2020 8th IEEE International Conference on Biomedical
Robotics and Biomechatronics (BioRob)
New York, USA. Nov 29 - Dec 1, 2020

Towards End-to-End Control of a Robot Prosthetic Hand via Reinforcement Learning*

Mohammadreza Sharif[1], Deniz Erdogmus[1], Christopher Amato[2], and Taskin Padir[1]

Abstract— **Robot prosthetic hands intend to replicate one's lost abilities through intuitive control. So far, control methods that rely heavily on the human input such as Electromyographic (EMG) and Electroneurographic (ENG) signals have been predominantly studied. However, these methods face issues such as lack of robustness resulting in abandonment of this technology by the users. There is a need for a paradigm shift in the robot prosthetic hand control methods. With this regard, we propose an end-to-end learning of control policy for a robot prosthetic hand through reinforcement learning. Imitation learning has been fostered to help with the sparse reward setting in the hard-to-explore state-space of the problem. The results in simulation show the feasibility of successfully learning an end-to-end policy for grasping objects by robot prosthetic hands, potentially increasing robustness for grasp control of future robot prosthetic hands.**

I. Introduction

There are about 1.6 million people in the United States alone with at least one limb amputation, from whom about 500 thousand suffer major or minor upper extremity amputation. This number is expected to reach double its amount by 2050 [1]. Creating prostheses to retrieve part of the lost ability for amputees has been a research topic for a long time in the human history. Functionality, intuitiveness and ease of use, as well as robustness are three major factors that drive the research in this field. State-of-the-art human-in-the-loop control methods rely heavily on human input, such as EMG and ENG signals, and infer human intent by pattern recognition techniques. However, the proposed solutions have rarely penetrated into the market, and even the older EMG-based controllers such as on-off methods have a high rejection rate, as statistics show [2], [3]. There is a need for a new paradigm of control, in which robot has more autonomy and human is less relied on. This work proposes a reinforcement learning (RL) with imitation learning (IL) framework for learning to grasp objects without relying on EMG signals.

The motivation for the proposed method is specialized environments in which we can have access to signals other than just the EMG signal, e.g. camera vision and hand tracking information. These environments can be built, for example, for work cells specialized for amputees in factories or amputees' homes. Our purpose is to provide a solution that does not need EMG information processing and that can provide faster and more reliable interaction with the environment by an amputee. For this, the amputee does not need to wear a new prosthetic in the specialized environment, but the myoelectric hand that the amputee already uses can switch to our controller once located in the specialized place.

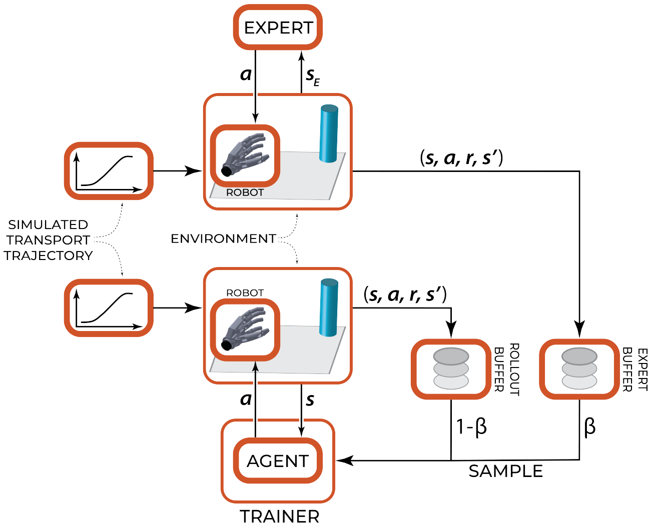


Fig. 1: The training loop of our RL problem. Expert and agent transitions are stored in expert and rollout buffers, respectively. For training, a mixture of samples from both buffers is used.

EMG is the main human signal which is relied on for controlling robot prosthetic hands. The main issue with EMG controllers is lack of robustness in real life. Lack of robustness is mainly attributed to the deterministic or stochastic variations between the ideal lab settings and real-life conditions, such as electrode number or shift [4], change of skin-electrode impedance over time, muscle fatigue [5], crosstalk effect [6], and stump posture change [7]. Frequent calibrations is a practical solution for the lack of robustness issue for commercial products, although reported as a source of user inconvenience [8]. EMG-based control also puts excessive mental and physical burden on the amputee to run the hand. Over-emphasizing an EMG pattern to be distinguished as a certain grasp or to maintain a grasp causes early fatiguing of wearers of myoelectric hands [9]. Moreover, users of prosthetic hands with EMG-based control need intense training sessions which is considered as another issue for EMG-based methods [10].

To solve the above issues, it has been suggested to include

*This work was supported by NSF under the award number 1544895, 1928654, 1935337 and 1944453.

[1] Electrical and Computer Engineering Department, Northeastern University, Boston, MA 02115, USA {sharif.mo,deniz,tpadir}@northeastern.edu

[2] Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115, USA c.amato@northeastern.edu

978-1-7281-5907-2/20/$31.00 ©2020 IEEE

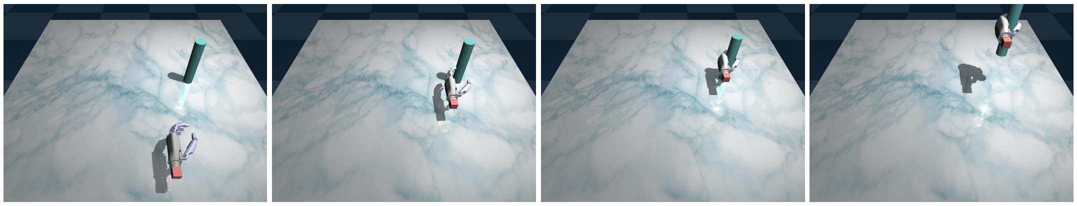


Fig. 2: Four frames of the trained policy performing a successful grasp.

signals besides EMG into the inference loop [11] in a hybrid way. Integration of EMG with gaze information [12], RGB camera [13], and inertial measurement units (IMUs) [14] have already shown improvements in the overall control process. There are also works which do not rely on EMG processing for control at all. Kim et al. [15] used egocentric cameras to control a wearable exo-glove using deep neural networks for quadriplegic subjects grasping an object. A particle-filter-based method is proposed in [16] to detect human intention for grasp based on hand trajectory alone. Ficuciello et al. [17] controls a robot hand in synergy space for grasping an object using IL and then RL to improve the results.

RL is a method to learn from interaction with environment in real world or in simulation. RL enables learning from data to deal with complexity of robot environments, without actually coding for every possible case. Recent applications of RL has demonstrated promising results in robot control such as robot manipulation [18], [19] and dexterous manipulation [20]–[22]. Among the drawbacks of RL-based methods are data-inefficiency, unstable training process, and exploration with little or no intelligence. There have been efforts to address these drawbacks by methods such as imitation learning [21] and maximum entropy reinforcement learning [23], [24].

Our work presented here introduces an RL-based shared-autonomy framework to control a robot prosthetic hand, using only the human hand trajectory as the input. Imitation learning (IL) is used to guide the exploration of the robot in the state space. It should be noted that this work is different from [17], in which IL and RL are used to initialize and refine synergy coefficients of a prosthetic hand and then use a controller to reach those values without considering the environment dynamics along the user hand transport trajectory. In our approach, we use RL to find an end-to-end controller that controls the hand actuators directly from the measurements with regards to the environment dynamics. By end-to-end learning, we mean learning actuator commands directly from measured system states. Thus, the main contributions of this work are: (1) formulating the problem of robot prosthetic hand control as an RL problem, and (2) introducing an IL-guided RL framework to learn to control a robot prosthetic hand in an end-to-end manner in simulation.

II. Methodology

A. Problem Statement

Consider a person with a transradial (below-elbow) amputation using a robot prosthetic hand to grasp an object from a table top. The hand has 5 fingers, with 1 degree of actuation each (5-DOF overall); however, for the sake of simplicity, we reduce the action space to the first synergy of the hand when all fingers are coupled to move equally, i.e. $\mathbb{R}^1$. Assume the system measures the states in real-time, which are the robot joint positions and velocities as well as robot and object 6D pose and velocity in the space, i.e. $\mathbb{R}^{48}$ (see Table.I). We are interested in calculating the robot joint trajectories to pick the object when the prosthetic hand controlled by the human approaches the object along a rectilinear hand transport trajectory. We have tested other variants as state observations as indicated in Table.III.

The robot prosthetic hand grasping problem is different from classic robot manipulation and grasping problem. In the latter, the robot controls both hand transport and grasping; however, in the former, the robot has no control over hand transport but only the grasping. Here, we assume we know *a priori* that the user intent is to grasp the object. Then, the robot's goal is to perform a successful grasp. We define a successful grasp as a practical measure when the object height is increased over a specified threshold. The hand transport trajectory can be arbitrary; however, we assume a rectilinear hand transport trajectory within the scope of this study. The simultaneous contribution and collaboration of the human and robot to the final goal, in a sense, categorizes the problem as a shared-control problem.

In order to demonstrate the feasibility of our approach, we focus on one grasp type, a cylindrical grasp, in which all finger actuators are controlled by only one variable, $\hat{v} \in [-1, 1]$, which is a normalized velocity command to all finger actuators. The object is also limited to one object type and size, i.e. a cylinder. A successful grasp is rewarded sparsely $+1$, which encourages the agent to achieve a successful grasp as fast as possible. Reward will be 0 otherwise. Different measurement sets are used as states in this paper (Table III). The $\mathcal{O}_0$ measurement set is used unless otherwise specified. The definitions of the measurement sub-elements are presented in Table I.

B. Reinforcement learning problem

Consider an agent (i.e. robot) in an environment that measures, at each time instant t, a state variable $s_t \in \mathcal{S}$ and performs an action $a_t \in \mathcal{A}$ which can influence the

TABLE I: Definition of measurement

Measurement	Size	Definition
position	25	Vector of all hand joint positions (11), hand pose (7), and object pose (7)
velocity	23	Vector of all hand joint velocities (11), hand velocity (6), and object velocity (6)
$\vec{r}_{\text{rel}}$	3	Relative position vector between the object and the hand
$\|\vec{r}_{\text{rel}}\|$	1	Relative hand/object distance
$\|\vec{r}_{\text{rel},xy}\|$	1	The norm of 2d projection of $\vec{r}_{\text{rel}}$ on table top
closure (h)	1	A measure of hand flexion status, $h = 0$ for fully open, $h = 1$ for fully closed.

environment state, and accordingly receive a bounded reward signal $r : \mathcal{S} \times \mathcal{A} \rightarrow [r_{\min}, r_{\max}]$. This setting is formalized by a Markov-Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, p, r)$, where $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, \infty)$ is the state transition function giving the distribution of the next state s_{t+1} given current state s_t and action a_t. The action space $\mathcal{A}$ is a continuous space.

If the agent chooses its actions according to a stochastic policy $\pi(a|s) : \mathcal{S} \times \mathcal{A} \rightarrow [0, \infty)$, our goal is to find the optimal policy $\pi^\star$, which maximizes return (i.e the expected sum of rewards) $J(\pi) = \sum_{t=0}^{T} \mathbb{E}_\pi \left[r_t(s_t, a_t) \right]$, where T is the final time instant and $\mathbb{E}_\pi$ indicates expectation with respect to distribution of the visited states/actions under policy π. For infinite horizon $T \rightarrow \infty$, a discount factor $\gamma \in [0, 1)$ is multiplied by the reward function, to ensure finite summation. We do not show the discount factor in the following formulations for the sake of simplicity. Starting from an arbitrary state s and following policy π, the expected return would be defined as the state-value function, $V(s) \triangleq \sum_{k=0}^{T-t} \mathbb{E}_\pi \left[r_{t+k} | s_t = s \right]$. This function evaluates the value of a policy from a given state, i.e. the reward expected to be achieved from that state given that policy. The action-value function, the expected return from state s, doing an action a, and then following policy π is then defined as $Q(s, a) \triangleq \sum_{k=0}^{T-t} \mathbb{E}_\pi \left[r_{t+k} | s_t = s, a_t = a \right]$.

In RL, the policy is learned through interaction with the environment. When the agent receives some rewards based on its current policy, the question arises whether to stay with the current policy to collect more rewards, or to deviate from it to explore more of the uncharted environment and to achieve an overall better policy. This is known as the exploration vs. exploitation dilemma [25]. In maximum entropy RL [23], this issue is addressed by maximizing the entropy of the policy function alongside the environment rewards, using the new objective function:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_\pi [r(s_t, a_t) + \alpha \mathcal{H}(\pi(.|s_t))]$$

where $\mathcal{H}(p(X)) \triangleq \mathbb{E}\left[-\log(p(X)) \right]$ is the entropy function and α is the temperature parameter which weighs the importance of the entropy term with respect to the reward function. The result of this objective function is a policy that explores the environment more broadly, learns multi-modal behaviors, and learns faster [24]. We use Soft Actor-Critic (SAC), which is an actor-critic method solving the maximum entropy objective in an off-policy setting [24]. As an off-policy method, SAC uses a replay buffer $\mathcal{D}$ to store all transition tuples (s_t, r_t, a_t, s_{t+1}) to be sampled later for stochastic gradient updates. For the rest of the paper we do not show α in equations since it can be scaled into reward function by scaling it appropriately [24].

SAC is an actor-critic method and it keeps track of both state-/action- value function (critic) and policy (actor). SAC is based on a soft policy iteration method and includes two steps: policy evaluations, i.e. calculating the state-/action-value functions for a given policy, and policy improvements, i.e. updating the policy so that the value functions increase. We assume that $V(s)$, $Q(s, a)$, and $\pi(a|s)$ are represented by neural networks parameterized by ψ, θ, and ϕ. These parameters will be updated by stochastic gradients by back-propagating through the respective loss functions. In the policy evaluation step, the state-value function parameters are updated by using the squared residual error

$$\begin{aligned} J_V(\psi) = \mathbb{E}_{s_t \sim \mathcal{D}} \Bigg[\frac{1}{2} \Big(V_\psi(s_t) - \\ \mathbb{E}_{a_t \sim \pi_\phi} \left[Q_\theta(s_t, a_t) - \log \pi_\phi(a_t|s_t) \right] \Big)^2 \Bigg]. \quad (1) \end{aligned}$$

Also, the action-value function is updated by minimizing the soft Bellman residual

$$J_Q(\theta) = \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}} \left[\frac{1}{2} \left(Q_\theta(s_t, a_t) - \hat{Q}(s_t, a_t) \right)^2 \right], \quad (2)$$

where

$$\hat{Q}(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1} \sim p} [V_{\bar{\psi}}(s_{t+1})]. \quad (3)$$

Here, $V_{\bar{\psi}}$ is a soft average of V_ψ, which uses the update rule $\bar{\psi} \leftarrow (1 - \tau)\bar{\psi} + \tau\psi$, where $\tau \in [0, 1]$ (we refer to τ as the Polyak coefficient.)

In the policy improvement step, the policy is pushed towards the exponentiation of the action-value function, which is then projected into a tractable policy space by using Kullback-Leibler divergence, resulting in the following loss

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim \mathcal{D}} \left[D_{KL} \left(\pi_\phi(.|s_t) \middle\| \frac{\exp(Q_\theta(s_t, .))}{Z_\theta(s_t)} \right) \right], \quad (4)$$

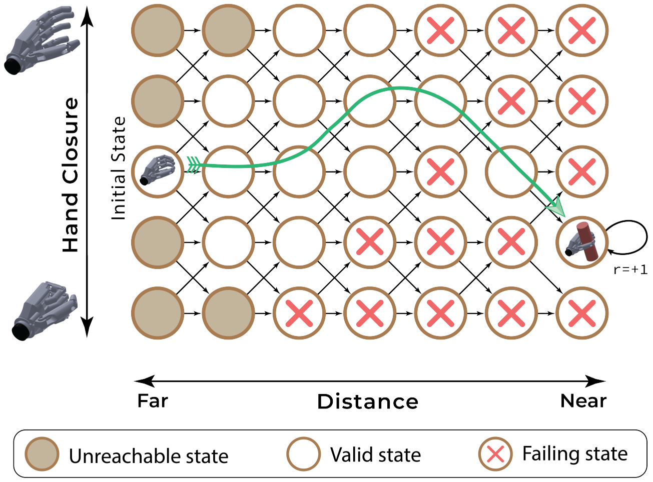


Fig. 3: Our problem is hard to explore. If the fingers are too flexed near the object, they will hit the object. If they are too open, they would not have the opportunity to close in time due to limited actuation speed of robot hands.

where Z_θ is the partition function which normalizes the action-value function, and is intractable to compute. However, it is independent from policy parameters and thus has no effect on the gradient.

C. Imitation learning

Due to the sparsity of rewards in this problem, it is very unlikely for the agent to discover the rewards by random exploration methods, e.g. ϵ-greedy [25]. Our problem is an instance of Keylock MDP where a sequence of actions are needed to reach a far rewarding state [26]. For instance, in the simplified state-space shown in Fig. 3 with the hand closure and distance as states, the hand should be open at least at the size of the object so it does not hit the object when approaching. Moreover, the hand should not be completely open, so that it does not have enough time to grasp the object when its time to (due to limited actuator speed). This means that only certain paths in the state-space will reach the desired state, making the exploration problem hard. With this regard, we use guided exploration through imitation learning (IL) to explore the rewarding states more readily. As the demonstrator, we use a scripted demonstrator with oracle access (i.e. access to all needed underlying states) that stores the generated transitions in an expert replay buffer $\mathcal{D}'$. A constant demo-use ratio $\beta \in [0, 1]$ determines the ratio of samples in the mini-batch to sample from the rollout buffer $\mathcal{D}$ and expert replay $\mathcal{D}'$, respectively. See Algorithm 1 for the complete algorithm. For a diagram of the overall process refer to Fig. 1.

This approach is known as Dataset Aggregation (DAGGER) method [26]. In DAGGER it is suggested to anneal the demo-use ratio β, whereas in our case we found it unnecessary in practice.

Algorithm 1: Integration of Soft Actor-Critic [24] with imitation learning. $\bar{\pi}$ indicates the demonstrator policy. M is the minibatch size. β is the ratio of demo transitions in the batch. Stochastic gradients on a sampled minibatch B with respect to ψ, θ, and ϕ are shown as $\hat{\nabla}^B_\psi$, $\hat{\nabla}^B_\theta$, and $\hat{\nabla}^B_\phi$, respectively. Learning rates for ψ, θ, and ϕ parameters are shown using λ_V, λ_Q, and λ_π, respectively.

```
Initialize parameter vectors $\psi$, $\bar{\psi}$, $\theta$, $\phi$.
foreach iteration do
    foreach environment step do
        $a_t \sim \pi_\phi(a_t|s_t)$
        $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(s_t, a_t, r(s_t, a_t), s_{t+1})\}$
    end
    foreach environment step do
        $a_t \sim \bar{\pi}(a_t|s_t)$
        $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$
        $\mathcal{D}' \leftarrow \mathcal{D}' \cup \{(s_t, a_t, r(s_t, a_t), s_{t+1})\}$
    end
    foreach gradient step do
        $B_{e,i} \sim \mathcal{D}'$ for $i \in \{1, \cdots, \lfloor \beta M \rfloor\}$
        $B_{r,i} \sim \mathcal{D}$ for $i \in \{1, \cdots, M - \lfloor \beta M \rfloor\}$
        $B \leftarrow B_r \cup B_e$
        $\psi \leftarrow \psi - \lambda_V \hat{\nabla}^B_\psi J_V(\psi)$
        $\theta \leftarrow \theta - \lambda_Q \hat{\nabla}^B_\theta J_Q(\theta)$
        $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}^B_\phi J_\pi(\phi)$
        $\bar{\psi} \leftarrow (1 - \tau)\bar{\psi} + \tau\psi$
    end
end
```

D. Scripted expert

As the expert in the IL, we use a scripted controller which loosely resembles human aperture control behavior [27]. The demonstrator is a two-phase proportional (P) controller which gets hand closure h and normalized relative hand-object distance $\hat{d}$ (with respect to initial distance) as inputs and generates joint velocities as output.

$$C(h, \hat{d}) = \begin{cases} K(h_{\text{open}} - h) & \hat{d} \geq \hat{d}_c \\ K(h_{\text{closed}} - h) & \hat{d} < \hat{d}_c \end{cases} \tag{5}$$

where K is the controller gain, h_{open} and h_{closed} are the targets for the hand closure in the first and second phases, respectively, and $\hat{d}_c$ is the critical normalized relative hand-object distance to switch controller phase. This controller first tries opening the hand up to some closure, then starts closing the hand when the distance is smaller than a threshold, so the object can be grasped. This controller will guide the exploration of the RL problem to the rewarding states through IL, thus making the agent to learn useful policies.

E. Hand transport model

As stated before, in the robot prosthetic hand control problem, the human is responsible for robot hand transport, while robot only controls the fingers/wrist. We use a human-based model to provide hand transport trajectories to the environment (simulated transport trajectory blocks in Fig. 1). In order to provide realistic trajectories for hand transport model, we use one of the classic hand transport models in the literature provided by Hoff and Arbib [27]. They provide an optimal controller based on time-to-arrive of hand D, i.e. the time remaining until the hand reaches the object. The cost function for this optimal control is the integral of the hand jerk over the overall hand transport duration T:

$$J = \int_0^T \left(\frac{d^3x}{dt^3} \right)^2 dt \tag{6}$$

where x is the hand displacement coordinate in 1D as a function of time. Flash and Hogan [28] have shown that transport components in higher dimensions are decoupled, i.e. the same controller can be applied to all displacement components. By assuming zero velocity and acceleration, both initially and at the end, the open-loop non-stationary controller fulfilling Eq. 6 in 1D is:

$$\dot{\mathbf{X}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -60/D^3 & -36/D^2 & -9/D \end{bmatrix} \mathbf{X} + \begin{bmatrix} 0 \\ 0 \\ 60/D^3 \end{bmatrix} u$$

where $\mathbf{X} = [x, \dot{x}, \ddot{x}]$ is the vector of hand displacement, velocity, and acceleration in 1D, $D = T - t$ is the remaining time to the goal, and u is the control input, here the reference value for x. Because $\lim_{t \to T} D = 0$ which leads to division by zero near the final time, we regularize D by $D = T - t + \delta$ where $\delta > 0$ ensures $D > 0$. This will deviate the trajectory slightly from the minimum jerk trajectory, however.

F. Implementation

For implementation purposes, we used Google Deepmind's dm_control [29], which is a wrapper for MuJoCo [30], to simulate the environment. A new hand model was designed based on MuJoCo HAPTIX [31]. The neural network architectures were selected as the original paper [24]. PyTorch [32] was used for implementation of the neural network policy and (action-) value functions. For implementation of the SAC method, Digideep package [33] was used. The initial position of the hand in cylindrical frame (r, θ) was uniformly sampled from $r \in [20cm, 45cm]$ and $\theta \in [\pi/14, \pi/7]$. The hand transport trajectory was set to a piecewise rectilinear motion; one piece from the initial point to the object grasp point, and the second piece going upwards for another $20cm$ in the direction of $+z$. Each piece was generated by a minimum-jerk motion model as was described in II-E. A $+1$ reward was given at each time instant where the object height was above a $15cm$ threshold. The simulation was terminated without a penalty if the object height dropped by $0.5cm$. We found that penalties make learning even harder due to their adverse effect on exploration; the agent will try to avoid danger zones by keeping the fingers always open.

TABLE II: Hyperparameters used for training

Hyperparameter	**Value**
SAC	
Learning rate (λ_Q, λ_V, λ_π)	3e-4
Replay buffer size ($\lvert\mathcal{D}\rvert = \lvert\mathcal{D}'\rvert$)	1e6
Mini-batch size (M)	128
Polyak coefficient (τ)	0.01
Epoch size	1000 frames
Discount factor (γ)	0.99
Demo-use ratio (β)	0.3
Expert	
Controller gain (K)	
Critical distance ($\hat{d}_c$)	0.8
Hand open target ($h_{\text{open}} = 0.8$)	0.8
Hand closed target ($h_{\text{closed}} = 0.8$)	0.2
Transport Model	
Motion duration (T)	6sec

III. Results

In this section we try to investigate the effectiveness of our RL + IL approach to solve the problem. Fig. 4a the learning graph for the original RL problem with and without IL. Fig. 4b shows the sensitivity of our method to the $beta$ hyperparameter that we introduced. In Fig. 4c we try to see the effect of different measurement sets (as introduced in Table. III) on the results.

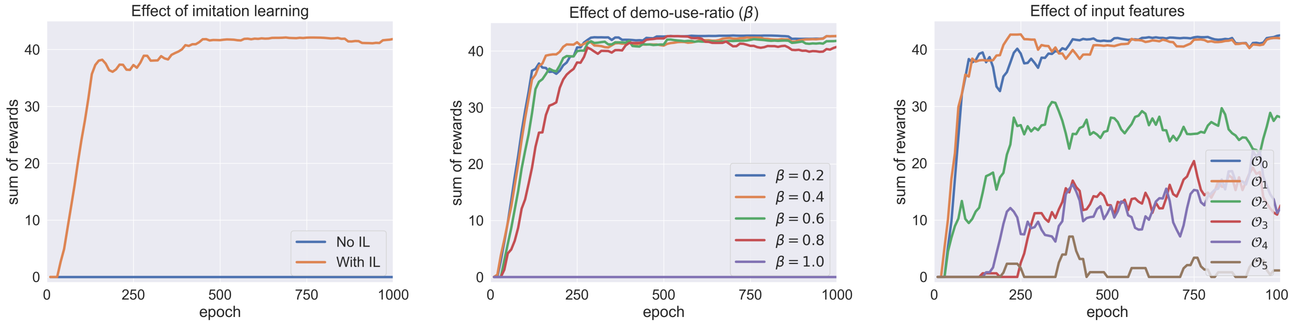
In Fig. 4, every epoch equals 1000 frames of simulation steps. The test graphs were generated by averaging rewards of rolling out the trained policy in the environment for 10 episodes every 10 epochs. During training, actions were sampled from the policy distribution whereas during tests the mean of action distribution was used. The results are smoothed by a moving average of window-size 15. Different hyperparameters were tested and the best was used for all simulations unless explicitly mentioned (see Table II). Four frames of a successful grasp selected from the trained policy using RL + IL is demonstrated in Fig. 2.

TABLE III: Specification of feature sets $\mathcal{O}_1 \dots \mathcal{O}_5$

	Set Name					
Feature	$\mathcal{O}_0$	$\mathcal{O}_1$	$\mathcal{O}_2$	$\mathcal{O}_3$	$\mathcal{O}_4$	$\mathcal{O}_5$
position (25×1)	✓	✓	✓			
velocity (23×1)	✓	✓				
$\vec{r}_{\text{rel}}$ (3×1)	✓			✓	✓	
$\lvert\lvert\vec{r}_{\text{rel}}\rvert\rvert$	✓				✓	
$\lvert\lvert\vec{r}_{\text{rel},xy}\rvert\rvert$	✓				✓	✓
closure (h)				✓	✓	✓

IV. Discussion

This paper presents an end-to-end RL framework to learn grasping policy for robot prosthetic hands. As shown in Fig. 4a, our RL + IL approach can learn a grasping policy which is always successful (43 is the highest possible sum of rewards in an episode). Without IL, it is shown that no useful policies are learned at all (the constant zero line in Fig. 4a). The reason is arguably the sparse rewards in our problem



(a) Importance of imitation learning (b) Effect of demo-use ratio (β) parameter (c) Effect of different input features

Fig. 4: Results of different settings on training. Results are shown just for a single random seed. The vertical axis is sum of all rewards recieved in an episode.

setting and the fact that the rewarding states are hard to explore, rendering random exploration methods ineffective.

The end-to-end nature of our methods entails learning actuator commands directly from the input states. End-to-end methods can learn feedback controllers that can react to environment changes or perturbations in real-time, a feature which is missing from other non-EMG control methods [13], [14].

The effect of introducing the β hyperparameter on the robustness of our RL + IL approach is investigated in Fig. 4b. The results show that our approach is robust to this hyperparameter as long as $0 < \beta < 1$. The $\beta = 1$ is indicative of the pure IL solution which cannot learn anything due to lack of interaction with the environment.

The role of the input feature sets on training was also studied in Fig. 4c. While testing all combinations of the feature sets is tedious, the point of this part is to highlight the importance of some feature inputs to successfully learning the optimal policy. It is shown that the combination of position and velocity is enough to learn the grasping policy. Interestingly, the agent cannot learn a useful policy from $\mathcal{O}_5$, which is the same input feature to the scripted expert. While this is an interesting finding, more investigations are required to state whether it is a weakness of the RL method used, as there are other influential factors like the neural network architecture. It is also shown that velocity has an important role in achieving a high performance ($\mathcal{O}_1$ vs. $\mathcal{O}_2$). This can be due to the partial observability and violation of Markov assumption when position is the only input state.

Most RL methods are known for poor statistical efficiency. Low iterations for convergence matters more when there are humans involved in the training loop, as is the case for shared control problems. In our settings, thanks to the combination of SAC method with the IL, the convergence happens in about 500 episodes, which conservatively is about 3 hours in real world settings, given $T = 6\ sec$ duration for each episode.

There are several advantages for not relying on EMG signals for controlling the robot prosthetic hand. The time and effort needed for training the user would be reduced. Also, due to independence from the highly variable EMG signals, the learned policy is easier to transfer to other users. Furthermore, the user does not need to be involved, physically or mentally, in generating the EMG signals and making them distinguishable for the pattern recognition models [9], [34]. Altogether, independence from EMG signal processing may potentially increase the usage time of the prosthetic hand due to increased robustness, and thus reduce the rejection rate of the robot prosthetic hands.

In this paper, we investigated a single grasping scenario in simulation with a constant object shape, size, and position. In order to capture object variabilities in real world, it is important to have input modalities, like RGB cameras, that can capture the associated information. Training the agent using camera input should be done as a future work. Constant environment dynamics, fixed robot geometry, and no noise in state readings can also cause discrepancy between simulation and real world and are considered as other limitations of the current simulation. As a future work, domain randomization will be used to address the real-world variations from the ideal case.

Another drawback of our approach is not using EMG or other notions of human intent in the control process. This compromises the applicability and responsiveness of our approach to activities of daily life where spontaneous intent inference is required usually outside of a pre-specified context. Furthermore, in this work, we studied only grasping but not releasing of objects. While EMG-based methods offer reliable solutions for releasing objects since only one gesture needs to be detected, task-related information can still be leveraged to perform the task. For instance, if the task is moving objects from a table into a bucket, then by using the hand trajectory, the robot can know when the object should be released if the position of bucket is known *a priori*. Overall, our approach is applicable to organized environments where task pre-knowledge together with hand trajectory serve as sufficient information to infer human intent. For more sophisticated scenarios, hybrid models can be created where one uses EMG for high-level inference and task and trajectory data for low-level robot control.

V. Conclusion

In this work, an RL-based framework is offered to learn end-to-end policies for controlling robot prosthetic hands for grasping. This work is proposed in response for a need to change of paradigm in controlling robot prosthetic hands due to shortage of current control methods. As opposed to the state-of-the-art EMG-based control methods for robot prosthetic hands, our method does not rely on EMG at all, which can lead to potentially more robust controllers with less training. However, this work only offers a feasibility study for the proposed RL-based end-to-end method in a simulated environment, which has yet to be applied to real life in future works.

References

[1] K. Ziegler-Graham, E. J. MacKenzie, P. L. Ephraim, T. G. Travison, and R. Brookmeyer, "Estimating the prevalence of limb loss in the united states: 2005 to 2050," *Arch. Phys. Med. Rehabil.*, vol. 89, no. 3, pp. 422–429, Mar. 2008.

[2] T. W. Wright, A. D. Hagen, and M. B. Wood, "Prosthetic usage in major upper extremity amputations," *J. Hand Surg. Am.*, vol. 20, no. 4, pp. 619–622, Jul. 1995.

[3] E. A. Biddiss and T. T. Chau, "Upper limb prosthesis use and abandonment: a survey of the last 25 years," *Prosthet. Orthot. Int.*, vol. 31, no. 3, pp. 236–257, Sep. 2007.

[4] S. Muceli, N. Jiang, and D. Farina, "Extracting signals robust to electrode number and shift for online simultaneous and proportional myoelectric control by factorization algorithms," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 3, pp. 623–633, May 2014.

[5] M. Hakonen, H. Piitulainen, and A. Visala, "Current state of digital signal processing in myoelectric interfaces and related applications," *Biomed. Signal Process. Control*, vol. 18, pp. 334–359, Apr. 2015.

[6] A. L. Ciancio, F. Cordella, R. Barone, R. A. Romeo, A. D. Bellingegni, R. Sacchetti, A. Davalli, G. Di Pino, F. Ranieri, V. Di Lazzaro, E. Guglielmelli, and L. Zollo, "Control of prosthetic hands via the peripheral nervous system," *Front. Neurosci.*, vol. 10, p. 116, Apr. 2016.

[7] H.-J. Hwang, J. M. Hahne, and K.-R. Müller, "Real-time robustness evaluation of regression based myoelectric control against arm position change and donning/doffing," *PLoS One*, vol. 12, no. 11, p. e0186318, Nov. 2017.

[8] D. Farina, N. Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl, and O. C. Aszmann, "The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 797–809, Jul. 2014.

[9] K. Z. Zhuang, N. Sommer, V. Mendez, S. Aryan, E. Formento, E. D'Anna, F. Artoni, F. Petrini, G. Granata, G. Cannaviello, W. Raffoul, A. Billard, and S. Micera, "Shared human–robot proportional control of a dexterous myoelectric prosthesis," *Nature Machine Intelligence*, vol. 1, no. 9, pp. 400–411, Sep. 2019.

[10] C. Castellini, P. Artemiadis, M. Wininger, A. Ajoudani, M. Alimusaj, A. Bicchi, B. Caputo, W. Craelius, S. Dosen, K. Englehart, D. Farina, A. Gijsberts, S. B. Godfrey, L. Hargrove, M. Ison, T. Kuiken, M. Marković, P. M. Pilarski, R. Rupp, and E. Scheme, "Proceedings of the first workshop on peripheral machine interfaces: going beyond traditional surface electromyography," *Front. Neurorobot.*, vol. 8, p. 22, Aug. 2014.

[11] N. Jiang, S. Dosen, K. R. Muller, and others, "Myoelectric control of artificial Limbs—Is there a need to change focus? [in the spotlight]," *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 152–150, Sep. 2012.

[12] A. Gigli, A. Gijsberts, V. Gregori, M. Cognolato, M. Atzori, and B. Caputo, "Visual cues to improve myoelectric control of upper limb prostheses," Aug. 2017.

[13] S. Došen, C. Cipriani, M. Kostić, M. Controzzi, M. C. Carrozza, and D. B. Popović, "Cognitive vision system for control of dexterous prosthetic hands: Experimental evaluation," *J. Neuroeng. Rehabil.*, vol. 7, no. 1, p. 42, Aug. 2010.

[14] G. K. Patel, J. M. Hahne, C. Castellini, D. Farina, and S. Dosen, "Context-dependent adaptation improves robustness of myoelectric control for upper-limb prostheses," *J. Neural Eng.*, vol. 14, no. 5, p. 056016, Oct. 2017.

[15] D. Kim, B. B. Kang, K. B. Kim, H. Choi, J. Ha, K.-J. Cho, and S. Jo, "Eyes are faster than hands: A soft wearable robot learns user intention from the egocentric view," *Science Robotics*, vol. 4, no. 26, p. eaav2949, Jan. 2019.

[16] M. Sharif, D. Erdogmus, and T. Padir, "Particle filters vs hidden markov models for prosthetic robot hand grasp selection," *International Journal of Robotic Computing*, vol. 1, no. 2, p. 25, Jul. 2019.

[17] F. Ficuciello, A. Migliozzi, G. Laudante, P. Falco, and B. Siciliano, "Vision-based grasp learning of an anthropomorphic hand-arm system in a synergy-based control framework," *Science Robotics*, vol. 4, no. 26, p. eaao4900, Jan. 2019.

[18] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "QT-Opt: Scalable deep reinforcement learning for Vision-Based robotic manipulation," Jun. 2018.

[19] Y. Zhu, Z. Wang, J. Merel, A. Rusu, T. Erez, S. Cabi, S. Tunyasuvunakool, J. Kramár, R. Hadsell, N. de Freitas, and N. Heess, "Reinforcement and imitation learning for diverse visuomotor skills," Feb. 2018.

[20] OpenAI, :, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous In-Hand manipulation," Aug. 2018.

[21] A. Gupta, C. Eppner, S. Levine, and P. Abbeel, "Learning dexterous manipulation for a soft robotic hand from human demonstrations," *Rep. U. S.*, vol. 2016-Noem, pp. 3786–3793, 2016.

[22] V. Kumar, E. Todorov, and S. Levine, "Optimal control with learned local models: Application to dexterous manipulation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 378–383.

[23] B. D. Ziebart, "Modeling purposeful adaptive behavior with the principle of maximum causal entropy," Ph.D. dissertation, figshare, 2010.

[24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy maximum entropy deep reinforcement learning with a stochastic actor," Jan. 2018.

[25] R. S. Sutton, A. G. Barto, and F. Bach, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[26] S. Ross, G. J. Gordon, and J. Andrew Bagnell, "A reduction of imitation learning and structured prediction to No-Regret online learning," Nov. 2010.

[27] B. Hoff and M. A. Arbib, "Models of trajectory formation and temporal interaction of reach and grasp," *J. Mot. Behav.*, vol. 25, no. 3, pp. 175–192, Sep. 1993.

[28] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *J. Neurosci.*, vol. 5, no. 7, pp. 1688–1703, Jul. 1985.

[29] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. de Las Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. Lillicrap, and M. Riedmiller, "DeepMind control suite," Jan. 2018.

[30] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 5026–5033.

[31] V. Kumar and E. Todorov, "MuJoCo HAPTIX: A virtual reality system for hand manipulation," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov. 2015, pp. 657–663.

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, High-Performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[33] M. Sharif, "Digideep: A DeepRL pipeline for developers," 2019.

[34] M. Kryger, A. E. Schultz, and T. Kuiken, "Pattern recognition control of multifunction myoelectric prostheses by patients with congenital transradial limb defects: a preliminary study," *Prosthet. Orthot. Int.*, vol. 35, no. 4, pp. 395–401, Dec. 2011.