3, Garching, Germany.

² German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchener Str. 20, 82234 Weßling, Germany.



(a) $\mathcal{M}_1: \mathbb{R}^3 \times SO(3)$. (b) $\mathcal{M}_2: S^1 \times \mathbb{R}^2 \times SO(3)$. (c) $\mathcal{M}_3: S^2 \times \mathbb{R}^1 \times SO(3)$.
Fig. 1: Charts on $SE(3)$ used in this work, proposed by [5]. Cylindrical (\mathcal{M}_2) or spherical (\mathcal{M}_3) coordinates allow to express some tasks more data efficient compared to Cartesian (\mathcal{M}_1) coordinates. Coordinate systems in each image depict the unit quaternion $(0, 0, 0, 1)^\top$ for different positions on the manifold.

Finally, Linear Quadratic Tracking (LQT) induces a strong motion bias and as such is very well-suited to derive smooth motion policies. A STS variant of this algorithm has been presented in [10]; based on a modified version of the cost function in [12], we derive a Riemannian formulation of LQT (Section III-C).

We compare the approaches in Section III-B and summarize the findings in Section V. Background on Riemannian geometry, optimization and statistics is provided in Section II.

II. BACKGROUND

Many quantities in robotics, such as orientations in $SO(3)$, cannot be modeled using Euclidean geometry. Riemannian manifolds allow for a principled treatment of such non-Euclidean geometries [10]. Product manifolds can be constructed to represent a full pose as seen in Fig. 1. Common to all smooth manifolds \mathcal{M} is that a logarithm function $\mathbf{u}_{12} = \text{Log}_{\mathbf{x}_1}^{\mathcal{M}}(\mathbf{x}_2)$ maps a point $\mathbf{x}_2 \in \mathcal{M}$ onto the tangent space $\mathcal{T}_{\mathbf{x}_1}\mathcal{M}$ at $\mathbf{x}_1 \in \mathcal{M}$, with the exponential function locally performing the inverse, calculating $\mathbf{x}_2 = \text{Exp}_{\mathbf{x}_1}^{\mathcal{M}}(\mathbf{u}_{12})$. *Parallel transport* moves tangent vectors between tangent spaces at different points on one manifold, while the Jacobian $\mathbf{J}_{\mathcal{M}}$ as given in [13] transforms tangent vectors between tangent spaces of different manifolds \mathcal{M} that are related via a smooth map $\mathcal{F}: \mathcal{M} \rightarrow \mathcal{M}_1$. Optimization on manifolds can be performed using geodesic regression [14] (multivariate case: [15]), making use of *parallel transport* to approximate a small update in the vicinity of the current linearization point.

The Riemannian Gaussian distribution [9], [10] uses manifold-specific distances of the log map and is given as

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D \det(\boldsymbol{\Sigma})}} e^{-\frac{1}{2} \text{Log}_{\boldsymbol{\mu}}^{\mathcal{M}}(\mathbf{x})^\top \boldsymbol{\Sigma}^{-1} \text{Log}_{\boldsymbol{\mu}}^{\mathcal{M}}(\mathbf{x})}, \quad (1)$$

with mean $\boldsymbol{\mu} \in \mathcal{M}$, covariance $\boldsymbol{\Sigma} \in \mathcal{T}_{\boldsymbol{\mu}}\mathcal{M} \otimes \mathcal{T}_{\boldsymbol{\mu}}\mathcal{M}$ and D the dimension of $\mathcal{T}_{\boldsymbol{\mu}}\mathcal{M}$. Using the default metric of each \mathcal{M} through its log map, we obtain different distances and thus different covariances depending on the chosen manifold.

III. RIEMANNIAN MOTION PRIMITIVES

A. Nadaraya-Watson Regression

Euclidean Nadaraya-Watson [6], [7] regression calculates the output ξ for a query s^* based on known outputs μ_i for s_i

$$\xi(s^*, s) = \sum_{i=1}^N w_i(s^*, s_i) \mu_i, \quad (2)$$

with weights calculated through a kernel function $k(\cdot, \cdot)$

$$w_i(s^*, s_i) = \frac{k(s^*, s_i)}{\sum_{j=1}^N k(s^*, s_j)}. \quad (3)$$

Lopez et al. [8] propose a covariance estimate

$$\Sigma(s^*, s_i) = \sum_{i=1}^N w_i(s^*, s_i) (\mu_i - \xi(s^*)) (\mu_i - \xi(s^*))^\top. \quad (4)$$

We reformulate (2) and (4) for the Riemannian Gaussian using a weighted Fréchet mean [9] and original weights (3)

$$\Delta = \sum_{i=1}^N w_i(s^*, s_i) \text{Log}_{\xi(s^*, s)}^{\mathcal{M}}(\mu_i), \quad \xi(s^*, s) \leftarrow \text{Exp}_{\mu}(\Delta),$$

where, upon convergence, the covariance is given as

$$\Sigma(s^*, s_i) = \sum_{i=1}^N w_i(s^*, s_i) \text{Log}_{\xi(s^*, s)}^{\mathcal{M}}(\mu_i) \text{Log}_{\xi(s^*, s)}^{\mathcal{M}}(\mu_i)^\top.$$

B. Kernelized Movement Primitives

KMPs model the distribution of a probabilistic variable ξ based on an input s^* . The goal is to closely approximate a probabilistic reference $\mathcal{D} = \{s_n, \mu_n, \Sigma_n\}_{n=1}^N$ for known input-output pairs which is often encoded in a Gaussian Mixture Model (GMM) and retrieved using Gaussian Mixture Regression (GMR). In Euclidean space, this mapping is derived from a parametric function using basis functions [16], where the *kernel trick* allows for the computation of outputs.

Observing the similarities between Gaussian Processes (GPs) with noisy observations [17] and KMPs, we reformulate KMPs to model a noisy function $\xi(s^*) = f(s^*) + \delta$ with process mean function value e and $\delta \sim \mathcal{N}(\mathbf{0}, \Sigma_{s^*})$. The prior on these noisy observations then becomes $\text{cov}(\xi(s^*)) = \mathbf{K} + \lambda \Sigma$ where $\Sigma = \text{blockdiag}(\Sigma_1, \Sigma_2, \dots, \Sigma_N)$. Similarly, parametric noise models have been used before [18]. This allows us to write

$$\begin{bmatrix} \mu \\ \xi(s^*) \end{bmatrix} = \mathcal{N}\left(e, \underbrace{\begin{bmatrix} \mathbf{K} + \lambda \Sigma & k^* \\ k^{*\top} & k(s^*, s^*) \end{bmatrix}}_{\Sigma'}\right) \quad (5)$$

which can be solved by minimizing the cost

$$J(\xi(s^*)) = c + \text{Log}_{\xi(s^*)}^{\mathcal{M}}(e)^\top \Lambda^{IO} \text{Log}_{\mu}^{\mathcal{M}}(e) + \frac{1}{2} \text{Log}_{\xi(s^*)}^{\mathcal{M}}(e)^\top \Lambda^{OO} \text{Log}_{\xi(s^*)}^{\mathcal{M}}(e) \quad (6)$$

where $\Lambda = \Sigma'^{-1}$ and superscripts I and O refer to partitions for μ and $\xi(s^*)$, respectively. We assume both Σ as well as the kernel function result to be in $\mathcal{T}_{\mu} \mathcal{M} \otimes \mathcal{T}_{\mu} \mathcal{M}$ respectively

$\mathcal{T}_{\xi(s^*)} \mathcal{M} \otimes \mathcal{T}_{\xi(s^*)} \mathcal{M}$. This holds trivially for Σ as the Riemannian Gaussian Distribution which is used as the probabilistic reference expresses its covariance in $\mathcal{T}_{\mu} \mathcal{M} \otimes \mathcal{T}_{\mu} \mathcal{M}$. For the kernel expressed as the Kronecker product between a scalar output kernel and the constant kernel \mathbf{I} this property also holds. We can perform a Taylor expansion

$$J(\xi(s^*)) \approx c + \left(\text{Log}_{\xi(s^*)}^{\mathcal{M}}(e) + \epsilon\right)^\top \Lambda^{OI} \text{Log}_{\mu}^{\mathcal{M}}(e) + \frac{1}{2} \left(\text{Log}_{\xi(s^*)}^{\mathcal{M}}(e) + \epsilon\right)^\top \Lambda^{OO} \left(\text{Log}_{\xi(s^*)}^{\mathcal{M}}(e) + \epsilon\right).$$

With $\epsilon \in \mathcal{T}_{\xi(s^*)} \mathcal{M}$, no parallel transport is required, giving

$$\begin{aligned} \epsilon &= -(\Lambda^{OO})^{-1} \left(\Lambda^{OO} \text{Log}_{\xi(s^*)}^{\mathcal{M}}(e) + \Lambda^{OI} \text{Log}_{\mu}^{\mathcal{M}}(e)\right) \\ &= -\text{Log}_{\xi(s^*)}^{\mathcal{M}}(e) + k^* (\mathbf{K} + \lambda \Sigma)^{-1} \text{Log}_{\mu}^{\mathcal{M}}(e). \end{aligned} \quad (7)$$

When starting from an initial estimate $\xi(s^*) = e$, convergence is given after one step through

$$\xi(s^*) = \text{Exp}_e^{\mathcal{M}} \left(k^* (\mathbf{K} + \lambda \Sigma)^{-1} \text{Log}_{\mu}^{\mathcal{M}}(e) \right). \quad (8)$$

Finally, the covariance evaluates to

$$\Sigma(s^*) = (\Lambda^{OO})^{-1} = k(s^*, s^*) - k^* (\mathbf{K} + \lambda \Sigma)^{-1} k^{*\top}. \quad (9)$$

C. Linear Quadratic Tracking

LQT as e.g. presented by [20] aims to track poses $\hat{\mu}_k$ with covariance $\hat{\Sigma}_k$ at time steps k for a double-integrator dynamical system. In Euclidean space, we get the system

$$\begin{bmatrix} \mathbf{x}_{t+1} \\ \hat{\mathbf{x}}_{t+1} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{I} \Delta t \\ \mathbf{0} & \mathbf{I} \end{bmatrix}}_{=\mathbf{A}} \underbrace{\begin{bmatrix} \mathbf{x}_t \\ \hat{\mathbf{x}}_t \end{bmatrix}}_{=\xi_t} + \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{I} \Delta t \end{bmatrix}}_{=\mathbf{B}} \mathbf{u}_t \quad (10)$$

with optimal control inputs $\hat{\mathbf{U}} = [\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, \dots, \hat{\mathbf{u}}_N]^\top$

$$\hat{\mathbf{U}} = (\mathbf{S}^{\mathbf{u}\top} \Sigma^{-1} \mathbf{S}^{\mathbf{u}} + \mathbf{R})^{-1} \mathbf{S}^{\mathbf{u}\top} \Sigma^{-1} (\mu - \mathbf{S}^{\xi} \xi_1), \quad (11)$$

where \mathbf{S}^{ξ} unrolls the state evolution and $\mathbf{S}^{\mathbf{u}}$ control actions. $\mathbf{R} = \text{blockdiag}(\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_N)$ denotes the cost of control actions at each time step.

On Riemannian manifolds, the state ξ_k consists of a pose $\mathbf{x}_k \in \mathcal{M}$ and corresponding velocity $\dot{\mathbf{x}}_k$ in $\mathcal{T}_{\mathbf{x}_k} \mathcal{M}$. The state evolution thus evaluates to

$$\dot{\mathbf{x}}_{k+1} = \dot{\mathbf{x}}_k + \mathbf{u}_k \cdot \Delta t \quad (12)$$

$$\mathbf{x}_{k+1} = \text{Exp}_{\mathbf{x}_k}^{\mathcal{M}}(\dot{\mathbf{x}}_k) \quad (13)$$

where Δt is the time between two consecutive steps and \mathbf{u}_k the control action at k . In order to minimize the control actions for the given tracking task, we consider the cost function

$$J(\mathbf{u}) = \sum_{k=1}^K \left(\text{Log}_{\hat{\mu}_k}^{\mathcal{M}}(\mathbf{x}_k)^\top \hat{\Sigma}_k \text{Log}_{\hat{\mu}_k}^{\mathcal{M}}(\mathbf{x}_k) + \mathbf{u}_k^\top \mathbf{R} \mathbf{u}_k \right).$$

Setting $\mathbf{u}_k^* = \mathbf{u}_k + \epsilon$, we can compute the Taylor expansion

$$\begin{aligned} J(\mathbf{u} + \epsilon) &= \sum_{k=1}^K \left(\text{Log}_{\hat{\mu}_k, \epsilon}^{\mathcal{M}}(\mathbf{x}_k)^\top \hat{\Sigma}_k \text{Log}_{\hat{\mu}_k, \epsilon}^{\mathcal{M}}(\mathbf{x}_k) \right. \\ &\quad \left. + (\mathbf{u}_k + \epsilon_k)^\top \mathbf{R} (\mathbf{u}_k + \epsilon_k) \right) \end{aligned} \quad (14)$$

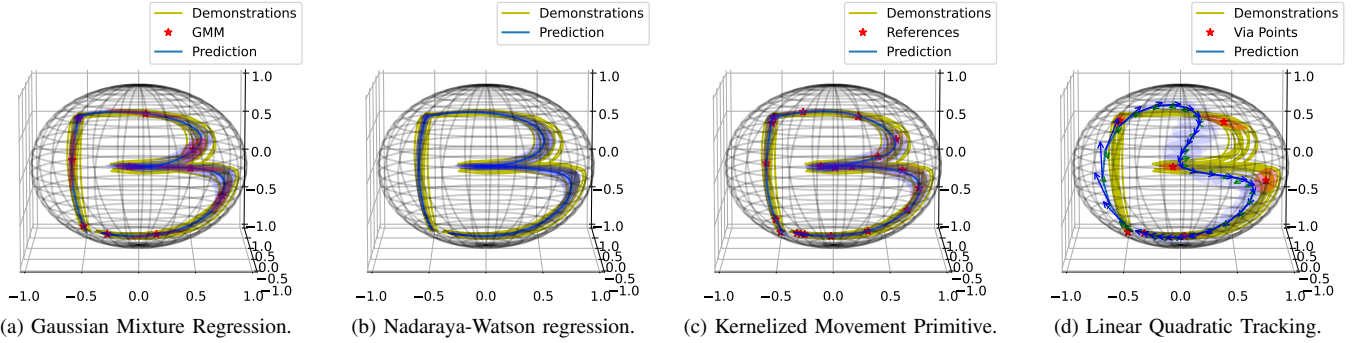


Fig. 2: Plots for writing the letter “B” of the handwriting dataset [19]. We fit the GMM using 10 Gaussians and use 20 reference points for KMP and 7 via points for LQT; all these Gaussians are depicted with their mean and covariance ellipsoid. For the LQT, we furthermore plot its velocity as blue and the command at each time step as green arrows. Note how it passes through the edge of the covariance ellipsoids, trading accuracy for control cost.

where we consider states modified by the update ϵ

$$\text{Log}_{\hat{\mu}_k, \epsilon}^{\mathcal{M}}(\mathbf{x}_k) = \text{Log}_{\hat{\mu}_k}^{\mathcal{M}}(\mathbf{x}_k) + \sum_{j=1}^{k-1} \Gamma_{\mathbf{x}_j \rightarrow \mathbf{x}_{k-1}} \epsilon_j \Delta t^2 \cdot (k-j-2).$$

$\Gamma_{\mathbf{x}_j \rightarrow \mathbf{x}_{k-1}}$ denotes the parallel transport with respect to the Levi-Civita connection from $\mathcal{T}_{\mathbf{x}_j} \mathcal{M}$ to $\mathcal{T}_{\mathbf{x}_{k-1}} \mathcal{M}$ along the geodesics connecting the system states. Calculating the derivative of (14) and setting to zero gives

$$\epsilon = - \left((\mathbf{S}^u \top \Sigma^{-1} \mathbf{S}^u + \mathbf{R})^{-1} \mathbf{S}^u \top \Sigma^{-1} \text{Log}_{\hat{\mu}_k}^{\mathcal{M}}(\mathbf{x}) + \mathbf{R}u \right),$$

for iterating $\mathbf{u} \leftarrow \mathbf{u} + \epsilon$, converging after a few steps with

$$\mathbf{S}^u = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \Gamma_{0,1} \Delta t^2 & 0 & \dots & 0 \\ 2\Gamma_{0,2} \Delta t^2 & \Gamma_{1,2} \Delta t^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ (T-2)\Gamma_{0,T-1} \Delta t^2 & (T-3)\Gamma_{1,T-1} \Delta t^2 & \dots & \Gamma_{T-2,T-1} \Delta t^2 \end{bmatrix}$$

where $\Gamma_{u,v}$ denotes the parallel transport from u to v along the trajectory, expressed as a rotation matrix for the manifolds we consider [9]. Subsequently, the covariance evaluates to

$$\Sigma = \mathbf{S}^u (\mathbf{S}^u \top \Sigma^{-1} \mathbf{S}^u + \mathbf{R})^{-1} \mathbf{S}^u \top. \quad (15)$$

IV. EVALUATION

In Fig. 2, we show results on the handwriting dataset [19] projected on \mathcal{S}^2 for all three methods. Additionally, we plot the output of Riemannian GMR [9] used to extract reference points for KMP and via points for LQT. We use a radial basis function kernel with length scales 0.02 (Nadaraya-Watson) and 0.1 (KMP) and set $\Delta t = 0.03$ and $\mathbf{R} = 1e-4 \cdot \mathbf{I}$ for the LQT. All methods except for LQT pass centrally through the demonstrations, modeling their uncertainty accurately. Contrarily, LQT prefers a trajectory requiring less control action, passing only through the edge of the covariance ellipsoids of the via points. Note that this is exactly the behavior we expect from LQT, balancing tracking accuracy and control cost.

Fig. 3 shows motion primitives on $\text{SE}(3)$ expressed through \mathcal{M}_1 and \mathcal{M}_2 fitted on 6 via points for a spiral trajectory. Accordingly, we plot both the position as well as the orientation using the coordinate axes. We use a radial basis function kernel with length scales 0.05 (Nadaraya-Watson) and 0.2 (KMP) and set $\Delta t = 0.01$ and $\mathbf{R} = 1e-4 \cdot \mathbf{I}$ for the LQT. Note

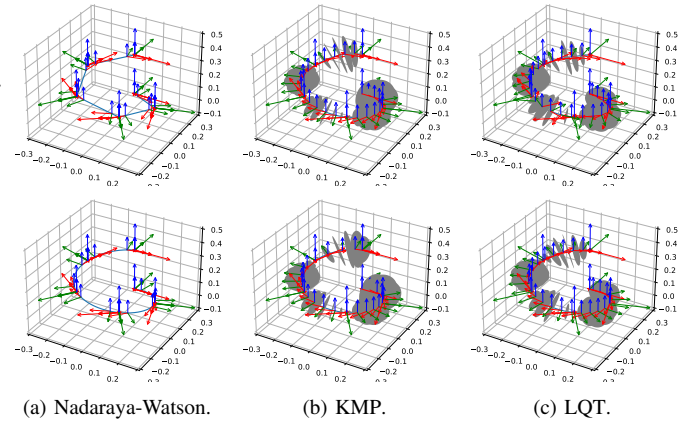


Fig. 3: Circular trajectory from 6 via points on \mathcal{M}_1 (top) and \mathcal{M}_2 (bottom). Coordinate systems show predictions, longer arrows belong to via points. Gray disks perpendicular to the trajectory depict the covariance estimate.

that Nadaraya-Watson regression has a strong bias towards the known data points, resulting in a poor approximation when only few data points are available. For time-based motions, this would result in the system spending much time around the via points and moving quickly in between. Not providing an *epistemic uncertainty* estimate, the covariance furthermore remains low even far from known points, unlike for KMP and LQT. Finally, also the choice of manifold acts as a prior on the estimated trajectory: on \mathcal{M}_2 , an approximation of the spiral trajectory is possible with less data as can especially be seen from the smoother result of the Nadaraya-Watson regression.

V. CONCLUSION

We have shown extensions of Nadaraya-Watson regression, KMPs and LQTs to the Riemannian setting. This enables their use on curved manifolds without inducing distortions, e.g., through a STS approximation. An evaluation on different task settings furthermore highlights *priors* induced both by method as well as by charts \mathcal{M} on $\text{SE}(3)$ – while Nadaraya-Watson regression allows for a smooth interpolation of dense measurements, both KMPs and LQT excel at trajectory modeling, with the latter furthermore introducing a strong motion prior. In future work, we plan to extend those methods to mesh-based manifolds [21] to leverage further geometry-induced priors.

REFERENCES

- [1] M. Mühlbauer, M. Chalon, J. Silvério, and A. Albu-Schäffer, “Control system for a torque-sensitive space robotic arm,” in *76th International Astronautical Congress, IAC 2025*, ser. IAC 2025. International Astronautical Federation (IAF), 2025, pp. 310–321.
- [2] A. Schmidt, D. Höhn, H. Singh, M. Walter, M. Mühlbauer, E.-U. Pap, M. Panzirsch, B. Lodewikus, H. van Strien, S. Zhang *et al.*, “Conceptual framework for teleoperated robotic recovery of large subsea uxos,” in *2025 IEEE Conference on Telepresence, Telepresence 2025*, 2025.
- [3] G. Quere, F. Stulp, D. Filliat, and J. Silvério, “A probabilistic approach for learning and adapting shared control skills with the human in the loop,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 05 2024, pp. 15 728–15 734.
- [4] N. Jaquier, L. Rozo, and T. Asfour, “Unraveling the single tangent space fallacy: An analysis and clarification for applying riemannian geometry in robot learning,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 242–249.
- [5] B. Ti, A. Razmjoo, Y. Gao, J. Zhao, and S. Calinon, “A geometric optimal control approach for imitation and generalization of manipulation skills,” *Robotics and Autonomous Systems*, vol. 164, p. 104413, 2023.
- [6] E. A. Nadaraya, “On estimating regression,” *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142.
- [7] G. S. Watson, “Smooth regression analysis,” *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, vol. 26, no. 4, pp. 359–372, 1964. [Online]. Available: <http://www.jstor.org/stable/25049340>
- [8] P. C. Lopez-Custodio, K. Bharath, A. Kucukyilmaz, and S. P. Preston, “Non-parametric regression for robot learning on manifolds,” 2024. [Online]. Available: <https://arxiv.org/abs/2310.19561>
- [9] M. J. A. Zeestraten, I. Havoutis, J. Silvério, S. Calinon, and D. G. Caldwell, “An approach for imitation learning on Riemannian manifolds,” *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1240–1247, 2017.
- [10] S. Calinon, “Gaussians on riemannian manifolds: Applications for robot learning and adaptive control,” *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 33–45, 2020.
- [11] Y. Huang, F. J. Abu-Dakka, J. Silverio, and D. G. Caldwell, “Toward orientation learning and adaptation in cartesian space,” *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 82–98, 2021.
- [12] L. Rozo, M. Guo, A. G. Kupcsik, M. Todescato, P. Schillinger, M. Giftthaler, M. Ochs, M. Spies, N. Waniek, P. Kesper, and M. Burger, “Learning and sequencing of object-centric manipulation skills for industrial tasks,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9072–9079.
- [13] M. Mühlbauer, B. Weber, S. Calinon, F. Stulp, A. Albu-Schäffer, and J. Silvério, “A unified framework for probabilistic dynamic-, trajectory- and vision-based virtual fixtures,” 2025. [Online]. Available: <https://arxiv.org/abs/2506.10239>
- [14] P. Thomas Fletcher, “Geodesic regression and the theory of least squares on riemannian manifolds,” *International Journal of Computer Vision*, vol. 105, no. 2, pp. 171–185, 2012.
- [15] H. J. Kim, N. Adluru, M. D. Collins, M. K. Chung, B. B. Bendin, S. C. Johnson, R. J. Davidson, and V. Singh, “Multivariate general linear models (mgm) on riemannian manifolds with applications to statistical analysis of diffusion weighted images,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 2705–2712.
- [16] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, “Kernelized movement primitives,” *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 833–852, 2019.
- [17] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2.
- [18] R. Murray-Smith and A. Girard, “Gaussian process priors with arma noise models,” in *Irish Signals and Systems Conference*, 2001.
- [19] J. Silvério and Y. Huang, “A non-parametric skill representation with soft null space projectors for fast generalization,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2988–2994.
- [20] S. Calinon, “A tutorial on task-parameterized movement learning and retrieval,” *Intelligent Service Robotics*, vol. 9, no. 1, pp. 1–29, Sep 2015.
- [21] M. Dyck, A. Sachtler, J. Klodmann, and A. Albu-Schaffer, “Impedance control on arbitrary surfaces for ultrasound scanning using discrete differential geometry,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7738–7746, 2022.