Online Reinforcement Learning Control for the Personalization of a Robotic Knee Prosthesis

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

***Abstract*—Robotic prostheses deliver greater function than passive prostheses, but we face the challenge of tuning a large number of control parameters in order to personalize the device for individual amputee users. This problem is not easily solved by traditional control designs or the latest robotic technology. Reinforcement learning (RL) is naturally appealing. The recent, unprecedented success of AlphaZero demonstrated RL as a feasible, large-scale problem solver. However, the prosthesis-tuning problem is associated with several unaddressed issues such as that it does not have a known and stable model, the continuous states and controls of the problem may result in a curse of dimensionality, and the human-prosthesis system is constantly subject to measurement noise, environmental change and human-body-caused variations. In this paper, we demonstrated the feasibility of direct heuristic dynamic programming, an approximate dynamic programming (ADP) approach, to automatically tune the 12 robotic knee prosthesis parameters to meet individual human users' needs. We tested the ADP-tuner on two subjects (one able-bodied subject and one amputee subject) walking at a fixed speed on a treadmill. The ADP-tuner learned to reach target gait kinematics in an average of 300 gait cycles or 10 min of walking. We observed improved ADP tuning performance when we transferred a previously learned ADP controller to a new learning session with the same subject. To the best of our knowledge, our approach to personalize robotic prostheses is the first implementation of online ADP learning control to a clinical problem involving human subjects.**

***Index Terms*—Approximate dynamic programming (ADP), direct heuristic dynamic programming (dHDP), reinforcement learning (RL), robotic knee prosthesis.**

I. Introduction

ADVANCES in robotic prostheses, compared to conventional passive devices, have shown great promise to further improve the mobility of individuals with lower limb amputation [1]–[5]. Robotic prosthesis control typically consists of a finite-state machine and a low-level controller to regulate the prosthetic joint impedance. Existing robotic prosthesis controllers rely on a large number of configurable parameters (i.e., 12–15 for knee prostheses [3], [5], [6] and 9–15 for ankle–foot prostheses [3], [4]) for a single locomotion mode such as level ground walking. The number of parameters grows when the number of included locomotion modes increases. These control parameters need to be personalized to individual user differences, such as height, weight, and physical ability. Currently, in clinics, prosthesis control parameters are personalized manually [7], [8], which can be time, labor, and cost intensive.

Researchers have attempted to improve the efficiency of prosthesis tuning through three major approaches. The first approach is to estimate the control impedance parameters with either a musculoskeletal model [9] or measurements of biological joint impedance [10], [11]. However, these methods have not been validated for real prosthesis control. The second solution does not directly address parameter tuning but aims at reducing the number of control parameters [7], [12]. The third method provides automatic parameter tuning by coding prosthetists' decisions [13], which can be time consuming to perform and potentially biased by individual prosthetist's experience. We, therefore, need new approaches to solve this prosthesis parameter tuning problem.

Personalizing wearable robots, that is, robotic prostheses and exoskeletons, requires optimal adaptive control solutions. Koller *et al.* [14] used gradient descent method to optimize an onset time of an ankle exoskeleton to enhance able-bodied (AB) persons' gait efficiency. Zhang *et al.* [15] used evolution strategy to optimize four control parameters for an ankle exoskeleton. Ding *et al.* [16] applied Bayesian optimization to identify two control parameters of hip extension assistance. These methods are promising, but they have not been used for personalizing robotic prostheses potentially because it is difficult to scale up to a high-dimensional (≥ 5) parameter space, adapt to changing conditions (e.g., weight change), or monitor the chosen performance measure in daily life (e.g., metabolic cost).

Reinforcement learning (RL) lends itself as an alternative approach to personalizing lower limb prostheses. Although it has defeated two thousand years of human GO knowledge by learning to play the game in hours. RL has not yet been applied in clinical situations with greater complexity and human interactions. For example, the control of wearable robotics introduces the additional challenge of the curse

Manuscript received May 1, 2018; revised September 26, 2018; accepted December 19, 2018. Date of publication January 16, 2019; date of current version May 7, 2020. This work was supported in part by the National Science Foundation under Grant 1563454, Grant 1563921, Grant 1808752, and Grant 1808898. This paper was recommended by Associate Editor H. Zhang. *(Corresponding authors: He (Helen) Huang; Jennie Si.)*

Y. Wen, A. Brandt, and H. Huang are with the UNC/NCSU Joint Department of Biomedical Engineering, North Carolina State University, Raleigh, NC 27695 USA, and the University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 USA (e-mail: hhuang11@ncsu.edu).

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

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2019.2890974

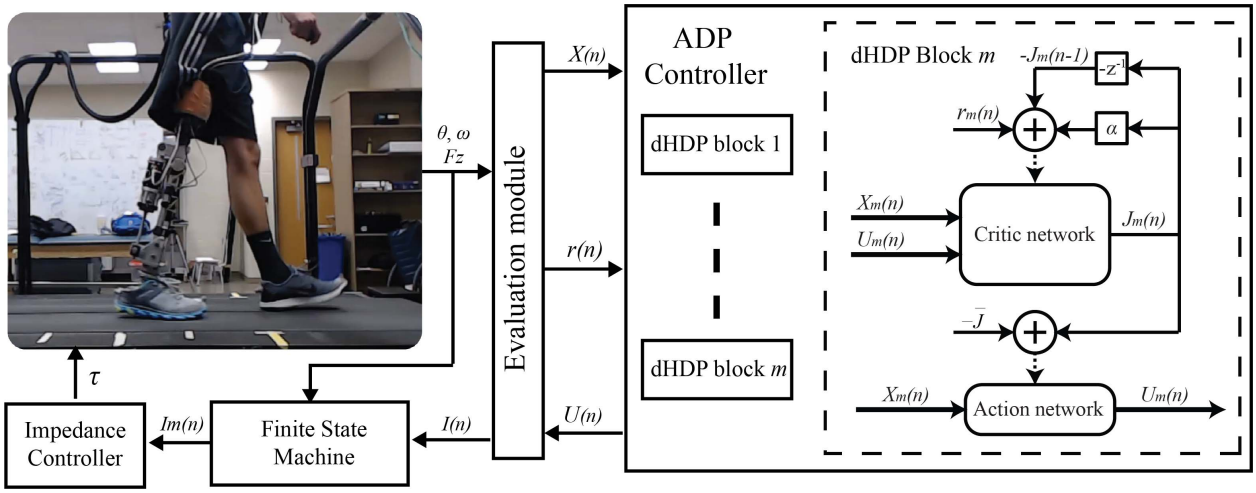


Fig. 1. Block diagram of ADP-tuner, an automatic robotic knee control parameter tuning scheme by dHDP with amputee in the loop. The learning control system operates at three different time scales: 1) real-time impedance controller provides outputs at 100 Hz to regulate the joint torque; 2) the finite-state machine runs at the gait frequency (denoted by time index g) with four phases per gait cycle; and 3) the dHDP generated control is updated $I_{m,n}$ every few gaits (denoted by time index n) to update the impedance parameters. The respective variables in the figure are defined and discussed in Sections II and III. The ADP-tuner consists of four dHDP blocks ($m = 1, 2, 3, 4$) corresponding to four gait phases in the finite-state machine impedance controller.

of high dimensionality in continuous state and control/action spaces, and the demand of meeting optimal performance objectives under system uncertainty, measurement noise, and unexpected disturbance. Approximate dynamic programming (ADP) [17]–[19] is synonymous to RL, especially, in controls and operations research communities, and it has shown great promise to address the aforementioned challenges.

Adaptive critic designs are a series of ADP designs that were originally proposed by Werbos [20]–[22]. In the last decade, the adaptive critic design has been developed and applied extensively to robust control [23], optimal control [24]–[26], and event-driven applications [27]–[29]. The action-dependent heuristic dynamic programming (ADHDP) is similar to Q-learning but with promising scalability [30]. New developments within the realm of ADHDP (e.g., neural fitted Q (NFQ), NFQ with continuous actions (NFQCA) [31], direct heuristic dynamic programming (direct HDP or dHDP) [32], the forward model for learning control [33], and fuzzy adaptive dynamic programming [25]) have emerged and demonstrated their feasibility for complex and realistic learning control problems. Furthermore, dHDP and NFQCA (noted as a batch variant of the dHDP [34]) algorithms are associated with perhaps most of the demonstrated applications of RL control for continuous state and control problems [34]–[43]. The focus of this paper is therefore to implement the dHDP [32] in real time for online learning control to adaptively tune the impedance parameters of the prosthetic knee.

Prior to real experimentation involving human subjects, we performed a simulation study [44]. We designed ADP-tuner for a prosthetic knee joint and validated this control on an established computational model, OpenSim [45], for dynamic simulations of amputee gait. We compared dHDP with NFQCA. Our simulation results showed that dHDP controller enabled the simulated amputee model to learn to walk within fewer gait cycles and with a higher success rate than NFQCA [44]. Although exciting and promising, it is unknown how dHDP performs with a real human in the loop. This is because the OpenSim model ignores human responses to actions of the prosthesis, natural gait variability, and most importantly, safety.

This paper reported herein include the following major contributions.

1) To our knowledge, this is the first study to realize an ADP learning controller for a real-life situation such as the personalization of robotic prostheses for human subjects. This application is novel in the rehabilitation field.
2) The model-free dHDP was tailored to be data and time efficient for this application and was implemented to automatically tune 12 impedance parameters through interactions with the human-prosthesis system online.
3) The study demonstrated, for the first time, that the proposed RL-based control is feasible and, with further development, can potentially be made safe and practical for clinical use.

The remaining of this paper is organized as follows. Section II describes the human-prosthesis system and formulates the human-prosthesis tuning/configuration problem. Section III presents an ADP-tuner design for online control of prosthetic knee. Section IV elaborates the design considerations for real human subjects. Section V explains the experimental evaluation of the ADP-tuner. The results are presented in Section VI. Remarks and discussions are presented in Section VII, followed by the conclusion in Section VIII.

II. Prosthetic Knee Control Problem Formulation

Fig. 1 shows our proposed automatic tuning approach of prosthetic knee control parameters with a human in the loop. In this section, we introduce the human-prosthesis system, namely an amputee wearing a robotic knee prosthesis.

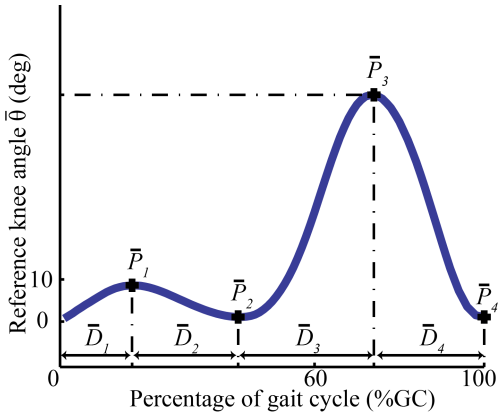


Fig. 2. Feature representation of near-normal knee kinematics during one gait cycle was used as learning control target, where $\bar{D}_m$ indicates the angle feature and $\bar{P}_m$ indicates the duration feature. The phase index is indicated by $m = 1, 2, 3, 4$. The start at 0% and the finish at 100% are the heel strike events, and 60% is approximate toe-off time.

A. Human-Prosthesis Configuration

Both the mechanical interface and control parameters of the robotic knee prosthesis need to be personalized to each user. Humans differ in their physical conditions, such as height, weight, and muscle strength. First, the length of the pylon, the alignment of the prosthesis, and the fit of the socket that interfaces the user and the prosthesis must be customized by a certified prosthetist. Then, the robotic knee control parameters must be tuned to provide personalized assistance from the knee prosthesis. Our proposed automatic tuning realized as an RL-based supplementary control is shown in Fig. 1.

B. Prosthetic Knee Finite-State Machine Impedance Controller

Finite-state machine impedance control (FSM-IC, Fig. 1) is an established framework for robotic knee prosthesis control [3], [5], [6], [46]. Based on the foot–ground contact and knee joint movement, a single gait cycle is divided into four phases (corresponding to $m = 1, \ldots, 4$ in Fig. 1): 1) the stance flexion phase (STF, $m = 1$); 2) stance extension phase (STE, $m = 2$); 3) swing flexion phase (SWF, $m = 3$); and 4) swing extension phase (SWE, $m = 4$). The phase transitions can be triggered by measurements from a load cell and an angle sensor in the prosthetic device. Then, the corresponding impedance parameters I_m as described in (1) are provided to impedance controller

$$I_m = [k_m, \theta e_m, b_m]^T. \tag{1}$$

Within each phase m, the robotic knee is regulated by a different impedance controller (2) to produce phase-specific dynamic properties. The impedance controller monitors the knee joint position θ and velocity ω and controls the knee joint torque τ in real time based on three impedance parameters: 1) stiffness k; 2) damping b; and 3) equilibrium position θ_e

$$\tau_m = k_m(\theta - \theta e_m) + b_m\omega. \tag{2}$$

Thus, with four gait phases, there are 12 total impedance parameters to be configured for each locomotion mode.

C. Representation of Knee Kinematics

Robotic knee kinematics are measured by an angle sensor mounted on the rotational joint. The angle sensor reads zero when the knee joint is extended to where the shank is in line with the thigh, and a positive value in degrees when the knee joint is flexed. Typically, the knee joint angle trajectory in one gait cycle has a local maximum during stance flexion and swing flexion, and a local minimum during stance extension and swing extension (Fig. 2). The peak value of each phase is primarily determined by the impedance parameters in that phase. Therefore, we represented the knee kinematics in one gait cycle with four pairs of peak angle values P and their respective duration values D : $[P_m, D_m]$, where $m = 1, 2, 3, 4$. Similarly, we extracted the same features from normative knee kinematics [47] as target features, denoted as $[\bar{P}_m, \bar{D}_m]$ (Fig. 2).

D. Human-Prosthesis System Tuning Process

The tuning process is built upon the commonly used FSM-IC framework, and the goal is to find a set of impedance parameters that allow the human-prosthesis system to generate normative target knee kinematics. As mentioned earlier, three impedance parameters took effect in each gait phase, and correspondingly, the knee kinematic features were extracted during each gait phase. For the ease of discussion, we will drop the subscript m for gait phase from hereon.

For the human-prosthesis system, the control inputs are the impedance parameters $I(n)$, and the outputs are the features $x(n)$ of prosthetic knee kinematics

$$\begin{aligned} I(n) &= [k(n), \theta e(n), b(n)]^T \\ x(n) &= [P(n), D(n)]^T \end{aligned} \tag{3}$$

where n denotes the update index of each parameter update, which is every seven gait cycles.

In the tuning procedure, the impedance parameters are updated as

$$I(n) = I(n-1) + \beta \odot U(n-1) \tag{4}$$

where U denotes the actions from the ADP-tuner, $\beta \in \mathbb{R}^{3\times 1}$ are the scaling factors to assign physical magnitudes to the actions, and $\odot$ is the Hadamard product of two vectors.

The states of the human-prosthesis system used in the learning controller are defined as

$$X(n) = \gamma \odot \left[x^T(n) - \bar{x}^T, x^T(n) - x^T(n-1)\right]^T \tag{5}$$

where $\gamma \in \mathbb{R}^{4\times 1}$ is a vector of scaling factors to normalize the states to $[-1, 1]$ and $\bar{x}$ are the features $[\bar{P}, \bar{D}]^T$ of the target knee kinematics. The feature errors $x(n) - \bar{x}$ capture the distance to the target knee kinematics, and the feature change rate $x(n) - x(n-1)$ obtain the dynamic change during the tuning procedure.

In the tuning process, the actions from the ADP-tuner are adjusted to the impedance parameters, which are continuous, and the states to the ADP-tuner are derived from the features of knee kinematics, which are also continuous. Therefore, the human-prosthesis tuning process has continuous states and controls. Equations (3)–(5) are implemented in

the "evaluation module" (Fig. 1) as an interface between the human-prosthesis system and the ADP-tuner. In addition, the evaluation module includes reinforcement signals provided to the ADP-tuner based on the outputs of the human-prosthesis system (described in Section IV-A).

III. ADP-Tuner

For the given human-prosthesis impedance parameter tuning problem, we implemented the ADP-tuner with four parallel dHDP blocks corresponding to four gait phases: 1) STF; 2) STE; 3) SWF; and 4) SWE (Fig. 1). Each dHDP block took in four state variables in (5) and tuned three impedance parameters for the respective phase. All dHDP blocks were identical, including one action neural network (ANN) and one critic neural network (CNN). Thus, without loss of generality, we present the detailed dHDP implementation without phase numbers.

A. Utility Function/Reinforcement Signal

The reinforcement signal $r(n) \in \mathbb{R}$ is defined as the instantaneous cost that is determined from the human-prosthesis system

$$r(n) = \begin{cases} -1, & \text{if } x(n) \notin [B_l, B_u] \\ -0.8, & \text{if } S^- > 4 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $[B_l, B_u]$ denotes the safety bounds as defined in Section IV-A, S^- is a penalty score, and the -0.8 reinforcement signal is imposed to the ADP block when the S^- value is greater than 4, indicating the dHDP block continues to tune the impedance parameter in an unfavorable direction (i.e., increasing the angle error and/or duration error) [44]. When the reinforcement signal is -1, the impedance parameters of the human-prosthesis system are reset to default values.

The total cost-to-go at ADP tuning time step n is given by

$$J(n) = r(n+1) + \alpha r(n+2) + \cdots + \alpha^N r(n+N+1) + \cdots \tag{7}$$

where α is a discount rate ($0 < \alpha < 1$) and N is infinite. It can be rewritten as

$$J(n) = r(n+1) + \alpha J(n+1). \tag{8}$$

B. Critic Neural Network

The CNN consisted of three layers of neurons (7-7-1) with two layers of weights, and it took the state $X \in \mathbb{R}^{4\times 1}$ and the action $U \in \mathbb{R}^{3\times 1}$ as inputs and predicted the total cost-to-go $\hat{J}$

$$\hat{J}(n) = W_{c2}(n)\varphi\Big(W_{c1}(n)\big[X^T(n), U^T(n)\big]^T\Big) \tag{9}$$

where $W_{c1} \in \mathbb{R}^{7\times 7}$ was the weight matrix between the input layer and the hidden layer, and $W_{c2} \in \mathbb{R}^{1\times 7}$ was the weight matrix between the hidden layer and the output layer. And

$$\varphi(v) = \frac{1-\exp(-v)}{1+\exp(-v)} \tag{10}$$

$$v_{c1}(n) = W_{c1}(n)\big[X^T(n), U^T(n)\big]^T \tag{11}$$

$$h_{c1}(n) = \varphi(v_{c1}(n)) \tag{12}$$

where $\varphi(\cdot)$ was the tan-sigmoid activation function and h_{c1} was the hidden layer output.

The prediction error $e_c \in \mathbb{R}$ of the CNN can be written as

$$e_c(n) = \alpha\hat{J}(n) - \Big[\hat{J}(n-1) - r(n)\Big]. \tag{13}$$

To correct the prediction error, the weight update objective was to minimize the squared prediction error E_c, denoted as

$$E_c(n) = \frac{1}{2}(e_c(n))^2. \tag{14}$$

The weight update rule for the CNN was a gradient-based adaptation given by

$$W(n+1) = W(n) + \Delta W(n). \tag{15}$$

The weight updates of the hidden layer matrix W_{c2} were

$$\begin{aligned}\Delta W_{c2}(n) &= l_c(n)\left[-\frac{\partial E_c(n)}{\partial W_{c2}(n)}\right] \\ &= l_c(n)\left[-\frac{\partial E_c(n)}{\partial e_c(n)}\frac{\partial e_c(n)}{\partial \hat{J}(n)}\frac{\partial \hat{J}(n)}{\partial W_{c2}(n)}\right].\end{aligned} \tag{16}$$

The weight updates of the input layer matrix W_{c1} were

$$\begin{aligned}\Delta W_{c1}(n) &= l_c(n)\left[-\frac{\partial E_c(n)}{\partial W_{c1}(n)}\right] \\ &= l_c(n)\left[-\frac{\partial E_c(n)}{\partial e_c(n)}\frac{\partial e_c(n)}{\partial \hat{J}(n)}\frac{\partial \hat{J}(n)}{\partial h_{c1}(n)}\frac{\partial h_{c1}(n)}{\partial v_{c1}(n)}\frac{\partial v_{c1}(n)}{\partial W_{c1}(n)}\right]\end{aligned} \tag{17}$$

where $l_c > 0$ was the learning rate of the CNN.

C. Action Neural Network

The ANN consisted of three layers of neurons (4-7-3) with two layers of weights, and it took in the state $X \in \mathbb{R}^{4\times 1}$ from the human-prosthesis system and output the actions $U \in \mathbb{R}^{3\times 1}$ to adjust the impedance parameters of the human-prosthesis system

$$U(n) = \varphi(W_{a2}(n)\varphi(W_{a1}(n)X(n))) \tag{18}$$

where $W_{a1} \in \mathbb{R}^{7\times 4}$ and $W_{a2} \in \mathbb{R}^{3\times 7}$ were the weight matrices and $\varphi(\cdot)$ was the tan-sigmoid activation function of the hidden layer and the output layer.

Under our problem formulation, the objective of adapting the ANN was to backpropagate the error between the desired ultimate objective, denoted by $\bar{J}$, and the approximated total cost-to-go $\hat{J}$. And $\bar{J}$ was set to 0 indicating "success." Thus, policy update goal was to minimize the absolute total cost-to-go value to 0. The weight update rule for the ANN was to minimize the following performance error:

$$E_a(n) = \frac{1}{2}\Big(\hat{J}(n) - \bar{J}\Big)^2. \tag{19}$$

Similarly, the weight matrix was updated based on gradient-descent

$$W(n+1) = W(n) + \Delta W(n). \tag{20}$$

The weight updates of the hidden layer matrix W_{a2} were

$$\Delta W_{a2}(n) = l_a(n)\left[-\frac{\partial E_a(n)}{\partial W_{a2}(n)}\right]. \tag{21}$$

Algorithm 1 Online ADP-Tuning of Impedance Parameters for Robotic Knee Prosthesis

Initialization of human-prosthesis system: $I(0), x(0)$, and Random initialization of weights of ANN and CNN.
Step 1:Value update
Get state $X(n)$ from (5) and reinforcement signal $r(n)$ from (6)
Update weights of CNN using (13)-(17)
Step 2:Policy improvement
Update weights of ANN using (19)-(22).
Calculate $U(n)$ from (18) and update $I(n)$ using (4).
Reset $I(n)$ if $r(n) == -1$ from (6).
Go to Step 1 until termination criteria (Section IV-E)

The weight updates of the input layer matrix W_{a1} were

$$\Delta W_{a1}(n) = l_a(n)\left[-\frac{\partial E_a(n)}{\partial W_{a1}(n)}\right] \tag{22}$$

where $l_a > 0$ is the learning rate of the ANN.

The above ANN and CNN weight updates and the ADP-tuner implementation is summarized in Algorithm 1. The weights of both ANN and CNN were initialized with uniformly distributed random numbers between -1 and 1. With mild and intuitive conditions, the dHDP with discounted cost has the property of uniformly ultimately boundedness [48].

IV. Design Considerations of Online Learning for Human Subjects

Human studies are different from simulation studies and, therefore, we modified and implemented the ADP-tuning algorithm to accommodate real-life considerations for human subjects wearing a prosthetic leg.

A. Safety Bounds

For weight-bearing prostheses, safety is the primary concern, so we included constraints to ensure the human-prosthesis system outputs remain within a safe range [denoted by $[B_l, B_u]$ in (6)]. First, to avoid potential harm to an amputee user, we set bounds on the feature errors of 1.5 times the standard deviation of the average knee kinematic features of people walking without a prosthesis (i.e., STF 10.5 degrees, STE 7.5 degrees, SWF 9 degrees, and SWE 6 degrees [47]). Second, to avoid collision of mechanical parts in a prosthesis that may damage the robotic prosthesis, we set bounds on the range of motion to -5 degrees and 60 degrees. These constraints defined the exploration range for the ADP controller to avoid damage or harm to the human-prosthesis system. When the features exceeded these ranges, we reset the control parameters to the default values determined at the beginning of each experimental session, which were known to result in safe operation. At the same time, a -1 reinforcement signal was sent to the ADP-tuner.

B. Robust Feature Extraction

Sensor signal noise is inevitable from real prostheses, so we implemented a robust feature extraction method to extract features of the knee joint angle. In reality, the knee joint angle trajectory is not ideal mainly because of two reasons: 1) inevitable noise in the angle sensor readings and 2) nearly flat angle trajectory at some places of a gait cycle where sensor readings remained steady. Under those conditions, the timing feature D varied greatly when obtaining the peak and duration values from a gait cycle. To address this, we first located the minimum or maximum features $[\tilde{P}_i, \tilde{D}_i]$ from the knee joint angle trajectory, where i denotes the sensor sample index (100 Hz). For each sample θ_j in the knee joint angle trajectory, there are two features $[P_j, D_j]$. We selected and used the features at index j to replace $[\tilde{P}_i, \tilde{D}_i]$, where

$$j = \arg\min\left(D_j - \bar{D}\right) \tag{23}$$

and index j is within $[i-10, i+10]$, and corresponding angle feature P_j is within $[\tilde{P}_i - 0.3, \tilde{D}_i + 0.3]$. This is to find robust and representative duration features based on real-time sensor measures.

C. Human Variability

To attenuate inevitable variations of human gait from step to step, the ADP-tuner processed the human-prosthesis system features the every gait cycle, but control updates were made for every seven gait cycles. This is to say, the human subjects walked with an updated set of impedance parameters for seven gait cycles. If the angle features of a particular gait cycle was greater than 1.5 standard deviations from the mean of the seven gait cycles, it was considered an outlier and removed. This eliminated accidental tripping or unbalanced gait cycles from influencing the control updates.

D. Prevention of Faulty Reinforcement Signal

As mentioned previously, the features of one gait phase impact the subsequent phases. To avoid propagating a faulty reinforcement signal, we provided a -1 reinforcement signal only to the dHDP block that exhibited out of bound angle/duration features. If multiple failure reinforcement signals were generated simultaneously, we prioritized (from high to low) the feedback reinforcement signal in the following order: STF, SWF, SWE, STE. In other words, if multiple phases generated -1 reinforcement signals in the same tuning iteration, we applied the -1 reinforcement signal to the dHDP block that had higher priority.

E. Termination Criteria

For practical applications with a human in the real-time control loop, termination criteria are necessary to avoid human fatigue in the tuning procedure. The tuning procedure was limited to 70 tuning iterations (i.e., $7 \times 70 = 490$ gait cycles) and terminated earlier if tuning was successful. Because the human-prosthesis systems are highly nonlinear, vulnerable to noise and disturbances, and subject to uncertainty, we introduced a tolerance range μ_m ($m = 1, \ldots, 4$ denotes the four gait phases) for acceptable ranges of feature errors, which was 1.5 times the standard deviation of the features from more than 15 gait cycles without supplemental impedance control inputs. Parameter tuning in a given phase is considered successful if the features of this phase meet the tolerance criterion for at least three of the previous five tuning iterations. When all the four phases are successful, the tuning procedure is considered as a success and consequently terminated.

V. Experimental Design

A. Participants

The Institutional Review Board at the University of North Carolina at Chapel Hill approved this paper. One male AB subject (age: 41 years, height: 178 cm, weight: 70 kg) and one male, unilateral transfemoral amputee (TF) subject (age: 21 years, height: 183 cm, weight: 66 kg, time since amputation: six years) were recruited. Both subjects provided written and informed consent before the experiments.

B. Prosthesis Fitting and Subject Training

A certified prosthetist aligned the robotic knee prosthesis for each subject. The TF subject used his own socket, and the AB subject used an L-shape adaptor (with one leg bent 90 degrees) to walk with the robotic knee prosthesis [49].

Each subject visited the laboratory for at least five 2-h training sessions, including set up and rest time, to practice walking with our robotic knee prosthesis on an instrumented treadmill (Bertec Corporation, Columbus, OH, USA) at 0.6 m/s. In the first training session, the impedance parameters were manually tuned based on the observation of the subject's gait and the subject's verbal feedback, similar to the tuning process in the clinic. In the second training session, a physical therapist trained the TF subject to reduce unwarranted compensation while walking with the robotic knee. The subjects were allowed to hold the treadmill handrails for balance when needed. The subject was ready for testing once he was able to walk comfortably for three minutes without holding the handrails.

C. Experiment Protocol

We conducted three testing sessions (over three days) for each subject to evaluate the learning performance of a naïve ADP, and an additional fourth testing session with the TF subject to evaluate the performance of an experienced ADP in prosthesis tuning.

1) Initializing the ADP-Tuner and Impedance Parameters: An ADP-tuner is naïve if the ANN and the CNN were randomly initialized. An ADP-tuner is experienced if the ANN and the CNN were transferred from a previously successful session. We randomly selected initial impedance parameters from a range obtained from our previous experiments conducted on 15 human subjects [13], but the resulting knee motion was not optimized to the target. We excluded the initial parameter sets that: 1) did not allow the subject to walk without holding the handrails; 2) generated prosthesis knee kinematics that were too close to the target knee kinematics (i.e., root-mean-squared error (RMSE) between those two knee trajectories in one gait cycle was less than 4 degrees); or 3) generated knee kinematic features were out of the safety range.

2) Testing Sessions With Naïve ADP-Tuner: In each of the three testing sessions, we first provided three minutes of acclimation time for the subject to walk with the newly initialized naïve ADP-tuner and the control parameters. Then, the subject walked on the treadmill at 0.6 m/s for no more than seven segments, each of which lasted no more than 3-min walking periods. Each segment was followed by a 3-min rest. These rest periods are typical in clinical settings, and they prevent potential confounding effects of fatigue. For all walking periods, we recorded the time series data of knee kinematics from the angle sensor and the loading force from the load cell.

The first 30 s of the first walking period served as our "pretuning" condition, in which the ADP-tuner was not enabled yet, and the impedance parameters remained constant (i.e., initial randomly selected impedance parameters). The last 30 s of their final walking period served as our "post-tuning" condition for performance evaluation, in which the ADP-tuner was disabled and the impedance parameters were again held constant (i.e., the impedance parameters were at the final parameters provided by the ADP-tuner).

During all other walking periods, we asked the subjects to walk in a consistent manner on the treadmill while the ADP controller was enabled and iteratively updated the prosthesis impedance parameters. Each update (defined as ADP learning iteration) was performed for every seven gait cycles. As said previously, this is to reduce step-to-step variability in the knee kinematics features of the peak angle and the phase duration. We paused the ADP-tuner during each rest period to prevent losing learned information.

We terminated the testing session when one of the two stop criteria were met: 1) the testing session reached 70 learning iterations to avoid subject fatigue or 2) errors of all four angle features were within their corresponding tolerance range μ for three out of the previous five ADP learning iterations.

3) Testing Session With Experienced ADP-Tuner: To evaluate if knowledge of the previously learned ADP-tuners would make learning more efficient, we conducted an additional testing session with the TF subject on another day with the same protocol. We instead started with an experienced ADP, which used ANN and CNN network coefficients derived from the previous session that generated the lowest RMSE.

D. Data Analysis

The time-series robotic knee kinematics data were segmented into gait cycles based on the events of heel-strike (Fig. 2), and were then normalized to 100 samples per gait cycle.

The accuracy of the naïve and experienced ADP-tuner was evaluated by the RMSE between measured and target knee kinematics and the feature errors obtained in each tuning iteration. To compare the pretuning and post-tuning performance, the averaged RMSE of knee kinematics and feature errors of 20 gait cycles in pretuning and post-tuning conditions were calculated and compared.

Data efficiency was quantified by the number of learning iterations in each testing session. Time efficiency was quantified by the subject's walking duration in each testing session.

Finally, the stability of the ADP-tuner was demonstrated by the tuned knee impedance parameters and knee kinematics (i.e., RMSE and feature errors averaged across seven gait cycles within each iteration) across learning iterations.

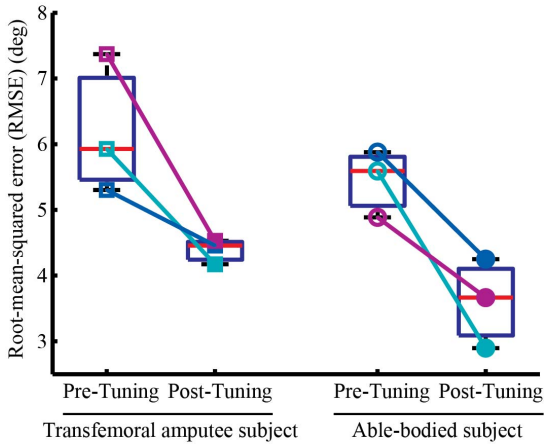


Fig. 3. Comparison of knee kinematics by RMSE between pretuning and post-tuning across multiple testing sessions. The square markers represent the testing sessions from the TF subject, and circle markers represent the testing sessions from AB subject. Open marker represents the pretuning condition, and closed marker represents the post-tuning condition.

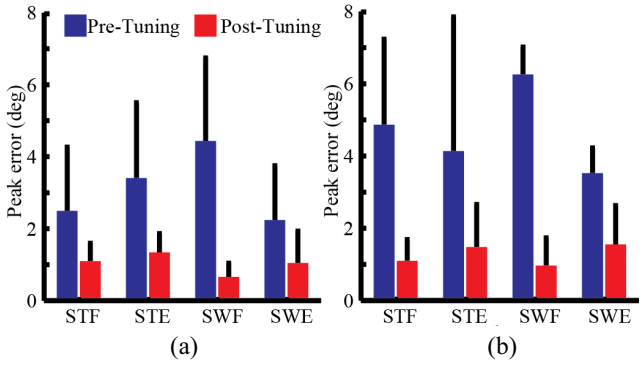


Fig. 4. Peak error comparison between pretuning and post-tuning conditions of the TF subject (a), and the AB subject (b) at each phase. Each bar represents the mean error of three testing sessions, and the error bars denote one standard deviation from the mean.

VI. Results

As a measure of accuracy of the ADP-tuner, the RMSE of the robotic knee angle (compared to the target) averaged across testing sessions and subjects decreased from 5.83 ± 0.85 degrees to 3.99 ± 0.62 degrees (Fig. 3, individual subject results). All the angle feature errors decreased after tuning by the ADP-tuner (Fig. 4). The duration feature errors did not show a consistent trend (Fig. 5) across these two subjects. This variability of the duration feature errors was no surprise because: 1) the duration of each phase is partially controlled by the human prosthesis user and 2) our ADP algorithm allowed more flexibility (or relatively larger acceptable range) of the duration feature errors than the angle feature errors to meet the target and complete tuning.

As measures of data and time efficiency, the ADP-tuner took an average of 43 ± 10 learning iterations to find the "optimal" impedance parameters, amounting to an average of 300 gait cycles and 10 ± 2 minutes of walking. The data and time efficiency were similar between the subjects (AB: 45 ± 9 iterations and amputee subject: 41 ± 12 iterations).

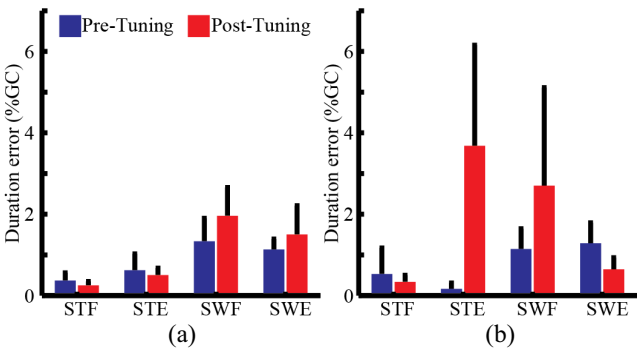


Fig. 5. Duration error comparison between pretuning and post-tuning conditions of the TF subject (a) and the AB subject (b) for each phase. Each bar represents the mean error of three testing sessions, and the error bars denote one standard deviation from the mean.

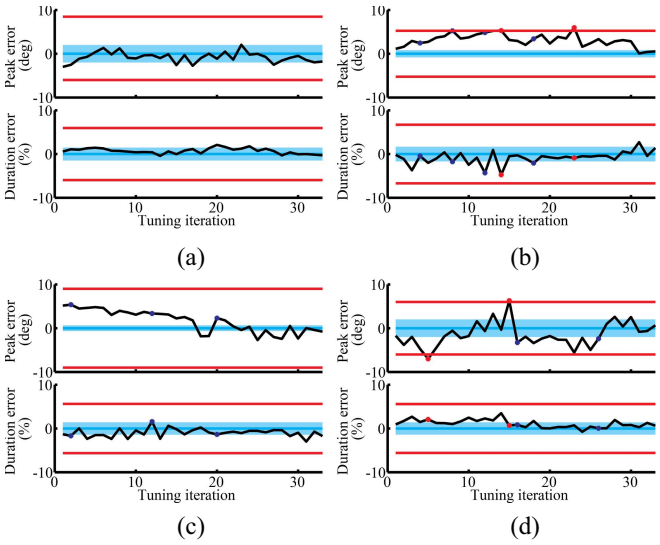


Fig. 6. Peak error and duration error during the four phases for a representative tuning procedure. (a) STF, (b) STE, (c) SWF, and (d) SWE. The red dots were times when the -1 reinforcement signals incurred, and the blue dots were times when the -0.8 reinforcement signals incurred. The horizontal blue areas, which centered at zero, indicate the tolerance ranges for each feature. The paired horizontal red lines indicate the allowed maximum and minimum exploration limits for each feature.

Both the feature errors and impedance parameters generally stabilized by the post-tuning period (Figs. 6 and 7, representative trial shown). In particular, both the feature errors and the impedance parameters of the swing flexion and swing extension gait phases stabilized. However, for stance flexion and stance extension, the feature errors stabilized, while the impedance parameters were still changing slowly. The final impedance parameters that the ADP-tuner selected to allow the user to walk with a near-normal knee motion, or the target knee profile, were not the same across all testing sessions (Table I). In general, the stiffness parameters and damping parameters at stance phases (2.33 ± 0.56 Nm/deg, 0.13 ± 0.05 Nms/deg) were greater than those of the swing phases (0.95 ± 0.83 Nm/deg, 0.03 ± 0.02 Nms/deg). In the experienced ADP test session, for all four gait phases, both the angle and duration feature errors followed a decreasing trend toward zero [Fig. 8(a) and (b)]. The $\hat{J}$ value of the CNN network decreased along the tuning

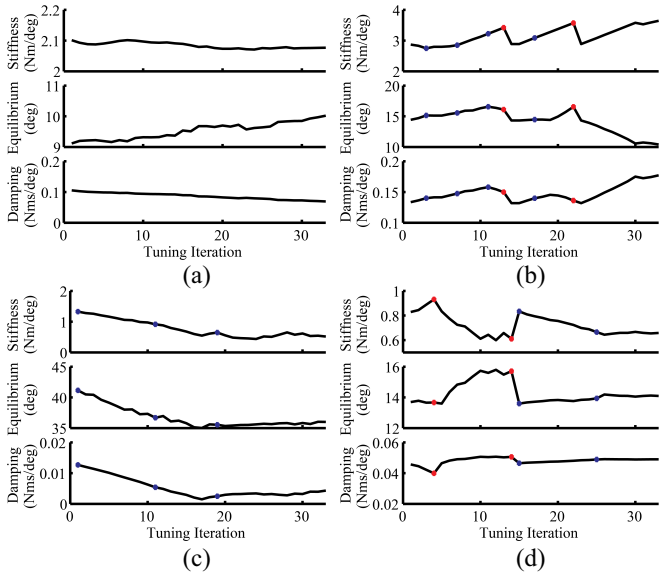


Fig. 7. Impedance parameters of the four phases during a representative tuning procedure. (a) STF, (b) STE, (c) SWF, and (d) SWE. The meanings of the red and blues dots are the same as in Fig. 6.

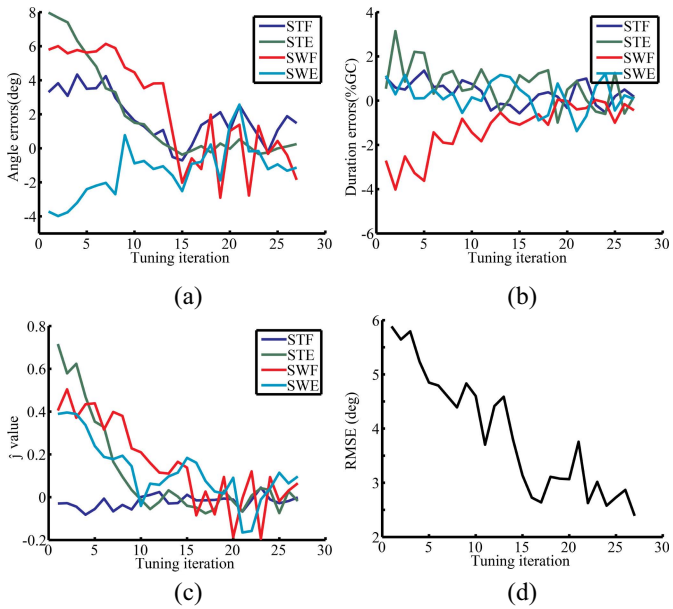


Fig. 8. Learned ADP auto-tuner online evaluation results. (a) Trends of angle error along tuning iterations. (b) Trends of duration error along tuning iterations. (c) Changing $\hat{J}$ values as learning proceeded. (d) RMSE along tuning iterations.

iteration [Fig. 8(c)], and the RMSE of the robotic knee kinematics decreased from 5.9 degrees to 2.5 degrees from pre- to post-tuning [Fig. 8(d)]. In this case evaluation, the experienced ADP-tuner took 28 iterations (approximately 7 min) to find the 12 optimal impedance parameters. No additional reinforcement signal occurred during this testing session with the experienced/learned ADP.

VII. Discussion

This paper aims at investigating the feasibility of a novel RL-based approach for personalization of the control of a robotic knee prosthesis. A total of 12 impedance parameters were tuned simultaneously using our ADP-tuner for two human subjects. Here, we will address the implications and remaining challenges of our proposed RL-based approach to achieve our design objective of automatically tuning of robotic prostheses for amputees.

A. Feasibility and Reliability

The accuracy of ADP-tuner to meet the target knee angle profile both for each gait phase (Figs. 4 and 5) and the entire gait cycle (Fig. 3) indicates that the ADP-tuner was feasible to optimize a large number of prosthesis control parameters. In this study, the ADP-tuner adjusted impedance parameters to allow both subjects to walk consistently toward near-normal knee kinematics. In addition, the ADP-tuner reliably reproduced similar results for all testing sessions, each of which began with different, randomly initialized ANN and CNN weight matrices (i.e., no prior knowledge built into the learning controller), and impedance parameters.

Variations in the final impedance parameters after ADP tuning indicated that multiple combinations of impedance parameters yielded similar prosthesis kinematics (Table I). This is not surprising because according to (2), the motor torque is underdetermined by a combination of three impedance parameters. It would be an interesting future study to investigate an optimal combination of control parameters subject to additional constraints or optimization objectives.

Even though the prosthetic knee kinematics were solely measured from the prosthesis, it represented an inherently combined effort from both the human and the machine or the prosthesis controller. Based on our results, we postulate that the robotic knee flexion/extension peaks are primarily influenced by the impedance parameters and thus affected by our ADP-tuner (Fig. 4), but the duration of each gait phase may be dominated by the human user (Fig. 5). Subjects were able to control the timing of their gait events likely because they can control when to place and lift the prosthetic foot on or off the treadmill with their ipsilateral hip and the entire body. In the feedback control of robotic prostheses, the feedback signals must be responsive to the control action. Therefore, we believe that using knee kinematics as the feedback and optimization state was reasonable as a first step, but questions regarding the appropriate control objective remain open. We plan to investigate this topic systematically with future studies.

B. Efficiency

Starting without any prior knowledge or a plant model, our ADP-tuner was able to gather information and gain understanding on how to simultaneously tune the 12 control parameters in 10 min of one test session, or 300 gait cycles for both subjects. As a reference, an advanced expert system tuning method required at least three days of systematic recording of a human experts tuning decisions and transferred those knowledge to a computer algorithm, which then took 96 gait cycles to tune the impedance parameters [13]. Note, however, this cyber-expert system is subjective (i.e., biased by prosthetists experience) and inflexible when the system input

TABLE I
POST-TUNING IMPEDANCE PARAMETERS OF THREE TESTING SESSIONS FOR TWO SUBJECTS

Phase	Stance Flexion			Stance Extension			Swing Flexion			Swing Extension		
Impedance Parameter[a]	k	θ_e	b	k	θ_e	b	k	θ_e	b	k	θ_e	b
TF S1	2.077	10.020	0.069	3.644	10.399	0.177	0.515	36.000	0.004	0.658	14.096	0.049
TF S2	2.423	9.842	0.089	2.534	13.366	0.165	0.740	35.098	0.020	0.609	12.546	0.046
TF S3	2.068	11.099	0.103	2.493	13.749	0.177	0.601	32.779	0.006	0.657	12.814	0.066
AB S1	1.948	9.914	0.066	2.542	12.875	0.133	1.853	36.754	0.013	0.967	24.068	0.011
AB S2	2.783	8.241	0.067	2.288	14.010	0.127	0.305	52.407	0.019	0.522	16.768	0.014
AB S3	1.578	10.774	0.107	1.571	19.581	0.237	3.294	36.444	0.042	0.625	16.334	0.057

[a] k (Nm/deg) is the stiffness coefficient; θ_e (deg) is the equilibrium position; b (Nms/deg) is the damping coefficient.

and output changes. Our ADP-tuner is objective and flexible in structure. Furthermore, the experienced ADP-tuner (i.e., with some prior knowledge) took only 210 gait cycles without additional reinforcement signals to learn, demonstrating the learned knowledge can be effectively transferred to tune the impedance parameters. Therefore, we believe the ADP-tuner is time and data efficient for potential clinical use.

In daily life, the ADP-tuner potentially can handle slow changes, such as body weight change. For environmental demand changes, like going from level ground walking to walking up a ramp or stair, the ADP-tuner could potentially find the optimal control parameters for each locomotion modes (e.g., ramp walking and stair climbing), which might take longer, but could store the impedance parameters and switch the parameters when the user encounters the task changes in real life. We will explore this in our future work.

C. Learning Outcome

The ADP-tuner learned through reinforcement signals (Figs. 6 and 7, colored point characters) and was able to tune the impedance parameters that in turn decreased the angle feature errors to meet the respective error tolerance. At the end of the tuning procedure, the feature errors also maintained within the tolerance range for at least three of the previous five ADP learning iterations in order to terminate the tuning session.

The feature errors clearly converged toward zero in two out of four phases [Fig. 6(c) and (d)], and the corresponding impedance parameters [Fig. 7(c) and (d)] stabilized. These results show promise that the ADP-tuner is able to generate a converged policy for these gait phases. However, in the remaining two phases, the impedance parameters were still adapting, but the feature errors were within the tolerance ranges. These results lead us to believe the feature errors were not very responsive to certain impedance parameters or combinations of parameters. This phenomenon may be also caused by our stop criteria of maximum 70 tuning iterations, enforced to keep this paper practical for clinical applications and to prevent amputee from fatigue. In the future, to achieve a converged policy quickly, we might address this challenge by adding small disturbances to the impedance parameters when the feature errors approach zero in order to test convergence properties of the ADP-tuner and by allowing ADP-tuner to accumulate more learning experiences.

Finally, we demonstrated that the experienced ADP-tuner, after only interacting with the human-prosthesis system for one testing session, effectively learned tuning knowledge to reach the target knee kinematics. With both human and interphase influence contributing to the robotic knee motion [49], we expected both the angle and the duration feature errors would oscillate about zero [Fig. 8(a) and (b)]. In addition, the experienced ADP tuned the prosthesis control parameters faster than the naïve ADP. This exciting result opens up the opportunity to make our prosthesis controller adaptive to users in their daily life.

D. Implications of the Results

In this paper, the ADP-tuner had no prior knowledge of: 1) the structure of the impedance controller and 2) the mechanical properties of the robotic knee prosthesis. The only information observed by the ADP was the state of the human-prosthesis system through measurements of the prosthetic knee angle, and reinforcement signals when the performance/features were out of allowed exploration range. Therefore, the ADP-tuner design potentially can be applied to knee prostheses with different mechanical structures and control methods and even possibly extended to the control parameter tuning problem for ankle prostheses and exoskeletons.

Further, our method may be applied to other control objectives to reach behavioral goals. For example, if the target knee kinematics is to generate a greater swing flexion angle for foot clearance, the experienced ADP-tuner may potentially tune the impedance parameters quickly to reach the new target. Therefore, our learned control policy may significantly enhance the tuning/personalization process of robotic prostheses, as well as the adaptability of the prosthesis to changes within a user and its environment.

E. Limitations and Future Work

In this paper, we focused on demonstrating the feasibility of RL-based control to automatically tune robotic prostheses. Individuals differ in their physical conditions and behaviors, and they interact with different terrains in their daily life. In order to reveal the full capability of this promising approach, we need to further evaluate the ADP-tuner with more human subjects and more locomotion tasks (e.g., ascending or descending stairs and walking on grass) to consolidate the reliability of this RL-based approach.

Another limitation is that in this proof-of-concept study, we chose the normative prosthetic knee kinematics as the tuning objective, which might not be perfectly aligned with every

amputee's preference and the human-prosthesis system's overall performance. However, from the parameter tuning point of view, this paper proved that the ADP-tuner can find a set of impedance parameters to meet a given tuning objective automatically. Future studies will focus on determining the desired knee target features that enhance the human-prosthesis gait performance, such as gait symmetry index, stability margin, or even the user's subjective preferences.

VIII. Conclusion

In this paper, we provided a significant leap forward from the traditional time-consuming and labor-intensive manual tuning of the prosthesis control parameters. We developed a novel RL-based control approach to automatically tune 12 impedance parameters of a robotic knee prosthesis. The new concept was validated on one AB subject and one TF through multiple testing sessions. The promising results illustrated that the ADP-tuner is a feasible and safe method to automatically configure a large number of control parameters within the scope of this paper. The algorithm learns efficiently through interaction with the human-prosthesis system in real time, without any prior tuning knowledge from either a trained clinician or a field expert. The learning also does not require a prior plant model of the human-prosthesis system.

The results of this paper might lead to a novel prosthesis control framework to personalize the robotic prostheses, optimize human-prosthesis system performance in gait, and make the prosthesis adaptive to users. In the future, we will further explore the learning control designs that automatically meet human-prosthesis' various performance goals, such as gait symmetry, stability, and even user perception.

Acknowledgment

The authors would like to thank M. Liu, Ph.D. and S. Huang, Ph.D. for help with the experimental design, D. Frankena, CPO and M. Soyars, PT for subject training and experimental setup, and both participants.

References

[1] J. K. Hitt, T. G. Sugar, M. Holgate, and R. Bellman, "An active foot–ankle prosthesis with biomechanical energy regeneration," *J. Med. Device.*, vol. 4, no. 1, Mar. 2010, Art. no. 011003.

[2] L. Ambrozic *et al.*, "CYBERLEGs: A user-oriented robotic transfemoral prosthesis with whole-body awareness control," *IEEE Robot. Autom. Mag.*, vol. 21, no. 4, pp. 82–93, Dec. 2014.

[3] F. Sup, H. A. Varol, J. Mitchell, T. J. Withrow, and M. Goldfarb, "Preliminary evaluations of a self-contained anthropomorphic transfemoral prosthesis," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 6, pp. 667–676, Dec. 2009.

[4] 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.

[5] 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, Oct. 2014.

[6] 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, nos. 3–4, pp. 461–474, Dec. 2014.

[7] A. M. Simon *et al.*, "Configuring a powered knee and ankle prosthesis for transfemoral amputees within five specific ambulation modes," *PLoS ONE*, vol. 9, no. 6, Jun. 2014, Art. no. e99387.

[8] A. Brandt, Y. Wen, M. Liu, J. Stallings, and H. H. Huang, "Interactions between transfemoral amputees and a powered knee prosthesis during load carriage," *Sci. Rep.*, vol. 7, no. 1, Nov. 2017, Art. no. 14480.

[9] S. Pfeifer, H. Vallery, M. Hardegger, R. Riener, and E. J. Perreault, "Model-based estimation of knee stiffness," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 9, pp. 2604–2612, Sep. 2012.

[10] 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.

[11] M. R. Tucker, C. Shirota, O. Lambercy, J. S. Sulzer, and R. Gassert, "Design and characterization of an exoskeleton for perturbing the knee during gait," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 10, pp. 2331–2343, Oct. 2017.

[12] R. D. Gregg, T. Lenzi, L. J. Hargrove, and J. W. Sensinger, "Virtual constraint control of a powered prosthetic leg: From simulation to experiments with transfemoral amputees," *IEEE Trans. Robot.*, vol. 30, no. 6, pp. 1455–1471, Dec. 2014.

[13] 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, Sep. 2016.

[14] J. R. Koller, D. H. Gates, D. P. Ferris, and C. D. Remy, "'Body-in-the-loop' optimization of assistive robotic devices: A validation study," in *Proc. Robot. Sci. Syst. XII*, Ann Arbor, MI, USA, Jun. 2016, pp. 1–10.

[15] J. Zhang *et al.*, "Human-in-the-loop optimization of exoskeleton assistance during walking," *Science*, vol. 356, no. 6344, pp. 1280–1284, Jun. 2017. [Online]. Available: http://www.sciencemag.org/lookup/doi/10.1126/science.aal5054

[16] 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, Feb. 2018, Art. no. eaar5438.

[17] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA, USA: Athena Sci., 1996.

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

[19] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 2nd ed., D. Balding, Ed. Hoboken, NJ, USA: Wiley, 2011.

[20] P. J. Werbos, "A menu of designs for reinforcement learning over time," in *Neural Networks for Control*. Cambridge, MA, USA: MIT Press, 1990, ch. 3, pp. 67–95.

[21] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," Ph.D. dissertation, Harvard Univ., Cambridge, MA, USA, 1974.

[22] P. J. Werbos, "Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research," *IEEE Trans. Syst. Man Cybern.*, vol. 17, no. 1, pp. 7–20, Jan. 1987.

[23] D. Wang, H. He, and D. Liu, "Adaptive critic nonlinear robust control: A survey," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3429–3451, Oct. 2017.

[24] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Dec. 2011.

[25] H. Zhang, J. Zhang, G.-H. Yang, and Y. Luo, "Leader-based optimal coordination control for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 1, pp. 152–163, Feb. 2015.

[26] J. Zhang, H. Zhang, and T. Feng, "Distributed optimal consensus control for nonlinear multiagent system with unknown dynamic," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3339–3348, Aug. 2018.

[27] D. Wang and D. Liu, "Learning and guaranteed cost control with event-based adaptive critic implementation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6004–6014, Dec. 2018.

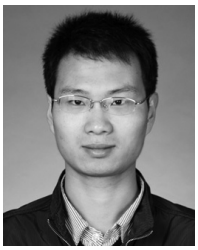
[28] D. Wang, H. He, X. Zhong, and D. Liu, "Event-driven nonlinear discounted optimal regulation involving a power system application," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 8177–8186, Oct. 2017.

[29] Y. Wang, W. X. Zheng, and H. Zhang, "Dynamic event-based control of nonlinear stochastic systems," *IEEE Trans. Autom. Control*, vol. 62, no. 12, pp. 6544–6551, Dec. 2017.

[30] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 997–1007, Sep. 1997.

[31] M. Riedmiller, "Neural fitted Q iteration—First experiences with a data efficient neural reinforcement learning method," in *Proc. 16th Eur. Conf. Mach. Learn*, 2005, pp. 317–328.

[32] J. Si and Y.-T. Wang, "Online learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 264–276, Mar. 2001.
[33] M. I. Jordan and R. A. Jacobs, "Learning to control an unstable system with forward modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 1990, pp. 324–331.
[34] R. Hafner and M. Riedmiller, "Reinforcement learning in feedback control: Challenges and benchmarks from technical process control," *Mach. Learn.*, vol. 84, nos. 1–2, pp. 137–169, Jul. 2011.
[35] M. Riedmiller, M. Montemerlo, and H. Dahlkamp, "Learning to drive a real car in 20 minutes," in *Proc. Front. Converg. Biosci. Inf. Technol*, 2007, pp. 645–650.
[36] M. Riedmiller, T. Gabel, R. Hafner, and S. Lange, "Reinforcement learning for robot soccer," *Auton. Robots*, vol. 27, no. 1, pp. 55–73, Jul. 2009.
[37] T. Gabel and M. Riedmiller, "Adaptive reactive job-shop scheduling with reinforcement learning agents," *Int. J. Inf. Technol. Intell. Comput.*, vol. 24, no. 4, pp. 1–30, 2008.
[38] R. Hafner and M. Riedmiller, "Neural reinforcement learning controllers for a real robot application," in *Proc. IEEE Int. Conf. Robot. Autom.*, Rome, Italy, Apr. 2007, pp. 2098–2103.
[39] R. Enns and J. Si, "Helicopter trimming and tracking control using direct neural dynamic programming," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 929–939, Jul. 2003.
[40] L. Yang, J. Si, K. S. Tsakalis, and A. A. Rodriguez, "Direct heuristic dynamic programming for nonlinear tracking control with filtered tracking error," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1617–1622, Dec. 2009.
[41] C. Lu, J. Si, and X. Xie, "Direct heuristic dynamic programming for damping oscillations in a large power system," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 1008–1013, Aug. 2008.
[42] R. Enns and J. Si, "Helicopter flight-control reconfiguration for main rotor actuator failures," *J. Guid. Control. Dyn.*, vol. 26, no. 4, pp. 572–584, Jul. 2003.
[43] W. Guo *et al.*, "Online supplementary ADP learning controller design and application to power system frequency control with large-scale wind energy integration," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1748–1761, Aug. 2016. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7124490
[44] 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.
[45] S. L. Delp *et al.*, "OpenSim: Open-source software to create and analyze dynamic simulations of movement," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 11, pp. 1940–1950, Nov. 2007.
[46] B. E. Lawson, H. A. Varol, A. Huff, E. Erdemir, and M. Goldfarb, "Control of stair ascent and descent with a powered transfemoral prosthesis," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 3, pp. 466–473, May 2013.
[47] 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.
[48] F. Liu, J. Sun, J. Si, W. Guo, and S. Mei, "A boundedness result for the direct heuristic dynamic programming," *Neural Netw.*, vol. 32, pp. 229–235, Aug. 2012, doi: 10.1016/j.neunet.2012.02.005.
[49] Y. Wen, M. Liu, J. Si, and H. H. Huang, "Adaptive control of powered transfemoral prostheses based on adaptive dynamic programming," in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Orlando, FL, USA, 2016, pp. 5071–5074.



Yue Wen received the B.S. degree in automation from the Wuhan University of Technology, Wuhan, China, in 2011 and the M.S. degree in control theory and engineering from the Huazhong University of Science and Technology, Wuhan, in 2014. He is currently pursuing the Ph.D. degree with the NCSU/UNC 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.

His current research interests include adaptive control of bionic limbs and assistive robotic devices, machine learning, and human motion analysis.

Jennie Si (F'08) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, and the Ph.D. degree from the University of Notre Dame, Notre Dame, IN, USA.

She has been a Faculty Member with the Department of Electrical Engineering, Arizona State University, Tempe, AZ, USA, since 1991. She consulted for Intel, Santa Clara, CA, USA, Arizona Public Service, Phoenix, AZ, USA, and Medtronic, Dublin, Ireland. Her current research interests include reinforcement learning-based control, namely, adaptive dynamic programming utilizing machine learning and neural-network approximation techniques. She also works on fundamental neuroscience studies of the frontal cortex and its role during learning behavior using electrophysiological techniques.

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 has served on several professional organizations executive boards and international conference committees. She was the Vice President for Education in the IEEE Computation Intelligence Society from 2009 to 2012. She was an Advisor to the NSF Social Behavioral and Economical Directory. She served on several proposal review panels. She is a past Associate Editor of the IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING and the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, and an Action Editor of *Neural Networks*. She is currently an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS.

Andrea Brandt received the B.Sc. degree in mathematics from the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, in 2013. She is currently pursuing the Ph.D. degree with the NCSU/UNC Joint Department of Biomedical Engineering, North Carolina State University, Raleigh, NC, USA, and the University of North Carolina at Chapel Hill.

Her current research interests include powered prostheses, amputee gait, and rehabilitation.

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. degree from Arizona State University, Tempe, AZ, USA, in 2014, where he is currently pursuing the Ph.D. degree with the School of Electrical, Computer and Energy Engineering.

His current research interests include machine learning, neural networks, and rehabilitation robotics.

He (Helen) Huang (S'03–M'06–SM'12) received the Ph.D. degree in biomedical engineering from Arizona State University, Tempe, AZ, USA.

She was a Post-Doctoral fellow in neural engineering with the Rehabilitation Institute of Chicago/Northwestern University, Evanston, IL, USA. She is currently a Professor with the NCSU/UNC Joint Department of Biomedical Engineering and the Director of the Closed-Loop Engineering for Advanced Rehabilitation Core, North Carolina State University, Raleigh, NC, USA, and the University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. Her current research interests include neural-machine interfaces for artificial limbs and exoskeletons, human–robot interaction, adaptive and optimal control of wearable robots, and human movement control.

Dr. Huang was a recipient of the Delsys Prize for Innovation in Electromyography, the Mary E. Switzer Fellowship with NIDILRR, the NSF CAREER Award, and was named NC State Faculty Scholar in 2015. She is also a member of the Society for Neuroscience and Biomedical Engineering Society.