2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Knowledge-Guided Reinforcement Learning Control for Robotic Lower Limb Prosthesis

Xiang Gao, Jennie Si, *Fellow, IEEE,* Yue Wen, Minhan Li, and He (Helen) Huang, *Senior Member, IEEE*

***Abstract*— Robotic prostheses provide new opportunities to better restore lost functions than passive prostheses for transfemoral amputees. But controlling a prosthesis device automatically for individual users in different task environments is an unsolved problem. Reinforcement learning (RL) is a naturally promising tool. For prosthesis control with a user in the loop, it is desirable that the controlled prosthesis can adapt to different task environments as quickly and smoothly as possible. However, most RL agents learn or relearn from scratch when the environment changes. To address this issue, we propose the knowledge-guided Q-learning (KG-QL) control method as a principled way for the problem. In this report, we collected and used data from two able-bodied (AB) subjects wearing a RL controlled robotic prosthetic limb walking on level ground. Our ultimate goal is to build an efficient RL controller with reduced time and data requirements and transfer knowledge from AB subjects to amputee subjects. Toward this goal, we demonstrate its feasibility by employing OpenSim, a well-established human locomotion simulator. Our results show the OpenSim simulated amputee subject improved control tuning performance over learning from scratch by utilizing knowledge transfer from AB subjects. Also in this paper, we will explore the possibility of information transfer from AB subjects to help tuning for the amputee subjects.**

I. Introduction

The rapid development of robotic prostheses in both research and commercial products brings them closer to real-life scenarios. Compared to passive devices, robotic lower limb prostheses promise to provide better functions to restore natural gaits for amputees, such as decreased metabolic consumption [1], improved adaptation to various terrains [2], [3] or walking speed [4], and enhanced balance and stability [5]. In robotic lower limb prosthetics, finite state impedance control (FS-IC) [6], [7] is still the most common approach in both prototypes or commercial devices. However, in order to maximize the performance for each user, there are a large number of control parameters in these devices that need to be tuned by experienced clinicians.

Reinforcement learning allows learning from interacting with the environment to generate suitable actions while maximizing a performance reward in a particular situation. Learning can take place under different formulations of a problem, including learning directly from data without requiring an explicit mathematical description of the environment and the interacting dynamics between the controller and the environment. This has given RL an expanded domain of control applications beyond the capacity of traditional control theory and practice. There have been several successful demonstrations of RL applications to solving challenging robotic control problems. Among those, deep RL methods attracted most attentions. For example, Nair *et al.* [8] employed deep deterministic policy gradients (DDPG) for a robotic arm block stacking task with sparse reward. The authors of [9] proposed deep latent policy gradient (DLPG) for learning locomotion skills. However, deep RL based methods may be not suitable for biomedical applications such as the human-prosthesis control problem being discussed in this paper, because training data involving amputee subjects are usually difficult to acquire and expensive to collect. Additionally, experimental sessions involving human subjects usually cannot last more than one hour because of human fatigue and safety considerations. To tackle this challenge, we proposed several sample-efficient and easy-to-implement RL methods in our previous works [10]-[13] allowing for directly learning from data. In our application of prosthesis control, it is very common that the robotic prosthesis need to be adapted for a new user. However, these RL methods, as well as most existing RL methods, are designed to learn from scratch whenever a new task or new model is presented, and thus not readily capable of storing and transferring knowledge gained from one subject to another.

It is therefore of high priority that the RL agent is designed to be training time and sample efficient when tuned for a new user. To take advantage of previous knowledge and information, we consider building a representation for potentially transferable knowledge across subjects. In the current study, we consider extracting knowledge from able-bodied (AB) subjects and use that for future RL control design for amputee subjects. It is known that transfer learning has attracted great attention in the machine learning field where it is typically considered for storing knowledge gained while solving one problem and applying it to a different but related problem [14]. In the context of general transfer learning in the literature, our prosthesis parameter tuning problem has the same state and action while the problem calls for gaining knowledge from tuning parameters for AB subjects (source task) and using that for tuning parameters for amputee subjects (target task).

Recently, many successful applications of structural knowledge transfer have been reported in the literature.

*This work was partly supported by National Science Foundation: #1563921 and #1808752 for J. Si, #1563454 and #1808898 for H. Huang. Correspondence: J. Si and H. Huang.

X. Gao and J. Si are with the the Department of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA si@asu.edu

Y. Wen, M. Li and H. Huang are with the Department of Biomedical Engineering, North Carolina State University, Raleigh, NC 27695 USA, and also with the University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA hhuang11@ncsu.edu

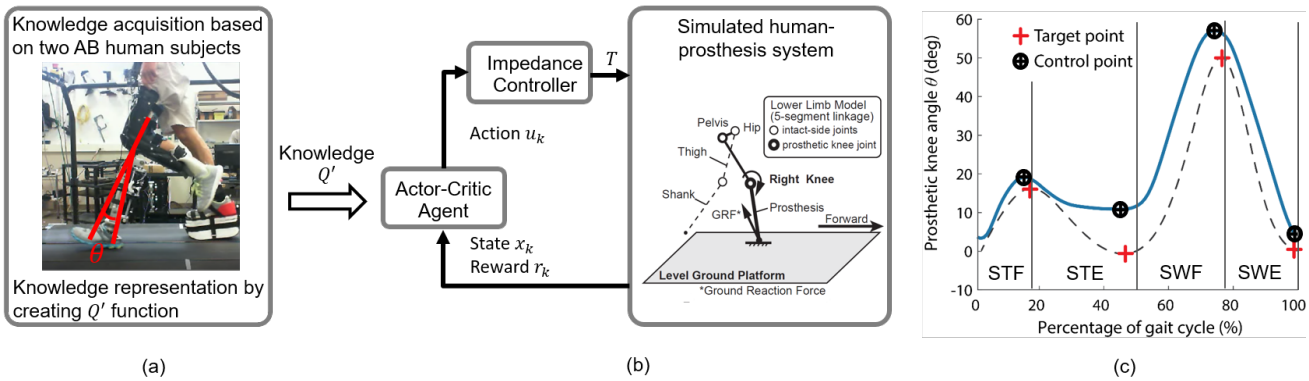


Fig. 1. Schematic of knowledge-guided RL control. (a) Knowledge representation by regression models of the system are obtained using data from two able-bodied subjects. (b) The online learning process in OpenSim: an actor-critic agent is trained to optimize the impedance parameters by interacting with the lower limb walking model in OpenSim, with the help of the transferred knowledge represented in the function of $Q'(x_k, u_k)$. (c) Gray dashed line: normal knee kinematics, blue line: actual measured knee kinematics.

Barreto *et al.* [15] solved the problem where rewards change but environments remain the same using successor features, a value function representation that decouples the dynamics of the environment from the rewards. Asadi *et al.* [16] proposed a learning architecture which transfers control knowledge in the form of behavioral skills and representation hierarchy, which separates the subgoals so that a more compact state space can be learned. In [17], researchers demonstrated that Schema network is capable to perform zero-shot transfer between tasks where cause-effect relationship remains unchanged, such as learning to play the breakout game with different maps. In [18], target apprentice learning is proposed for cross-domain transfer, e.g. from balancing a cart-pole to balancing a bike.

Unlike the above approaches, we propose a new knowledge transfer framework for the class of problems that transfer from a source task to a target task while maintaining the same state and control problems. We built a knowledge representation from AB subjects into the actor-critic update, and the knowledge transfer schedule results in a diminishing influence of previous knowledge, which simultaneously allows for increased attention to learning of the target task on hand. Specifically, we first collected data from AB subjects, then we built regression models on these data, which then became transferred knowledge to guide a Q-learning process, namely our proposed knowledge guided Q-learning (KG-QL) process. Our method introduces two new advances from the existing transfer learning methods. First, we provided a flexible framework where the representation of the transferred knowledge can be either a value function or a regression model or both. Second, the amount of transferred knowledge into a new task can be programmed in a convenient way to address different applications' needs.

II. Methods

Fig. 1 is a schematic diagram of our proposed transfer learning-based or knowledge guided reinforcement learning framework for prosthesis parameter tuning.

A. Finite State Impedance Control of Human-Prosthesis System

From the perspective of a RL agent, the integrated human-prosthesis system can be treated as a nonlinear dynamic system of the form

$$x_{k+1} = F(x_k, u_k), k = 0, 1, 2, \ldots \quad (1)$$

where k is the discrete time index or gait cycle in this study, $x_k \in \mathbb{R}^2$ is the state vector, $u_k \in \mathbb{R}^3$ is the action or control vector, and F describes the intrinsic human-prosthesis system dynamics of how a new state at $k+1$ evolves from a current state and control at k. Specifically, state x_k is defined as the differences (errors) between the measured knee angle profile and the target knee profile at the feature points. The target knee profile is identical to those normative knee kinematics reported in [19]. In Fig. 1(c), for each of the four phases there is a pair of such feature points with black and red markers, where their vertical and horizontal differences are peak error $\Delta P_k \in \mathbb{R}$ and duration error $\Delta D_k \in \mathbb{R}$, respectively:

$$x_k = [\Delta P_k, \Delta D_k]^T. \quad (2)$$

The RL controller is realized within an established FS-IC platform. In FS-IC, a complete gait cycle is divided into four sequential gait phases based on knee joint kinematics and ground reaction force (GRF) by a finite state machine (FSM). These four gait phases are stance flexion (STF), stance extension (STE), swing flexion (SWF) and swing extension (SWE). In real-time experiments, phase transitions are realized as those in [7] based on Dempster-Shafer theory (DST). In each phase, the prosthetic system mimics a passive spring-damper-system with a group of three predefined impedance parameters as

$$I_k = [K_k, B_k, \theta_{e,k}] \in \mathbb{R}^3, \quad (3)$$

where K_k is stiffness, B_k is damping coefficient and $\theta_{e,k}$ is equilibrium position. For all four phases, there are 12 impedance parameters in total. Four RL controllers sharing identical structure are designed separately for the four phases

, and each of them has its own parameters. The knee joint torque $T \in \mathbb{R}$ is generated based on the following impedance control law

$$T_k = K_k(\theta - \theta_{e,k}) + B_k\omega. \tag{4}$$

Correspondingly, the action u_k of the RL agent is defined as adjustments ΔI_k to the impedance parameters I_k,

$$u_k = \Delta I_k = [\Delta K_k, \Delta B_k, \Delta\theta_{e,k}]. \tag{5}$$

B. Human Gait Data Collection

To perform knowledge transfer from a source task to a target task, first we need to obtain the transferable knowledge, which can be represented in the form of raw data, policy, value function, or others. Here we define a function $Q'(x_k, u_k)$ to store such information for transfer. It takes state and action as input to generate the state-action value function or Q-value function $Q(x, u)$ as in the RL literature. Although Q' can be a previously learned Q-value function from a RL agent, it can also be represented in other forms. In this work, we construct Q' using regression model based on the source task data.

Source task data was collected from two AB participants (both male, 25-35 years old) while walking at a constant speed of 0.6 m/s on a treadmill with force platforms embedded within each belt. All participants provided written informed consent prior to participating according to protocols approved by the Institutional Review Board at North Carolina State University. A certified prosthetist aligned the robotic knee prosthesis for each subject. The AB subjects used an L-shape adaptor (with one leg bent 90 degrees) to walk with the robotic knee prosthesis (Fig. 1(a)) [11].

The gait data used in this study includes a total of $N = 1120$ pairs of state-action tuples (x_k, u_k) from the two AB subjects (AB1: 480 pairs, AB2: 640 pairs) using the same prosthesis device. During data collection, the prosthesis impedance parameters were controlled by the dHDP based RL approach that we investigated previously [11], [20]. Note that the dHDP was only to provide some control to the prosthesis instead of providing optimal control to achieve a performance measure. In other words, the data were drawn from the online learning process of the dHDP RL controller rather than generated by a well-learned policy to provide sufficient exploration of the control policy space. Actually, a RL controller is not unique for data collection. Any sampling method is acceptable as long as it sufficiently samples the control parameter space, and it maintains practical stability of the human-prosthesis system. Note that during data collection, the impedance parameters I_k were updated every seven gait cycles, and state x_k was averaged by the seven gait cycles conditioned on the same impedance parameters I_k. That is to say, to reduce step-by-step variability in feature measurements, the time index k here corresponds to every seven gait cycles.

C. Extracting Knowledge from Human Gait Data

We performed linear regression to establish a relationship between prosthesis impedance parameters and the human-prosthesis system kinematics as follows,

$$x_{k+1} = \mathcal{F}(x_k, u_k) = \mathcal{F}(z_k) = \beta_0 + \beta_1 z_k + e, \tag{6}$$

where $x_{k+1} \in \mathbb{R}^2$ is the next state, $\beta_0 \in \mathbb{R}$ is the intercept, $\beta_1 \in \mathbb{R}^{2\times 5}$ is the regression coefficient (or the slope), $z_k = [x_k, u_k] \in \mathbb{R}^5$ is the predictor variable formed by the current state x_k and action u_k, and $e \in \mathbb{R}^2$ is the error term. Least-square solution of the coefficients β_0 and β_1 can be found using the (x_k, u_k, x_{k+1}) tuples. Equation (6) characterizes the human-prosthesis system qualitatively because when a controller enables the human-prosthesis system to generate improved locomotion performance, we generally observe that $|x_{k+1}| \leq |x_k|$.

After the regression model $\mathcal{F}$ is obtained, we can formulate $Q'(x_k, u_k)$ based on $\mathcal{F}(x_k, u_k)$. How Q' is formulated also relates to the stage reward or cost in RL. In our work, we set the stage cost variable $r_k = 0$ for success and $r_k = 1$ for failure (see (9)), which implied that the goal for the RL agent was to minimize the total cost-to-go. Accordingly, inspired by LQR control objective function, Q' was formulated as a quadratic form such that $Q' \geq 0$,which was consistent with $r_k \geq 0$ and $Q_i \geq 0$ (Q_i is the iterative Q-value function defined in (14) and (15)):

$$Q' = 0.02x_{k+1}^2 = 0.02(\mathcal{F}(x_k, u_k))^2. \tag{7}$$

Note that here the form of Q' was manually defined and was not unique. The ratio of 0.02 was set manually to make Q' within a comparable range of the stage cost r_k. As shown later, knowledge represented in Q' can be adopted by the designer at a preferred rate. Fig. 2 Illustrates kinematics and Q-values as knowledge representations.

D. Knowledge Guided Reinforcement Learning

Algorithm 1 Knowledge Guided Q-Learning (KG-QL) for prosthesis control with a human in the loop

Input: Transferred knowledge Q' from a source task
Initialization: Random actor NN weights and critic NN weights. $i = 0, k = 0$.

1: Start from a random initial state x_0.
2: **repeat**
3: Get u_k from x_k according to actor NN (ϵ-greedy).
4: Take action u_k, observe cost r_k and next state x_{k+1}.
5: Update actor NN weights to minimize (22).
6: Update critic NN weights to minimize (24).
7: $x_{k+1} \leftarrow x_k$.
8: $i \leftarrow i + 1, k \leftarrow k + 1$.
9: **until** x_k is a terminal state

We have introduced how the transferred knowledge Q' is obtained through regression. Now we can move onto the online learning process of the RL agent as shown in Fig. 1(b). We call this RL algorithm a knowledge-guided Q-learning

algorithm (KG-QL) because when determining a best action for the next step, its decision is guided and biased by the transferred knowledge Q'.

At time index k, the RL agent starts from state x_k and takes the action u_k. Then it ends up at the next state x_{k+1}, and receives a cost r_k. This process repeats for $k = 1, 2, ...$ until a terminal state is reached. The total cost-to-go function or value function is defined as

$$J(x_k, u_k) = \sum_{j=k}^{\infty} \gamma^{j-k} r_k, \tag{8}$$

where $r_k = r(x_k, u_k)$ is the stage cost, and γ is the discount factor with $0 < \gamma < 1$. In our work, we defined r_k as

$$r_k = r(x_k, u_k) = \begin{cases} 0, & \text{if } x_{k+1} \text{ is a success state} \\ 1, & \text{if } x_{k+1} \text{ is a failure state} \\ 0.01, & \text{otherwise} \end{cases} \tag{9}$$

In this work, a success state is defined as when the state is in the target range, and a failure state is defined as the state is out of the safety range. Further details can be found in III-A.

Equation (8) can be written as

$$J(x_k, u_k) = r_k + \gamma J(x_{k+1}, u_{k+1}). \tag{10}$$

According to Bellman's optimality principle [21], the optimal cost function J^* satisfies the action dependent discrete time Hamilton–Jacobi–Bellman (HJB) equation

$$J^*(x_k, u_k) = r_k + \gamma \min_{u_{k+1}} J^*(x_{k+1}, u_{k+1}). \tag{11}$$

Besides, the optimal control π^* can be expressed as

$$\pi^*(x_k) = \arg\min_{u_k} J^*(x_k, u_k). \tag{12}$$

By substituting (12) into (11), the discrete time HJB equation becomes

$$J^*(x_k, u_k) = r_k + \gamma J^*(x_{k+1}, \pi^*(x_{k+1})). \tag{13}$$

For an actor-critic agent, we have the following structure,

$$\pi_i(x_k) = \arg\min_{u_k} Q_i(x_k, u_k), \tag{14}$$

$$Q_{i+1}(x_k, u_k) = r_k + \gamma Q_i(x_{k+1}, \pi_i(x_{k+1})), \tag{15}$$

where i is the iterative index, π_i and Q_i are the iterative control policy and iterative Q-value function, respectively. Combining (14) and (15), we have

$$Q_{i+1}(x_k, u_k) = r_k + \gamma \min_{u_{k+1}} Q_i(x_{k+1}, u_{k+1}). \tag{16}$$

Accordingly, the *knowledge-guided* form of actor-critic learning can be written as

$$\pi_i(x_k) = \arg\min_{u_k} [Q_i(x_k, u_k) + \alpha_i Q'(x_k, u_k)], \tag{17}$$

$$\begin{aligned} Q_{i+1}(x_k, u_k) = r_k + \gamma [Q_i(x_{k+1}, \pi_i(x_{k+1})) \\ + \alpha_i Q'(x_{k+1}, \pi_i(x_{k+1}))], \end{aligned} \tag{18}$$

where Q' is an positive semi-definite function that represents previously learned knowledge, and $0 \leq \alpha_i \leq 1$ is a weighting factor such that $\alpha_{i+1} \leq \alpha_i$, and $\alpha_i \to 0$ when $i \to \infty$. Here we simply let α_i be a uniformly decreasing sequence of 0.5,0.49,0.48,...,0 as i increases. Combining the above two equations, we have

$$\begin{aligned} Q_{i+1}(x_k, u_k) = r_k + \gamma \min_{u_{k+1}} [Q_i(x_{k+1}, u_{k+1}) \\ + \alpha_i Q'(x_{k+1}, u_{k+1})] \end{aligned} \tag{19}$$

E. Actor-Critic Implementation

Algorithm 1 summarizes our implementation of the the proposed KG-QL method. Note that i increases with k at the same time in our implementation. These two indexes i and k have different meanings, and they are not equal in general (though they are equal in this work), so we did not combine them. We implemented KG-QL with an actor-critic structure [22], [23], where (17) was implemented by an actor, and (18) was implemented by a critic. Both actor and critic were feed-forward neural networks (NN) with one hidden layer (5-6-1 for the critic, and 2-6-3 for the actor). The critic has the state x_k and the action u_k as inputs, and outputs an approximation of the Q-value function, denoted by $\hat{Q}(x_k, u_k)$. The actor has state x_k as inputs, and outputs the control action u_k. The actor used a tangent sigmoid activation function $\varphi(v)$ in both the hidden layer and output layer,

$$\varphi(v) = \frac{1 - \exp(-v)}{1 + \exp(-v)} \tag{20}$$

where v is the input vector for the activation function. Note that $-1 < \varphi(v) < 1$. For the critic, it also used the same tan-sigmoid function $\varphi(v)$ in its hidden layer. But it used a linear activation function $\phi(v) = v$ in its output layer.

During training, the actor and critic back-propagated their prediction error to update their weights (Steps 5 & 6 in Algorithm 1). The prediction error of the actor $e_{a,k} \in \mathbb{R}$ is to realize (17),

$$e_{a,k} = \hat{Q}_i(x_k, u_k) + \alpha_i Q'(x_k, u_k). \tag{21}$$

Then the squared error E_a for the actor is

$$E_a = \frac{1}{2} e_{a,k}^2. \tag{22}$$

The prediction error of the critic $e_{c,k} \in \mathbb{R}$ is the temporal difference (TD) error of (18),

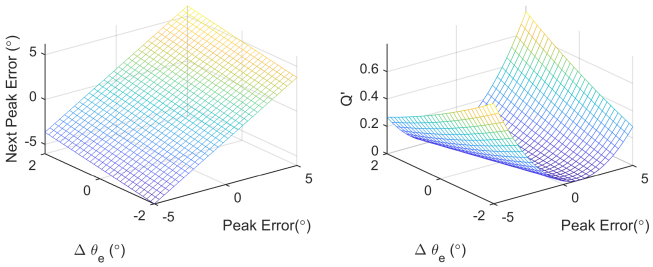
$$\begin{aligned} e_{c,k} = r_k + \gamma [\hat{Q}_i(x_{k+1}, \pi_i(x_{k+1})) \\ + \alpha_i Q'(x_{k+1}, \pi_i(x_{k+1}))] - \hat{Q}_{i+1}(x_k, u_k) \end{aligned} \tag{23}$$

which is the difference between the left-hand side and right-hand side of (18). To correct the prediction error, the weight update objective was to minimize the squared performance error E_c,

$$E_c = \frac{1}{2} e_{c,k}^2. \tag{24}$$

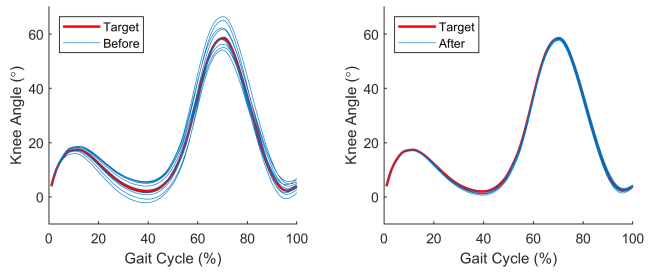
III. RESULTS

In the following experiments, knowledge was extracted from AB subjects and then transferred to an OpenSim simulated amputee subject.



(a) (b)

Fig. 2. Knowledge extraction and representation based on AB human subjects. Data shown here is from the SWF phase. (a) The regression model in (6). (b) Knowledge representation in the form of Q' in (7).



(a) (b)

Fig. 3. Knee angle profiles. (a) Before Tuning (b) After Tuning

A. OpenSim Experiment Setup

The OpenSim lower limb walking model (Fig. 1(a)) used in this work is adopted from [24] and identical to the one in [10]. In this model, five rigid-body segments linked by one degree-of-freedom pin joints are used to model human walking dynamics. For the tuning task, we defined a target range of $\pm 1°$ and ± 0.01 s for peak error ΔP_k and duration error ΔD_k, respectively. Only if for all four phases both $|\Delta P_k| < 1°$ and $|\Delta D_k| < 0.01$s were met then we said the state x_k was in the target range. If $|\Delta P_k|$ or $|\Delta D_k|$ are greater than some preset values, then state x_k was out of the safety range and the control system resets to the default position of initial impedance parameters to ensure human subject safety. More details about the target range and safety range can be found in our previous work [10, Table I]. An episode is the process from learning step $k = 0$ until termination which can either be that the state x_k enters the target range for 10 consecutive gait cycles or runs out of safety range. If terminated, the state x_k was reset with the initial impedance and initial state as the next episode began. Each OpenSim session consisted of multiple episodes with a total of maximum 500 gait cycles.

The common parameters used in the OpenSim experiment are listed as follows except those mentioned elsewhere. The discount factor γ was set to 0.95, the initial NN weights for both actor and critic were uniformly distributed between -1 and 1.

B. Knowledge Representation Results

Fig. 2 depicts the regression results data from two AB subjects in the SWF phase. In Fig 2(a), the z-axis is the next peak error ΔP_{k+1}, which is the first element of the next state x_{k+1}. Its values were obtained from the linear regression model (6) by varying one of the state variable peak error ΔP_k and one of the action variable $\Delta\theta_{e,k}$, while other state and action variables remain unchanged. We can learn how the next peak angle θ_{k+1} may change from Fig 2(a). For example, suppose $\Delta P_k = -5°$. If $\Delta\theta_{e,k} = -2°$, then $\Delta P_{k+1} < -5°$ according to Fig 2(a). Vice versa, if $\Delta\theta_{e,k} = 2°$, then $\Delta P_{k+1} > -5°$. So $2°$ may be a better choice than $-2°$ for $\Delta\theta_{e,k}$ in this case, as it makes the deviation of the next peak error ΔP_{k+1} smaller. Fig 2(b) shows the values of Q' which are computed from (7). Q' has a minimum value 0 at $(0, 0)$. Larger Q' value indicates greater cost, which is unfavorable by the RL agent.

C. Results of Reinforcement Learning with Knowledge Transfer

Fig. 3 shows the knee kinematics with different initial impedance parameters in the 10 simulation sessions were distant from the target profile, especially the peak angle errors. Clearly, after the impedance parameters were adjusted by the proposed RL controller, knee kinematics of the final acclimation stages approached the target points. Specifically, the averaged absolute values of the peak errors over the three sessions deceased from $1.23° \pm 0.77°$ to $0.36° \pm 0.32°$ for STF, from $3.13° \pm 0.31°$ to $0.52° \pm 0.24°$ for STE, from $5.53° \pm 0.89°$ to $0.63° \pm 0.68°$ degrees for SWF and from $2.72° \pm 1.67°$ to $0.12° \pm 0.25°$ for SWE. The results indicate that the proposed knowledge guided QL controller is able to adjust the prosthetic knee kinematics to reproduce the target knee profile under different initial conditions.

Fig. 4 illustrates the evolution of the state, i.e. peak errors ΔP_k and duration errors ΔD_k, during the experimental session under one of the sets of initial parameters. Since similar results were obtained from other experiment sessions, hereafter we only present the result from the first session as an example. Because all four phases were tuned simultaneously, the parameter changes in one phase would affect its subsequent phases. In Fig. 4, notice that the sharp edges on the curves indicate the impedance parameters being reset to their initial values, because failure occurred. In this example episode, the KG-QL agent was able to reduce all peak errors and duration errors to zero after approximately 150 gait cycles.

Fig. 5 illustrates the averaged root-mean-square error (RMSE) of the gait knee profile over the 10 experimental sessions. With the transferred knowledge from AB subjects, the RMSE of the proposed KG-QL algorithm drops faster than the QL without knowledge transfer, i.e., learning with $\alpha_i = 0$ in (18). Our proposed KG-QL achieved a RMSE performance less than $1°$ after only 100 gait cycles, however, QL without knowledge transfer can only achieved similar performance after about 400 gait cycles. Fig. 4 and Fig. 5

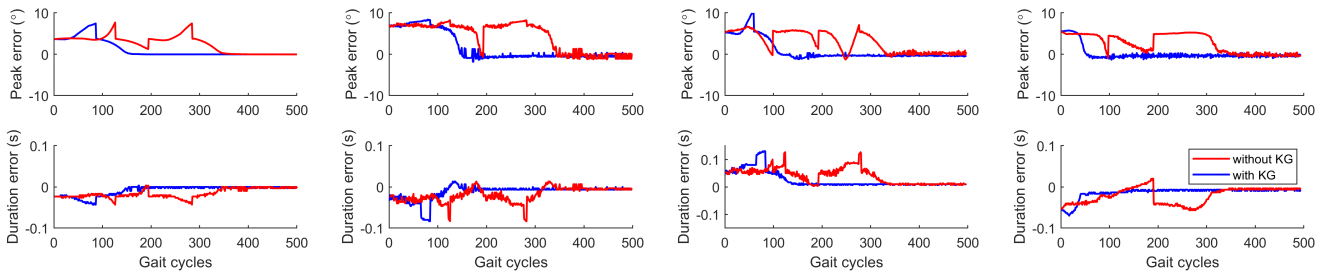


Fig. 4. Evolution of states in the four gait phases (a) phase STF (b) phase STE (c) phase SWF (d) phase SWE.

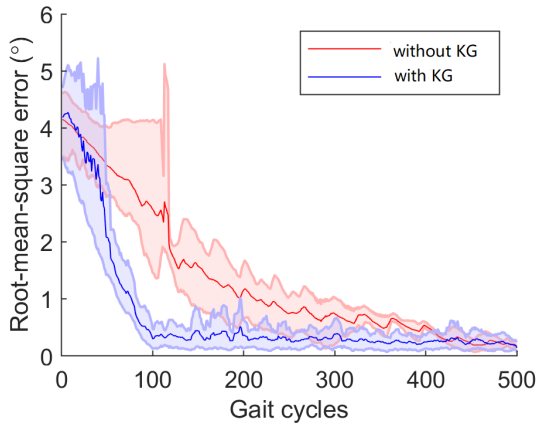


Fig. 5. Comparison of root-mean-square error (RMSE) for the with knowledge guide case and the without knowledge guide case.

show that the target task time was significantly reduced with knowledge transfer.

IV. Discussions and Conclusions

We developed a new KG-QL framework to integrate and transfer knowledge from AB subjects to OpenSim simulated amputee subjects with a common goal of optimizing impedance parameters for robotic knee prosthesis. The knowledge for transfer can be obtained offline using historical data, aka, from AB subjects in our study, to facilitate online reinforcement learning for amputee subjects. Our OpenSim simulation results validated this new approach and showed that our new scheme can help restore near-normal knee kinematics, in a time and sample-efficient manner compared to the naive learner. Our results suggested that the proposed KG-QL controller is a promising new framework when performing the cross-subject learning for the robotic knee prosthesis with human in the loop. Our demonstrated effectiveness of transfer learning from AB subjects to OpenSim simulated amputee subject may be due to the fundamental principle guiding human gaiting. Or in other words, the underlying physiology and physics represented in the relationships from impedance parameters in the FS-IC to knee joint torque and further to locomotion, should be preserved in both AB subjects and the OpenSim simulated amputee subjects, where in the latter case, the forward dynamics model should capture such relationships.

Based on experimental measurements from two AB subjects, we established a knowledge representation in the form of a regression model of the human-prosthesis dynamics, and a Q-value integration of this knowledge for transferring to the target task. We demonstrated the effectiveness of this KG-QL control framework. Our results show that, with transferred knowledge, QL was able to reach a comparable performance in the target task of an OpenSim simulated subject, but saving at least 60% of the learning time.

Our contribution is not limited to the demonstration of the feasibility of such transfer learning. It also includes our proposed RL control design framework that allows for flexible knowledge representation in the value function or system dynamics or both. In addition, we provided additional flexibility by allowing for a designer to determine how much information can be transferred from the source task to the target task. In addition, our KG-RL control framework provides a principled way to solving transfer learning problems that involve the same states and controls. Thus, it can be integrated with other TD-based methods such as SARSA and value iteration, as well as their deep learning variants, to name a few.

In this work, we demonstrated the feasibility of KG-QL based control to automatically tune robotic prostheses. To validate the full potential of our approach, we need to further evaluate it with transfers between AB subjects as well as amputees. Also, the normative (target) knee kinematics being used in this paper may not be an ideal design goal for the RL agent. We will explore other design goals that better quantify the human-prosthesis gait performance, such as gait symmetry index and stability margin.

References

[1] S. K. Au, J. Weber, and H. Herr, "Powered ankle-foot prosthesis improves walking metabolic economy," *IEEE Trans. Robot.*, 2009.

[2] S. Au, M. Berniker, and H. Herr, "Powered ankle-foot prosthesis to assist level-ground and stair-descent gaits," *Neural Networks*, vol. 21, no. 4, pp. 654–666, May 2008.

[3] A. H. Shultz and M. Goldfarb, "A Unified Controller for Walking on even and Uneven Terrain with a Powered Ankle Prosthesis," *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2018.

[4] D. Quintero, D. J. Villarreal, D. J. Lambert, S. Kapp, and R. D. Gregg, "Continuous-Phase Control of a Powered Knee-Ankle Prosthesis:

Amputee Experiments Across Speeds and Inclines," *IEEE Trans. Robot.*, 2018.
[5] B. E. Lawson, H. A. Varol, and M. Goldfarb, "Standing stability enhancement with an intelligent powered transfemoral prosthesis," *IEEE Trans. Biomed. Eng.*, 2011.
[6] F. Sup, A. Bohara, and M. Goldfarb, "Design and Control of a Powered Transfemoral Prosthesis." *Int. J. Rob. Res.*, vol. 27, no. 2, pp. 263–273, Feb 2008.
[7] M. Liu, F. Zhang, P. Datseris, and H. H. Huang, "Improving Finite State Impedance Control of Active-Transfemoral Prosthesis Using Dempster-Shafer Based State Transition Rules," *J. Intell. Robot. Syst. Theory Appl.*, vol. 76, no. 3-4, pp. 461–474, Dec 2014.
[8] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming Exploration in Reinforcement Learning with Demonstrations," in *Proc. - IEEE Int. Conf. Robot. Autom.*, 2018.
[9] S. Choi and J. Kim, "Trajectory-based Deep Latent Policy Gradient for Learning Locomotion Behaviors," in *IEEE Int. Conf. Robot. Autom.*, 2019.
[10] Y. Wen, J. Si, X. Gao, S. Huang, and H. H. Huang, "A New Powered Lower Limb Prosthesis Control Framework Based on Adaptive Dynamic Programming," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 9, pp. 2215–2220, Sep 2017.
[11] Y. Wen, J. Si, A. Brandt, X. Gao, and H. Huang, "Online Reinforcement Learning Control for the Personalization of a Robotic Knee Prosthesis," *IEEE Trans. Cybern.*, pp. 1–11, Jan 2019.
[12] M. Li, X. Gao, W. Yue, S. Jennie, and H. He, "Offline Policy Iteration Based Reinforcement Learning Controller for Online Robotic Knee Prosthesis Parameter Tuning," in *Proc. - IEEE Int. Conf. Robot. Autom.*, 2019.
[13] X. Gao, Y. Wen, M. Li, J. Si, and H. Huang, "Robotic Knee Parameter Tuning Using Approximate Policy Iteration," in *Cogn. Syst. Signal Process.*, F. Sun, H. Liu, and D. Hu, Eds. Singapore: Springer, Singapore, Nov 2019, pp. 554–563.
[14] M. E. Taylor and P. Stone, "Transfer Learning for Reinforcement Learning Domains : A Survey," *J. Mach. Learn. Res.*, vol. 10, pp. 1633–1685, 2009.
[15] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. Van Hasselt, and D. Silver, "Successor features for transfer in reinforcement learning," in *Adv. Neural Inf. Process. Syst.*, 2017.
[16] M. Asadi and M. Huber, "Effective control knowledge transfer through learning skill and representation hierarchies," in *IJCAI Int. Jt. Conf. Artif. Intell.*, 2007.
[17] K. Kansky, T. Silver, D. A. Mély, M. Eldawy, M. Lázaro-Gredilla, X. Lou, N. Dorfman, S. Sidor, S. Phoenix, and D. George, "Schema networks: Zero-shot transfer with a generative causal model of physics intuitive," in *34th Int. Conf. Mach. Learn. ICML 2017*, 2017.
[18] G. Joshi and G. Chowdhary, "Cross-Domain Transfer in Reinforcement Learning Using Target Apprentice," in *Proc. - IEEE Int. Conf. Robot. Autom.*, 2018.
[19] M. P. Kadaba, H. K. Ramakrishnan, and M. E. Wootten, "Measurement of lower extremity kinematics during level walking," *J. Orthop. Res.*, vol. 8, no. 3, pp. 383–392, May 1990.
[20] J. Si and Y. T. Wang, "Online learning control by association and reinforcement," *IEEE Trans. Neural Networks*, vol. 12, no. 2, pp. 264–76, Mar. 2001.
[21] R. Bellman, *Dynamic programming*. Princeton University Press, 1957.
[22] R. S. Sutton and A. G. Barto, *Reinforcement learning : an introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018.
[23] J. Si, A. G. Barto, W. B. Powell, and D. C. Wunsch, *Handbook of learning and approximate dynamic programming*. IEEE Press, 2004.
[24] Jennifer Hicks, "From the Ground Up: Building a Passive Dynamic Walker Model," 2014. [Online]. Available: https://simtk-confluence.stanford.edu:8443/display/OpenSim33/From+the+Ground+Up{%}3A+Building+a+Passive+Dynamic+Walker+Model