Safe and stable motion primitives via imitation learning and geometric fabrics

Saray Bakker[1], **Rodrigo Pérez-Dattari**[1], **Cosimo Della Santina**[1], **Wendelin Böhmer**[2], **Javier Alonso-Mora**[1]
1. Department of Mechanical Engineering,
2. Department of Electrical Engineering, Mathematics & Computer Science,
Delft University of Technology, The Netherlands
s.bakker-7@tudelft.nl

Abstract: Using the language of dynamical systems, Imitation learning (IL) provides an intuitive and effective way of teaching stable task-space motions to robots with goal convergence. Yet, these techniques are affected by serious limitations when it comes to ensuring safety and fulfillment of physical constraints. With this work, we propose to solve this challenge via TamedPUMA, an IL algorithm augmented with a recent development in motion planning called geometric fabrics. We explore two variations of this approach, which we name the forcing policy method and the compatible potential method. The result is a stable imitation learning strategy within which we can seamlessly blend geometrical constraints like collision avoidance and joint limits. Beyond providing a theoretical analysis, we demonstrate TamedPUMA with simulated and real-world tasks, including a 7-DoF manipulator.

Keywords: Imitation Learning, Dynamical Systems, Geometric Motion Planning, Fabrics, Movement Primitives

1 Introduction

As robotic solutions rapidly enter unstructured environments such as the agriculture sector, homes, and the food industry, there is a critical need for methods that allow non-experts to easily adapt robots for new tasks. These sectors demand that robots safely interact with dynamic, fragile environments where humans are present. Currently, experts manually program these tasks, a method that is costly and not scalable for widespread use.

A possible solution to this societal challenge comes from Imitation Learning (IL). Using this technique, robots can learn motion profiles from demonstrations provided by non-expert users. Furthermore, by encoding the learned trajectories as solutions of a dynamical system, established mathematical tools from dynamical system theory can be used to guarantee global convergence to the goal [1, 2]. An influential methodology in learning stable dynamical systems is Dynamical Movement Primitives (DMPs) [3] ensuring convergence towards a simple manually-designed dynamical system. This approach is extended to non-Euclidean state spaces [4], probabilistic environments [5, 6], and in the context of deep neural network (DNN) [7, 8]. To ensure stability, several approaches enforce a specific structure on the function approximators, such as enforcing positive or negative definiteness [9, 10, 11], or invertibility [12, 13, 14]. In contrast, [15, 16] enforce stability via additional loss functions derived using tools from the deep metric learning literature [17]. Importantly, [16] extends the ideas of [15] to more general scenarios, achieving better results in non-Euclidean state spaces and 2^{nd}-order dynamical systems.

In a robotics context, these learned dynamical systems commonly encode the navigation policy towards a goal within a task space - such as the evolution of the end-effector pose of a manipulator while pouring water in a glass from all possible initial locations. However, this focus on task

8th Conference on Robot Learning (CoRL 2024), workshop MRM-D, Munich, Germany.

space motions renders IL fundamentally limited when it comes to considering physical constraints involving the body of the robot impacting with itself or interacting with the external environment. Substantial recent research has looked into the problem, e.g. [18, 19, 20, 21, 22, 23], although most methods lack simulatenously addressing stability guarantees and real-time fulfillment of physical constraints for a system with many degrees of freedom.

This work introduces TamedPUMA, a safe and stable extension of the IL algorithm PUMA [16], which allows for an increased motion expressiveness in contrast to [24, 25]. The key enabling idea behind this new method is to look at the learned stable motion primitives as the navigation policy within a recently introduced geometrical framework for motion planning called geometric fabrics [26]. To make this possible, the learned model has to be formulated as a 2^{nd}-order (neural) dynamical system as fabrics operate within the Finsler Geometry framework where vector fields must be defined at the acceleration level [27]. Also, the learned dynamics must admit an *aligned* potential, which roughly means a scalar function whose gradient is aligned with the acceleration field when the velocity is null. In the paper, we show how we can ensure that both conditions are met. We propose two variations on TamedPUMA: the forcing policy method, treating the learned DNN as a forcing term within the geometric fabrics formulation, and the compatible potential method, introducing an energy-regulation term requiring the design of a so-called compatible potential. We evaluate the performance of TamedPUMA with a simulated and real-world 7-DoF manipulator, where we also benchmark it against vanilla geometric fabrics, vanilla learned stable motion primitives and a modulation-based IL approach leveraging collision-aware IK.

2 TamedPUMA: Combining learned stable motion primitives and fabrics

With learned stable motion primitives, complex tasks can be learned from demonstrations, while converging to the goal. By incorporating these learned dynamical systems into the navigation policy of geometric fabrics, stable and safe motions are generated respecting whole-body obstacle avoidance and physical constraints of the robot. In the following, we propose two variations of TamedPUMA, the Forcing Policy Method (FPM) and Compatible Potential Method (CPM).

The Forcing Policy Method (FPM): First, we introduce the FPM. Via Policy via neUral Metric leArning (PUMA), a dynamical system is learned in task space $\mathcal{T}$, $\boldsymbol{f}_\theta^{\mathcal{T}}(\boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{\varphi}_\theta(\boldsymbol{\rho}_\theta(\boldsymbol{x}, \dot{\boldsymbol{x}}))$, where $\boldsymbol{\rho}_\theta : \mathcal{X} \to \mathcal{L}$ encodes the first $1, ..., l$ layers and $\boldsymbol{\varphi}_\theta : \mathcal{L} \to \mathcal{X}$ the last $l+1, ..., L$ layers of the DNN. We define the dynamical system $\boldsymbol{f}_\theta^{\mathcal{C}}$ in configuration space $\mathcal{C}$ resulting from applying a pullback operation [28] to the learned system:

$$\ddot{\boldsymbol{q}} = \boldsymbol{f}_\theta^{\mathcal{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathrm{pull}_{\phi^{\mathcal{T}}}\left(\boldsymbol{f}_\theta^{\mathcal{T}}(\boldsymbol{x}, \dot{\boldsymbol{x}})\right), \tag{1}$$

Then, leveraging the definition of a forced system of Definition II.4 in [28], in the FPM we propose to use the pulled system obtained via PUMA as the policy that forces the energy-conservative fabric $\tilde{\boldsymbol{h}} : \mathcal{S} = (\tilde{\boldsymbol{M}}, \tilde{\boldsymbol{\xi}})_{\mathcal{X}}$ where $\tilde{\boldsymbol{h}}$ conserves a Finsler energy,

$$\ddot{\boldsymbol{q}} = \tilde{\boldsymbol{h}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{f}_\theta^{\mathcal{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}}). \tag{2}$$

Assuming the loss function of PUMA has already been minimized, the system $\boldsymbol{f}_\theta^{\mathcal{T}}$ comes to rest at the equilibrium $\boldsymbol{x}_{\mathrm{g}}$, implying that $\boldsymbol{f}_\theta^{\mathcal{C}}$ converges to the configuration $\boldsymbol{q}_{\mathrm{g}}$ where multiple values of $\boldsymbol{q}_{\mathrm{g}}$ may exist in the case of a redundant system. This collection of states $\boldsymbol{q}_{\mathrm{g}}$ corresponds to the *zero set* of $\boldsymbol{f}_\theta^{\mathcal{C}}$. From Proposition II.17 in [28], we know that if the system in Eq. (2) reaches the zero set of $\boldsymbol{f}_\theta^{\mathcal{C}}$, it will stay there (which comes from the observation that fabrics are conservative). However, we cannot make any claims regarding the convergence of Eq. (2) to the zero set of $\boldsymbol{f}_\theta^{\mathcal{C}}$. Therefore, we propose the CPM, a method with stronger convergence guarantees.

The Compatible Potential Method (CPM): As a second approach, we propose the CPM that exploits the concept of *compatible potentials* to obtain a stronger notion of convergence. A potential compatible with a dynamical system *generally* points in the same direction as the system's vector field. More formally:

Definition 1 (Compatible potential [28])**.** *A potential function ψ is compatible with $\boldsymbol{f}$ if: (1) $\partial\psi(\boldsymbol{q}) = \boldsymbol{0}$ if and only if $\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{0}) = \boldsymbol{0}$, and (2) $-\partial\psi^\top \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{0}) > \boldsymbol{0}$ wherever $\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{0}) \neq \boldsymbol{0}$.*

From this, [28] introduces Theorem III.5, which states that given a dynamical system with a compatible potential, then the following system converges to the zero set of $\boldsymbol{f}$,

$$\ddot{\boldsymbol{q}} = \text{energize}_{\mathcal{H}}[\boldsymbol{h} + \boldsymbol{f}] + \gamma(\boldsymbol{q}, \dot{\boldsymbol{q}}) \quad \text{with} \quad \gamma(\boldsymbol{q}, \dot{\boldsymbol{q}}) = -\left(\frac{\dot{\boldsymbol{q}}\dot{\boldsymbol{q}}^\top}{\dot{\boldsymbol{q}}^\top M_{\mathcal{L}_e} \dot{\boldsymbol{q}}}\right) \partial\psi - \beta\dot{\boldsymbol{q}}. \tag{3}$$

Consequently, we aim to leverage this result by using $\boldsymbol{f}_\theta^{\mathcal{C}}$ (Eq. 1) as the system $\boldsymbol{f}$ in Eq. (3) with the compatible potential. From the previous section, we already concluded that the zero set of this system maps to the equilibrium $\boldsymbol{x}_{\mathrm{g}} \in \mathcal{T}$. Hence, it remains to find a compatible potential for this function to employ the result from Theorem III.5 [28].

Notably, if the stability loss of PUMA is successfully minimized, it is possible to design a compatible potential for the system $\boldsymbol{f}_\theta^{\mathcal{T}}$ in the latent space $\mathcal{L}$ using the mapping $\boldsymbol{\rho}_\theta$ by setting $\dot{\boldsymbol{x}} = \boldsymbol{0}$. Specifically, we can construct a potential using the latent-space variable $\boldsymbol{y}$ and the encoder $\boldsymbol{\rho}_\theta$ of PUMA,

$$\psi(\boldsymbol{x}) = \|\boldsymbol{y}_{\mathrm{g}} - \boldsymbol{y}\|^2 = \|\boldsymbol{\rho}_\theta(\boldsymbol{x}_{\mathrm{g}}, \boldsymbol{0}) - \boldsymbol{\rho}_\theta(\boldsymbol{x}, \boldsymbol{0})\|^2. \tag{4}$$

To observe that this is a compatible potential of $\boldsymbol{f}_\theta^{\mathcal{T}}$, first, we highlight that since $\boldsymbol{x}_{\mathrm{g}}$ is asymptotically stable, we have $\partial\psi(\boldsymbol{x}_{\mathrm{g}}) = \boldsymbol{0}$ if and only if $\boldsymbol{f}(\boldsymbol{x}_{\mathrm{g}}, \boldsymbol{0}) = \boldsymbol{0}$. This satisfies the first condition of Definition 1. Second, we note that for all $\boldsymbol{x} \neq \boldsymbol{x}_{\mathrm{g}}$, the stability loss enforces the value of $\psi(\boldsymbol{x})$ to decrease as $\boldsymbol{f}_\theta^{\mathcal{T}}$ evolves over time, provided that $\boldsymbol{\rho}_\theta$ has a Lipschitz constant that is not large, which can be controlled through regularization. Thus, this potential also satisfies the second condition of Definition 1. Finally, it only remains to express the gradient of this potential in configuration space, previously denoted as $\partial\psi$. For clarity, we will henceforth write this as $\partial\psi/\partial\boldsymbol{q}$. To achieve this, we require the forward kinematics from configuration space $\mathcal{C}$ to task space $\mathcal{T}$, denoted $\phi^{\mathcal{T}}$. Then, we obtain

$$\partial\psi = \frac{\partial\psi}{\partial\boldsymbol{q}} = \frac{\partial\psi}{\partial\boldsymbol{x}} \cdot J_{\phi^{\mathcal{T}}}(\boldsymbol{x}), \tag{5}$$

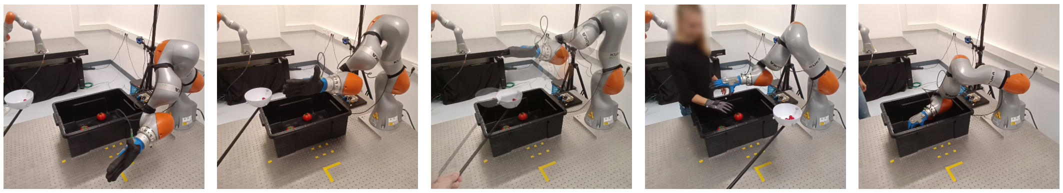
where $J_{\phi^{\mathcal{T}}}$ is the Jacobian matrix of the forward kinematics. The matrix $J_{\phi^{\mathcal{T}}}$ is commonly available in robotic frameworks, and the term $\partial\psi/\partial\boldsymbol{x}$ can be approximated via automatic differentiation tools for DNNs.

3 Experimental Results

Experimental setup and performance metrics: To showcase the performance of the two variations of TamedPUMA, FPM and CPM, simulations using the Pybullet physics simulation [29] and real-world experiments are performed on a 7-DoF KUKA iiwa manipulator. Two tasks are analyzed, picking a tomato from a crate and pouring liquid from a cup, where a DNN is trained for each task using 10 demonstrations. The proposed FPM and CPM, are compared against vanilla geometric fabrics, vanilla PUMA and *modulation-IK*. Modulation-IK modifies PUMA to be obstacle-free within the task space using a modulation matrix [30]. Then, whole-body collision avoidance is achieved by tracking the modified desired pose using a collision-aware Inverse Kinematics (IK) [31]. All methods are evaluated based on their *success rate* and *time-to-success*, which respectively indicate the ratio of successful scenarios and the time required for the robot to reach the goal pose. The methodologies are also compared on the average *minimum clearance* between the collision shapes of the robot and obstacles over all scenarios and *computation time*. In addition, the *path difference* to the desired path by PUMA is denoted as $\frac{1}{P}\sum\|\boldsymbol{x}_{\mathrm{ee}} - \boldsymbol{x}_{\mathrm{PUMA}}\|_2$ where $\boldsymbol{x}_{\mathrm{ee}}$ and $\boldsymbol{x}_{\mathrm{PUMA}}$ correspond

Table 1: Statistics for 30 simulated scenarios. The path difference to PUMA is measured in an obstacle-free environment, while all other metrics are compared in an obstacle-rich environment.

	Success-Rate	Time-to-Success [s]	Min Clearance [m]	Computation time [ms]	Path difference to PUMA
PUMA	0.5	5.2 ± 1.6	0.02 ± 0.03	4.7 ± 0.5	0
Modulation-IK	0.6	2.7 ± 1.8	0.01 ± 0.01	13.5 ± 8.5	0.38 ± 0.42
Fabrics	0.9	6.5 ± 3.1	0.06 ± 0.04	0.41 ± 0.04	0.15 ± 0.19
FPM	1	7.3 ± 5.0	0.05 ± 0.02	5.9 ± 0.8	0.04 ± 0.07
CPM	1	10.9 ± 6.7	0.05 ± 0.02	7.1 ± 0.5	0.06 ± 0.09



(a) Initial pose (b) Bowl approaches (c) Avoid the bowl (d) Avoid the hand (e) Goal reached

Figure 1: Selected time frames of CPM during a tomato-picking task.

to the end-effector poses along the path with length P. The path difference is computed in obstacle-free scenarios, where we aim to track the DNN as closely as possible and in obstacle-rich settings, allowing for deviations from this path.

Simulation experiments on a 7-DOF manipulator: In simulation, 30 realistic scenarios are explored, including 15 scenarios of a tomato-picking task and 15 scenarios of a pouring task. In each task, the initial robot configuration and obstacle locations change. Moreover, 10 scenarios included moving goals, and 7 scenarios included moving obstacles. As depicted in Table 1, the two variations of TamedPUMA improve the success rate with respect to PUMA as it allows for whole-body obstacle avoidance. In contrast to geometric fabrics, FPM and CPM can track the desired motion profile leading to a smaller path difference with PUMA of 0.04 $\pm$ 0.07 and 0.06 $\pm$ 0.09 respectively, compared to geometric fabrics, 0.15 $\pm$ 0.19, in an obstacle-free environment. In an obstacle-rich environment, geometric fabrics result in a deadlock in 3 of the 30 scenarios where the robot does not reach the goal as it is unable to move around the edge of the crate or object. The benchmark Modulation-IK is also unable to achieve all tasks due to collisions or deadlocks. In comparison to FPM and CPM, Modulation-IK has a higher computation time due to the optimization-based IK solver with high varieties depending on the difficulty of the scenario. Although the CPM has stronger theoretical guarantees compared to FPM, performance is similar when comparing the two proposed approaches in Table 1. Even though we do not optimize over the time-to-success or minimum clearance, both FPM and CPM achieve the task within a reasonable time while remaining collision-free. Computation times are within the order of 4-7 ms on a standard laptop (i7-12700H) making the methodologies well suitable for real-time reactive motion planning.

Real-world experiments on a 7-DOF manipulator: Experiments are performed on the real 7-DOF for the tomato-picking and pouring task where all obstacles, e.g. a bowl, a person's hand and a helmet, are dynamically tracked in real-time via an optitrack system. Snapshots of a real-world experiment of the CPM are illustrated in Fig. 1. If the obstacles are not blocking the trajectory of the robot, the observed behavior of the proposed methods, FPM and CPM, are similar to PUMA and showcases clearly the learned behavior as demonstrated by the human. The user can push the robot away from the goal or change the goal online, and TamedPUMA recovers from this disturbance. In the presence of obstacles, FPM and CPM achieve collision avoidance between the considered links on the robot and the obstacles while reaching the goal pose, as illustrated in Fig. 1.

4 Conclusion

Imitation learning via stable motion primitives is a well-suitable approach for learning motion profiles from demonstrations while providing convergence to the goal. We introduced TamedPUMA, a safe and stable extension of learned stable motion primitives augmented with the recently developed geometric fabrics for safe and stable operations in the presence of obstacles. We proposed two variations, the Forcing Policy Method and Compatible Potential Method, ensuring respectively that the goal is stable, or the stronger notion that the system converges towards the reachable goal. Experiments were carried out both in simulation and in the real world. When trained on a tomato-picking task or pouring task, the proposed TamedPUMA generates a desired motion profile using a DNN while taking whole-body collision avoidance and joint limits into account, with a computation time of just 4-7 ms.

Acknowledgments

This project has received funding from the European Union through ERC, INTERACT, under Grant 101041863, and by the NXTGEN national program. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the NXTGEN national program. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] A. Billard, S. Mirrazavi, and N. Figueroa. *Learning for adaptive and reactive robot control: a dynamical systems approach*. Mit Press, 2022.

[2] Y. Hu, F. J. Abu-Dakka, F. Chen, X. Luo, Z. Li, A. Knoll, and W. Ding. Fusion dynamical systems with machine learning in imitation learning: A comprehensive overview. *Information Fusion*, page 102379, 2024.

[3] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2):328–373, 2013.

[4] A. Ude, B. Nemec, T. Petrić, and J. Morimoto. Orientation in cartesian space dynamic movement primitives. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2997–3004. IEEE, 2014.

[5] 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*, 8 (4):2325–2332, 2023.

[6] M. Przystupa, F. Haghverd, M. Jagersand, and S. Tosatto. Deep probabilistic movement primitives with a bayesian aggregator. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3704–3711. IEEE, 2023.

[7] A. Pervez, Y. Mao, and D. Lee. Learning deep movement primitives using convolutional neural networks. In *2017 IEEE-RAS 17th international conference on humanoid robotics (Humanoids)*, pages 191–197. IEEE, 2017.

[8] B. Ridge, A. Gams, J. Morimoto, A. Ude, et al. Training of deep neural networks for the generation of dynamic movement primitives. *Neural Networks*, 127:121–131, 2020.

[9] S. M. Khansari-Zadeh and A. Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.

[10] A. Lemme, K. Neumann, R. F. Reinhart, and J. J. Steil. Neural learning of vector fields for encoding stable dynamical systems. *Neurocomputing*, 141:3–14, 2014.

[11] N. F. Fernandez and A. Billard. A physically-consistent bayesian non-parametric mixture model for dynamical system learning. *Proceedings of Machine Learning Research*, 2018.

[12] N. Perrin and P. Schlehuber-Caissier. Fast diffeomorphic matching to learn globally asymptotically stable nonlinear dynamical systems. *Systems & Control Letters*, 96:51–59, 2016.

[13] 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*, pages 630–639. PMLR, 2020.

[14] J. Urain, M. Ginesi, D. Tateo, and J. Peters. Imitationflow: Learning deep stable stochastic dynamic systems by normalizing flows. In *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5231–5237, 2020.

[15] R. Pérez-Dattari and J. Kober. Stable motion primitives via imitation and contrastive learning. *IEEE Transactions on Robotics*, 39(5):3909–3928, 2023.

[16] R. Pérez-Dattari, C. Della Santina, and J. Kober. Deep metric imitation learning for stable motion primitives. *arXiv preprint*, 2023.

[17] M. Kaya and H. Ş. Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[18] H. B. Mohammadi, S. Hauberg, G. Arvanitidis, N. Figueroa, G. Neumann, and L. Rozo. Neural contractive dynamical systems. In *The Twelfth International Conference on Learning Representations*.

[19] M. Ginesi, D. Meli, A. Calanca, D. Dall'Alba, N. Sansonetto, and P. Fiorini. Dynamic movement primitives: Volumetric obstacle avoidance. In *2019 19th international conference on advanced robotics (ICAR)*, pages 234–239. IEEE, 2019.

[20] L. Huber, J.-J. Slotine, and A. Billard. Avoidance of concave obstacles through rotation of nonlinear dynamics. *IEEE Transactions on Robotics*, 2023.

[21] K. Ahn, Z. Mhammedi, H. Mania, Z.-W. Hong, and A. Jadbabaie. Model predictive control via on-policy imitation learning. In *Learning for Dynamics and Control Conference*, pages 1493–1505. PMLR, 2023.

[22] Y. Hu, M. Cui, J. Duan, W. Liu, D. Huang, A. Knoll, and G. Chen. Model predictive optimization for imitation learning from demonstrations. *Robotics and Autonomous Systems*, 163: 104381, 2023.

[23] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger. Learning-based model predictive control: Toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:269–296, 2020.

[24] M. Xie, A. Li, K. Van Wyk, F. Dellaert, B. Boots, and N. Ratliff. Imitation learning via simultaneous optimization of policies and auxiliary trajectories. *arXiv preprint arXiv:2105.03019*, 2021.

[25] M. Xie, A. Handa, S. Tyree, D. Fox, H. Ravichandar, N. D. Ratliff, and K. Van Wyk. Neural geometric fabrics: Efficiently learning high-dimensional policies from demonstration. In *Conference on Robot Learning*, pages 1355–1367. PMLR, 2023.

[26] N. D. Ratliff, K. Van Wyk, M. Xie, A. Li, and M. A. Rana. Optimization fabrics. *arXiv preprint arXiv:2008.02399*, 2020.

[27] D. Bao, S.-S. Chern, and Z. Shen. *An introduction to Riemann-Finsler geometry*, volume 200. Springer Science & Business Media, 2012.

[28] N. Ratliff and K. Van Wyk. Fabrics: A foundationally stable medium for encoding prior experience. *arXiv preprint:2309.07368*, 2023.

[29] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016.

[30] S. M. Khansari-Zadeh and A. Billard. A dynamical system approach to realtime obstacle avoidance. *Autonomous Robots*, 32:433–454, 2012.

[31] R. Tedrake. *Robotic Manipulation*. 2023. URL http://manipulation.mit.edu.