MotorNet-Probabilistic: Stochastic Model-based RL for Reach Control of a Realistic Arm

Amirhossein Asadian
Western Institute for Neuroscience
aasadia3@uwo.ca

Pranshu Malik
Western Institute for Neuroscience
pranshu.malik@uwo.ca

Abstract

We are inherently able to produce movements that help us achieve our goals in the world. In order to do so, we make use of our knowledge of the environment and our own body to produce optimal-looking movements. This self- and environmental-knowledge can be thought to be encoded in the form of internal models, which are central to inferring sensorimotor control. These models can be forumlated various ways, implicitly or explicitly, depending on the modelling perspective and level of abstraction (e.g. control-theoretic modules, algorithmic procedures, or implicit transformations in artificial neural networks). This work aims to externalize the iterative internal models for the control of a realistic arm model by using the model-based reinformcent learning (RL) framework, which is contrary to the predominant and performant recurrent neural network (RNN) controllers wherein the internal models and control hierarchy are entirely obscured. In this study, we present the integration of a realistic and differentiable arm model and then train a baseline model-free stochastic RL policy to control the arm. We also propose a model-based RL formulation, through which we may be able to ask further scientific questions on the organization of naturalistic control, as well as achieve better artificial control methods. Altogether, these form the first ideas for MotorNet-Probabilistic (or MotorNet-Pro in short).

Keywords: sensorimotor control, model-based continuous feedback control, stochastic policy

1. Introduction

o Why is it an important problem? o Why can't any of the existing techniques effectively tackle this problem? o What is the intuition behind the technique that you developed?

Motor behaviors that may come intuitively to us are not easy to achieve synthetically. The body is a complex system, all the way from the anatomy – featuring underactuated and nonlinear musculoskeletal dynamics – to noisy and delayed sensorimotor modalities – with indirect and distinct feedback spaces as well as distributed control structures. To overcome such complexities, our bodies use internal models to predict and control movements in an adaptive manner, producing optimal and robust control even in unpredictable and uncertain environments (McNamee and Wolpert 2019). However, our model-based control policies, can also display model-free *strategic* changes to adapt to unexpected environmental conditions even though reaction to perturbations through reflexes is model- and policy-based (Crevecoeur, Scott, and Cluff 2019). This shows the need for a more general framework and formulation that models the organization of control as not as a single model-based policy output but also policy modulation, adaptation (e.g. continual learning), and online multi-level planning and control. Added to that, the framework should be grounded in stochastic control, as this is a key feature of naturalistic movements.

To this end, most neuroscientifically-motivated modelling attempts have included an RNN-based controller, which lends itself to a neuron-population level analyses of activity patterns which are similar to what is done on experimental recordings from the motor-related cortical areas. MotorNet by (Codol et al. 2023) is one such realistic arm model coupledwith a controller that is trained in delayed continuous feedback and continuous action spaces, following the optimal feedback control (OFC) framework laid out by (Todorov 2004, Scott 2004). This is a promising direction, especially when studying neural dynamics in cortical control, however, we are interested in the organizational aspect of motor control in a more *interpretable* fashion, for which the model-based RL framework is more suitable. In a study by (Almani and Saxena 2022), performant control is elegantly achieved with an RNN actor in soft actor-critic (SAC) by (Haarnoja et al. 2018) so that the policy can learn to do online control by maintaining a stateful representation of the ongoing movement and task demands. However, similar to MotorNet, the purpose of this controller was to investigate proximity of neural activity in the
RNN-pool and with neural recordings of the motor cortex. A study by (F. Fischer et al. 2021) also trained a SAC policy, but for a much simpler arm model, and was focussed on explaining certain movement laws that arise through in the stochastic policy that prove certain properties of the assumed noise model, proposing that the biomechanical system can be understood at a simpler scale. However, we are interested in how control is produced in the first place, which would also help address many open questions on the same front as well as provide clues on how to achieve better robot control.

2. Problem Formulation

Hi o Brief review of previous work concerning this problem (i.e., the 4-8 papers that you read) o Brief description of the techniques chosen and why o Describe the technique that you developed o Brief description of the existing techniques that you will compare to

Both forward and inverse models are used in control: policy usually acts as an iterative inverse model that is directed towards the goal. We would like to fuse the feedforward capability (inv model) to use feedback control (forward model) to be able to create an online optimal robust controller; for this, the policy should have distilled the knowledge about the stateprediction and subsequent control – the dreaming architecture is good for that;

For that, it should have an idea of what future states (short term) I might be in, and thus plan for those … miniplans/ chunks. For this, a multilevel policy net can be trained; one for the objective space and short-term plan and another for the control in that; this enables implicit chunking and subsequent local robust control; this is also a good way to train an RNN to replicate this – easier way to bake-in interpretability into the RNN models.

Planning and control in latent feedback space: all pixel and dreaming models will be useful. Planning predicts the future statespace of potential interest and then control is done on that space by understanding the reward/objective landscape for the short landscape; this iterates cleanly as states progress, quite naturally.

Latent predicted future feedback manifold/landscape on which the actor level of the policy is trained to control; (Gosztolai et al. 2023) (among various others) discuss this in RNNs and this is also a dominant idea in neuroscience literature under the banner of optimal feedback control (Todorov 2004, Scott 2004).

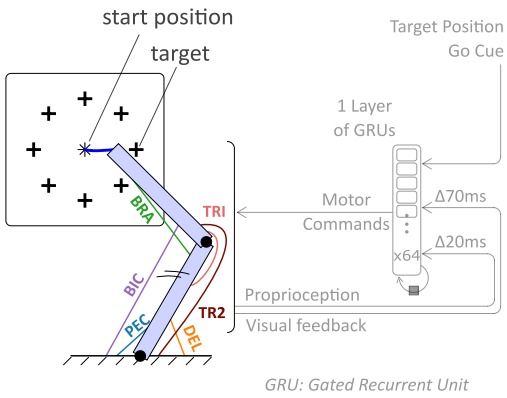

Figure 1: MotorNet Plant

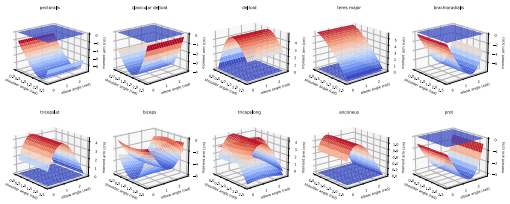

Figure 2: Moment Arms

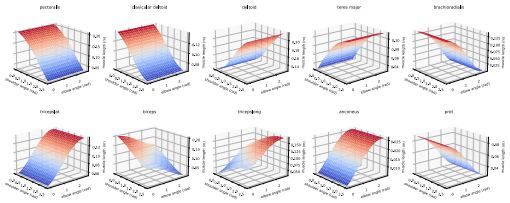

Figure 3: Muscle Lengths

Noteably, the forward state prediction can act at different timescales, a.nd for timepoints with complex dynamics, the states can evolve with a zoomed in parametrization so that objectively lesser feedback space is included in the plannning/inferring control, but it leads to more precise manuevers. This is similar to how numerical integration of ODEs work; closer to more error-prone timepoints, there are more iterations of the solver. It seems logical that online optimal feedback control should do the same.

Should also relate to some *knowledge distillation* ideas where the forward/transition model applies to first level of the policy and the reward model applies to the second level of the policy to learn an evolving local reward landscape. Should perturbations happen, and the actor be thrown outside the current state landscape, it will automatically have to reiterate the bilevel processing, but if the perturbation is small enough and it fits the current local computation, then it should be able to produce a robust correction based on the reward landscape with subsequent recomputation at the first level within a few more steps. Stiffness control can also be introduced during training to include the ability of the model to adapt to new environment dynamics (Shadmehr and Mussa-Ivaldi 1994, Conditt, Gandolfo, and Mussa-Ivaldi 1997) or also produce like in (Crevecoeur, Scott, and Cluff 2019). Since the literature on this front is sparse in neuroscience, modeling this adhoc is one way how such a model can be used immediately in sensorimotor neuroscience to generate new hypotheses and experiments, test and then update our understanding. Similar to the model-free response above, we can similarly also use this bilevel policy to study the emergence of multiple strategies (and their selection) in motor control.

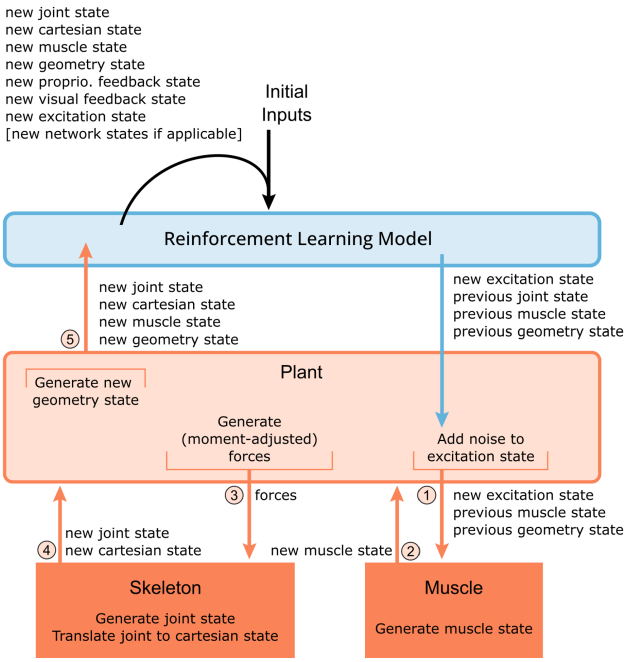

Figure 4: MotorNet-Pro Architecture

- All of these "snapshots" can be replayed to train an RNN, which will regularize learning of representations and provide us some control over interpretability of the same – going from encoded and learnt rules to a universal differential equation.
- Such a control policy can then also be applied to robots and control of other limbs and the whole body.

2.1. Related Work

This has been done. More conventional and early approaches first began with the (T. P. Lillicrap et al. 2015) algorithm, then many came on to use MPC to have an implicit policy requiring a sub-routine to get the current action by solving the optimal-control problem (as a nonlinear program NLP) at each time step (or state). Even though MPC-based policy and value functions approximation may offer a high explainability about the policy behaviour in addition to being equipped with a broad set of theoretical tools for formal verification of safety and stability e.g. (Chua et al. 2018), it is not a good fit for our problem. This is computationally expensive and especially not meeting our criteria for explicit representation of the policy and internal models. Give MPC formulation and parametrization by policy param θ. By using off-the-shelf non-differentiable solvers, we have the added disadvantage of losing the ability to backpropagate gradients through the policy. give citations too: PSRL etc..

Some model based RL approaches: (Janner et al. 2019, Clavera, V. Fu, and Abbeel 2020, Fan and Ming 2020, Amos et al. 2020). Some other ideas are: (Nikishin et al. 2021). (Chen et al. 2021) also looks promising, but Pytorch implementation – needs to be done in Tensorflow. After this, this would be our goto baseline for not only comparison but also gaining insights into the learnt policy. What is suggested by and contemporary motor control theories (Selen, Corneil, and Medendorp 2023) is that there is benefit in including a small ensemble of planning models to eventually reduce the variance. This could also allow for modelling several strategies in these policy/actor models.

There are also image-based control algorithms that infer from a latent state, this should also generally apply, however, to our best knowledge the literature does not seem to have shown that, particularly it would be nice to see if the latent space is better to generalize (A. X. Lee et al. 2019, Hafner, T. Lillicrap, I. Fischer, et al. 2018, Yarats et al. 2021) pixel-to-control. However, most of engineering innovations made in the algorithms (e.g. displacing image by ± 4 pixels) seem to also not fit any biological detail involving change of feedback structure or processing. But, noteably, PlaNet (Hafner, T. Lillicrap, I. Fischer, et al. 2018) makes an important Recurrent State Space Model (RSSM) that predicts forward in latent space split the state into stochastic and deterministic parts, allowing the model to robustly learn to predict multiple futures. A different, but similar model strategy, Dreamer, was developed by (Hafner, T. Lillicrap, Ba, et al. 2019) that, through its formulation, is able to analytically calculate and exploit gradients from state transitions to speed up the RSSM-model learning. This was implemented for real-time, real-world learning in robotics by (Wu et al. 2022). Adapting these latent-space ideas for the bilevel distilled policy network is a promising direction.

Along the topic of planning and control in latent space, work by (Ghugare et al. 2022) on aligning these latent space models each of which were previously with their own auxiliary objectives, making the submodel alignment unclear. Single objective which jointly optimizes a latent-space model and policy to achieve high returns while remaining self-consistent. This is a very close analogue to our bilevel policy formulation, as it a model that predicts in representation space for the feedback instead of high-dimensional observation space, and a policy that acts based on those representations.

This is different than the proposed conceptual formulation Figure 5 is the implicit inclusion of online planning through continual state prediction in the latent space, to a variable extent conditioning the actor on the goal(s) thereby setting the context for action. This naturally falls into the scope of, however, more thought is needed on the mechanism and appropriate setup and formulation for adaptive contraction and expansion of feedback prediction space. An alternative would be to possible that the transformer and variational autoencoder (VAE) based models.

MBPO: ensemble of models trained using MLE, trained on environment data only and then used to produce cheaper rollouts in each episode to train the policy further from the sampled states. (some criticism from MAAC and PSRL). When the model class is misspecified or has a limited representational capacity, model parameters with high likelihood might not necessarily result in high performance of the agent on a downstream control task (Nikishin et al. 2021). Nishkin et al. propose a method to address this issue by … What it may mean for us…

2.2. Exploratory Formulation

Based on the recent innovation novel algorithm Action Chunking with Transformers (ACT) which reduces the effective horizon by simply predicting actions in chunks, we would also like that as a generalization of the policy formulation (Zhao et al. 2023). This can also be supported by the neuroscience literature in the sense that a near short horizon plan is known at the time of execution although the cost landscape for actions beyond the current may evolving in an online manner. Bringing this formulation into the RL framework can be hard and for now is beyond the scope of this report. But this is something we will definitely try to incorporate simply in the future.

Similar to how optimal control is done in the brain, with a higher-level context coming in, also recently demonstrated to be critical to a (Schimel, Kao, and Hennequin 2023). This highlights the need for a plan and also the fits the introspection of having constructed a general plan or strategy of movement before.

In our model, actions are defined as the activation of each muscle, which results in a 10-dimensional action space. The state is determined by a variety of observations, including proprioceptive feedback (muscle length and velocity for each muscle), visual feedback (position of the arm), perceived target, and muscle activations (affordance copy of each muscle), resulting in a 34-dimensional state space. To account for the delay in information processing in the central nervous system, each feedback changes the state after a certain amount of time. Specifically, we have set a proprioceptive delay of 20 ms and a visual delay of 70 ms. Additionally, to account for noise in neural pathways, we have added a normal noise with a standard deviation of 0.01 to each feedback information (proprioceptive feedback, visual feedback of arm location, and visual feedback of target location).

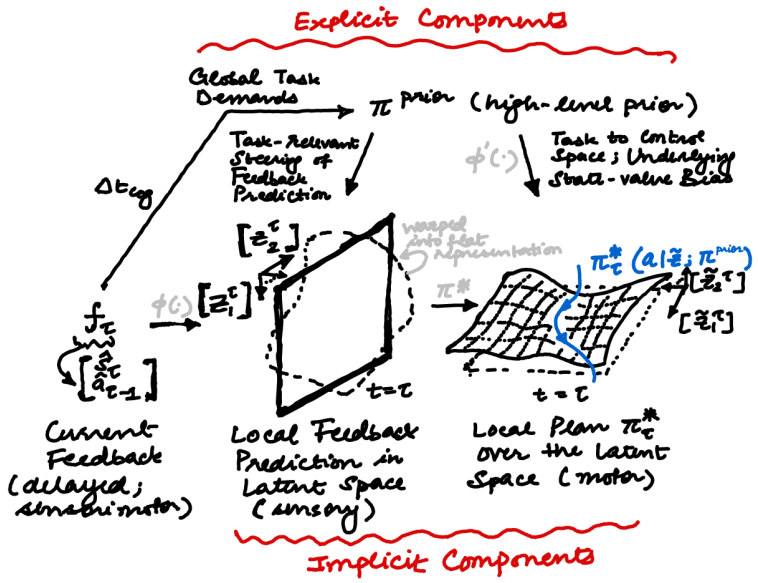

Figure 5: Latent OFC Policy

3. Baseline Algorithm and Considerations

As a starting point, we have decided to implement the soft actor-critic (SAC) algorithm for policy optimization. SAC is a popular model-free deep reinforcement learning algorithm used for continuous control tasks. It is based on the maximum entropy reinforcement learning framework, which aims to maximize the expected return of a policy while also maximizing the entropy of the policy distribution. By maximizing the entropy, SAC encourages exploration and prevents the policy from getting stuck in local optima.

At each iteration, SAC performs two main steps: policy evaluation and policy improvement. In the policy evaluation step, SAC estimates the state-action value function Q_π, which represents the expected return starting from a given state-action pair under the current policy. The Q-function is updated using the Bellman backup operator:

$$Q(s,a) = \mathbb{E}[r + \gamma V(s')]$$

where r is the immediate reward, γ is the discount factor, and $V(s')$ is the value function of the next state. The policy improvement step involves training a stochastic policy that minimizes the expected Kullback-Leibler (KL) divergence between the current policy and the exponential of the Q-function minus a value function V_π. The objective function for policy improvement is given by:

$$J(\pi) = \mathbb{E}\Big[Q_{\pi}(s,a) - \alpha \log(\pi(a|s))\Big]$$

where α is a temperature parameter that controls the balance between maximizing the expected return and maximizing the entropy (Haarnoja et al. 2018, Janner et al. 2019).

We have chosen SAC because it is easy to implement and has shown promising results in a variety of domains. While we will be using SAC as our baseline algorithm, we are also considering the approach proposed by Chen et al. (2021) for future use. They have suggested an ensemble of SAC models combined with in-target minimization as an improved method, and we will keep this in mind for potential future optimization.

It is worth noting that our goal is not only to achieve high performance levels and quick convergence but also to develop a model-based policy that is interpretable and can provide insights into the underlying processes. We believe that this combination of performance and interpretability could serve as a valuable oracle for recurrent neural networks and model trans-cortical computation for movement control, including long-latency reflexes (LLRs) and optimal and robust voluntary motion.

Algorithm 1: Soft Actor-Critic

Initialize parameter vectors ψ, $\bar{\psi}$, θ, ϕ.

for each iteration **do**

  for each environment step **do**

    $\boldsymbol{a}_t \sim \pi_{\phi}(\boldsymbol{a}_t|\boldsymbol{s}_t)$

    $\boldsymbol{s}_{t+1} \sim p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$

    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\boldsymbol{s}_t, \boldsymbol{a}_t, r(\boldsymbol{s}_t, \boldsymbol{a}_t), \boldsymbol{s}_{t+1})\}$

  end for

  for each gradient step **do**

    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_{\psi} J_V(\psi)$

    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$

    $\phi \leftarrow \phi - \lambda_{\pi} \hat{\nabla}_{\phi} J_{\pi}(\phi)$

    $\bar{\psi} \leftarrow \tau\psi + (1-\tau)\bar{\psi}$

  end for

end for

3.1. Preliminary Results

We extensively trained our model over 10,000 episodes to achieve the target within one second. Figure 5 showcases the learning curve, and the best episode rewarded our model with -249.65. Interestingly, we observed optimal performance after 2000 episodes, but then a gradual decline occurred. This decline may be attributed to "catastrophic forgetting," where the model's success causes it to forget what failure looks like. As a result, the model predicts high values for all states and features, regardless of their relevance.

When the model encounters unexpected situations with incorrect predicted values, the error rate can be high, and recovery can be challenging. Additionally, the model may incorrectly link features of the state representation, making it difficult to distinguish between various parts of the feature space. This creates unusual effects on the model's learning about the values of all states. While the RL model may behave

incorrectly for a few episodes before relearning optimal behavior, it may also break down entirely and never recover.

Catastrophic forgetting is an active research area, and one potential solution is to set aside some percentage of replay memory with the initial poor-performing random exploration. We are actively considering various approaches to tackle this issue.

Other implementations exist for reaching in environments with more than one arm, such as the one using 22 double-joint arms with a 33-dimensional observation space (cite git). However, they control simplistic arms lacking realistic and non-linear features. They utilize a single agent to control multiple arms in order to promote generalization. In contrast, we are exploring the use of the DDPG algorithm to enable the model to reach different targets from varying starting positions. Our primary objective is to gain insight into the internal models of this agent as the model learns to reach different locations in a 2D space.

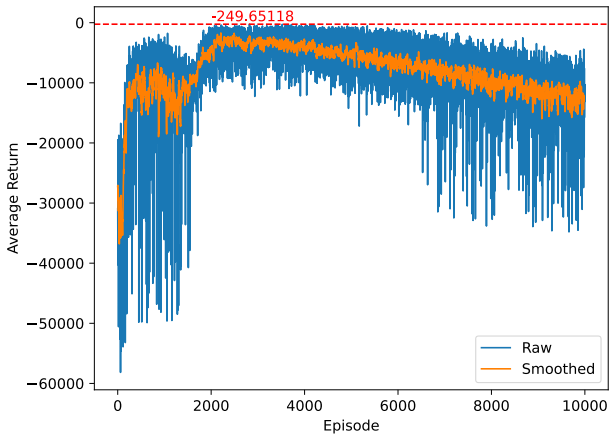

Figure 6: SAC Learning Curve

4. Discussion and Future Work

Many possibilities. First to improve the baseline, we can do Dyna-style (Sutton 1990) virtual rollouts to improve the policy, however, doing this early on may be detrimental to the learning as the model will not be accurate enough. Also adding virtual rollouts to the replay buffer may be problematic for the same reason. Some theoretical work on need- and risk-aware model learning also exists, such as (Abachi, Ghavamzadeh, and Farahmand 2021), but it will need to be adapted to evolving stochastic continuous control framework. Overall, this approach is mostly an empirical tweak and its efficacy will depend on the extent of fine-tuning the baseline algorithm and its features. One missing thing is the inclusion of signal dependent noise in the MotorNet plant model.

We can use the same points of improvement discussed above for our original algorithm (to be developed).

5. Conclusion

Test o What is the best technique? o Is any technique good enough to declare the problem solved? o What future research do you recommend?

Going forward, we have gotten the intuition that an ensemble will help for at least the action-level policy net, which also automatically allows for informed variance during control and maybe even different strategies. Then, we also see that latent space planning is a promising avenue, which gives our problem formulation some basis. However, we would also like to keep the eventual model simplistic without too many engineering tricks to keep it well motivated and based in neuroscience. More than model's time-to-performance, we are interested in its design details and suitability to scientific ideas, as well as eventual performance and overall interpretability.... Modularization to test different hypotheses (adhoc or not), like model-free and model-based robust control,stiffness strategies, etc. will also be an important factor in the final design. What we hope to achieve is a mildly speculative model of robust optimal feedback control for a realistic arm, and more importantly, one that will help us simulate various hypotheses and organization of hierarchical and nested control systems in the nervous system.

Notes

Our work (in-progress) on MotorNet-Pro can be found in the following repository: https://github.com/asadian98/motornet-pro. We would also like to highlight the brand-new Typst open-source typesetting system using which this report was gracefully written.

References

Abachi, R., Ghavamzadeh, M., & Farahmand, A.-m. (2021). Policy-aware model learning for policy gradient methods. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.2003.00030

Almani, M. N., & Saxena, S. (2022, July). Recurrent neural networks controlling musculoskeletal models predict motor cortex activity during novel limb movements [Paper presentation]. In *IEEE Engineering in Medicine & Biology Society (EMBC)*.

Amos, B., Stanton, S., Yarats, D., & Wilson, A. G. (2020, August). On the model-based stochastic value gradient for continuous reinforcement learning. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.2008.12775

Chen, X., Wang, C., Zhou, Z., & Ross, K. (2021, January). Randomized ensembled double q-learning: learning fast without a model. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.2101.05982

Chua, K., Calandra, R., McAllister, R., & Levine, S. (2018, May). Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.1805.12114

Clavera, I., Fu, V., & Abbeel, P. (2020, May). Model-augmented actor-critic: backpropagating through paths. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.2005.08068

Codol, O., Michaels, J. A., Kashefi, M., Pruszyski, J. A., & Gribble, P. L. (2023). Motornet: a python toolbox for controlling differentiable biomechanical effectors with artificial neural networks. *Biorxiv*. https://doi.org/10.1101/2023.02.17.528969

Conditt, M. A., Gandolfo, F., & Mussa-Ivaldi, F. A. (1997, July). The motor system does not learn the dynamics of the arm by rote memorization of past experience. *J. neurophysiol.*, 78(1), 554–560.

Crevecoeur, F., Scott, S. H., & Cluff, T. (2019, October). Robust control in human reaching movements: a model-free strategy to compensate for unpredictable disturbances. *J. neurosci.*, 39(41), 8135–8148.

Fan, Y., & Ming, Y. (2020, November). Model-based reinforcement learning for continuous control with posterior sampling. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.2012.09613

Fischer, F., Bachinski, M., Klar, M., Fleig, A., & Müller, J. (2021, July). Reinforcement learning control of a biomechanical model of the upper extremity. *Sci. rep.*, 11(1), 14445.

Ghugare, R., Bharadhwaj, H., Eysenbach, B., Levine, S., & Salakhutdinov, R. (2022, September). Simplifying model-based rl: learning representations, latent-space models, and policies with one objective. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.2209.08466

Gosztolai, A., Peach, R. L., Arnaudon, A., Barahona, M., & Vandergheynst, P. (2023, April). Interpretable statistical representations of neural population dynamics and geometry. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.2304.03376

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018, January). Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.1801.01290

Hafner, D., Lillicrap, T., Ba, J., & Norouzi, M. (2019, December). Dream to control: learning behaviors by latent imagination. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.1912.01603

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., & Davidson, J. (2018, November). Learning latent dynamics for planning from pixels. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.1811.04551

Janner, M., Fu, J., Zhang, M., & Levine, S. (2019, June). When to trust your model: model-based policy optimization. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.1906.08253

Lee, A. X., Nagabandi, A., Abbeel, P., & Levine, S. (2019, July). Stochastic latent actor-critic: deep reinforcement learning with a latent variable model. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.1907.00953

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015, September). Continuous control with deep reinforcement learning. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.1509.02971

McNamee, D., & Wolpert, D. M. (2019, May). Internal models in biological control. *Annu rev control robot auton syst*, 2, 339–364.

Nikishin, E., Abachi, R., Agarwal, R., & Bacon, P.-L. (2021, June). Control-oriented model-based reinforcement learning with implicit differentiation. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.2106.03273

Schimel, M., Kao, T.-C., & Hennequin, G. (2023, April). When and why does motor preparation arise in recurrent neural network models of motor control? *Biorxiv*, 2023. https://doi.org/10.1101/2023.04.03.535429

Scott, S. H. (2004, July). Optimal feedback control and the neural basis of volitional motor control. *Nat. rev. neurosci.*, 5(7), 532–546.

Selen, L. P. J., Corneil, B. D., & Medendorp, W. P. (2023, April). Single-Trial dynamics of competing reach plans in the human motor periphery. *J. neurosci.*, 43(15), 2782–2793.

Shadmehr, R., & Mussa-Ivaldi, F. A. (1994, May). Adaptive representation of dynamics during learning of a motor task. *J. neurosci.*, 14(5 Pt 2), 3208–3224.

Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming [Paper presentation]. In *Machine learning proceedings 1990*. Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-1-55860-141-3.50030-4

Todorov, E. (2004, September). Optimality principles in sensorimotor control. *Nat. neurosci.*, 7(9), 907–915.

Wu, P., Escontrela, A., Hafner, D., Goldberg, K., & Abbeel, P. (2022, June). Daydreamer: world models for physical robot learning. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.2206.14176

Yarats, D., Fergus, R., Lazaric, A., & Pinto, L. (2021, July). Mastering visual continuous control: improved data-augmented reinforcement learning. *Arxiv*. https://doi.org/https://doi.org/10.48550/arXiv.2107.09645

Zhao, T., Kumar, V., Levine, S., & Finn, C. (2023, March). Learning fine-grained bimanual manipulation with low-cost hardware. *Github*. https://tonyzhaozh.github.io/aloha/