# Neural-Dynamics Optimization and Repetitive Learning Control for Robotic Leg Prostheses

Qinjian Li, Tao Zhang, Guoxin Li, Zhijun Li, *Senior Member, IEEE*,
Haisheng Xia, and Chun-Yi Su, *Senior Member, IEEE*

***Abstract*—Rapid development in robotics and bionics make it possible for robotic leg prostheses to help amputees while impose challenges on duplicating the motion characteristics of amputees' healthy leg. One critical problem in prosthetic control is joint angle drift problem, that is, a repetitive motion trajectory in task space cannot guarantee that the generated motion trajectories in joint space are also repetitive. In order to solve the problem, we propose neural-dynamics optimization for robotic leg prostheses to generate the repetitive joint trajectories in real-time. Our proposed method duplicates the self-selected walking speed of the amputees' healthy leg. The online motion generation is formulated as a constrained quadratic programming (QP) optimization problem whose objective function adopts the kinetic energy and joint displacement performance criterion. The varying parameter convergent differential neural network is developed as a real-time QP solver which can globally converge to the optimal solution of the constrained QP problem. Then, a repetitive learning controller is designed for robotic leg prostheses to reduce the tracking errors while following the repetitive motion. Through the physical experiments on the developed two-degree-of-freedom robotic leg prosthesis worn on an amputee, the neural-dynamics optimization is substantiated to be a remedy of online motion generation, and the effectiveness of repetitive learning control for reducing the prosthetic tacking errors is verified.**

***Index Terms*—neural-dynamics optimization, repetitive learning control, robotic leg prostheses.**

## I. Introduction

The latest development in robotics and bionics make it possible for robotic leg prostheses to duplicate the motion characteristics of lower-limb amputees' healthy legs in level walking [1]. This ability to duplicate the motion characteristics can significantly improve the motion symmetry between amputees' healthy legs and the prostheses, which increase the mobility of the lower-limb amputees [2]. In current researches, the healthy leg's angular positions or velocities in joint space are usually selected to be duplicated [3]–[5]. However, there are differences in the structure of the prosthesis and human leg, even if the prosthesis duplicate the joint position/velocity as the healthy leg, it cannot produce the same walking speed (horizontal velocity in task space) as the healthy leg. The motion characteristics of human healthy legs in task space, such as the walking speed or step length, are easier to measure and more significant in practical applications [6]. Inspired by these points, we propose to duplicate the horizontal velocity of the healthy foot in task space (which can be regarded as the walking speed of lower-limb amputee) to control the robotic leg prostheses while assisting amputees in level walking. To achieve this goal, the motion generation and tracking control for robotic leg prostheses need to be investigated.

In order to duplicate the level walking speed of amputees in real-time, the robotic leg prostheses need to online generate the desired trajectories in joint space according to the healthy leg's horizontal velocity in task space. Without considering the non-periodic disturbance, the level walking can be regarded as the repetitive motion in task space of human healthy legs [7]. However, the joint angle drift problem [8] exists in robotic leg prostheses when only tracking the horizontal velocity in task space. Therefore, the generated joint trajectories need to be repeatable. In this paper, the repetitive motion generation of robotic leg prostheses is formulated as a constrained quadratic programming (QP) optimization problem. To handle the small and middle size QP problems, among traditional QP numerical optimization approaches, the interior-point method [9]–[11] and active-set method [12]–[14] can usually achieve acceptable speed when running them on a computer or server with good calculated performance [15]. However, consider the mobility and portability of the robotic leg prostheses, the control device can not choose the heavy and large computer or server, and only one option is embedded control board which has a limited computation power. Thus, we need exploit an QP optimization method for fulfilling the real-time requirement of generating repetitive joint trajectories for robotic leg prostheses under the limited calculated performance. In recent years, the neural-dynamics optimization is used for solving QP problems in robotic systems due to its parallel computation and high efficiency. In [16], a recurrent neural network was utilized to calculate the optimal solution of the constrained motion generation scheme of humanoid robots. In [17], a dual neural network based on linear variational inequality was used to ensure the model predictive controller obtain the effective tracking performance of a non-holonomic mobile robot. The convergence speed of these neural networks in neural-dynamics optimization can be accelerated if the convergence parameters are changed into time-varying parameters. In [18], a varying parameter convergent differential neural network (VPCDNN) was developed to solve the established QP problems of robot manipulators. However, all experimental

This work was supported in part by the National Natural Science Foundation of China under Grant 61625303 and Grant U1913601. Corresponding author: Zhijun Li (zjli@ieee.org).

Q. Li, T. Zhang, G. Li, Z. Li and H. Xia are with the Department of Automation, University of Science and Technology of China, Hefei 230026, China. Email: lqj0414@mail.ustc.edu.cn, sa010096@mail.ustc.edu.cn, lgxin16@mail.ustc.edu.cn, zjli@ieee.org, hsxia@ustc.edu.cn.

C.-Y. Su is with the School of Aeronautical Engineering (School of Intelligent Manufacturing), Taizhou University, Taizhou 318000, China, on leave from Concordia University, Montreal, QC H3G 1M8, Canada. E-mail:chunyi.su@gmail.com.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2021.3071936, IEEE/ASME Transactions on Mechatronics



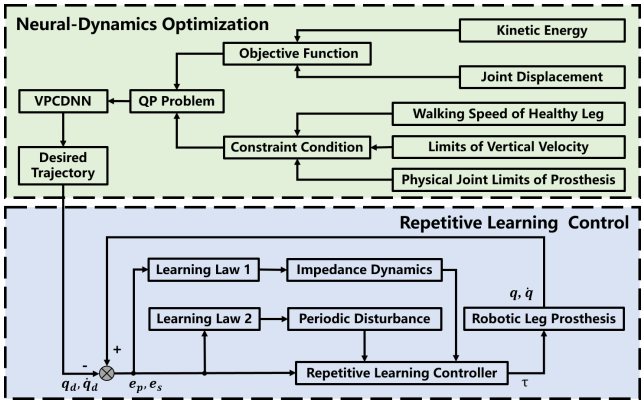



Fig. 1. The control scheme in this paper.

results in [18] were obtained through off-line optimization, and real-time optimization was not considered and discussed thoroughly. Inspired by above works, we develop a VPCDNN as the QP solver for online motion generation of robotic leg prostheses.

The next important work is developing a suitable prosthetic controller for tracking the repetitive motion. In [19], several schemes have been introduced for robotic leg prosthetic control. The commonly used robotic leg prosthetic controllers are PD controller or impedance controller based on the finite-state machines, while lots of the control parameters must be adjusted [20], [21]. In order to avoid the troublesome parameters adjusting, a new control strategy for robotic leg prostheses that unifies the gait in an entire cycle using the continuous sense of phase was presented in [22]. In [23], a high performance decoupled control approach was developed to control the series elastic actuator as an ideal torque source, which provided the basis of designing torque controller for some robotic leg prostheses with compliance structure. However, it should be noted that all the controllers introduced above neglected the repeatability characteristics of robotic leg prosthetic motion. Repetitive learning control is suitable for handling the repetitive motion tasks in robotic systems that have periodic motion characteristics [24]–[28]. Although repetitive learning control has been widely utilized in many robotic systems and has achieved outstanding control performance, to the best of the authors' knowledge, there is no research work on developing repetitive learning control for robotic leg prostheses. It is worth emphasizing that the repeatability characteristics plays a critical role in walking assistance for robotic leg prostheses. Designing a repetitive learning controller would be beneficial for robotic leg prostheses to improve the tracking performance of repetitive joint trajectories. Motivated by this point, a repetitive learning controller is proposed for tracking control of robotic leg prostheses.

The control scheme in this paper is summarized in Fig. 1. In the block diagram above, the objective function which consists of kinetic energy and joint displacement performance criterion is utilized. The walking speed of healthy leg, the limits of vertical velocity, and the physical joint limits of the prosthesis are incorporated into the constraint condition. Thus, the online motion generation is formulated as a QP problem with the above-mentioned objective function and constraint condition. Then, a VPCDNN is used for solving the constrained QP problem in real-time. The repetitive joint trajectories generated by neural-dynamics optimization are taken as the desired joint trajectories of robotic leg prostheses. In the bottom part of the block diagram, the impedance dynamic parameters are estimated by a predefined learning law in real-time. Meanwhile, the periodic disturbance compensation is online estimated using another learning law. Finally, they constitute the proposed repetitive learning controller together with another feedback part related to tracking errors. The contributions of this paper mainly include the following aspects:

1) A neural-dynamics optimization method with a VPCDNN is developed for robotic leg prostheses to generate the repetitive joint trajectories based on the collected level walking data from healthy leg, our proposed method has lower computational complexity than traditional QP numerical optimization methods, and can fulfill the real-time requirement of generating the desired joint trajectories;

2) A repetitive learning controller is proposed for robotic leg prostheses to continuously improve the control performance without explicit knowledge of the prosthetic dynamic model parameters, the effectiveness of the proposed controller for reducing the tracking errors under the repetitive motion is demonstrated in the physical experiments;

3) With the neural-dynamics optimization and repetitive learning controller, we have achieved online duplicating the walking speed of lower-limb amputees' healthy leg in real world walking experiment for the first time, which can achieve the motion symmetry between amputees' own healthy legs and their prostheses in task space.

## II. Preliminaries and Problem Formulation

### A. Kinematic Modeling

As shown in Fig. 2, the two-degree-of-freedom (2-DOF) robotic leg prosthesis can be viewed as a three-link structure. The attach point represents a rectangular pyramid at the junction of the amputee's recipient cavity and leg prosthesis. The link between the attach point and the knee joint can be seen as the base of the prosthesis. The end-effector point is assumed to be the prosthetic foot's center of mass.

Let us define a vector of the end-effector position in task space $r = [r_x \ r_y]^T \in \mathbb{R}^2$ and a vector of joint position $q = [q_1 \ q_2]^T \in \mathbb{R}^2$. Forward kinematics of the prosthesis can be expressed as

$$r = f_k(q) \tag{1}$$

where $f_k$ is a non-linear continuous mapping function with the known $l_1$ (the length from point $O_0$ to point $O_1$) and $l_{c2}$ (the length from point $O_1$ to the end-effector point). The relationship between the velocity of the end-effector in task space $\dot{r} = [\dot{r}_x \ \dot{r}_y]^T \in \mathbb{R}^2$ and the joint velocity $\dot{q} = [\dot{q}_1 \ \dot{q}_2]^T \in \mathbb{R}^2$ can be obtained by taking the partial differentiation of $f_k$:

$$\dot{r} = \frac{\partial f_k(q)}{\partial q}\dot{q} = J(q)\dot{q} \tag{2}$$

where $J(q) = [J_x(q) \in \mathbb{R}^{1\times 2}; J_y(q) \in \mathbb{R}^{1\times 2}] \in \mathbb{R}^{2\times 2}$ is the Jacobian matrix of this 2-DOF robotic leg prosthesis, $J_x(q)$ is the first row in $J(q)$, and $J_y(q)$ is the second row in $J(q)$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2021.3071936, IEEE/ASME Transactions on Mechatronics



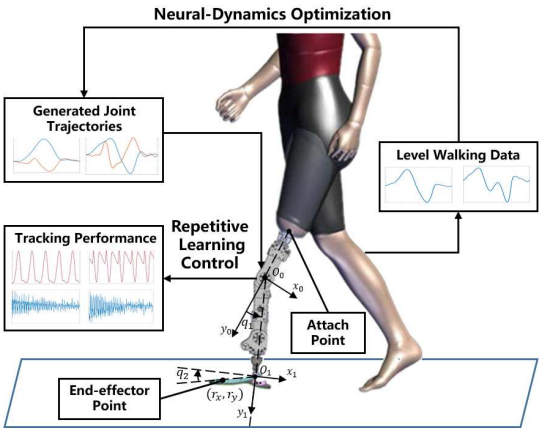



Fig. 2. Model of the two-degree-of-freedom robotic leg prosthesis system.

### B. Dynamic Modeling

The dynamic model of the robotic leg prosthetic can be modeled as the following equation derived from [29]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + E(q) + V(\dot{q}) + D = \tau \quad (3)$$

where $M(q) \in \mathbb{R}^{2\times 2}$ represents the robotic inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{2\times 2}$ represents the centripetal and Coriolis matrix, $G(q) \in \mathbb{R}^2$ represents the gravity force vector, $E(q) \in \mathbb{R}^2$ represents the elastic force vector due to joint stiffness associated with angular position, $V(\dot{q}) \in \mathbb{R}^2$ represents the viscous force vector due to joint damping associated with angular velocity, $D \in \mathbb{R}^2$ represents unknown bounded disturbance vector, $\tau \in \mathbb{R}^2$ represents the joint torque vector of knee and ankle. Besides, there are two properties of the dynamic model:

*Property 1:* The matrix $M(q)$ is symmetric and positive definite.

*Property 2:* The matrix $\dot{M}(q) - 2C(q,\dot{q})$ is skew-symmetric.

### C. Problem Formulation

Consider the velocity of the healthy foot during level walking, its horizontal component in task space could be regarded as the walking speed. When only tracking the horizontal velocity, the numbers of DOF $n = 2$ and the space dimensionality $m = 1$, thus the 2-DOF robotic leg prosthesis becomes kinematically redundant. The relationship between the horizontal velocity of end-effector point in task space $\dot{r}_x \in \mathbb{R}$ and the joint velocity $\dot{q} \in \mathbb{R}^2$ can be obtained from (2):

$$\dot{r}_x = J_x(q)\dot{q}. \quad (4)$$

Now, the differential inverse kinematic problem of (4) is solving the joint velocity $\dot{q}(t) \in \mathbb{R}^2$ under the premise of the horizontal velocity in task space $\dot{r}_x(t) \in \mathbb{R}$ is known. Since the space dimensionality $m$ is less than the numbers of DOF $n$, there are multiple solutions to this differential inverse kinematic problem. The conventional pseudoinverse-type solution to (4) can be expressed as a homogeneous solution plus one minimum-norm particular solution [30], i.e.,

$$\dot{q} = J_x^+\dot{r}_x + (I - J_x^+ J_x)z \quad (5)$$

where $J_x^+ \in \mathbb{R}^{2\times 1}$ denotes the pseudoinverse of $J_x$, $I \in \mathbb{R}^{2\times 2}$ denotes the unit matrix, and $z \in \mathbb{R}^2$ is an arbitrary vector that can be selected as a gradient of a performance criteria to be optimized [31]. By selecting the appropriate vector $z$, we can obtain the desired joint trajectories of robotic leg prostheses according to the amputee's self-selected walking speed. So far, we have transformed the differential inverse kinematic problem into the optimization problem of robotic leg prostheses.

## III. Neural-Dynamics Optimization

### A. Objective Function

To make the inverse kinematics repeatable, the displacement of joint position between the initial state and the current state should be minimized. Hence, the vector $z$ can be selected as $z = \delta(q(t) - q(0))$ where $\delta$ is a positive constant to scale the prosthetic response magnitude to the joint displacement [32]. On the premise of the inverse kinematics is repeatable, the kinetic energy expected to be minimum. Therefore, the objective function of the QP problem is formulated as

$$\text{minimize} \quad \frac{1}{2}\dot{q}^T W \dot{q} + z^T\dot{q} \quad (6)$$

where $W = M(q)$, $z = \delta(q(t) - q(0))$.

### B. Constraint Condition

For the safety of amputees, the robotic leg prostheses should have limits in prosthetic joints and vertical velocity $\dot{r}_y \in \mathbb{R}$. Thus, it is necessary to consider the constrained optimization scheme for robotic leg prostheses. Considering the physical joint position and velocity limits

$$q_{lb} \leqslant q \leqslant q_{ub}, \quad (7)$$

$$\dot{q}_{lb} \leqslant \dot{q} \leqslant \dot{q}_{ub} \quad (8)$$

where the subscript $_{lb}$ and $_{ub}$ denote the lower and upper bounds, respectively. Meanwhile, consider the limits of vertical velocity

$$\dot{r}_{ylb} \leqslant \dot{r}_y \leqslant \dot{r}_{yub},\ \dot{r}_y = J_y(q)\dot{q} \quad (9)$$

where $\dot{r}_{ylb}$ and $\dot{r}_{yub}$ denote the lower and upper bound of vertical velocity, respectively. The limited joint position range $[q_{lb}, q_{ub}]$ and vertical velocity range $[\dot{r}_{ylb}, \dot{r}_{yub}]$ need to be transformed into the joint velocity constraint which is updated dynamically since the problem is solved at the velocity level:

$$k(\mu q_{lb} - q) \leqslant \dot{q} \leqslant k(\mu q_{ub} - q), \quad (10)$$

$$J_y^+\dot{r}_{ylb} \leqslant \dot{q} \leqslant J_y^+\dot{r}_{yub} \quad (11)$$

where $k$ is a positive coefficient selected to scale the feasible region of $\dot{q}$, $\mu \in (0,1]$ is a coefficient that ensures the robotic leg prosthesis would decelerate when it enters the critical region $(q_{lb}, \mu q_{lb})$ or $(\mu q_{ub}, q_{ub})$, and $J_y^+ \in \mathbb{R}^{2\times 1}$ denotes the pseudoinverse of $J_y$. It should be noted that $k$ should be chosen to achieve the condition that the original feasible region of $\dot{q}$ in joint velocity limits (8) is not larger than the one converted by the joint position limits (7). Besides, sharp deceleration will happen with the large values of $k$ when the prosthesis

is close to its joint limits, which is extremely unsafe for the amputees. That is, $k \geqslant \max_{1\leqslant i\leqslant n}\{(\dot{q}_{ubi}-\dot{q}_{lbi})/(q_{ubi}-q_{lbi})\}$. The lower and upper bounds of vertical velocity $\dot{r}_{ylb}$ and $\dot{r}_{yub}$ can be selected appropriately according to the actual walking gait. Thus, the dynamically updated bound constraint should be combined into $\zeta_{lb} \leqslant \dot{q} \leqslant \zeta_{ub}$ from (8), (10) and (11), where the $i$th joint limits of $\zeta_{lb}$ and $\zeta_{ub}$ are defined as

$$\zeta_{lbi} = \max\{\dot{q}_{lbi}, k(\mu q_{lbi} - q_i), J_{yi}^{+}\dot{r}_{ylb}\}, \tag{12}$$

$$\zeta_{ubi} = \min\{\dot{q}_{ubi}, k(\mu q_{ubi} - q_i), J_{yi}^{+}\dot{r}_{yub}\}. \tag{13}$$

Then, the constrained optimization problem of robotic leg prostheses is formulated as the following general QP problem:

$$\text{minimize} \quad \frac{1}{2}\theta^T W\theta + z^T\theta, \tag{14}$$

$$\text{subject to} \quad Q\theta = b, \tag{15}$$

$$\zeta_{lb} \leqslant \theta \leqslant \zeta_{ub} \tag{16}$$

where $\theta$ is the decision vector utilized to replace the joint velocity $\dot{q}$, $W = M(q)$, $z = \delta(q(t) - q(0))$, $Q = J_x(q)$, $b = \dot{r}_x$. The equality constraint (15) denotes a relationship between the desired walking speed $\dot{r}_x \in \mathbb{R}$ and the joint velocity $\dot{q} \in \mathbb{R}^2$ which need to be solved. Using the inequality constraint (16), it is convenient to keep all the joint variables and vertical velocity within their limit bounds.

### C. VPCDNN Solver

A VPCDNN is developed as the real-time QP solver to calculate the optimal solution of the aforementioned QP problem (14)–(16). The deisred joint trajectories can be generated when the objective function (14) is minimized. To achieve this goal, the Lagrange function for (14)–(16) can be designed with the aid of dual decision variables by duality theory. The dual decision variable is usually defined as the Lagrangian multiplier for each constraint (15) and (16). However, to reduce the QP-solver complexity, we could use an elegant treatment to cancel the dual decision variable for bound constraint (16). This means that we only need to define the dual decision vector $\gamma \in \mathbb{R}$ for equality constraint (15). Thus, let $\gamma$ represents the Lagrange multiplier, the augmented primal-dual decision variable is then defined as $u = \begin{bmatrix} \theta \\ \gamma \end{bmatrix} \in \mathbb{R}^{(n+m)}$, and its bounds $u^{\pm}$ are defined as

$$u^{+} = \begin{bmatrix} \zeta^{+} \\ \varpi 1_v \end{bmatrix} \in \mathbb{R}^{(n+m)}, u^{-} = \begin{bmatrix} \zeta^{-} \\ -\varpi 1_v \end{bmatrix} \in \mathbb{R}^{(n+m)} \tag{17}$$

where $\varpi \gg 0$ is a large constant, and $1_v = [1, ..., 1]^T$ is a vector of ones with appropriate dimensions. Then, the convex set is defined as $\Omega = \{u \in \mathbb{R}^{(n+m)} \mid u^{-} \leq u \leq u^{+}\} \subset \mathbb{R}^{(n+m)}$. The Lagrange function is then designed as

$$G(\theta, \gamma) = \frac{1}{2}\theta^T W\theta + z^T\theta + \gamma^T(Q\theta - b). \tag{18}$$

According to the Lagrangian theorem in [9], if the derivatives of $\frac{\partial G}{\partial \theta}$, $\frac{\partial G}{\partial \gamma}$ are continuous, by setting the partial derivatives of $G(\theta, \gamma)$ to zero, we can obtain the algebraic equations of the Lagrange necessary condition as

$$\begin{cases} \frac{\partial G}{\partial \theta} = W\theta + z + Q^T\gamma = 0, \\ \frac{\partial G}{\partial \gamma} = Q\theta - b = 0. \end{cases} \tag{19}$$

We can rewrite (19) as the following matrix form as

$$Hu = d \tag{20}$$

where $H = \begin{bmatrix} W & Q^T \\ Q & 0 \end{bmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$, $d = \begin{bmatrix} -z \\ b \end{bmatrix} \in \mathbb{R}^{(n+m)}$.

Through the above transformation, resolving the constrained QP problem in (14)–(16) is transformed into find the solution of matrix equation (20). Note that the optimal solution is obtained when the difference between both sides of (20) are equivalent to zero, the error can be defined as

$$\phi(t) = H(t)u(t) - d(t) \tag{21}$$

where $\phi(t) \in \mathbb{R}^{(n+m)}$. The variable $\phi$ is expected to be close to zero. Thus, the VPCDNN is proposed for (21):

$$\dot{\phi}(t) = -\lambda\alpha^{\nu t}P_{\Omega}(\phi(t)), \tag{22}$$

$$P_{\Omega}(\phi(t)) = \begin{cases} \phi_i^{-} & if \quad \phi_i < \phi_i^{-}, \\ \phi_i & if \quad \phi_i^{-} \leq \phi_i \leq \phi_i^{+}, \\ \phi_i^{+} & if \quad \phi_i > \phi_i^{+} \end{cases} \tag{23}$$

where the parameters $\lambda$, $\alpha$ and $\nu$ are the positive constants to scale the convergence rate, and $P_{\Omega}$ represents the monotonically increasing odd activation function. Substituting (21) into (22), the following implicit-dynamic equation of the QP solver can be obtained:

$$H(t)\dot{u}(t) = -\dot{H}(t)u(t) - \lambda\alpha^{\nu t}P_{\Omega}(H(t)u(t) - d(t)) + \dot{d}(t). \tag{24}$$

Finally, equation (24) can be rewritten as

$$\begin{aligned} \dot{u}(t) = (I - H(t))\dot{u}(t) - \dot{H}(t)u(t) \\ - \lambda\alpha^{\nu t}P_{\Omega}(H(t)u(t) - d(t)) + \dot{d}(t). \end{aligned} \tag{25}$$

The $i$th-neural dynamic equation of equation (25) can be written as

$$\begin{aligned} \dot{u}_i = \sum_{j=1}^{n+m}(\eta_{ij} - h_{ij})\dot{u}_j - \sum_{j=1}^{n+m}\dot{h}_{ij}u_j \\ - \lambda\alpha^{\nu t}P_{\Omega}(\sum_{j=1}^{n+m}(h_{ij}u_j - d_i)) + \dot{d}_i \end{aligned} \tag{26}$$

where $u_i$, $d_i$ are the $i$-th elements of $u$, $d$ respectively, and $h_{ij}$, $\eta_{ij}$ are the $i$-th row and the $j$-th column elements of $H$, $I$ respectively. Now, we can obtain the following theorem.

*Theorem 1:* If there exists an optimal solution $\theta^*$ to the constrained QP problem (14)–(16) while the monotone non-decreasing activation function $P_{\Omega}$ acts on the error $\phi$, the state vector $u(t) = [\theta^T(t), \gamma^T(t)]$ of VPCDNN can globally converge to the equilibrium point $u^*(t) = [\theta^{*T}(t), \gamma^{*T}(t)]$ beginning with any original state $u(0) \in \mathbb{R}^{n+m}$, and the optimal solution $\theta^*$ to the QP problem is composed of the first $n$ elements of $u^*(t)$.

*Proof*: Considering the Lyapunov function as

$$B(t) = \frac{1}{2}\phi^T(t)\phi(t). \tag{27}$$

It is obvious that $B(t) = 0$ if and only if $\phi(t) = 0$, $B(t) > 0$ when $\phi(t) \neq 0$. The time derivative of $B(t)$ can be obtained as

$$\dot{B}(t) = \frac{dB(t)}{dt} = \phi^T(t)\dot{\phi}(t). \tag{28}$$

Substituting (22) into (28), we obtain

$$\dot{B}(t) = -\lambda\alpha^{\nu t}\phi^T(t)P_\Omega(\phi(t)) = -\lambda\alpha^{\nu t}\sum_{i=1}^{n+m}\phi_i(t)P_\Omega(\phi_i(t)) \tag{29}$$

where $\phi_i(t)$ represents the $i$th element of variable $\phi(t)$, and $P_\Omega(\phi_i(t)))$ is the $i$th projection element of variable $P_\Omega(\phi(t))$. Then, we have the following equation because of the monotone non-decreasing property of $P_\Omega(\phi(t))$:

$$\phi_i(t)P_\Omega(\phi_i(t))) = \begin{cases} >0, & if \quad \phi_i(t)\neq 0, \\ =0, & if \quad \phi_i(t)=0. \end{cases} \tag{30}$$

Since $\lambda$, $\alpha$ and $\nu$ are the positive constants, we can obtain

$$\dot{B}(t) = \begin{cases} <0, & if \quad \phi_i(t)\neq 0, \\ =0, & if \quad \phi_i(t)=0. \end{cases} \tag{31}$$

According to (31), we can see that $\dot{B}(t)=0$ if and only if $\phi(t)=0$, $\dot{B}(t)<0$ when $\phi(t)\neq 0$. Therefore, $u(t)-u^*(t)$ globally converges to 0 based on the Lyapunov theory [33], which also means $\theta(t)-\theta^*(t)$ globally converges to zero. Thus, the proof of Theorem 1 is finished.

*Remark 1:* In general, quadratic programming can be employed to solve (14)–(16). The minimum computational cost of a QP optimization method is generally proportional to the dimension of the decision vector. For the traditional quadratic programming numerical algorithm, such as the interior point method [34], it is necessary to repeatedly calculate the Hessian matrix of Lagrangian. In order to compute the solution of the QP problem in (14)–(16), the interior point method includes $O(n^4+n+(n+m)n^2+(2n+m)^3)$ operations. While the VPCDNN requires performing $8(n+m)$ additions/subtractions, $3(n+m)^2$ multiplications, $(n+m)$ limiters, and $3(n+m)$ integrators, so it only includes $O(12(n+m)+3(n+m)^2)$ operations. Obviously, the VPCDNN optimization method greatly reduces the computational cost.

## IV. Repetitive Learning Control

### A. Control Design

The robotic leg prosthesis needs to track the repetitive motion in task space with a fixed period of time under the task of assisting the amputees to walk on the level ground. Therefore, the given desired joint trajectories, which can be represented as $q_d, \dot{q}_d, \ddot{q}_d$, need to be bounded and periodic with a cycle period $T$:

$$q_d(t+T)=q_d(t), \dot{q}_d(t+T)=\dot{q}_d(t), \ddot{q}_d(t+T)=\ddot{q}_d(t). \tag{32}$$

The control objective of the controller is converge the tracking errors with all signals in the closed-loop robotic system remain bounded.

In order to design this controller, the position tracking error is defined as

$$e_p = q - q_d. \tag{33}$$

Then, the reference velocity error is defined as

$$e_s = \dot{q} - \dot{q}_r \tag{34}$$

where $q_r$ is the reference position defined as

$$q_r = q_d - \Lambda\int_0^t e_p\ dt \tag{35}$$

with a constant matrix $\Lambda$ whose real part eigenvalues are positive strictly. Then, we have

$$\dot{q}_r = \dot{q}_d - \Lambda e_p. \tag{36}$$

Thus, the reference velocity error can be represented as

$$e_s = \dot{e}_p + \Lambda e_p. \tag{37}$$

Using (34), the tracking error dynamics of the system (3) is given by

$$M(q)\dot{e_s} + C(q,\dot{q})e_s = \tau(t) - F(q,\dot{q}) - D(t) \tag{38}$$

where

$$F(q,\dot{q}) = M(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + G(q) + E(q) + V(\dot{q}). \tag{39}$$

Obviously, $F(q,\dot{q})$ is an unknown function of $\dot{q}_r$ and $\ddot{q}_r$. Meanwhile, the unknown period disturbances $D(t)$, which satisfies $D(t+T)=D(t)$, are ubiquitous in the robotic systems under repetitive motion task and assumed to be bounded with a certain constant. Hence, if the truth values of $F(q,\dot{q})$ and $D(t)$ can be obtained, we can get an ideal control law of robotic leg prostheses for tracking the repetitive motion.

On the one hand, it should be noted that $M(q), C(q,\dot{q}), G(q), E(q)$ and $V(\dot{q})$ in (3) can be viewed as impedance dynamics since these terms are related to the stiffness and damping of prosthetic joints. Then, there exists the following property [26]:

*Property 3:* For $\forall q \in \mathbb{R}^2$ and $\forall \dot{q} \in \mathbb{R}^2, \exists\ l_w > 0\ (w=1,2,3,4)$ such that $\|M(q)\| \leqslant l_1, \|F(q,\dot{q})\| = \|M(q)\ddot{q}_r + C(q,\dot{q})\dot{q}_r + G(q) + E(q) + V(\dot{q})\| \leqslant L^T\Phi = l_1 + l_2\|\dot{q}\| + l_3\|\dot{q}\|^2 + l_4\|\dot{q}_r|$ with $L=[l_1,l_2,l_3,l_4]^T$ and $\Phi=[1,\|\dot{q}\|,\|\dot{q}\|^2,\|\dot{q}_r\|]^T$ .

Let $L^*$ represents the real value of stiffness and damping coefficient $L$ whose value is learned as the estimated value $\hat{L}$. Obviously, the estimated value $\hat{L}$ is expected to approximate the real value $L^*$.

On the other hand, in order to approximate the real value of the feed-forward compensate torque for periodic disturbance $D(t)$, we estimate $D_i(t)$, the $i$th component of $D(t)$ in the $i$th joint, by a linear combination. Each term in the linear combination is the product of a shape function $\xi_j(t)$ and a corresponding constant coefficient $\kappa_j^i$. The approximation can be expressed as

$$D_i(t) = \sum_{j=0}^{N}\kappa_j^i\xi_j(t) \tag{40}$$

where the unknown constant coefficient $\kappa_j^i$ of each shape function $\xi_j(t)$ for each $D_i(t)$ is the quantity that can be estimated in repetitive learning control, and $N$ represents the selected numbers of shape functions. Note that $D(t)$ is periodic and continuous, according to the Stone-Weirstrass theorem in [35], there exit abundant shape functions that can be chosen to approximate $D(t)$. One option is defined below:

*Definition 1:* Assume that $S(T)$ represents the space of continuous and periodic functions whose cycle period is $T$,

consider a countable set of $\{\xi_j \in S(T)\}$ which is linearly independent such that:

1) unity can be expressed as a linear combination includes finite terms $\xi_j$,

2) the span of $\{\xi_j\}$ is dense in $S(T)$, i.e., for any $D \in S(T)$ and $\zeta > 0$, there exist $\kappa_j \in \mathbb{R}^n$ and $N$ such that

$$\sup_{t\in[0,T]} |D(t) - \sum_{j=0}^{N} \kappa_j \xi_j(t)| < \zeta. \tag{41}$$

A piecewise linear function fits Definition 1 is chosen as the shape function in this work. It has an advantage in computational speed because only two parameters need to be updated at the same time. Assuming that the $j$th piece linear function be

$$\hat{D}_{i_j}(t) = s_1 t_l + s_0 \tag{42}$$

where $t_l$ represents the local time of the $j$th piece linear function. In order to depict the chosen piecewise linear function, the following notation is used:

$$\beta_j \equiv \frac{t}{T} N - j. \tag{43}$$

Then, the shape functions $\xi_j(t)$ for $t \in [0, T]$ can be expressed by

$$\xi_j(t) = \begin{cases} 1 - \beta_j, & if \quad 0 \leq \beta_j < 1, \\ 1 + \beta_j, & if \quad -1 \leq \beta_j < 0, \\ 0, & else, \end{cases} \tag{44}$$

$$\xi_j(t + cT) = \xi_j(t) \tag{45}$$

where $c$ is a positive integer that represents the number of cycles. Let $\kappa_j^*$ represents the real value of the coefficients $\kappa_j$ corresponding to the shape functions $\xi_j(t)$ and it is learned as the estimated value $\hat{\kappa_j}$. The estimated value $\hat{\kappa}_j$ is also expected to approximate the real value $\kappa_j^*$.

Now, the repetitive learning controller which determines the joint torque $\tau(t)$ can be proposed according to Property 3 and (40):

$$\tau = -K_p e_p - K_s e_s - \mathrm{sgn}(e_s)\hat{L}^T \Phi + \sum_{j=0}^{N} \hat{\kappa}_j \xi_j(t) \tag{46}$$

where $K_p$, $K_s$ are the positive gain matrixes, and $\mathrm{sgn}(e_s) = \frac{e_s}{\|e_s\|}$ is denoted as the signum function. The estimated $\hat{L}$ is online updated according to the following learning law designed as

$$\dot{\hat{L}} = -\Psi(h(t)\hat{L}(t) - \Phi\|e_s\|) \tag{47}$$

where $\Psi$ is the positive constant corresponding to the learning rate of $L^*$, and $h(t)$ is an optional function of time which should satisfy $h(t) > 0$, $\lim_{t\to\infty} h(t) = 0$ and $\int_0^\infty h(s)ds = a < \infty$. Meanwhile, the estimated $\hat{\kappa_j}$ are online updated according to the following learning law designed as

$$\dot{\hat{\kappa}}_j = -\Theta_j \xi_j(t) e_s \tag{48}$$

where $\Theta_j$ are the positive constants corresponding to the learning rates of $\kappa_j^*$.

The proposed controller (46) composes of three parts: 1) The first term and the second term, it is a feedback law of tracking errors and has constant gains; 2) The third term is utilized to estimate impedance dynamics parameters; 3) The fourth term is adopted for compensating the periodic disturbances. While the robotic leg prosthesis tracking the desired repetitive trajectories, the real value of estimated impedance dynamics parameters $L^*$ are learned by (47), and the real value of estimated coefficient vector $\kappa_j^*$ associated with the feedforward disturbance compensation $D(t)$ is learned using (48).

### *B. Stability Analysis*

We are ready to derive the following stability theorem for our proposed controller.

*Theorem 2:* Consider the dynamics (3) with desired repetitive motion (32) under the proposed controller (46) with the repetitive learning laws (47) and (48), from each origin compact set $\Upsilon(0) = (q(0), \dot{q}(0), \hat{L}(0), \hat{\kappa}_j(0))$, the tracking errors $e_p$ and $e_s$ can converge to zero and all signals are bounded.

*Proof*: Considering the Lyapunov stability, we choose the following Lyapunov function as

$$V = V_p + V_s + V_l + V_d, \tag{49}$$

$$V_p = \frac{1}{2} e_p^T K_p e_p, \tag{50}$$

$$V_s = \frac{1}{2} e_s^T M(q) e_s, \tag{51}$$

$$V_l = \frac{1}{2} \int_{t-T}^{t} \tilde{L}^T(s) \Psi^{-1} \tilde{L}(s) ds, \tag{52}$$

$$V_d = \frac{1}{2} \sum_{j=0}^{N} \tilde{\kappa}_j^T \Theta_j^{-1} \tilde{\kappa}_j \tag{53}$$

where $\tilde{L} = \hat{L} - L^*$, and $\tilde{\kappa}_j = \hat{\kappa}_j - \kappa_j^*$. Firstly, we can obtain the time derivative of (50) as

$$\dot{V}_p = -\Lambda e_p^T K_p e_p + e_p^T K_p e_s. \tag{54}$$

Then, $\dot{V}_s$ can be obtained by differentiating (51), substituting (46) into (38), and then using Property 2:

$$\begin{aligned} \dot{V}_s &= \frac{1}{2} e_s^T \dot{M}(q) e_s + e_s^T M(q) \dot{e}_s \\ &= \frac{1}{2} e_s^T (\dot{M}(q) - 2C(q, \dot{q})) e_s + e_s^T (-K_p e_p - K_s e_s \\ &\quad - \mathrm{sgn}(e_s)\hat{L}^T \Phi + \sum_{j=0}^{N} \hat{\kappa}_j \xi_j(t) - F(q, \dot{q}) - D(t)) \\ &\leqslant -e_s^T K_p e_p - e_s^T K_s e_s - \|e_s\| \tilde{L}^T \Phi + e_s^T \sum_{j=0}^{N} \tilde{\kappa}_j \xi_j(t). \end{aligned} \tag{55}$$

Differentiating (52) and utilizing the learning law (47), $\dot{V}_l$ can be obtained as

$$\begin{aligned} \dot{V}_l &= \frac{1}{2\Psi} \tilde{L}^T(t) \tilde{L}(t) - \frac{1}{2\Psi} \tilde{L}^T(t - T) \tilde{L}(t - T) \\ &= \frac{1}{2\Psi} (\tilde{L}(t) - \tilde{L}(t - T))^T (\tilde{L}(t) + \tilde{L}(t - T)) \\ &= \frac{1}{2\Psi} \dot{\tilde{L}}^T (2\tilde{L}(t) - \dot{\tilde{L}}) = \frac{1}{\Psi} \dot{\tilde{L}}^T \tilde{L}(t) - \frac{1}{2\Psi} \dot{\tilde{L}}^T \dot{\tilde{L}} \\ &= \Phi\|e_s\| \tilde{L} - \frac{1}{2\Psi} \dot{\tilde{L}}^T \dot{\tilde{L}} + h(t) \hat{L}^T(t)(L^* - \hat{L}(t)) \end{aligned}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2021.3071936, IEEE/ASME Transactions on Mechatronics



$$
\begin{aligned}
&= \Phi\|e_s\|\tilde{L} - \frac{1}{2\Psi}\dot{\tilde{L}}^T\dot{\tilde{L}} + h(t)(\frac{1}{4}L^{*T}L^* - \frac{1}{4}L^{*T}L^* \\
&+ \hat{L}^T(t)L^* - \hat{L}^T(t)\hat{L}(t)) \\
&\leqslant \Phi\|e_s\|\tilde{L} - \frac{1}{2\Psi}\dot{\tilde{L}}^T\dot{\tilde{L}} + \frac{1}{4}h(t)L^{*T}L^*. \qquad (56)
\end{aligned}
$$

Similarly, $\dot{V}_d$ can be obtained from differentiating (53) and utilizing the learning law (48):

$$
\dot{V}_d = \sum_{j=0}^{N}\tilde{\kappa}_j^T\Theta_j^{-1}\dot{\tilde{\kappa}}_j \quad = -\sum_{j=0}^{N}\tilde{\kappa}_j^T\xi_j(t)e_s. \qquad (57)
$$

Therefore, $\dot{V}$ can be obtained by differentiating (49) and merging (54) – (57):

$$
\begin{aligned}
\dot{V} &= \dot{V}_p + \dot{V}_s + \dot{V}_l + \dot{V}_d \\
&\leqslant -\Lambda e_p^T K_p e_p + e_p^T K_p e_s - e_s^T K_p e_p - e_s^T K_s e_s \\
&- \|e_s\|\tilde{L}^T\Phi + e_s^T\sum_{j=0}^{N}\tilde{\kappa}_j\xi_j(t) + \Phi\|e_s\|\tilde{L} \\
&- \frac{1}{2\Psi}\dot{\tilde{L}}^T\dot{\tilde{L}} + \frac{1}{4}h(t)L^{*T}L^* - \sum_{j=0}^{N}\tilde{\kappa}_j^T\xi_j(t)e_s \\
&\leqslant -\Lambda e_p^T K_p e_p - e_s^T K_s e_s + \frac{1}{4}h(t)L^{*T}L^*.
\end{aligned} \qquad (58)
$$

Since $\Lambda, K_p, K_s$ are selected as positive constant matrix and $\lim_{t\to\infty}(\frac{1}{4}h(t)L^{*T}L^*) = 0$ which is determined by the properties of $h(t)$, we can obtain $\dot{V} < 0$ when $e_p, e_s \neq 0$ as $t \to \infty$. According to the Lyapunov stability criterion and the Barbalat's lemma [36], $e_p$ and $e_s$ can converge to 0 as $t \to \infty$. Integrating both sides of (58), we have

$$
\begin{aligned}
V(t) - V(0) \leqslant &- \int_0^t(\Lambda e_p^T K_p e_p + e_s^T K_s e_s)ds \\
&+ \int_0^t(\frac{1}{4}h(t)L^{*T}L^*)ds.
\end{aligned} \qquad (59)
$$

Since $L^*$ is a constant vector and $\int_0^\infty h(s)ds = a$, we can obtain

$$
\begin{aligned}
V(t) - V(0) \leqslant &- \int_0^t(\Lambda e_p^T K_p e_p + e_s^T K_s e_s)ds \\
&+ \frac{a}{4}L^{*T}L^* < \infty.
\end{aligned} \qquad (60)
$$

Thus, $V$ is bounded. Since $q_d, \dot{q}_d$ are bounded and $e_p, e_s$ converge to zero as $t \to \infty$, we can obtain that $q, \dot{q}, \dot{q}_r$ are also bounded. Besides, since $l_w^*$ $(w = 1, 2, 3, 4)$ and $\kappa_j^*$ $(j = 0, 1, ..., N)$ are constants, we can conclude that $\hat{p}_w$ and $\hat{\kappa}_j$ are bounded. Hence, all signals in the closed-loop dynamics of robotic leg prosthetic system are bounded. The proof of Theorem 2 has been finished.

## V. Experiment Verification

### A. Experimental Setup

A 36-year-old male (172 cm, 55 kg, 4 years post-amputation) unilateral amputee subject shown in Fig. 3a was recruited through local medical institution. The calf of his healthy leg is 0.4 m long, and the length from his ankle joint to the end of the metatarsal bone is 0.12 m. The subject uses a passive prosthetic leg developed by the Ottobock company in his daily life. The informed consent was obtained from the subject before the experiment, and the ethic application was authorized by Institutional Review Board of the University of Science and Technology of China.

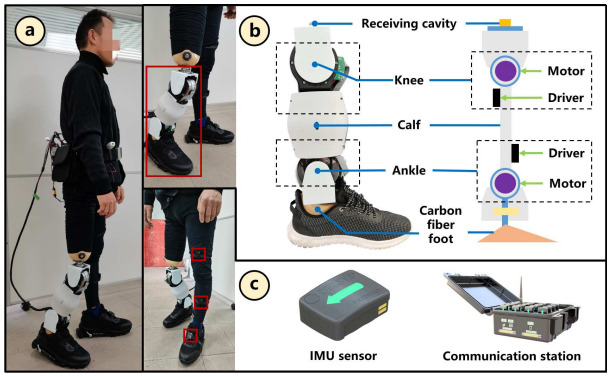



Fig. 3. Experimental setup. (a) The recruited unilateral transfemoral amputee subject. (b) The structure of the 2-DOF robotic leg prosthesis developed for test verification. (c) The IMU sensors mounted on the subject's healthy leg for collecting the level walking data.

TABLE I
THE DEVICES USED IN THE EXPERIMENTS

| Device | Brand | Type | Amount |
|---|---|---|---|
| Motor | Maxon | EC 90 flat 607322 | 2 |
| Reducer | HarmonicDrive | SHD-20-160-2SH | 2 |
| Encoder | Maxon | Encoder MILE 607515 | 2 |
| Servo Driver | Elmo | WHI-A20/60 | 2 |
| CAN Bus | Kvaser | Leaf Light v2 | 1 |
| Control Board | LattePanda | Delta | 1 |
| IMU | Delsys | Trigno Wireless | 3 |

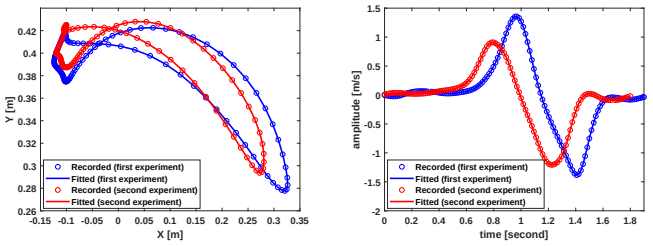
The structure of the 2-DOF robotic leg prosthesis developed for test verification is shown in Fig. 3b. The prosthesis is constructed with aluminum alloy and nylon fiber. The total weight of the prosthesis is 4.2 kg. The length of the robotic leg prosthesis can be adjusted from 0.35 m to 0.6 m. The flexion range of the knee joint is 0° to 110°. The maximum planterflexion and dorsiflexion of the ankle joint are both 45°.

The devices used in the experiments have been listed out in Table I. The two joints are driven by the direct current brushless motors (nominal voltage: 24 V; nominal current: 6.39 A; nominal torque: 457 mNm) and the harmonic reducers (reduction: 160:1). The encoders (counts per turn: 2048; maximum speed: 5000 rpm) and the servo drivers (maximum current: 20 A; maximum voltage: 60 V) connect with the control board (LattePanda Delta board which has the Intel(R) Celeron(TM) N4100 CPU @ 1.1 GHz and 4 GB memory) via a CAN bus (support 8000 messages/s). As shown in Fig. 3c, the Inertial Measurement Unit (IMU) sensors (Delsys Trigno Wireless sensors) are mounted on the thigh, calf, and instep of the subject's healthy leg in the experiments to collect the level walking data. The proposed methods are realized by C++ programming and run on the control board with a sampling interval of 10 ms.

Before the experiments, the desired joint trajectories of prosthesis in a gait cycle were predefined based on the previous

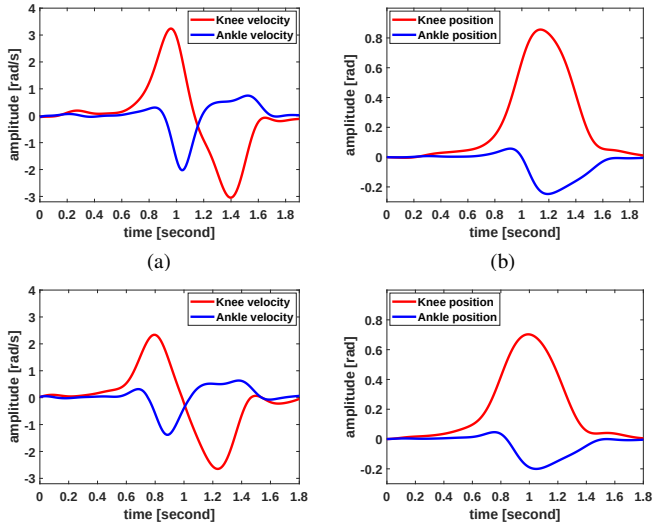
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2021.3071936, IEEE/ASME Transactions on Mechatronics





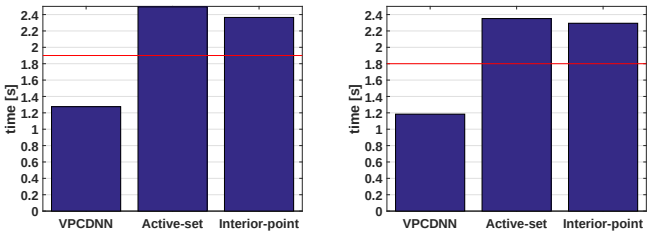

(a) (b)

Fig. 4. The level walking data in task space obtained from the healthy leg in the first stage. (a) The foot's position relative to the knee. (b) The foot's horizontal velocity.




(a) (b) (c) (d)

Fig. 5. The generated joint trajectories using proposed VPCDNN in the first stage. (a) The velocity trajectories generated in the first experiment. (b) The position trajectories generated in the first experiment. (c) The velocity trajectories generated in the second experiment. (d) The position trajectories generated in the second experiment.




(a) (b)

Fig. 6. The calculation time of the proposed VPCDNN compared with that of the active set method and interior point method. (a) The calculation time in the first experiment. (b) The calculation time in the second experiment.

collected gait data. Then, two experiments were carried out while the subject wearing the developed robotic leg prosthesis, and each experiment was divided into two stages. The first stage included one gait cycle. At the beginning of the gait cycle, the prosthesis began to move from the swing phase with the predefined desired joint trajectories, the healthy leg began to move from the support phase at a self-selected walking speed, and the IMU sensors began to collect the level walking data from the healthy leg. During the gait cycle, based on the desired walking speed collected from healthy leg, the prosthesis used the proposed neural-dynamics optimization method to online generate the repetitive joint trajectories which were used as the desired joint trajectories in the second stage. The second stage consisted of 10 consecutive gait cycles. During the 10 gait cycles, the prosthesis used the proposed repetitive learning control to track the desired joint trajectories generated in the first stage, and the control performance of the proposed controller was observed.

### *B. Experimental Results and Discussions*

*1) Level Walking Data:* The raw data of joint position and velocity collected from the IMU sensors during level walking are filtered using the 4th order Butterworth low pass filter, and the moving-average filtering are performed on the sampled data. The recorded data in joint space are processed by the human lower limb kinematics to obtain the position and horizontal velocity of the healthy foot relative to the knee in task space.

Fig. 4 shows the level walking data in task space obtained from the healthy leg in the first stage. As shown in Fig. 4a, the foot's position trajectories relative to the knee are closed paths, which prove that the level walking can be seen as the repetitive motion of human legs indeed. The foot's horizontal velocity shown in Fig. 4b are used as the subject self-selected walking speed to be duplicated in the neural-dynamics optimization method (the desired walking speed $b$ in (15)) in the first stage.

*2) Neural-dynamics Optimization:* The proposed VPCDNN is employed to online generate the repetitive joint trajectories in the first stage based on the desired walking speed collected from healthy leg. At one gait cycle, the prosthesis should start from initial position $q(0) = [0, 0]^T rad$ and return to the original position finally. We choose the limits of joint position and joint velocity as $[0, -\pi/4]^T rad$ to $[11\pi/18, \pi/4]^T rad$ and $[-3, -2]^T rad/s$ to $[4, 2]^T rad/s$, respectively. The vertical velocity limit bounds are selected as $\dot{r}_{ylb} = -0.8m/s$ and $\dot{r}_{yub} = 0.8m/s$. Besides, the parameters $\mu = 1.0$, $k = 4.0$, $\lambda = 2.0$, $\alpha = e$ and $\nu = 1.0$ are selected. $\delta$ is an important parameter to scale the prosthetic response magnitude to the joint drift, we find that satisfactory drift-free result can be achieved with $\delta = 50$ after the several actual verification.

According to Fig. 4b, the cycle time of the joint trajectories to be generated in the first and second experiment are 1.9 s and 1.8 s, respectively. The two cycle time were divided into 190 and 180 intervals, respectively, then the time length of each interval equals to the sampling time (10 ms). The optimal joint velocity vector $\dot{q}^*(w)(w = 1, 2, 3, ...)$ is solved in the $w$th interval and is used for calculating the joint position vector $q(w + 1)$ of the next $(w + 1)$th interval. Therefore, there are 190 and 180 QP problems in (14)–(16) need to be solved in the first stage of the first and second experiment, respectively. As shown in Figs. 5b and 5d, it can be observed that the final positions of the generated position trajectories coincide with their initial positions in joint space. In addition, under the same conditions, the calculation time of the proposed VPCDNN is compared with that of the traditional QP numerical optimization methods include active set method and interior point method. Although the other two traditional methods can generate the same repetitive joint trajectories

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMECH.2021.3071936, IEEE/ASME Transactions on Mechatronics



shown in Fig. 5, the proposed VPCDNN has the shortest calculation time as shown in Fig. 6. Considering the system has hard constraints of sampling time of 10 ms, the red line in Fig. 6 represents the maximum calculation time required to fulfill the real-time requirement in the first stage. Thus, the proposed VPCDNN method can satisfy the hard real-time requirement in our developed control system while the active set method and interior point method cannot satisfy the hard real-time requirement for our robotic leg prosthesis to online generate the repetitive joint trajectories.

*3) Repetitive Learning Control:* The repetitive joint trajectories shown in Figs. 5a, 5b and Figs. 5c, 5d generated in the first stage are used as the desired joint trajectories in the second stage. The coefficient matrix in (35) is given by $\Lambda$ = diag [1, 1]. The control gains in (46) are chosen as $K_p$ = diag [10, 34] and $K_s$ = diag [10, 2]. The number of shape functions is selected as $N = 9$. In the learning law (47), $\Psi$= 0.02, and $h(t) = 1/(1+t)^2$. In the learning law (48), $\Theta_j$ in the two joints are all set to 5. In order to verify the performance of the proposed controller, a controller proposed in our previous work [37] is used for comparison. The position and velocity error gains of the compared controller in [37] are also set to the same values as the proposed repetitive learning controller.

The control performance in joint space using the proposed controller in the second stage of the first experiment is shown in Fig. 7. As shown in Figs. 7a and 7e, the knee and ankle position tracking errors continuously decrease at each cycle as time progresses when using the proposed controller. Whereas, the joint position tracking errors in Figs. 7b and 7f are almost the same at each cycle when using the controller in [37]. It can be observed from Figs. 7c, 7d and Figs. 7g, 7h that the joint position tracking errors can be reduced remarkably after 10 cycles of learning when using the proposed repetitive learning controller. The consistent results and conclusions can be observed and drew from the control performance in joint space in the second stage of the second experiment shown in Fig. 8. The foot's horizontal velocity shown in Fig. 4b are used as the desired walking speed trajectories, i.e., the desired horizontal velocity trajectories of the prosthesis shown as blue solid lines in Figs. 9a and 9c. From Figs. 9b and 9d, it can be illustrated that the control objective of duplicating the walking speed of the amputee's own healthy leg with the prosthesis has been achieved by using the proposed controller. Fig. 10 shows the gait of the amputee subject in an entire gait cycle in the experiments.

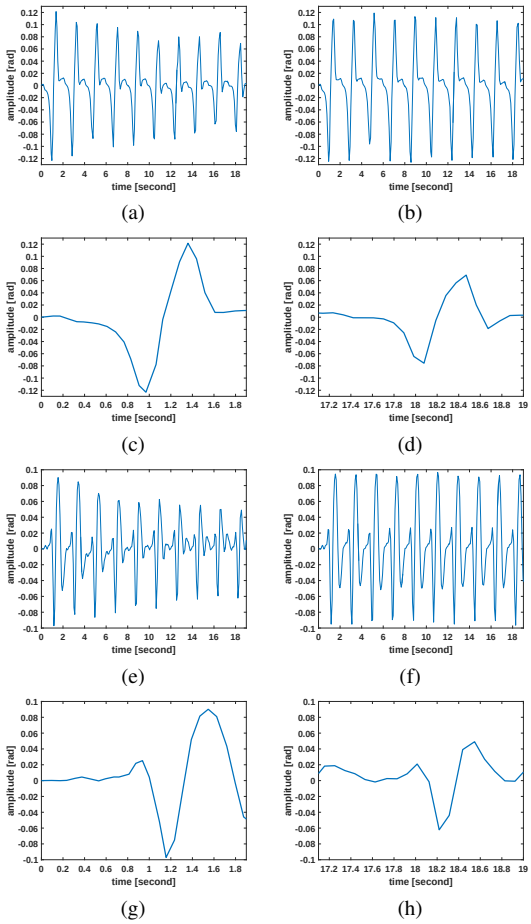



Fig. 7. The control performance in joint space using the proposed controller in the second stage of the first experiment. (a) Knee position tracking error. (b) Knee position tracking error using the controller in [37]. (c) Knee position tracking error at the first gait cycle. (d) Knee position tracking error at the last gait cycle. (e) Ankle position tracking error. (f) Ankle position tracking error using the controller in [37]. (g) Ankle position tracking error the first gait cycle. (h) Ankle position tracking error the last gait cycle.

## VI. Conclusions

In this paper, we have realized the online motion generation for robotic leg prostheses to solve the joint drift problem by using the neural-dynamics optimization. The online motion generation was formulated as a QP problem, and then found solution via the VPCDNN under the constraints in real-time. Meanwhile, the repetitive learning controller has been proposed for robotic leg prostheses to reduce the tracking errors while following the repetitive motion. The effectiveness and practicability of proposed neural-dynamics optimization and repetitive learning controller have been verified through the physical experiments based on a developed 2-DOF robotic leg prosthesis.

## References

[1] M. Goldfarb, B. E. Lawson, and A. H. Shultz, "Realizing the promise of robotic leg prostheses," *Sci. Tranl. Med.*, vol. 5, 2013.

[2] M. K. Shepherd, and E. J. Rouse, "The VSPA foot: a quasi-passive ankle-foot prosthesis with continuously variable stiffness," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 12, pp. 2375–2386, 2017.

[3] U.-J. Yang, and J.-Y. Kim, "Mechanical design of powered prosthetic leg and walking pattern generation based on motion capture data," *Adv. Robot.*, vol. 29, no. 16, pp. 1061–1079, Jul. 2015.

[4] F. Sup, H. A. Varol, J. Mitchell, T. 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.

[5] B. E. Lawson, J. Mitchell, D. Truex, A. Shultz, E. Ledoux, and M. Goldfarb, "A robotic leg prosthesis: Design control and implementation," *IEEE Robot. Autom. Mag.*, vol. 21, no. 4, pp. 70–81, Dec. 2014.

[6] P. Yang, H. Yue, L. Chen, and Y. Geng, "Intelligent lower limb prosthesis following healthy leg gait based on fuzzy control," in *Proc. IEEE 24th Chinese Control and Decision Conference*, pp. 3729-3731, 2012.

[7] D. Winter, *The Biomechanics and Motor Control of Human Gait: Normal, Elderly and Pathological*, 2nd ed. Waterloo, ON: Univ. Waterloo Press, 1991.

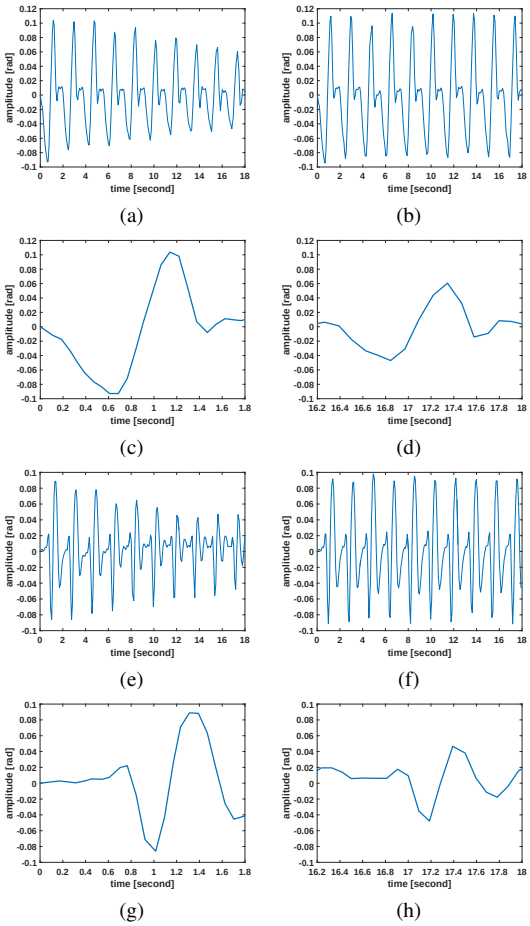



Fig. 8. The control performance in joint space using the proposed controller in the second stage of the second experiment. (a) Knee position tracking error. (b) Knee position tracking error using the controller in [37]. (c) Knee position tracking error at the first gait cycle. (d) Knee position tracking error at the last gait cycle. (e) Ankle position tracking error. (f) Ankle position tracking error using the controller in [37]. (g) Ankle position tracking error at the first gait cycle. (h) Ankle position tracking error at the last gait cycle.

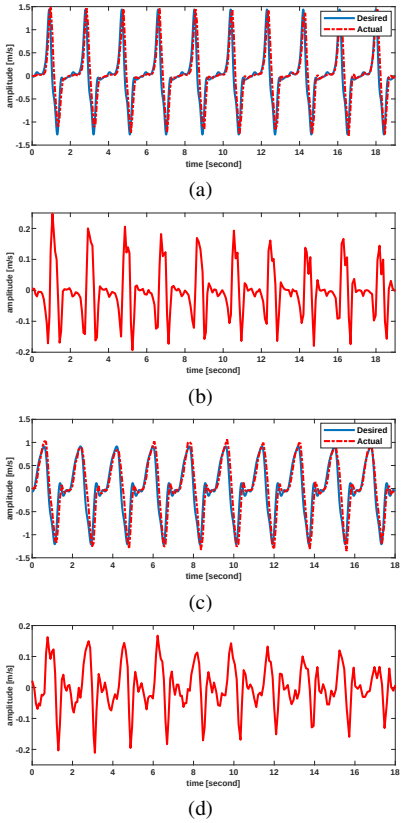



Fig. 9. The horizontal velocity tracking performance in task space using the proposed controller in the second stage. (a) The desired and actual horizontal velocity of the prosthesis in the first experiment. (b) The error of horizontal velocity between desired and actual trajectory in the first experiment. (c) The desired and actual horizontal velocity of the prosthesis in the second experiment. (d) The error of horizontal velocity between desired and actual trajectory in the second experiment.

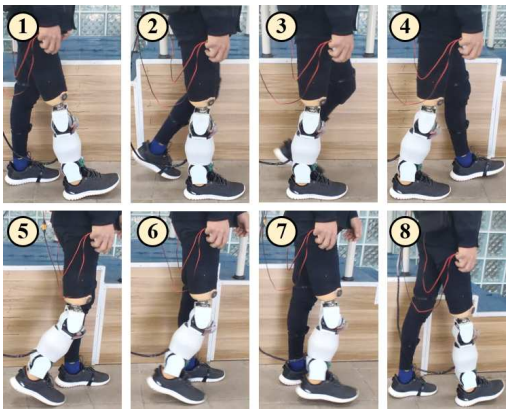



Fig. 10. The gait of the amputee subject in an entire cycle in the experiments.


[8] C. A. Klein, and K.-B. Kee, "The nature of drift in pseudoinverse control of kinematically redundant manipulators," *IEEE Trans. Robot. Autom.*, vol. 5, no. 2, pp. 231-234, Apr. 1989.

[9] S. Boyd, and L. Vandenberghe, *Convex Optimization.*, Cambridge, U.K.: Cambridge Univ. Press, 2004.

[10] J. Mattingley, and S. Boyd, *Automatic Code Generation for Real-Time Convex Optimization. Convex Optimization in Signal Processing and Communications*, Y. Eldar and D. Palomar, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[11] J. Mattingley, and S. Boyd, "CVXGEN: a code generator for embedded convex optimization," *Optimization and Engineering*, vol.13, no.1, pp.1–27, Mar. 2012.

[12] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *Int. J. Robust Nonlinear Control*, vol. 18, no. 8, pp. 816-830, 2008.

[13] R. A. Bartlett and L. T. Biegler, "QPSchur: A dual, active-set, Schur-complement method for large-scale and structured convex quadratic programming," *Optim. Eng.*, vol. 7, no. 1, pp. 5-32, 2006.

[14] A. Forsgren, P. Gill, and E. Wong, "Primal and dual active-set methods for convex quadratic programming," *Math. Program.*, vol. 159, pp. 469–508, 2016.

[15] A. Bemporad, "A quadratic programming algorithm based on nonnegative least squares with applications to embedded model predictive control," *IEEE Trans. Automat. Control*, vol. 61, no. 4, pp. 1111-1116, Apr. 2016.

[16] Z. Zhang, Z. Li, Y. Zhang, Y. Luo, and Y Li, "Neural-Dynamic-Method based Dual-Arm CMG Scheme with Time-Varying Constraints Applied to Humanoid Robots," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3251 – 3262, Dec. 2015.

[17] Z. Li, J. Deng, R. Lu, Y. Xu, J. Bai, and C.-Y. Su, "Trajectory Tracking Control of Mobile Robot Systems Incorporating Neural-dynamic Optimized Model Predictive Approach," *IEEE Trans. Syst., Man Cybern., Syst.*, vol. 46, no. 6, pp. 740 – 749, Jun. 2016.

[18] Z. Zhang, T. Fu, Z. Yan, L. Jin, L. Xiao, Y. Sun, Z. Yu, and Y. Li, "A varying parameter convergent-differential neural network for solving joint angular-drift problems of redundant robot manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 2, pp. 679 – 689, Apr. 2018.

[19] M. R. Tucker, J. Olivier, A. Pagel, H. Bleuler, M. Bouri, O. Lambercy, J. D. R. Millan, R. Riener, H. Vallery, and R. Gassert, "Control strategies for active lower extremity prosthetics and orthotics: a review," *J. Neuroeng. Rehabil.*, vol. 12, no. 1, pp. 1–29, 2015.

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

[21] 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, 2014.

[22] D. Quintero, D. J. Villarreal, D. Lambert, S. Kapp, and R. D. Gregg,

"Continuous-phase control of a powered knee-ankle prosthesis: Amputee experiment sacross speeds and inclines," *IEEE Trans. Robot.*, vol. 34, no. 3, pp. 686–701, Jun. 2018.

[23] Y. Lin, Z. Chen, and B. Yao, "Decoupled torque control of series elastic actuator with adaptive robust compensation of time-varying load-side dynamics," *IEEE Trans. Ind. Electron.*, vol. 67, no. 7, pp. 5604–5614, Jul. 2020.

[24] D. Huang, J. Xu, S. Yang, and X. Jin, "Observer based repetitive learning control for a class of nonlinear systems with non-parametric uncertainties," *Int. J. Robust Nonlinear Control*, vol. 25, pp. 1214-1229, 2015.

[25] J. Xu, and R. Yan, "On repetitive learning control for periodic tracking tasks," *IEEE Trans. Autom. Control*, vol. 51, no. 11, pp. 1842-1848, Nov. 2006

[26] R. Lu, Z. Li, C. Su, and A. Xue, "Development and learning control of a human limb with a rehabilitation exoskeleton," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3776-3785, Jul. 2014.

[27] Y. Yang, L. Ma, and D. Huang, "Development and repetitive learning control of lower limb exoskeleton driven by electro-hydraulic actuators," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4169–4178, May. 2017.

[28] Y. Yang, D. Huang, and X. Dong, "Enhanced neural network control of lower limb rehabilitation exoskeleton by add-on repetitive learning," *Neurocomputing*, vol. 323, pp. 256–264, Jan. 2019.

[29] N. Sharma, K. Stegath, C. M. Gregory, and W. E. Dixon, "Nonlinear neuromuscular electrical stimulation tracking control of a human limb," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 17, no. 6, pp. 576-584, Dec. 2009.

[30] A. Colome, "Closed-loop inverse kinematics for redundant robots: Comparative assessment and two enhancements," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 944-955, Apr. 2015.

[31] Y. Zhang, S. S. Ge, and T. H. Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Trans. Syst., Man, Cybern., Cybern.*, vol.34, no.5, pp. 2126–2132, Oct. 2004.

[32] Y. Zhang, Z. Tan, K. Chen, Z. Yang, and X. Lv, "Repetitive motion of redundant robots planned by three kinds of recurrent neural networks and illustrated with a four-link planar manipulator's straight-line example," *Robot. Auton. Syst.*, vol. 57, no. 6, pp. 645–651, Jun. 2009.

[33] E.A. Coddington, and N. Levinson, *Theory of ordinary differential equations*, New York, USA: McGraw-Hill, 1956.

[34] H. Wei, H. Sasaki, and R. Yokoyama, "An application of interior point quadratic programming algorithm to power system optimization problems," *IEEE Trans. Power Syst.*, vol. 11, no. 1, pp. 260–266, Feb. 1996.

[35] H. L. Royden, *Real Analysis.*, New York, USA: McMillan, 1968.

[36] J.J.E. Slotine, and W. Li, *Applied Nonlinear Control*, NJ, USA: Prentice Hall, 1991.

[37] M. Pi, Z. Li, Q. Li, Z. Kan, C. Xu, Y. Kang, C-Y. Su, and C. Yang, "Biologically Inspired Deadbeat Control of Robotic Leg Prostheses," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 6, pp. 2733–2742, Apr. 2020.

**Qinjian Li** received the B.S. degree in Automation from Nanjing Institute of Technology, Nanjing, Jiangsu, China, in 2015. He is currently working toward the Ph.D. degree in automation with the University of Science and Technology of China, Hefei, Anhui, China. His current research interests include the design and control of robotic leg prostheses and soft exosuits.

**Tao Zhang** received the B.S. degree in Automation from Xidian Univ., Xian, China, in 2019. He is currently working toward the Master degree in automation with the University of Science and Technology of China, Hefei, Anhui, China. His current research interests include the design and control of exoskeleton, robotic leg prostheses, and soft exosuits.

**Guoxin Li** received the B.S. degree in Mechanical Engineering and Automation from Hefei University of Technology, Hefei, Anhui, China, in 2016. He is currently working toward the Ph.D. degree in automation with the University of Science and Technology of China, Hefei, Anhui, China. His current research interests include human-machine interaction, myoelectric prostheses, and robotic systems.

**Zhijun Li** (M'07-SM'09) received the Ph.D. degree in mechatronics, Shanghai Jiao Tong University, P. R. China, in 2002. From 2003 to 2005, he was a postdoctoral fellow in Department of Mechanical Engineering and Intelligent systems, The University of Electro-Communications, Tokyo, Japan. From 2005 to 2006, he was a research fellow in the Department of Electrical and Computer Engineering, National University of Singapore, and Nanyang Technological University, Singapore. From 2017, he is a Professor in Department of Automation, University of Science and Technology, Hefei, China. From 2019, he is the Vice Dean of School of Information Science and Technology, University of Science and Technology of China, China.

From 2016, he has been the Co-Chairs of IEEE SMC Technical Committee on Bio-mechatronics and Bio-robotics Systems ($B^2S$), and IEEE RAS Technical Committee on Neuro-Robotics Systems. He is serving as an Editor-at-large of Journal of Intelligent & Robotic Systems, and Associate Editors of several IEEE Transactions. Dr. Li's current research interests include wearable robotics, tele-operation systems, nonlinear control, neural network optimization, etc.

**Haisheng Xia** received the B.S. in Mechatronics Engineering from Northwest AF University in 2014 and Ph.D. degree in Mechanical Engineering from Shanghai Jiao Tong University in 2020. He is currently a postdoctoral research fellow in the Department of Automation, University of Science and Technology of China. His research interests include wearable robotics with a focus on human gait modification and rehabilitation.

**Chun-Yi Su** (SM'99) received his Ph.D. degrees in control engineering from South China University of Technology in 1990. After a seven-year stint at the University of Victoria, he joined the Concordia University in 1998. Dr. Su conducts research in the application of automatic control theory to mechanical systems. He is particularly interested in control of systems involving hysteresis nonlinearities. Dr. Su is the author or co-author of over 300 publications, which have appeared in journals, as book chapters and in conference proceedings. Dr. Su has served as Associate Editor of IEEE Transactions on Automatic Control and IEEE Transactions on Control Systems Technology, and Journal of Control Theory and Applications. He is on the Editorial Board of 14 journals, including IFAC journals of Control Engineering Practice and Mechatronics. Dr. Su has also severed for many conferences as an organizing committee member, including General Co-Chair of the 2012 IEEE International Conference on Mechatronics and Automation, and Program Chair of the 2007 IEEE Conference on Control Applications.