This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Toward Expedited Impedance Tuning of a Robotic Prosthesis for Personalized Gait Assistance by Reinforcement Learning Control

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

***Abstract*—Personalizing medical devices such as lower limb wearable robots is challenging. While the initial feasibility of automating the process of knee prosthesis control parameter tuning has been demonstrated in a principled way, the next critical issue is to improve tuning efficiency and speed it up for the human user, in clinic settings, while maintaining human safety. We, therefore, propose a policy iteration with constraint embedded (PICE) method as an innovative solution to the problem under the framework of reinforcement learning. Central to PICE is the use of a projected Bellman equation with a constraint of assuring positive semidefiniteness of performance values during policy evaluation. Additionally, we developed both online and offline PICE implementations that provide additional flexibility for the designer to fully utilize measurement data, either from on-policy or off-policy, to further improve PICE tuning efficiency. Our human subject testing showed that the PICE provided effective policies with significantly reduced tuning time. For the first time, we also experimentally evaluated and demonstrated the robustness of the deployed policies by applying them to different tasks and users. Putting it together, our new way of problem solving has been effective as PICE has demonstrated its potential toward truly automating the process of control parameter tuning for robotic knee prosthesis users.**

***Index Terms*—Impedance control, knee prosthesis, policy iteration, rehabilitation robotics, reinforcement learning (RL).**

I. Introduction

ROBOTIC prostheses have emerged with recent breakthroughs in mechanical design, control theory, and biomechanics [1]–[4]. These robotic prostheses have manifested exceptional potentials to benefit lower limb amputees in various ways, such as reducing metabolic consumption [5], enhancing balance and stability [6], augmenting adaptability to varying walking speeds and inclines [7], and enabling walking on changing terrains seamlessly [8]–[10]. The finite-state machine impedance control (FSM-IC) has been the most adopted control framework for prosthetic devices [11]–[14], because studies have suggested that the human nervous system modulates the impedance of lower limb joints in order to realize stable and robust dynamics when walking on various terrains [15], [16]. In addition to the traditional FSM-IC, Azimi *et al.* [17] recently proposed to estimate the ground reaction force (GRF) [17], and integrated it into impedance controller [18], achieving improved tracking performance of a prosthetic knee.

The challenges in applying FSM-IC in robotic prostheses are as follows: 1) tuning of a large number of impedance control parameters (e.g., 12–15 in a knee prosthesis for level walking only) in order to achieve safe human–machine–environment interaction and sufficient characterization of limb movement within a gait cycle [1], [13], [19], [20]; 2) these control parameters must be personalized to assist individual amputees' gait. To find a modest set of parameters, in current clinical practice, highly trained prosthetists need to spend hours to arduously hand-tune the parameters for each amputee user and each locomotion mode (e.g., level walking, ramp ascent/descent) based mainly on the subjective observations of the user's gait performance [21]. Not only does it lack precision, but also it needs intensive time and efforts due to the inability of humans to tune high-dimensional parameters simultaneously.

Given a growing need for facilitating the assistive device personalization, the research community has developed various solutions to automate the process. A few estimation approaches were proposed to determine the impedance parameters by mimicking the nature of biological joints via modeling [22], [23]. Furthermore, optimization approaches were developed and validated on able-bodied subjects to minimize metabolic cost by using a small number of control parameters for exoskeletons (no more than 4) [24], [25]. Beyond merely identifying a set of optimal parameters, a couple of studies have attempted to learn the optimal sequential decision-making for optimizing high-dimensional prosthesis control parameters. Employing knowledge and skills from experienced prosthetists, an expert system was developed to encode human decisions as automatic tuning rules [26]. The method was challenged by the lack of sufficient data collected from the prosthetists in device tuning.

Manuscript received November 12, 2020; revised February 22, 2021; accepted April 30, 2021. This work was supported in part by the National Science Foundation under Grant 1563454, Grant 1563921, Grant 1808752, and Grant 1808898. This article was recommended for publication by Associate Editor S. Oh and Editor E. Yoshida upon evaluation of the reviewers' comments. *(Corresponding authors: He (Helen) Huang; Jennie Si.)*

Minhan Li, Yue Wen, and He Huang are with the NCSU/UNC Department of Biomedical Engineering, North Carolina State University, Raleigh, NC 27695-7115 USA, and with the University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA (e-mail: mli37@ncsu.edu; ywen3@ncsu.edu; hhuang11@ncsu.edu).

Xiang Gao and Jennie Si are with the Department of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85281 USA (e-mail: xgao29@asu.edu; si@asu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TRO.2021.3078317.

Digital Object Identifier 10.1109/TRO.2021.3078317

1552-3098 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Alternatively, an actor–critic reinforcement learning (RL) based method, called direct Heuristic Dynamic Programming (dHDP), was designed to directly obtain the impedance tuning policy via interaction with the human-prosthesis system in an online manner [27], [28] without a closed-form model of the system.

Although the aforementioned studies have demonstrated the feasibility of applying automatic tuning to wearable robots with human-in-the-loop, little attention has been paid to address the efficiency and robustness of the tuning algorithms from a user's perspective. First, efficiency of a tuning algorithm (i.e., the ability to safely complete the online tuning rapidly in time) is critical for the clinical translation of a new method due to patient-in-the-loop. Second, the robustness of the tuning algorithm quantifies whether the optimal control parameters or learned prosthesis tuning policy can handle situations when walking condition (e.g., treadmill walking versus level-ground walking) or user has changed. Additionally, a robust policy is expected to alleviate computational burden in online learning or continued customization, improve user safety during automated prosthesis tuning, and expedite the tuning process in clinics. Therefore, the objective of this study was to develop an efficient and robust automatic tuning method for a robotic prosthetic knee to reproduce near-normal knee kinematics during walking in clinic settings, which include level-ground and ramp. The tuning goal stemmed from the fact that having amputees walk normally as the able-bodied people has been widely used as the design goal or the evaluation criteria for knee prosthesis control [20], [29]. To this end, the objective of this study includes as follows: 1) developing a learning algorithm with enhanced data and time efficiency; 2) investigating the robustness of trained policies against changes in task and user.

To address the efficiency in learning a control parameter tuning policy for robotic prostheses, policy iteration, a classical RL method, lends itself as a promising candidate. This is because, such as general RL-based control framework, it has an excellent capability of learning optimal sequential decisions in high-dimensional space [30]. Also, our approach is data-driven, which learns directly from interactions between the actions and the consequences measured from the human–robot system. In another word, there is no need of explicitly performing a system identification procedure, either online or offline, prior to or during controller design as most control theoretic approaches do, including adaptive control. In addition, as the process of customizing prosthesis control parameter design for a human user does not render abundance of data, which is a necessary feature in those deep RL applications [31]–[34], the classic policy iteration framework with moderate demands on data amount is a suitable approach. Furthermore, previous evidences suggest that the policy iteration has the advantage of fast convergence over other classical RL algorithms, such as value iteration and gradient-based policy search [35], [36] (including our previously reported dHDP [37], which is a stochastic gradient method). The idea of policy iteration is to iteratively improve the policy by alternately carrying out the policy evaluation and the policy improvement steps [38], [39]. As the efficiency in the policy evaluation step significantly influences the overall learning algorithm efficiency, the problem boils down to improving the efficiency of policy evaluation.

Therefore, in this study, based upon the policy iteration framework, we proposed an improved solution, namely the policy iteration with constraint embedded (PICE), to enhance policy training efficiency. Such enhancement was enabled by the following special designs in the algorithm tailored for our application. First, we leveraged a projected Bellman equation from which the performance values can be approximately solved during each policy evaluation step. This is to ensure positive semidefiniteness of the value function to avoid incorrect outcome of negative values due to approximation errors. Second, inspired by a widely utilized quadratic cost formulation in successful applications of RL [40]–[42], we adopted simple yet efficient quadratic basis functions to approximately evaluate policies rather than using complex structures such as actor–critic networks. Additionally, we provided flexible implementations of PICE to perform either offline or online learning, a proper use of which can further improve learning efficiency. The PICE algorithm was implemented and tested on human-prosthesis systems; the efficiency and robustness of the tuning policy were evaluated quantitatively.

The main contributions of this study are as follows.

1) We developed a new problem solution PICE based on policy iteration RL for improving the efficiency of automatic tuning of the robotic knee control parameters. Our new design entails constraining the performance values from going into the incorrect range of having negative values.
2) The proposed PICE algorithm was implemented and tested in real time on human subjects for prosthesis control parameter tuning. We successfully demonstrated its efficiency and effectiveness in experiments involving human subjects.
3) The robustness of learned control parameter tuning policies against changes of tasks and users were tested on human subjects. The successful demonstration of robustness of PICE suggests its potential values for clinical application.

The remainder of this article is organized as follows. Section II describes the problem to be solved and shows how it relates to the theory of optimal sequential decision. Section III presents details of the proposed RL algorithm for improving data and time efficiency. Section IV elaborates on the considerations regarding the implementations of the algorithm. Results are presented in Section V. Finally, we discuss these results and limitations of this study in Section VI. Section VII concludes this article.

II. Problem Formulation

In this study, the proposed PICE algorithm aims at determining optimal control parameter tuning policies to supplement the impedance controller of a robotic knee prosthesis in order for its user to restore a near-normal knee motion. The algorithm is implemented within a well-established FSM-IC framework [1], [13], as shown in Fig. 1.

A. Finite-State Machine Impedance Controller

As depicted in the FSM-IC block of Fig. 1, a single gait cycle during walking is decomposed into four distinct phases in the FSM-IC: stance flexion (STF), stance extension (STE),

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

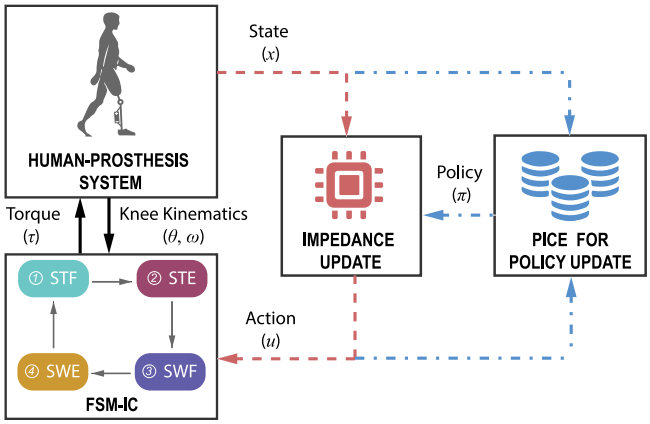


Fig. 1. Schematic illustration of the RL-based impedance tuning for a robotic knee prosthesis system within the FSM-IC framework. Red dashed lines denote inputs and outputs in the impedance update loop, whereas blue dash-dotted lines stand for those in the policy update loop. A tuning policy acts to adjust impedance parameters, according to the state of the human-prosthesis system, in the FSM-IC to regulate the interaction force with users. The policy can be obtained from and further updated by the proposed PICE algorithm. In the block of FSM-IC, STF, STE, SWF, and SWE stand for stance flexion, stance extension, swing flexion, and swing extension, respectively.

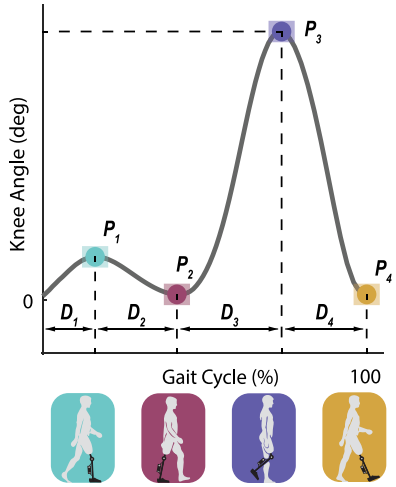


Fig. 2. Features of knee kinematics in a single gait cycle. The subscript numbers 1 through 4 denote respective phases (i.e., STF, STE, SWF, and SWE) of a gait cycle, to which the features correspond.

swing flexion (SWF), and swing extension (SWE). The major gait events determining the phase transitions are identified by utilizing the measurements of knee angle and GRF together using the Dempster–Shafer theory, as described in [13].

For each phase in a single gait cycle, the FSM selects the corresponding set of impedance parameters for the impedance controller to generate a torque τ at the prosthetic knee joint based on the impedance control law

$$\tau = K(\theta_e - \theta) - C\omega \tag{1}$$

where the impedance controller consists of three control parameters: the stiffness K, the equilibrium angle θ_e and the damping C. Real-time sensor feedback includes the knee joint angle θ and angular velocity ω. Therefore, a total of 12 impedance parameters need to be regulated in a gait cycle.

B. Dynamic Process of Impedance Update

As shown in Fig. 1, the impedance update loop is executed by specified policies to adjust impedance parameters for the FSM-IC. Without loss of generality, the following formulation toward describing the dynamic process of impedance update for a robotic prosthesis is applicable to all four phases in the FSM-IC. This is owing to the fact that, despite sharing the identical framework for learning the tuning policy, each phase is associated with an independent tuning policy running in parallel.

We consider the human-prosthesis system as a discrete-time system with unknown dynamics f, which was also studied in [28] and [43]

$$\begin{aligned} x_{(k+1)} &= f(x_{(k)}, u_{(k)}),\ k = 0, 1, \ldots \\ u_{(k)} &= \pi(x_{(k)}) \end{aligned} \tag{2}$$

where k denotes the discrete index in the impedance update loop in Fig. 1. We denote x and u as state and action variables of the process, respectively, while the tuning policy π represents a mapping to determine actions according to current states.

In the context of impedance update, the abovementioned state variables are defined based on features extracted from the knee kinematic profiles for each segmented phase in the FSM-IC. Specifically, the continuous knee profile within a single gait cycle (from the heel strike to the next heel strike of the same foot) is characterized by four discrete points, each of which is a local extrema in the corresponding phase along the profile as shown in Fig. 2. Each point is associated with two features, the angle feature P and the duration feature D, respectively. Similarly, target features (P^d and D^d) in each phase can be determined from the representative data of knee kinematics in the able-bodied population [44].

Consequently, state variables $x \in \mathbb{R}^2$ are defined as the differences between measured features and target features (referred as the peak error and the duration error, respectively) in a specific phase at each impedance update as follows:

$$x = [P - P^d, D - D^d]^T. \tag{3}$$

Meanwhile, action variables $u \in \mathbb{R}^3$ are defined in the following form:

$$u = [\Delta K, \Delta\theta_e, \Delta C]^T \tag{4}$$

where $\Delta K, \Delta\theta_e$, and ΔC are the adjustments of impedance parameters for the corresponding phase at each instance of impedance update.

C. Policy Update

The policy update loop is carried out by the proposed PICE algorithm, as shown in Fig. 1, to progressively approach optimal policies with respect to specified objectives. In this study, the objective is to regulate states with minimal control energy expenditure over the process of impedance update in order to keep the peak error and duration error as close to zero as possible.

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Hence, at each instance of impedance update, we assign a scalar stage cost with a quadratic form

$$g(x_{(k)}, u_{(k)}) = {x_{(k)}}^T R_s x_{(k)} + {u_{(k)}}^T R_a u_{(k)} \tag{5}$$

where $R_s \in \mathbb{R}^{2\times 2}$ and $R_a \in \mathbb{R}^{3\times 3}$ are both positive semidefinite (PSD) matrices. Thereby, given a policy π^i after the ith policy update, the corresponding discounted cost-to-go in an infinite horizon, namely the action-dependent value function $Q^{(i)}$, is given by

$$\begin{aligned} Q^{(i)}(x_{(k)}, u_{(k)}) &= g(x_{(k)}, u_{(k)}) + \sum_{t=k+1}^{\infty} \alpha^{t-k} g(x_{(t)}, u_{(t)}) \\ &= g(x_{(k)}, u_{(k)}) + \alpha Q^{(i)}(x_{(k+1)}, \pi^i(x_{(k+1)})) \end{aligned} \tag{6}$$

where α is the discount factor. This value function is nonnegative due to the definition of $g(x_{(k)}, u_{(k)})$ in (5), and the value function reflects a measure of the performance when action $u_{(k)}$ is applied at state $x_{(k)}$ and the control policy π^i is followed thereafter.

The goal for PICE, as in value-based RL algorithms [38], is to seek an optimal policy that minimizes the cost-to-go by solving the Bellman optimality equation approximately

$$\begin{aligned} Q^*(x_{(k)}, u_{(k)}) &= \min_{\pi} Q(x_{(k)}, u_{(k)}) \\ &= g(x_{(k)}, u_{(k)}) + \alpha \min_{u(k+1)} Q^*(x_{(k+1)}, u_{(k+1)}) \\ &= g(x_{(k)}, u_{(k)}) + \alpha Q^*(x_{(k+1)}, \pi^*(x_{(k+1)})) \end{aligned} \tag{7}$$

where π^* and Q^* denote the optimal tuning policy and the associated optimal value, respectively.

III. Policy Iteration With Constraint Embedded

To solve the abovementioned Bellman equation approximately and effectively, we propose the PICE algorithm. Instead of achieving a close approximation to the value function as in most general policy iteration algorithms, the PICE makes use of simple quadratic polynomial basis functions, which are expected to provide simplified, albeit with approximation errors and, thus, efficient solutions during policy evaluation. Meanwhile, aiming at improving the quality of policy evaluation, the PICE introduces a PSD constraint on the approximated value function to prevent the value function from becoming negative due to approximation errors.

A. Value Function Approximation

To represent the approximate value function $\hat{Q}^{(i)}$ associated with the policy π^i being evaluated (referred to as target policy hereafter), a linear parametric combination of basis functions is often used in classic policy iterations as follows, because of its virtues of easy-to-implement and fairly transparent behavior [35], [36]

$$\hat{Q}^{(i)}(x, u) = \phi(x, u)^T r^{(i)} \tag{8}$$

where $\phi(x, u) \in \mathbb{R}^m$ is the vector of fixed basis functions of states and actions, and the weight parameter vector $r^{(i)} \in \mathbb{R}^m$ varies as policy updates. Hereafter, we ignore subscript k in state and action and replace them with x and u, respectively, for notation simplicity. Instead of the usual universal approximators, such as multilayer perceptron neural networks, radial basis functions, and splines, we opt for a simple structure of quadratic polynomials as the basis functions to practically further simplify the basis functions, therefore, potentially reducing uncertainties associated with large number of free parameters used in the approximation.

As a result, the approximating value function can be rewritten in the following equivalent form of weighted inner product, which yields all possible quadratic basis functions of states and actions

$$\begin{aligned} \hat{Q}^{(i)}(x, u) &= \begin{bmatrix} x \\ u \end{bmatrix}^T H^{(i)} \begin{bmatrix} x \\ u \end{bmatrix} \\ &= \begin{bmatrix} x \\ u \end{bmatrix}^T \begin{bmatrix} H_{xx}^{(i)} & H_{xu}^{(i)} \\ H_{ux}^{(i)} & H_{uu}^{(i)} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \end{aligned} \tag{9}$$

where $H^{(i)}$ is a PSD matrix, and $H_{xx}^{(i)}$, $H_{xu}^{(i)}$, $H_{ux}^{(i)}$, and $H_{uu}^{(i)}$ are submatrices of $H^{(i)}$ with proper dimensions. By rearranging and grouping like terms in (9), we can convert the weight parameter vector $r^{(i)}$ to the matrix $H^{(i)}$ and vice versa.

B. Policy Iteration Under Constraint

Two iterative procedures, policy evaluation and policy improvement, are alternately performed in a standard approximate policy iteration. The policy evaluation is to find an approximated value function $\hat{Q}^{(i)}$ satisfying the Bellman equation under the target policy π^i as follows:

$$\hat{Q}^{(i)}(x, u) = g(x, u) + \alpha \hat{Q}^{(i)}(f(x, u), \pi^i(f(x, u))). \tag{10}$$

Replacing the approximate function $\hat{Q}^{(i)}$ with parametric basis functions in (8), we obtain the following equivalent form and its vector-matrix version

$$\phi(x, u)^T r^{(i)} = g(x, u) + \alpha \phi(f(x, u), \pi^i(f(x, u)))^T r^{(i)} \tag{11}$$

$$\boldsymbol{\Phi} r^{(i)} = \boldsymbol{g} + \alpha T \boldsymbol{\Phi} r^{(i)} \triangleq B(\boldsymbol{\Phi} r^{(i)}) \tag{12}$$

where the matrix $\boldsymbol{\Phi}$ consists of basis functions for every possible state-action pair in its rows, and the corresponding stage costs make up the vector $\boldsymbol{g}$. In addition, the T and B denote the state transition matrix and the Bellman operator, respectively, under the target policy.

The policy improvement then follows to seek an optimal mapping from states to actions as an improved target policy, with which the next iteration starts

$$\pi^{i+1}(x) = \arg\min_{u\in U} \hat{Q}^{(i)}(x, u) \tag{13}$$

where U is the admissible action space. With the choice of quadratic basis functions, the improvement procedure (13) is equivalent to solving a quadratic programming (QP) problem. The equivalence can be easily observed by formulating the minimization problem of (9) over the actions with any given states. As such, the equivalent QP problem can be readily written

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

as

$$\pi^{i+1}(x) = \arg\min_{u\in U}\{u^T H_{uu}^{(i)} u + 2x^T H_{xu}^{(i)} u\}. \tag{14}$$

Directly solving the Bellman equation (12) based on the abovementioned standard policy iteration framework, we obtained a preliminary proof-of-concept result for offline policy training [45]. To be more efficient and also to accommodate both offline and online scenarios, we proposed the PICE algorithm with the following details.

As a result of the approximation error, some of the value functions solved from (12) may yield negative values. This clearly indicates poor approximation given the positive-valued stage cost and the definition of value function. Stemming from insights on the formulated problem, we, therefore, impose a PSD constraint on the solved value functions from (12) toward an improved solution. Specifically, we seek an approximated value function $\hat{Q}^{(i)}$ satisfying the following projected Bellman equation that is to be solved by PICE

$$\mathbf{\Phi} r^{(i)} = \text{proj}_{S_+}(B(\mathbf{\Phi} r^{(i)})) \tag{15}$$

where proj_{S_+} denotes the operator of projection onto a closed convex subset S_+. The closed convex subset S_+ is contained in a subspace spanned by the columns of $\mathbf{\Phi}$

$$S_+ = \mathbf{\Phi} R_+ \tag{16}$$

where $R_+ \subset \mathbb{R}^m$ is the PSD cone in the vector space of $\mathbb{R}^m$.

The idea of solving the projected Bellman equation was also used in the least square policy iteration (LSPI) algorithm [35]. The PICE algorithm, however, imposes a new and tighter constraint and, thus, results in a different projected Bellman equation. Specifically, the PICE requires the solved value function from the projected Bellman to be positive semidefnite. Similar to the LSPI, the PICE can be used as either on-policy or off-policy learning schemes. In general, the behavior policy governing sample distributions for policy evaluation is different from the target policy being evaluated for off-policy learning scheme, whereas they are the same for on-policy learning scheme. Our discussions in the following on PICE cover both learning schemes.

Inspired by established results [46], we convert the problem of solving (15) to the one that corresponds to the solution of the following variational inequality:

$$(\mathbf{\Phi} r^{(i)} - B(\mathbf{\Phi} r^{(i)}))^T \Xi (\mathbf{\Phi} r - \mathbf{\Phi} r^{(i)}) \geq 0 \quad \forall r \in R_+ \tag{17}$$

where Ξ is a diagonal matrix and its diagonal elements are the steady-state probabilities of the Markov chain under the behavior policy for each corresponding state-action pair in $\mathbf{\Phi}$.

For notation simplicity, an equivalent form of inequality (17) can be readily written as

$$\begin{aligned} &(A^{(i)} r^{(i)} - b^{(i)})^T (r - r^{(i)}) \geq 0 \quad \forall r \in R_+ \\ &A^{(i)} = \mathbf{\Phi}^T \Xi (I - \alpha T) \mathbf{\Phi} \\ &b^{(i)} = \mathbf{\Phi}^T \Xi \boldsymbol{g}. \end{aligned} \tag{18}$$

Involving every single possible state-action pair in terms $A^{(i)}$ and $b^{(i)}$, the inequality is intractable to solve in closed form. Instead, in practice, we can replace the two terms with approximated $\hat{A}^{(i)}$ and $\hat{b}^{(i)}$ by using observational samples, as shown in [46]

$$\begin{aligned} \hat{A}^{(i)} &= \frac{1}{N+1} \sum_{n=0}^{N} \phi(s_n) \left(\phi(s_n) - \alpha \frac{p_{s_n,s'_n}}{q_{s_n,s'_n}} \phi(s'_n) \right)^T \\ \hat{b}^{(i)} &= \frac{1}{N+1} \sum_{n=0}^{N} \frac{p_{s_n,s'_n}}{q_{s_n,s'_n}} \phi(s_n) g(s_n) \end{aligned} \tag{19}$$

where n and N denote sample index and sample size of the collected data, respectively. The variable $s_n \triangleq (x_n, u_n)$ denotes a sample of a state-action pair, and the variable $s'_n \triangleq (x'_n, u'_n)$ denotes the next sample pair following s_n in the sampling trajectory. In addition, the ratio term $p_{s_n,s'_n}/q_{s_n,s'_n}$ is in place to correct any mismatch in state transition probability matrix T between the behavior policy and the target policy. Importance sampling can be readily applied to address the mismatch. Specifically, p_{s_n,s'_n} and q_{s_n,s'_n} denote transition probability from the sample s_n to the sample s'_n under the target policy and behavior policy, respectively. In the context of Q value function, the ratio can be further simplified to the following form as shown in [39]:

$$\frac{p_{s_n,s'_n}}{q_{s_n,s'_n}} = \frac{\delta\left(u'_n = \pi^i(x'_n)\right)}{\nu\left(u'_n | x'_n\right)} \tag{20}$$

where $\delta(\cdot)$ denotes the indicator function (i.e., equals to 1 if $u' = \pi^i(x')$ and 0 otherwise), and $\nu(\cdot)$ denotes the conditional probability of taking action u of the behavior policy in state x.

C. Iterative Approach for Solving Policy Evaluation

To solve the variational inequality (18), the following iterative approach has been proposed in previous studies [46], [47] to approximate $r^{(i)}$ with $\hat{r}_j^{(i)}$

$$\hat{r}_{j+1}^{(i)} = \text{proj}_E[\hat{r}_j^{(i)} - \gamma_j(\hat{A}_j^{(i)} \hat{r}_j^{(i)} - \hat{b}_j^{(i)})] \tag{21}$$

where j denotes iterative steps and E is the constraint set for the solution.

To result in a convergent sequence, the approach also requires constraint set E to be closed, bounded, and convex [47]. In our case, however, the PSD cone is not bounded. To address this issue, we construct a convex set with an intersection between the PSD cone and an Euclidean ball as follows:

$$E = R_+ \cap Z_\delta \tag{22}$$

where Z_δ denotes a closed Euclidean ball centered at the origin with the radius of δ, the choice of radius can be as large as needed to cover a sufficient subset of the original PSD cone. Since (21) involves a projection onto the intersection of two convex sets, the Dykstra's projection algorithm is applied [48].

Furthermore, the step size γ_j also needs to be decreasing on the order of $1/j$ to guarantee a convergent sequence resulted from (21) as follows:

$$\frac{\gamma_j - \gamma_{j+1}}{\gamma_j} = \mathcal{O}\left(\frac{1}{j}\right). \tag{23}$$

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

IV. Implementation

To apply PICE for tuning the impedance control parameters of a prosthetic knee on human subjects, some practical issues need to be considered during implementation. Hereafter, an experimental trial with a human subject, namely a trial, refers to a single experiment with impedance and policy initializations that allow prosthetic knee control parameters to adapt until reaching a stopping criterion.

A. Human Variability and Stopping Criterion

Due to variations in physical conditions and fatigue of human subjects, measurement noise, and other uncertainties associated with the environment, data recorded from the human-prosthesis system need to be processed and tuning target set needs to be realistically specified. Specifically, impedance update is set to take place every four gait cycles to reduce noise introduced by human stride-to-stride variance. That is to say, knee features are averaged over the four gait cycles with a single impedance update to form a state-action pair to be used in policy update. In addition, we introduce a target set as tolerance levels of error (specifically, $\pm 1.5^\circ$ for peak errors and $\pm 3\%$ for duration errors) to account for the inherent walking variability [44], [49]. Consequently, we consider an impedance parameter tuning procedure in a given phase a success if the errors stay within the target set for 8 out of 10 consecutive impedance updates. If all four phases become successful, a trial is successful and is considered reaching the stopping criterion.

B. Safety Bounds for Impedance Tuning

In each trial, a set of initial impedance parameters are selected for the prosthetic knee, and then subjects experience a series of impedance updates guided by tuning policies for each phase of the FSM. While the initial impedance parameters are randomly selected, they need to be feasible for walking. Such feasible initial impedance parameter setting is validated prior to the start of a trial and is verified by the experimenter either via visual inspection if the subject is capable of walking without holding on a handrail, or via the subject's verbal expression. To avoid any potential harm to human subjects caused by unsafe parameters and associated knee kinematics, we set a safety range within which peak error is allowed us to vary. Once a peak error is beyond the safety range, impedance parameters will be reset to the initial ones, which are known to be safe. Herein, the peak error bounds are set to $\pm 12^\circ$ for all four phases since they cover two standard deviations of knee kinematic features in normal walking among different test subjects [44]. More importantly, the safety range also defines a compact set for states and actions, which consequently guarantees our implementation to fulfill the requirement of initial admissible policy for general policy iteration algorithms because the zero-output policy is always an eligible initial policy.

C. Implementations of PICE

Prior to feeding data into the PICE algorithm for policy training, feature scaling was first performed on state and action variables for all four phases. To normalize them into a comparable unit magnitude, following scaling factors were selected in this study. Specifically, the state variables x in (3) were normalized with a scaling factor of 8 and 0.24 for the respective peak error and duration error. Similarly, the action variables u in (4) were normalized with a scaling factor of 0.05, 0.5, and 0.0005 for respective adjustments of stiffness, equilibrium angle, and damping. The only one exception was that the value for equilibrium angle in the SWF was set to 1 when considering phase differences. Meanwhile, to keep actions staying in a reasonable range, we set the admissible space U in (14) for normalized action variables to a range between -1 and 1.

Algorithm 1: Offline Off-Policy PICE.

Initialization:

Random initial target policy π^0, policy update index $i \leftarrow 0$;
Empty replay buffer *DS* with capacity N;
Choose a tolerance for offline training ε_a.

Offline Data Preparation:

Populate the buffer *DS* with N samples of 4-tuple (x_n, u_n, g_n, x'_n) generated by a behavior policy from previous experiments, where n is sample index and x' denotes the next state in each sample.

Iteration:

repeat
  (Update Next Actions) Compute u'_n with current target policy $\pi^i(x'_n)$ for every sample in *DS*;
  (Update Sampling Weight) Compute the ratio $\frac{p_{s_n,s'_n}}{q_{s_n,s'_n}}$ by (20) for every sample in *DS*;
  (Policy Evaluation) Evaluate policy π^i by solving (21) for $\hat{r}^{(i)}$ and $\hat{Q}^{(i)}$, along with approximating (19) using all samples in *DS*;
  (Policy Improvement) Update policy π^{i+1} by (14), $i \leftarrow i+1$;
until $\|\hat{r}^{(i)} - \hat{r}^{(i-1)}\|_2 \leqslant \varepsilon_a$;
return $\hat{Q}^* \leftarrow \hat{Q}^{(i)}$ and $\hat{\pi}^* \leftarrow \pi^i$.

The PICE features a flexibility that can be implemented in both offline and online manners by taking off-policy and on-policy learning schemes, respectively. The procedures of both implementations are described as pseudocodes in Algorithms 1 and 2. A summary of value selections for parameters in PICE implementations is listed as follows. Penalty matrices for states R_s and actions R_a were set to $\mathrm{diag}(1, 0.5)$ and $\mathrm{diag}(0.01, 0.01, 0.01)$, respectively, while discount factor α was selected as 0.9. The radius of Euclidean ball δ was assigned to 100. The tolerance for offline training ε_a was set to 10^{-4}. The batch size N_b for online training samples was selected as 15.

The two implementations differ from each other in two major aspects: training data and termination condition. For offline PICE, the training data consist of a fixed set of four-tuple samples, which were collected from previous studies and already stored in a buffer beforehand. The same data are repeatedly utilized until the iteration terminates. The off-policy scheme

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 2: Online On-Policy PICE.

Initialization:

Choose a batch size N_b, and empty replay buffer *DS*;
Random initial target policy π^0, policy update index $i \leftarrow 0$;
Random initial state $x_{(0)}$, impedance update index $k \leftarrow 0$;
Choose an initial action $u_{(0)}$ using policy $\pi^0(x_{(0)})$;

Iteration:

repeat
  (Online Data Collection) Execute the action $u_{(k)}$, observe stage cost $g_{(k)}$ and next state $x_{(k+1)}$, choose the next action $u_{(k+1)}$ by following current target policy π^i, form a 5-tuple sample $(x_{(k)}, u_{(k)}, g_{(k)}, x_{(k+1)}, u_{(k+1)})$ and store it in *DS*;
  if $k = iN_b (i \in \mathbb{N})$ **then**
    (Policy Evaluation) Evaluate policy π^i by solving (21) for $\hat{r}^{(i)}$ and $\hat{Q}^{(i)}$, along with approximating (19) using all samples in *DS*;
    (Policy Improvement) Update policy π^{i+1} by (14), $i \leftarrow i + 1$;
    (Reset Buffer) Empty *DS*;
  end if
  $k \leftarrow k + 1$;
until Early-stopping termination condition is fulfilled
return $\hat{Q}^* \leftarrow \hat{Q}^{(i)}$ and $\hat{\pi}^* \leftarrow \pi^i$.

is adopted for offline training as the original behavior policy generating the training samples is different from the target policy to be evaluated for every iteration. On the other hand, the online PICE follows an on-policy scheme. In each online iteration, the target policy keeps interacting with the human-prosthesis system to generate on-policy samples of up the specified batch size, thereby performing the policy evaluation correspondingly. The sample will be discarded after usage and replaced with new ones for the new target policy.

In addition, in contrast to offline PICE, which terminates iterations until the parameters in the parameterized value function get converged, we used an early stopping termination condition to deactivate an online training process. It is not only to prevent over training, but also to take into consideration that human subjects can only walk for about 30–60 min during an experiential trial due to physical constraints. Specifically, during online training, we analyze the trend in evolution of the stage cost based on the current policy every time when we have newly collected N_b samples of state-action pairs. When either of the two conditions listed below is fulfilled, the online training is deactivated and the rest of impedance update is carried out with the current policy until policy update is triggered again.

1) The case of no occurrence of impedance reset due to hitting the safety bounds. We fit a linear regression model between the time series of stage cost and that of impedance update. From the model, we obtain a confidence interval (specifically 95% confidence interval) around the slope of the regression line. If the interval falls below zero, which signals a rigorously decreasing stage cost as impedance parameters updated according to the current policy, we deactivate the online training.
2) The case of using stage costs. We averaged the stage costs over samples being analyzed. If it is smaller than a threshold value ε_b, online training is deactivated. We set the threshold to 0.043, which is equivalent to the stage cost of the largest tolerated errors within target set (i.e., 1.5° for peak error and 3% for duration error).

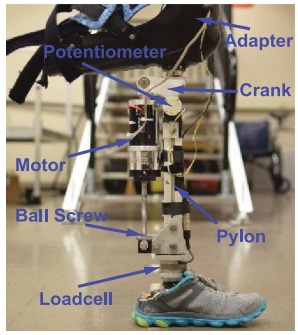


Fig. 3. Hardware setup for the prototype of robotic knee prosthesis.

V. Experiments and Results

We performed three tests involving four human subjects (two able-bodied and two amputees) to evaluate the performances of the proposed PICE.

A. Hardware Setup

A prototype of robotic knee prosthesis designed in our lab was used in this study [13]. The prosthesis utilizes a slider-crank mechanism, in which the slider is driven by the rotation of a dc motor (Maxon, Switzerland) through a ball screw (THK, Japan), and the crank rotation mimics the knee motion. The whole mechanism is integrated with a pylon as shown in Fig. 3. A maximum of 80 N · m torque output at the joint is ensured with such a design. The rotational motion of the prosthetic knee joint is recorded by a potentiometer (ALPS, Japan). A load cell (Bertec, USA) is attached to the pylon to measure the GRF. All the analog readings are converted to digital signals through a DAQ board (NI, USA) and then fed back to the control system, which is implemented by LabVIEW and MATLAB on a desktop PC.

B. Participants

We recruited four male subjects, two able-bodied individuals (AB1 and AB2), and two transfemoral amputees (TF1 and TF2), in this study. The participants' information is summarized in Table I. An L-shaped adapter (see Fig. 3) and a daily socket were used by AB and TF subjects, respectively, to allow them to walk with the knee prosthesis. The alignment of prosthesis for each subject was done by a certified prosthetist. All the subjects received training with the powered prosthesis until they can walk comfortably and confidently without holding the handrail. All

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
PARTICIPANT INFORMATION

Subject	Age	Height	Body Weight*	Prosthetic Side
AB1	43	1.78 m	74 kg	Right
AB2	32	1.78 m	65 kg	Left
TF1	63	1.64 m	64 kg	Left
TF2	59	1.83 m	94 kg	Left

*The body weight includes the robotic knee prosthesis.

the subjects were provided written informed consent before any procedures, and this study was approved by the Institutional Review Board of University of North Carolina at Chapel Hill.

C. Experimental Protocols

We carried out three experimental tests to validate and analyze the performance of the proposed PICE. They are, respectively, associated with the following the following three goals:

1) to experimentally assess convergence properties during offline training and the effect of training data size;
2) to quantitatively assess potential gains by using an offline pretrained policy as the initial policy for online training, and compare its performance to randomly initialized online training;
3) to investigate the robustness of a set of well-trained policies as tasks and users change.

1) Test of Offline Training: We used five sets of offline data all collected from AB1 to perform the offline policy training and obtained five sets of policies accordingly. The numbers of data samples in the five sets were 15, 45, 75, 105, and 135, respectively, each sample was a four-tuple (x_n, u_n, g_n, x'_n). We then evaluated each policy in five independent trials using AB1 as the test subject. AB1 walked on a treadmill at a speed of 0.6 m/s, while the offline trained policy adjusted the prosthesis impedance. The same set of initial impedance was applied to all the five trials. To eliminate the confounding effect of fatigue resulting from prolonged walking, for each tuning trial, AB1 performed several 3-min walking segments followed by a rest period. Additionally, a maximum of 135 impedance updates were allowed in consideration of the subject's limited enduring with walking. If training did not complete within this limit, the trial was considered a failure.

Two outcome measures were captured in each trial. The first measure was the L^2 distance (i.e., $\|r^{(i)} - r^{(i-1)}\|_2$) between the two consecutive weight parameter vectors in (18). It is used to quantify changes in the series of value outcomes in (10). The second measure was to explore the relationship between the number of offline training samples and the number of phases in which the success as defined in Section IV-A was reached without any online policy updates beyond offline training.

2) Test of Online Training: We conducted online training under two different initial policy conditions: 1) randomly initialized; 2) offline pretrained. Two subjects, AB1 and TF1, were asked to perform the treadmill walking task at the speed of 0.6 m/s. We used their own available offline data to obtain the pretrained policies. For both subjects, the offline training data had 105 samples. The same pretrained policies were used to serve as initial policies across trials for each subject, while randomly initialized policies varied. A few blocks of experimental sessions were conducted, each including two online training trials for comparison purpose. Specifically, in each block, we randomly selected the initial impedance parameters with the only requirement of being feasible for walking. Then, two online training trials with different initial policy conditions were performed. For AB1, three blocks of experimental sessions (each of which used different initial impedance parameter) were conducted. For TF1, one block was tested. The same walk-rest experimental protocol and the same maximum number of impedance update were applied as discussed in the first test.

The evaluation for the test of online training included efficiency, effectiveness, and impedance tuning convergence. Herein, the efficiency of online training was quantified by the following: 1) the number of phases needed for online policy updates beyond the initial policies until meeting the stopping criteria defined in Section IV-A; 2) the number of impedance updates to meet the stopping criterion for prosthesis tuning. To understand the effectiveness of tuning prosthesis control for producing desired knee motion, the knee kinematics were measured to reflect how the prosthetic knee joint moved when it interacted with the human users as the impedance varied with the guidance of policies. Finally, the impedance tuning convergence was analyzed by checking the evolution of peak errors and duration errors of knee kinematics (states) and prosthesis impedance values (control parameters) during the tuning.

3) Test of Policy Robustness: We investigated the robustness of well-trained policies and studied how well they behaved against changes of task and human subject. Two sets of well-trained policies were used, which were obtained from AB1 and AB2 in their respective treadmill walking tasks at the speed of 0.6 m/s prior to trials in this test. We applied them as initial policies for three new trials. Specifically, the trials consisted of the following:

1) up-ramp walking with slope of 4° performed by subject AB1 starting with his own policy;
2) self-paced level-ground walking performed by subject AB2 starting with his own policy;
3) treadmill walking (0.6 m/s) performed by subject TF2 but starting with AB2's policy.

To investigate how policies acted when they were applied to different tasks or subjects, we monitored sequences of both impedance and policy updates in each trial and the associated evolution of stage cost.

D. Experimental Results

1) Offline Training Assessment: Since similar results were obtained from all trials in the test of offline training, we only demonstrate the representative results (trained by data with the size of 105 samples collected from AB1) in Fig. 4(a). As shown in the data that changes in the weight parameter vectors of the approximating value function in (18) reduced to within the tolerance (10^{-4}) in six offline policy updates, which is equivalent

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

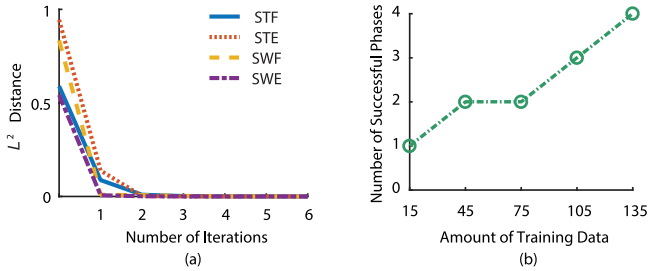


Fig. 4. Result of offline training evaluation. (a) Evolution of the L^2 distance between weight parameter vectors in two value function consecutive updates (i.e., $\|r^{(i)} - r^{(i-1)}\|_2$) in a representative trial of offline training test. At each instance of offline policy update, the vector is updated correspondingly. (b) Number of phases being able to reach a success increases with the increase of amount of training data.

TABLE II
EFFICIENCY COMPARISONS OF ONLINE TRAINING BY USING PRETRAINED VERSUS RANDOM INITIAL POLICIES

Experimental Block Number*	Phase Requiring Policy Updates†		Overall Number of Impedance Updates	
	Pre-trained	Random	Pre-trained	Random
B1	2	1, 2, 3, 4	39	126
B2	2	1, 2, 3, 4	43	94
B3	2	1, 2, 3, 4	42	111
B4	4	1, 2, 3, 4	28	53

* The experimental block B1 through B3 were tested with subject AB1, while the block B4 was associated with the subject TF1.† The numbers 1 through 4 under the column represent STF, STE, SWF, and SWE, respectively, in the FSM-IC.

to 10 s for performing the computation. The results suggest that the approximate value functions, as well as the policies, were convergent given the offline training data.

For the effect of the size of training dataset, we see in Fig. 4(b) that the number of phases being able to reach a success increased as the amount of training data increased. Particularly, when we performed the offline training with 135 samples, the number of phases reached up to four and no more policy updates were needed to accomplish the tuning. The evidence implies that, with the offline implementation alone, the proposed PICE algorithm is able to obtain policies ready to deploy as sufficient offline data of good quality are available to use.

2) Online Training Assessment: We first looked into its improvement in tuning efficiency by employing the pretrained initial policies obtained from offline training. As shown in Table II, the comparison results reveal that online training starting with pretrained policies obtained from offline training, albeit not perfect, were significantly more efficient than those starting with random policies. On average, the former cases only resulted in 1 phase that required online policy update, whereas the number amounted to 4 in the latter cases. Meanwhile, pretrained cases were observed to have less overall number of impedance updates than random cases did to meet the stopping criterion by an average of 58, which was equivalent to about 7 min of walking time of a subject.

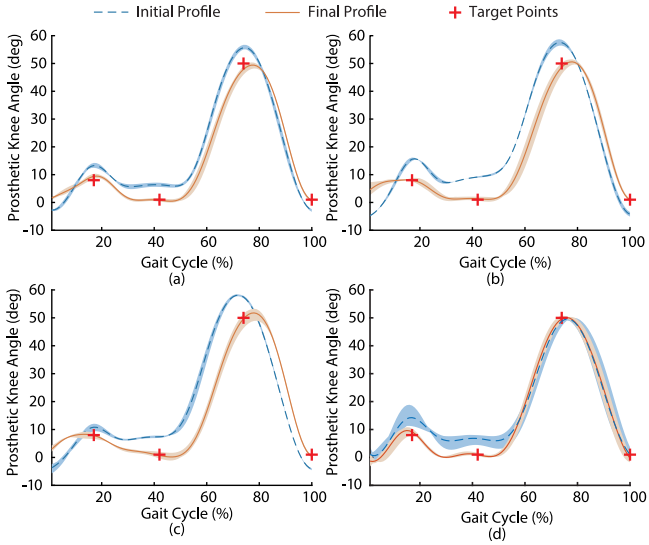


Fig. 5. Prosthetic knee kinematics with initial and final tuned impedance parameters in the test of online training. (a)–(d) Trials with pretrained initial policies in experimental block B1, B2, B3, and B4, respectively. Time series of kinematics are divided and normalized to multiple profiles in individual gait cycles based on the timing of heel strike. Shaded areas along profiles indicate the real motion ranges across four gait cycles performed by subjects walking with the same impedance parameters. The associated lines (dashed and solid) denote the averaged kinematics.

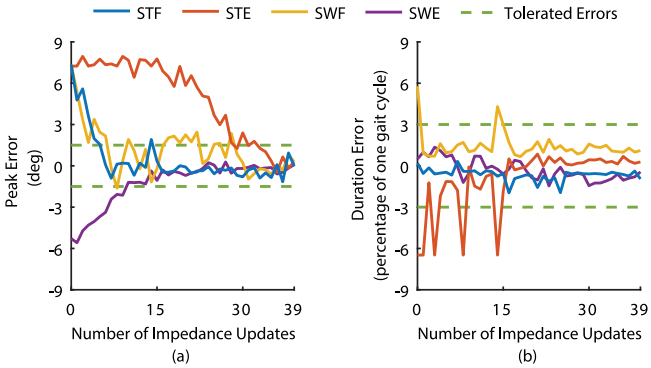


Fig. 6. Evolution of states as impedance parameters was updated. (a) Peak errors. (b) Duration errors.

Apart from the efficiency, for the trials staring with pretrained initial policies, we studied the tuning effectiveness. Fig. 5 displays the overall effect of tuning by comparing the knee profiles generated by initial impedance parameters before tuning with those produced by adjusted parameters at the end of tuning trials. We noted that, though differed in shape, initial knee profiles in all trials deviated from the targets, especially peak angle features. However, going through the tuning process under the guidance of final policies, the final parameters enabled knee profiles to approach the targets.

To inspect the impedance tuning convergence, we present representative results here (the experimental block B1 with pretrained initial policies in the Table II) as similar results across trials were observed. As revealed in Fig. 6, no matter how large the initial errors were, they all progressively converged into

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

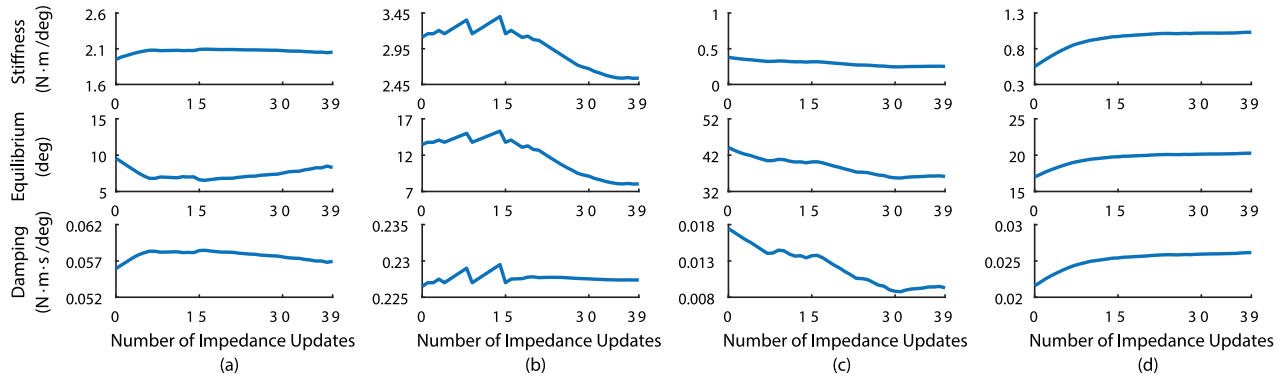


Fig. 7. Evolution of the impedance parameters (i.e., stiffness, equilibrium angle, and damping) in different phases. (a) STF, (b) STE, (c) SWF, (d) SWE.

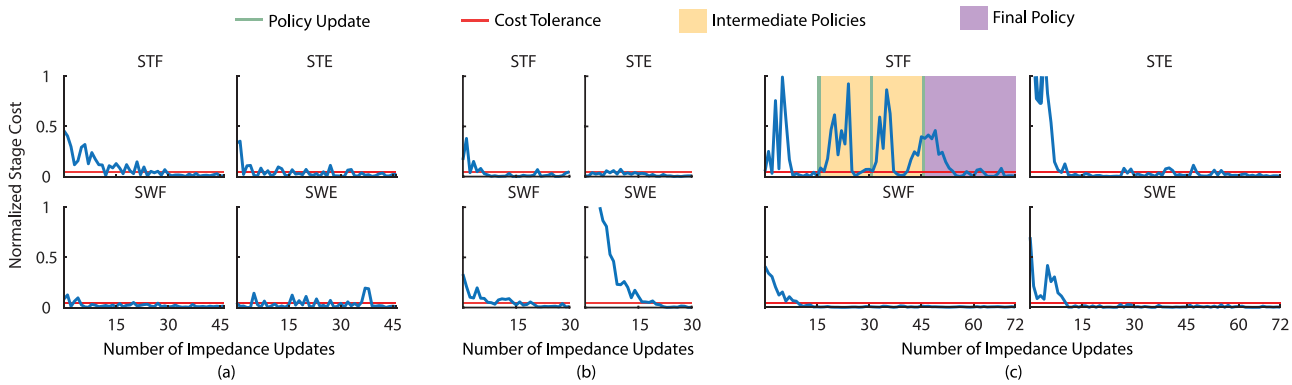


Fig. 8. Normalized stage cost over the number of impedance updates in testing policy robustness. The vertical lines indicate the instances where policy updates took place. The areas highlighted in yellow denote the periods of time when the online learning was activated and intermediate policies were employed, the areas highlighted in purple describe the phases where online learning was deactivated and final policies were deployed. Remaining areas without highlights indicate that initial policies trained from the treadmill walking of one subject were sufficient for successful impedance tuning when applied directly to other walking tasks or subjects. (a) Four phases in a trial of AB2 level-ground walking. (b) Four phases in a trial of AB1 up-ramp walking with slope of 4°. (c) Four phases in a trial of TF2 treadmill walking.

the tolerance range of errors ($\pm 1.5°$ for peak error, $\pm 3\%$ for duration error) and eventually remained within the range. Correspondingly, in Fig. 7, we observed that impedance parameters converged to constant values at the end of the trial (i.e., last ten updates) for most phases, except for the STF where the momentum of impedance adjustment lingered. The difference may be attributed to varying perturbations introduced by more dynamical interactions occurring in the STF among the human, the robotic prosthesis and the ground. As a result, the final policy for STF needed to respond by adjusting the impedance to accommodate such disturbances and stabilize errors within tolerances.

3) Robustness Investigation: As seen in Fig. 8(a), subject AB2 used a pretrained policy obtained from his own treadmill walking to perform level-ground walking with no difficulty as no further policy update was needed, and after 46 impedance updates (about 6 min of subject walking time), the subject knee kinematics met stopping criterion. Similar results were observed from Fig. 8(b) in the trial of AB1 up-ramp walking with even fewer impedance updates (i.e., 30 impedance updates). As for the trial of TF2 treadmill walking using the AB2's pretrained policy, despite not being completely successful in deploying policies to all four phases, only three updates of policy occurred in the STF phase, as shown in Fig. 8(c). Although 72 impedance updates (about 9 min) were needed to meet the stopping criterion, it only took 45 updates of impedance (about 6 min) to obtain the final policy refined for the STF phase of the new subject.

Note that a cyclic pattern of change in the cost was displayed in Fig. 8(c). This was caused by following initial or intermediate policies, which led to cost value sloping upward until the safety bound was hit, thereby triggering the reset of impedance parameters and getting the cost drop back to the initial value. The results suggest that policies we obtained from AB2 treadmill walking were, to some extent, robust against the changes of tasks and subjects.

VI. Discussions

In this study, we proposed a promising solution of PICE for tuning high-dimensional robotic knee prosthesis control parameters in order to provide efficient personalized assistance in walking. The tuning efficiency stemmed partly from our

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

innovation that enables offline policy training, beside online training, via policy iteration. To the best of authors' knowledge, few studies have successfully addressed wearable robot personalization in such an offline–online manner.

Our proposed RL method, as suggested in Fig. 4, demonstrated the feasibility of obtaining policies from a sufficient amount of existing offline data by the offline training, which can be deployed directly without interacting with the real human-prosthesis system. Clearly the offline implementation of PICE enables a maximal utility of existing parameter-performance data and is a new way to improve training efficiency in obtaining prosthesis tuning policies. Nevertheless, pure offline implementation has no guarantees to obtain accurate and robust policies despite being convergent in the sense of offline training, especially when the training data quality is poor. Note that the quality of data has two meanings, which include the amount of data and the extent of mismatch in data distribution [50]. In this study, however, we only investigated the influence of the amount of data on the offline training. Therefore, the number of training data we examined in this study might not be applicable to other datasets due to the confounding effect caused by data distributions, and it is actually difficult to determine the exact number in practice. Hence, an RL algorithm that is capable of performing offline–online training, such as our proposed PICE, became especially intriguing in order to ensure efficiency and effectiveness of autotuning algorithm for learning the prosthesis tuning policy. In this article, we demonstrated that when offline learned policies cannot handle realistic human-prosthesis interaction or were not robust enough to handle the variation across human users, as shown in Fig. 8(c), PICE can trigger online training that further update the policy to achieve the desired tuning goal.

In addition, the investigation of robustness associated with policies learned by the proposed algorithm showed other indirect benefits to potentially scale up the training outcome. As shown in Fig. 8, most deployed polices possessed exceptional robustness, in spite of the fact that policy refinements happened in the STF through online training to further accommodate for the changes in users. A potential reason to explain the phenomenon could be the fact that the underlying physical principles in prosthesis control have no drastic changes across different subjects and walking tasks as well as the associated variations in gait patterns and GRFs. The promising discovery may enable us to collect data and obtain pretrained initial policies, albeit not optimal, from more available users and relatively easier tasks. From there, further user-specific or task-specific refinements, if needed, could be accomplished by online training. As opposed to learning from scratch, such an approach is more likely to result in higher training efficiency, and thus, it is of great clinical value when applied at scales.

As a generic and efficient learning framework, the proposed PICE could also potentially shed light on similar problems for other assistive wearable machines, such as exoskeletons and neuroprosthetics. These devices are also in need of identifying the optimal control parameters for individual users with motor deficits [51], [52]. By unleashing the potentials demonstrated in this study, translations of the proposed approach into other human–machine systems are expected to be valuable because they all call for high training efficiency and being model-free due to patient-in-the-loop. However, specific modifications regarding problem formulations or implementations need to be properly considered before the translations, such as how to define states, actions, and costs for each application.

The successful implementation of the proposed PICE in the human-prosthesis system would encourage future studies to explore more application-specific solutions to an efficient approximation of the value function in RL. We demonstrated, in this study that leveraging simple basis functions (e.g., quadratic basis functions) fueled by insights on the control problem (e.g., the PSD constraint for the value function presented in this article) is likely to yield a satisfying approximation for the value function with limited amount of data. This is because, with fewer number of unknown parameters to estimate, such a choice alleviates the high demands for persistent excitation [53] or data richness [54] required by generic basis functions, which are often difficult to meet in practice; meanwhile the prestructured treatment is able to compensate for a poor approximation due to the lack of data.

Our proposed design and study, although promising, also had several limitations. The primary limitations of this study were the limited evaluation of the algorithm on human subjects and walking terrains in daily environments, because the focuses of this study were on developing a new automatic prosthesis tuning solution and demonstrating its promising advantages in clinic settings. Systematic evaluation of proposed tuning algorithm on more human subjects and terrain types (sand, grass, foam, etc.) is needed in order to show the clinical value in the future. In addition, we need more designed experiments to further validate the robustness of policies trained by the proposed approach against realistic conditions of uncertainties or disturbances. Another limitation arose from the feature extraction of the continuous knee profile. We selected four discrete points, as a means of dimension reduction, to characterize knee kinematics in each phase of a single gait cycle. Such a selection dropped the information of kinematics between these points, and thus, we had little control over the entire profile except for the feature points. To really reproduce target knee kinematics, we need to explore more advanced feature extraction methods that can better characterize a continuous profile. Lastly, in this study, the impedance tuning goal was to drive the prosthetic knee kinematics to approach the predefined knee profile extracted from representative motion in able-bodied population. This is limiting as well as such a goal might not align with user's preference, thus, may not be optimal with respect to other measures of the gait performance (e.g., gait symmetry, energy expenditure, balance, etc.). How to set up target profiles will be investigated in our future study.

VII. Conclusion

In this article, we proposed an improved solution of an RL-based algorithm, PICE, to learn impedance tuning policies for a robotic knee prosthesis efficiently. The tuning objective was to reproduce near-normal knee kinematics during walking tasks. The PICE algorithm benefited from an ability of offline–online

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

training and from making a compromise in using a simplified value function approximation structure. The problem of falsely yielded negative performance values due to approximation errors was avoided by imposing a PSD constraint to keep the approximated value function qualitatively correct. Therefore, it has great advantage on improving efficiency of the policy training.

We directly tested the proposed idea on human subjects. Our results showed that PICE successfully provided impedance tuning policy to the prosthetic knee with a human in the loop, and it significantly reduced policy training time especially for online training after initializing with an offline pretrained policy. In addition, the deployed policy is robust across human subjects and modifications in tasks. These promising results suggest great potential for future clinical application of our proposed methods on automatically personalizing assistive wearable robots.

APPENDIX
ERROR BOUND ANALYSIS

The focus of the analysis is to obtain a qualitative error bound for our proposed PICE algorithm. The analysis is performed along the line of [35] and [55]. We assume the following sufficient condition that the contraction property holds for any $r \subset \mathbb{R}^m$ under the general projected Bellman operation without constraints

$$\|\text{proj}_S(T\mathbf{\Phi}r)\|_\Xi \leq \|\mathbf{\Phi}r\|_\Xi.$$

The rationale of the assumption is that the property is theoretically guaranteed by adopting an on-policy scheme, while it can also be practically fulfilled by means of optimized sampling when utilizing an off-policy scheme as demonstrated in [55].

With the abovementioned assumption and nonexpansive property of projections shown as follows:

$$\|\text{proj}_{S_+}(\cdot)\|_\Xi \leq \|\text{proj}_S(\cdot)\|_\Xi \leq \|\cdot\|_\Xi$$

we have the following bounded error between the approximate value function $\hat{Q}^\pi$ and the ground truth Q^π for the target policy π in the context of PICE

$$\begin{aligned}
\|\hat{Q}^\pi - Q^\pi\|_\Xi &\leq \|\hat{Q}^\pi - \text{proj}_{S_+}(Q^\pi)\|_\Xi + \|\text{proj}_{S_+}(Q^\pi) - Q^\pi\|_\Xi \\
&= \|\text{proj}_{S_+}B(\hat{Q}^\pi) - \text{proj}_{S_+}B(Q^\pi)\|_\Xi \\
&\quad + \|\text{proj}_{S_+}(Q^\pi) - Q^\pi\|_\Xi \\
&\leq \alpha\|\text{proj}_S(T\hat{Q}^\pi) - \text{proj}_S(TQ^\pi)\|_\Xi \\
&\quad + \|\text{proj}_{S_+}(Q^\pi) - Q^\pi\|_\Xi \\
&\leq \alpha\|\text{proj}_S(T\hat{Q}^\pi) - \text{proj}_S(T\text{proj}_S(Q^\pi))\|_\Xi \\
&\quad + \alpha\|\text{proj}_S(T\text{proj}_S(Q^\pi)) - \text{proj}_S(TQ^\pi)\|_\Xi \\
&\quad + \|\text{proj}_{S_+}(Q^\pi) - Q^\pi\|_\Xi \\
&\leq \alpha\|\hat{Q}^\pi - Q^\pi\|_\Xi + \alpha\kappa(\Theta)\|\text{proj}_S(Q^\pi) - Q^\pi\|_\Xi \\
&\quad + \|\text{proj}_{S_+}(Q^\pi) - Q^\pi\|_\Xi \\
&\leq \frac{1 + 2\alpha\kappa(\Theta)}{1-\alpha}\|\text{proj}_{S_+}(Q^\pi) - Q^\pi\|_\Xi
\end{aligned}$$

where $\kappa(\Theta)$ denotes the condition number of the matrix $\Theta \triangleq (\Xi\Lambda)^{-\frac{1}{2}}$. The Ξ and Λ are diagonal matrices consisting of steady-state probabilities of Markov chains under the behavior and target policy, respectively.

Let ξ be the largest upper bound of evaluation errors over all iterations of policy evaluation. According to [35] and [56], when the policy improvement is performed exactly without incurring errors (which is guaranteed in this problem setting due to a use of QP solution), the following bound is yielded:

$$\limsup_{i\to\infty} \| \hat{Q}^{(i)} - Q^* \| \leq \frac{2\alpha\xi}{(1-\alpha)^2}.$$

This result for off-policy learning may result in a less tight bound if the condition number is poor. However, it still provides performance guarantees for the solution quality of PICE with the error being bounded, as opposed to being arbitrarily large. The bound implies that the proposed algorithm will eventually either converge or oscillate within a suboptimal policy space where the resulted policy is at most a constant away from the true optimality.

REFERENCES

[1] F. Sup, A. Bohara, and M. Goldfarb, "Design and control of a powered transfemoral prosthesis," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 263–273, 2008.

[2] T. Lenzi, M. Cempini, L. Hargrove, and T. Kuiken, "Design, development, and testing of a lightweight hybrid robotic knee prosthesis," *Int. J. Robot. Res.*, vol. 37, no. 8, pp. 953–976, 2018.

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

[4] V. Azimi, T. Shu, H. Zhao, R. Gehlhar, D. Simon, and A. D. Ames, "Model-based adaptive control of transfemoral prostheses: Theory, simulation, and experiments," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 2, pp. 1174–1191, Feb. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/8643092

[5] H. M. Herr and A. M. Grabowski, "Bionic ankle-foot prosthesis normalizes walking gait for persons with leg amputation," *Proc. Biol. Sci.*, vol. 279, pp. 457–464, 2012.

[6] B. E. Lawson, H. A. Varol, and M. Goldfarb, "Standing stability enhancement with an intelligent powered transfemoral prosthesis," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 9, pp. 2617–2624, Sep. 2011.

[7] 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.*, vol. 34, no. 3, pp. 686–701, Jun. 2018.

[8] L. J. Hargrove *et al.*, "Robotic leg control with EMG decoding in an amputee with nerve transfers," *New England J. Med.*, vol. 369, no. 13, pp. 1237–1242, 2013.

[9] M. Liu, D. Wang, and H. Huang, "Development of an environment-aware locomotion mode recognition system for powered lower limb prostheses," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 4, pp. 434–443, Apr. 2016.

[10] F. Zhang and H. Huang, "Source selection for real-time user intent recognition toward volitional control of artificial legs," *IEEE J. Biomed. Health Informat.*, vol. 17, no. 5, pp. 907–914, Sep. 2013.

[11] S. K. Au, J. Weber, and H. Herr, "Powered ankle-foot prosthesis improves walking metabolic economy," *IEEE Trans. Robot.*, vol. 25, no. 1, pp. 51–66, Feb. 2009.

[12] A. H. Shultz, B. E. Lawson, and M. Goldfarb, "Running with a powered knee and ankle prosthesis," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 23, no. 3, pp. 403–412, May 2015.

[13] M. Liu, F. Zhang, P. Datseris, and H. Huang, "Improving finite state impedance control of active-transfemoral prosthesis using Dempster-Shafer based state transition rules," *J. Intell. Robot. Syst.*, vol. 76, no. 3, pp. 461–474, 2014.

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[14] K. Kronander and A. Billard, "Stability considerations for variable impedance control," *IEEE Trans. Robot.*, vol. 32, no. 5, pp. 1298–1305, Oct. 2016.

[15] E. Burdet, R. Osu, D. W. Franklin, T. E. Milner, and M. Kawato, "The central nervous system stabilizes unstable dynamics by learning optimal impedance," *Nature*, vol. 414, pp. 446–449, 2001.

[16] N. Hogan, "Adaptive control of mechanical impedance by coactivation of antagonist muscles," *IEEE Trans. Autom. Control*, vol. AC-29, no. 8, pp. 681–690, Aug. 1984.

[17] S. Fakoorian, V. Azimi, M. Moosavi, H. Richter, and D. Simon, "Ground reaction force estimation in prosthetic legs with nonlinear Kalman filtering methods," *J. Dyn. Syst. Meas. Control*, vol. 139, no. 11, 2017, Art. no. 111004.

[18] V. Azimi, T. T. Nguyen, M. Sharifi, S. A. Fakoorian, and D. Simon, "Robust ground reaction force estimation and control of lower-limb prostheses: Theory and simulation," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 50, no. 8, pp. 3024–3035, Aug. 2020.

[19] D. A. Winter, *Biomechanics and Motor Control of Human Gait: Normal, Elderly and Pathological*. Waterloo, ON, Canada: Univ. Waterloo Press, 1991.

[20] E. J. Rouse, L. M. Mooney, and H. M. Herr, "Clutchable series-elastic actuator: Implications for prosthetic knee design," *Int. J. Robot. Res.*, vol. 33, no. 13, pp. 1611–1625, 2014.

[21] A. M. Simon *et al.*, "Configuring a powered knee and ankle prosthesis for transfemoral amputees within five specific ambulation modes," *PLoS One*, vol. 9, pp. 1–10, 2014.

[22] N. Aghasadeghi, H. Zhao, L. J. Hargrove, A. D. Ames, E. J. Perreault, and T. Bretl, "Learning impedance controller parameters for lower-limb prostheses," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 4268–4274.

[23] E. J. Rouse, L. J. Hargrove, E. J. Perreault, and T. A. Kuiken, "Estimation of human ankle impedance during the stance phase of walking," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 870–878, Jul. 2014.

[24] J. Zhang *et al.*, "Human-in-the-loop optimization of exoskeleton assistance during walking," *Science*, vol. 356, no. 6344, pp. 1280–1284, 2017.

[25] Y. Ding, M. Kim, S. Kuindersma, and C. J. Walsh, "Human-in-the-loop optimization of hip assistance with a soft exosuit during walking," *Sci. Robot.*, vol. 3, no. 15, 2018, Art no. eaar5438.

[26] H. Huang, D. L. Crouch, M. Liu, G. S. Sawicki, and D. Wang, "A cyber expert system for auto-tuning powered prosthesis impedance control parameters," *Ann. Biomed. Eng.*, vol. 44, no. 5, pp. 1613–1624, 2016.

[27] 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 Netw. Learn. Syst.*, vol. 28, no. 9, pp. 2215–2220, Sep. 2017.

[28] 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.*, vol. 50, no. 6, pp. 2346–2356, Jun. 2020.

[29] E. C. Martinez-Villalpando and H. Herr, "Agonist-antagonist active knee prosthesis: A preliminary study in level-ground walking," *J. Rehabil. Res. Develop.*, vol. 46, pp. 361–374, 2009.

[30] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, *Handbook of Learning and Approximate Dynamic Programming*. Piscataway, NJ, USA: Wiley, 2004.

[31] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2018, pp. 4238–4245.

[32] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2017, pp. 3389–3396.

[33] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "DeepMimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, 2018, Art. no. 143.

[34] J. Hwangbo *et al.*, "Learning agile and dynamic motor skills for legged robots," *Sci. Robot.*, vol. 4, no. 26, 2019, Art no. eaau5872.

[35] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *J. Mach. Learn. Res.*, vol. 4, no. 6, pp. 1107–1149, 2003.

[36] L. Busoniu, R. Babuška, B. de Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Boca Raton, FL, USA: CRC Press, 2010.

[37] J. Si and Yu-Tsung Wang, "Online learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.

[38] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[39] D. P. Bertsekas, "Approximate policy iteration: A survey and some new methods," *J. Control Theory Appl.*, vol. 9, no. 3, pp. 310–335, 2011.

[40] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2006, pp. 1–8.

[41] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Jul.–Sep. 2009.

[42] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Trans. Syst. Man Cybern. B. Cybern.*, vol. 41, no. 1, pp. 14–25, Feb. 2011.

[43] X. Gao, J. Si, Y. Wen, M. Li, and H. Huang, "Knowledge-guided reinforcement learning control for robotic lower limb prosthesis," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 754–760.

[44] M. P. Kadaba, H. K. Ramakrishnan, and M. E. Wootten, "Measurement of lower extremity kinematics during level walking," *J. Orthop. Res.*, vol. 8, pp. 383–392, 1990.

[45] M. Li, X. Gao, Y. Wen, J. Si, and H. Huang, "Offline policy iteration based reinforcement learning controller for online robotic knee prosthesis parameter tuning," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2019, pp. 2831–2837.

[46] D. P. Bertsekas, "Temporal difference methods for general projected equations," *IEEE Trans. Autom. Control*, vol. 56, no. 9, pp. 2128–2139, Sep. 2011.

[47] H. Yu, "Least squares temporal difference methods: An analysis under general conditions," *SIAM J. Control Optim.*, vol. 50, no. 6, pp. 3310–3343, 2012.

[48] R. Escalante and M. Raydan, *Alternating Projection Methods*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2011.

[49] D. A. Winter, "Kinematic and kinetic patterns in human gait: Variability and compensating effects," *Human Movement Sci.*, vol. 3, no. 1, pp. 51–76, 1984.

[50] Y. Liu *et al.*, "Representation balancing MDPs for off-policy policy evaluation," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, Dec. 2018, pp. 2649–2658.

[51] A. U. Pehlivan, D. P. Losey, and M. K. O'Malley, "Minimal assist-as-needed controller for upper limb robotic rehabilitation," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 113–124, Feb. 2016.

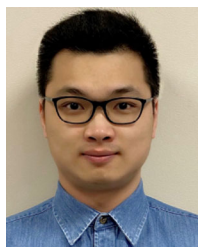
[52] A. J. Bergquist, J. Clair, O. Lagerquist, C. S. Mang, Y. Okuma, and D. F. Collins, "Neuromuscular electrical stimulation: Implications of the electrically evoked sensory volley," *Eur. J. Appl. Physiol.*, vol. 111, pp. 2409–2426, 2011.

[53] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, 2010.

[54] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.

[55] J. Z. Kolter, "The fixed points of off-policy TD," in *Proc. 24th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2011, pp. 2169–2177.

[56] D. P. Bertsekas, *Dynamic Programming and Optimal Control: Volume II*. Nashua, NH, USA: Athena Scientific, 2012.



Minhan Li received the B.S. and M.S. degrees in mechanical engineering from Tianjin University, Tianjin, China, in 2015 and 2018, respectively. He is currently working toward the Ph.D. degree in biomedical engineering with the Joint Department of Biomedical Engineering, North Carolina State University, Raleigh, NC, USA, and with the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.

His current research interests include wearable robots, intelligent control, machine learning, computer vision, and rehabilitation engineering.

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Yue Wen received the B.S. degree in Automation from Wuhan University of Technology, Wuhan, China, in 2011, the M.S. degree in Control Theory and Engineering from Huazhong Univerisity of Science and Technology, Wuhan, China, in 2014, and the Ph.D. degree in Biomedical Engineering in the NCSU/UNC Joint Department of Biomedical Engineering at North Carolina State University and the University of North Carolina at Chapel Hill, Raleigh, NC, USA, in 2019.

He is currently a Research Associate in the Shirley Ryan AbilityLab, Chicago, IL. His research interests include adaptive control of bionic limbs and assistive robotic devices, machine learning, and human motion analysis.

Xiang Gao received the B.S. degree in automation from the Huazhong University of Science and Technology, Wuhan, China, in 2011, and the M.S. and Ph.D. degrees in electrical engineering from Arizona State University, Tempe, AZ, USA, in 2014 and 2020, respectively.

He is currently a Research Fellow with the Intelligent Robot Research Center, Jihua Laboratory, Foshan, China. His current research interests include robot learning, machine learning, computer vision, and rehabilitation robotics.

Jennie Si (Fellow, IEEE) received the B.S. in 1985 and M.S. in 1988 from Tsinghua University, Beijing, China, and the Ph.D. degree in 1992 from the University of Notre Dame, Notre Dame, IN, USA.

She has been a Faculty Member with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA, since 1991. Her research focuses on reinforcement learning control which utilizes machine learning and neural network approximations. She is also interested in fundamental neuroscience of the frontal cortex and its role in decision and control processes.

Dr. Si was a recipient of the NSF/White House Presidential Faculty Fellow Award in 1995 and the Motorola Engineering Excellence Award in 1995. She is a Distinguished Lecturer of the IEEE Computational Intelligence Society. She consulted for Intel, Arizona Public Service, and Medtronic. She has served on several professional organizations' executive boards and international conference committees. She was an Advisor to the NSF Social Behavioral and Economical Directory. She served on several proposal review panels. She was an Associate Editor for the IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING, the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, and an Action Editor for *Neural Networks*. She is currently an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS.

He (Helen) Huang (Senior Member, IEEE) received the B.S. degree in electronic and information engineering from Xi'an Jiao-tong University, Xi'an, China, in 2000, and the M.S. and Ph.D. degrees in biomedical engineering from Arizona State University, Tempe, AZ, USA, in 2002 and 2006, respectively.

She is currently the Jackson Family Distinguished Professor with the joint Department of Biomedical Engineering, North Carolina State University, Raleigh, NC, USA, and the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, and the Director for the Closed-Loop Engineering for Advanced Rehabilitation (CLEAR) core. Her research interest lies in neural-machine interfaces for prostheses and exoskeletons, human–robot interaction, adaptive and optimal control of wearable robots, and human movement control.

Dr. Huang was the recipient of the Delsys Prize for Innovation in Electromyography, the Mary E. Switzer Fellowship with NIDRR, and a NSF CAREER Award. She is a Fellow of AIMBE and Member of the Society for Neuroscience, BMES, and ASB. She is currently an Associate Editor for the IEEE TRANSACTIONS ON NEURAL SYSTEMS AND REHABILITATION ENGINEERING, the *Journal of Neuroengineering and Rehabilitation*, and the Wearable Techologies, and is a Guest Associate Editor for the IEEE TRANSACTIONS ON ROBOTICS.

Authorized licensed use limited to: N.C. State University Libraries - Acquisitions & Discovery S. Downloaded on July 21,2021 at 17:01:59 UTC from IEEE Xplore. Restrictions apply.