Utilizing Human Feedback for Primitive Optimization in Wheelchair Tennis

Arjun Krishna, Zulfiqar Zaidi, Letian Chen, Rohan Paleja, Esmaeil Seraj, Matthew Gombolay

Georgia Institute of Technology

Abstract: Agile robotics presents a difficult challenge with robots moving at high speeds requiring precise and low-latency sensing and control. Creating agile motion that accomplishes the task at hand while being safe to execute is a key requirement for agile robots to gain human trust. This requires designing new approaches that are flexible and maintain knowledge over world constraints. In this paper, we consider the problem of building a flexible and adaptive controller for a challenging agile mobile manipulation task of hitting ground strokes on a wheelchair tennis robot. We propose and evaluate an extension to the work done on learning striking behaviors using a probabilistic movement primitive (ProMP) framework by (1) demonstrating the safe execution of learned primitives on an agile mobile manipulator setup, and (2) proposing an online primitive refinement procedure that utilizes evaluative feedback from humans on the executed trajectories.

Keywords: Learning from demonstrations and feedback, Movement primitives, Agile Mobile manipulator

1 Introduction

Over the years, roboticists have sought to develop robots that can play various sports to demonstrate and test the capabilities of their systems [1, 2, 3]. Sporting applications serve as a very natural motivator for the development of autonomous systems, serving as a milestone in developing systems that achieve human-level performance. Furthermore, to be effective at a sport, the system needs to be able to reason about the state of the game, be agile in its response to changing observations, and be safe by being mindful of objects in its vicinity. As autonomously learning effective robot behavior is challenging for sports, prior work has sought to learn behaviors from experts.

Learning for Demonstrations (LfD) [4] is a framework for learning a policy from a set of demonstrations provided by an expert. Prior work has shown how movement primitives obtained from kinesthetic teaching can be used to teach robots to hit table tennis strokes [5, 6, 7, 8]. It has also been shown that kinesthetic demonstrations can be used successfully to teach robots to play various styles of strokes for table tennis [9, 10]. More recently, the problem of improving a policy learned from expert demonstrations using reinforcement learning and inverse reinforcement learning has also been explored [10, 11, 12]. However, most of these works are evaluated on either simulated environments or on robot arms mounted on a stationary platform.

The need for algorithms that support learning robot behavior for versatile robot platforms is exacerbated in larger-scale racket sports, such as tennis. Tennis is a more challenging problem for robots as it requires a responsive agile mobile base and higher racket head speeds than table tennis. In this work, we demonstrate the first attempt to extend the ProMP framework to an agile mobile manipulator to achieve successful tennis groundstrokes with a wheelchair tennis robot. We additionally, describe an approach to refine the learned primitives from demonstrations online based on human feedback.

2 Preliminaries

In this section, we provide an overview of the Probabilistic Movement Primitive (ProMP) and the notations we will use in the paper. Interested readers are encouraged to refer to [8] for more details.

The ProMP is a modeling technique that compactly represents a probability distribution over robot trajectories [13]. Let q_t represent the joint states of the robot at time t. The ProMP defines a set of time-dependent basis functions (represented by Φ_t) and a weight vector, w, that compactly encodes the robot trajectory, $\tau = \{q_t\}$ by representing $q_t \sim \mathcal{N}(\Phi_t w, \Sigma_y)$, where Σ_y models any white noise.

Given a dataset of demonstrations, the ProMP learns a distribution over weights, $P(w \mid \{\mu_w, \Sigma_w\}) = \mathcal{N}(\mu_w, \Sigma_w))$, that captures the common features across trajectories while factoring in the variance that captures variations in demonstrations. The parameters, $\{\mu_w, \Sigma_w, \Sigma_y\}$, of this Hierarchical Bayesian Model (HBM) formulation can be computed from demonstrations by obtaining Maximum Likelihood Estimates (MLE) via exact methods or Expectation-Maximization (EM) procedures. A key property of ProMP that we leverage in this work is to *condition* it to pass through desired end-effector waypoints. Since the ProMP representation is in the joint space, some form of inverse kinematics (IK) is required to perform this conditioning operation.

3 Method

In Section 3.1, we provide a brief overview of our experiment setup: a wheelchair tennis robot. In Section 3.2, we describe the details of the deployed stroke controller that safely executes the learned primitives. In Section 3.3, we present our proposed method for improving motion primitives based on human feedback.

3.1 System Overview

We mounted a 7-DOF high-speed Barrett WAM arm on a motorized Top End Pro Tennis Wheelchair to build an agile mobile manipulator system [14]. The system is designed to emulate the athletic gameplay of regulation wheelchair tennis, where players need to react quickly in the order of a few hundred milliseconds.

To sense and track the movement of the tennis ball, we make use of a decentralized array of stereo cameras that provides measurements from different perspectives. These estimates are fused by an Extended Kalman Filter (EKF) [15] to output the ball's estimated state which is propagated forward in time to predict the ball's future trajectory.

The problem of whole-body control of mobile manipulators is challenging – particularly in an agile robotics setting – so we limit the scope of our study by constraining the wheelchair to move along one dimension. We choose to allow the robot to move laterally, as human players exhibit lateral movements only for the majority of the strokes [16]. We model the lateral movement as a prismatic joint and obtain a kinematics model for the system (illustrated in Appendix A).

3.2 Stroke Controller

We build our stroke controller upon ProMP, which was originally proposed for table tennis in [7]. We make two key advances for our wheelchair tennis robot.

First, since the strokes executed often reach racket-head speeds $\sim 10\,\mathrm{m\,s^{-1}}$, it is critical to ensure that the conditioned joint space configuration at the time of impact is safe and achievable without any self-collisions. Thus, we propose to perform a *constrained IK* starting from the unconditioned trajectory mean to identify the joint configuration to condition on, and we explicitly restrict the difference $(q_{\text{conditioned}} - q_{\text{unconditioned}})$ to be within specified lower and upper limits, doing so alleviates the risk of getting bad but feasible IK solutions. When a ball is launched, the desired hit point for conditioning is identified by computing where the predicted ball trajectory crosses a pre-specified hit plane, and this point is then transformed into the frame of the mobile base. Iterative IK updates to

the available Degrees of Freedom (DoFs) are performed to reach this desired hit point, and the total updates are clipped to be within the specified limits. The pseudocode of this procedure is presented in Appendix B.

Second, based on the observation from [14] that early positioning movement for tennis-playing robots can improve the chances of a successful return, we allow the wheelchair to continuously adjust its position during the conditioning process and execute the stroke independently based on the anticipated ball arrival time. A flow chart explaining the state-machine of the stroke controller is illustrated in Appendix B.

3.3 Refining primitives through human feedback

In [8], the parameters of ProMP are trained through a dataset of successful demonstrations obtained through kinesthetic teaching or engineered controllers. While the ProMP model is initially trained to recreate a demonstrated behavior (i.e., a robot arm trajectory that hits a ball), the result is suboptimal due to both the suboptimality of the demonstration itself, out-of-distribution incoming flights of the tennis ball, and the hardware-software system's inability to perfectly execute a commanded trajectory.

Nonetheless, the trained primitive does serve as an excellent starting point for exploring better trajectories that can be executed on the hardware. Therefore, we propose to iteratively improve the primitives by having a human evaluator assign scalar feedback indicative of the quality of the executed trajectory. We could collect human feedbacks $\{r_n\}_{n=1}^{N}$ from the evaluator for N executed trajectories $\{\{q_{nt}\}_{t=1}^{T_n}\}_{n=1}^{N}$. We then construct a dataset $\mathcal{D}$ which consists of trajectories and the associated importance weight (α_n, obtained as softmax over feedbacks), to optimize the weighted log-likelihood objective:

$$\texttt{WeightedLogLikelihood}(\mathcal{D} \mid \theta) = \sum_{i=1}^{N} \alpha_n \log(\texttt{Likelihood}(\ \{q_{nt}\}_{t=1}^{T_n} \mid \theta\ )) \tag{1}$$

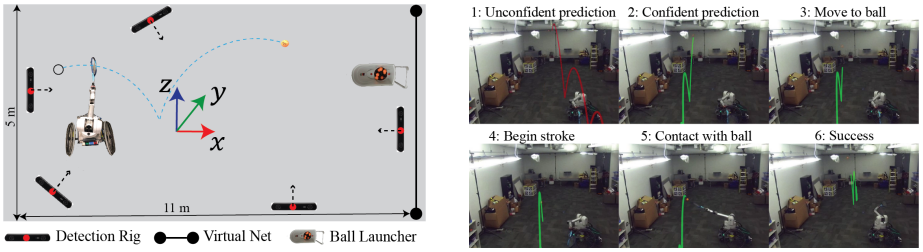
The parameters are optimized with the EM procedure outlined in [8]. In the M-step we compute the weighted average of the estimates from the E-step as a consequence of the weighted log-likelihood objective. This algorithm can be considered a part of the TAMER[17] class of algorithms, as we are interactively shaping the distribution of executed trajectories using human feedback.

Algorithm 1 Iterative refinement of ProMP parameters

Require: ProMP parameters $\theta = (\mu_w, \Sigma_w, \Sigma_y)$ trained from human demonstrations
  repeat
    Execute N trajectories $\{\tau_n\}$ from conditioned execution of θ and obtain human feedback $\{r_n\}$
    Compute importance weights over trajectories, $\{\alpha_n\} \leftarrow \texttt{SoftMax}_{n=1}^{N}(\{r_n\})$
    With dataset $\mathcal{D} = \{(\tau_n, \alpha_n)\}$, perform $\theta \leftarrow \texttt{EM-WeightedLogLikelihood}(\mathcal{D}, \theta_{\text{init}} = \theta)$
  until convergence

3.4 Experiment Setup

For our experiments, we evaluate the performance of the ProMP stroke controller in the lab setup illustrated in Figure 1a. We initialize the ProMP parameters by training it on a dataset of successful demonstration obtained from a manually engineered stroke (Figure 1b), this serves as our base primitive. To evaluate the proposed fine-tuning algorithm (Section 3.3) we collect a dataset by running the base primitive. For each ball launched, we record the joint states to a ROS[18] bag file and store the associated reward based on criteria listed in Table 1. We segment the trajectories from the bag files by analyzing the maximum of all joint velocities to determine points of inflection, this constitutes the start and end of the trajectory. The hit phase parameter is chosen based on where the end-effector approximately crosses the pre-specified hit plane. Figure 2 illustrates the process of trajectory extraction. We also evaluate the impact of the number of trajectories used for refining the



(a) (b)

Figure 1: (a) Overview of the lab setup. (b) Wheelchair Tennis Robot executing a stroke.

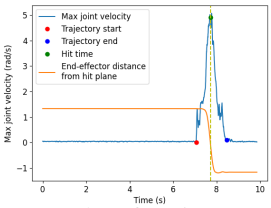


Figure 2: Example of trajectory segmentation from recorded joint states.

Table 1: Reward Criteria

criteria	**reward**
miss by a large margin	0
miss but close ($\leq 5\,\mathrm{cm}$)	0.25
hit but not good enough	0.5
good hit (hit side pillar)	1
good hit (above net)	2

primitives by comparing the performance of the primitives trained on datasets of 20 and 50 trajectories. Performance measures hit rate, success rate (success corresponds to good hits), and the average reward on a consecutive sequence of ball launches.

4 Results

We report the base primitive performance and the performance post-refinement in Table 2 based on the setup described in Section 3.4. Performance is reported over a consecutive run of 10 balls.

# trajectories	0 (base primitive)	20	50
Hit Rate	60%	40%	50%
Success Rate	40%	40%	40%
Avg. Reward	0.75	0.85	0.85

Table 2: Performance over the number of trajectories used for refining the base primitive.

5 Conclusion

We have successfully demonstrated a safely executed ground strike primitive on a wheelchair tennis robot. We proposed a formulation to fine-tune learned primitives online and conducted an evaluation. While we do not observe significant improvements in terms of success rates with fine-tuning, we do see a small increase in the average feedback rewarded to the primitives, so most balls have been missed by small margins. Future work in this direction can consider different reward designs and study how the choice of the reward impacts the learned primitive.

Future Work In future, we would like to explore methods to improve upon learned primitives to 1) ensure safe behavior and 2) increase task performance. Improving learned robot behavior has relations across Active Learning with Human Feedback [4], Reinforcement Learning of robot skills [19, 20], and Human-Robot Interaction. We would additionally like to consider several feedback paradigms in improving robot motion including natural language, kinesthetic teaching, third-person demonstration, etc.

Acknowledgments

References

[1] D. Calkins. An overview of robogames [competitions]. *IEEE Robotics & Automation Magazine - IEEE ROBOT AUTOMAT*, 18:14–15, 03 2011. doi:10.1109/MRA.2010.940146.

[2] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, and H. Matsubara. Robocup: A challenge problem for ai. *AI magazine*, 18(1):73–73, 1997.

[3] D. Büchler, S. Guist, R. Calandra, V. Berenz, B. Schölkopf, and J. Peters. Learning to play table tennis from scratch using muscular robots. *IEEE Transactions on Robotics*, 2022.

[4] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009. ISSN 0921-8890. doi:https://doi.org/10.1016/j.robot.2008.10.024. URL https://www.sciencedirect.com/science/article/pii/S0921889008001772.

[5] J. Kober, K. Mülling, O. Krömer, C. H. Lampert, B. Schölkopf, and J. Peters. Movement templates for learning of hitting and batting. In *2010 IEEE International Conference on Robotics and Automation*, pages 853–858, 2010. doi:10.1109/ROBOT.2010.5509672.

[6] K. Mülling, J. Kober, O. Kroemer, and J. Peters. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32 (3):263–279, 2013. doi:10.1177/0278364912472380. URL https://doi.org/10.1177/0278364912472380.

[7] S. Gomez-Gonzalez, G. Neumann, B. Schölkopf, and J. Peters. Using probabilistic movement primitives for striking movements. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 502–508, 2016. doi:10.1109/HUMANOIDS.2016.7803322.

[8] S. Gomez-Gonzalez, G. Neumann, B. Schölkopf, and J. Peters. Adaptation and robust learning of probabilistic movement primitives. *IEEE Transactions on Robotics*, 36(2):366–379, 2020.

[9] L. Chen, R. Paleja, M. Ghuy, and M. Gombolay. Joint goal and strategy inference across heterogeneous demonstrators via reward network distillation. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, pages 659–668, 2020.

[10] L. Chen, R. Paleja, and M. Gombolay. Learning from suboptimal demonstration via self-supervised reward regression. In J. Kober, F. Ramos, and C. Tomlin, editors, *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pages 1262–1277. PMLR, 16–18 Nov 2021. URL https://proceedings.mlr.press/v155/chen21b.html.

[11] X. Chen, Z. Zhou, Z. Wang, C. Wang, Y. Wu, and K. Ross. Bail: Best-action imitation learning for batch deep reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18353–18363. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d55cbf210f175f4a37916eafe6c04f0d-Paper.pdf.

[12] C.-A. Cheng, X. Yan, N. Wagener, and B. Boots. Fast policy learning through imitation and reinforcement. In *UAI*, 2018.

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

[14] **Anonymous**. Athletic mobile manipulator system for robotic wheelchair tennis. *arXiv preprint arXiv:2210.02517*, 2022.

[15] T. Moore and D. Stouch. A generalized extended kalman filter implementation for the robot operating system. In *Intelligent autonomous systems 13*, pages 335–348. Springer, 2016.

[16] M. S. Kovacs. Movement for tennis: The importance of lateral training. *Strength & Conditioning Journal*, 31(4):77–85, 2009.

[17] W. B. Knox and P. Stone. Interactively shaping agents via human reinforcement: The tamer framework. In *Proceedings of the fifth international conference on Knowledge capture*, pages 9–16, 2009.

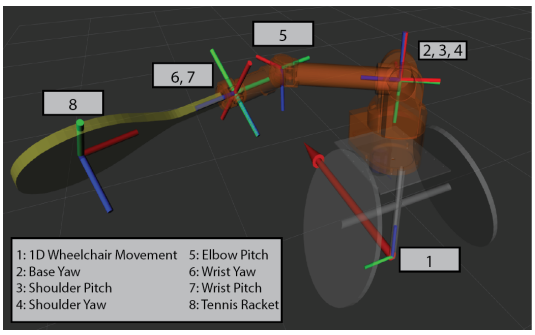
[18] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[19] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 40:698 – 721, 2021.

[20] J. Carvalho, D. Koert, M. Daniv, and J. Peters. Residual robot learning for object-centric probabilistic movement primitives. *ArXiv*, abs/2203.03918, 2022.

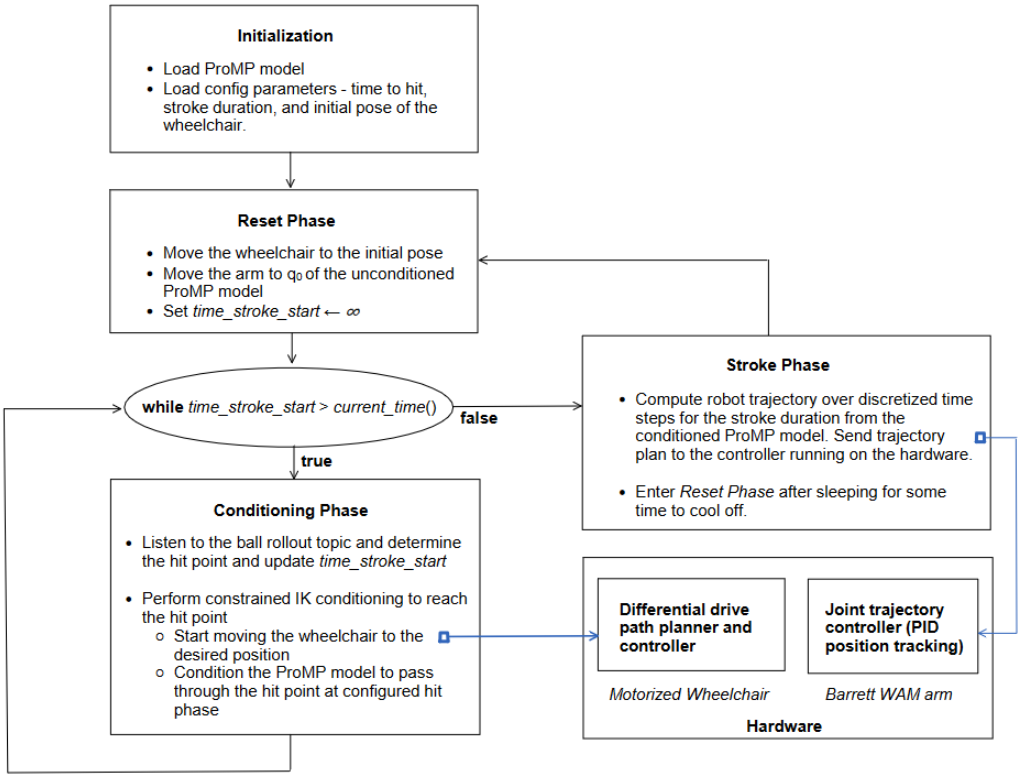
A Kinematic Model of the Robot

Figure 3: The kinematic model used for hit-point conditioning of primitive



B Details of the stroke controller

Figure 4: Flowchart depicting the state-machine of the stroke controller



Algorithm 2 Constrained Inverse-Kinematics for conditioning of stroke primitives

Require:

- Desired hit point position x_d in local mobile-base frame
- seed configuration q_{seed}
- Forward kinematics model $f : (r, q) \rightarrow \mathbb{R}^3$ and the corresponding manipulator jacobian $J(r, q)$, where r is the wheelchair position and q is the joint state
- lower (`LL`) and upper (`UL`) limits of allowed wheelchair and joint movements.

Let $q_c = q_{\text{seed}}$
Current end-effector position $x_c = f(q_c, 0)$
Let $\texttt{net\_dq} = 0$ and $\texttt{net\_dr} = 0$
repeat
    $\Delta r, \Delta q \leftarrow J(\texttt{net\_dr}, q_c)^{\dagger}(x_d - x_c)$
    $\texttt{net\_dr}, \texttt{net\_dq} \leftarrow \texttt{ClipedIncrement}(\texttt{net\_dr}, \texttt{net\_dq}, \Delta r, \Delta q, \texttt{LL}, \texttt{UL})$
    $q_c \leftarrow q_{\text{seed}} + \texttt{net\_dq}$
until iteration $<$ N **and** !$\texttt{allclose}(x_c, x_d, \epsilon)$

command wheelchair to move: $\texttt{net\_dr}$
condition stroke primitive to pass through: $q_{\text{seed}} + \texttt{net\_dq}$