

The Journal of Healthcare Engineering

# Robotics in Biomedical and Healthcare Engineering

Lead Guest Editor: Yajing Shen

Guest Editors: Uche Wejinya, Yasuhisa Hasegawa, Lianqing Liu, Qing Shi, and Chengzhi Hu





---

# **Robotics in Biomedical and Healthcare Engineering**

Journal of Healthcare Engineering

---

## **Robotics in Biomedical and Healthcare Engineering**

Lead Guest Editor: Yajing Shen

Guest Editors: Uche Wejinya, Yasuhisa Hasegawa, Lianqing Liu,  
Qing Shi, and Chengzhi Hu



---

Copyright © 2017 Hindawi. All rights reserved.

This is a special issue published in “Journal of Healthcare Engineering.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

---

## Editorial Board

S. Affatato, Italy  
W. Bertucci, France  
Tiago H. Falk, Canada  
Mostafa Fatemi, USA  
Joseph Finkelstein, USA  
Antonio Gloria, Italy  
Philippe Gorce, France  
Andreas H. Hielscher, USA  
Zhongwei Jiang, Japan

John S. Katsanis, Greece  
Feng-Huei Lin, Taiwan  
Maria Lindén, Sweden  
Francisco Lopez-Valdes, Spain  
Andreas Maier, Germany  
Rafael Morales, Spain  
David Moratal, Spain  
E. Ng Yin Kwee, Singapore  
J. Joaquin Rieta, Spain

H. A. Santos, Finland  
Emiliano Schena, Italy  
Maurizio Schmid, Italy  
Jiann-Shing Shieh, Taiwan  
Vinoy Thomas, USA  
Ioannis G. Tollis, Greece  
Jesús Favela Vara, Mexico

# Contents

---

## **Robotics in Biomedical and Healthcare Engineering**

Chengzhi Hu, Qing Shi, Lianqing Liu, Uche Wejinya, Yasuhisa Hasegawa, and Yajing Shen  
Volume 2017, Article ID 1610372, 2 pages

## **Hierarchical Shared Control of Cane-Type Walking-Aid Robot**

Chunjing Tao, Qingyang Yan, and Yitong Li  
Volume 2017, Article ID 8932938, 11 pages

## **Surgeon Training in Telerobotic Surgery via a Hardware-in-the-Loop Simulator**

Xiao Li, Homa Alemzadeh, Daniel Chen, Zbigniew Kalbarczyk, Ravishankar K. Iyer, and Thenkurussi Kesavadas  
Volume 2017, Article ID 6702919, 13 pages

## **Iterative Learning Impedance for Lower Limb Rehabilitation Robot**

Chenhui Guo, Shuai Guo, Jiancheng Ji, and Fengfeng Xi  
Volume 2017, Article ID 6732459, 9 pages

## **Development of a Prototype Robotic System for Radiosurgery with Upper Hemispherical Workspace**

Sun Young Noh, Kyungmin Jeong, Yong-chil Seo, Chang-hoi Kim, Jongwon Park, Yoo Rark Choi, Sung Uk Lee, Yeong-Geol Bae, and Seungho Kim  
Volume 2017, Article ID 4264356, 9 pages

## **Inertial Sensor-Based Motion Analysis of Lower Limbs for Rehabilitation Treatments**

Tongyang Sun, Hua Li, Quanquan Liu, Lihong Duan, Meng Li, Chunbao Wang, Qihong Liu, Weiguang Li, Wanfeng Shang, Zhengzhi Wu, and Yulong Wang  
Volume 2017, Article ID 1949170, 11 pages

## **Upper Limb Rehabilitation Robot Powered by PAMs Cooperates with FES Arrays to Realize Reach-to-Grasp Trainings**

Xikai Tu, Hualin Han, Jian Huang, Jian Li, Chen Su, Xiaobo Jiang, and Jiping He  
Volume 2017, Article ID 1282934, 15 pages

## **An Open-Structure Treadmill Gait Trainer: From Research to Application**

Jian Li, Diansheng Chen, and Yubo Fan  
Volume 2017, Article ID 9053630, 12 pages

## Editorial

# Robotics in Biomedical and Healthcare Engineering

**Chengzhi Hu,<sup>1</sup> Qing Shi,<sup>2</sup> Lianqing Liu,<sup>3</sup> Uche Wejinya,<sup>4</sup> Yasuhisa Hasegawa,<sup>5</sup> and Yajing Shen<sup>6</sup>**

<sup>1</sup>*Department of Mechanical and Process Engineering, ETH Zurich, Zurich, Switzerland*

<sup>2</sup>*Beijing Institute of Technology, Beijing, China*

<sup>3</sup>*Chinese Academy of Sciences, Shenyang, China*

<sup>4</sup>*University of Arkansas, Fayetteville, NC, USA*

<sup>5</sup>*Nagoya University, Nagoya, Japan*

<sup>6</sup>*Mechanical and Biomedical Engineering Department, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong*

Correspondence should be addressed to Yajing Shen; yajishen@cityu.edu.hk

Received 26 July 2017; Accepted 31 July 2017; Published 28 August 2017

Copyright © 2017 Chengzhi Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The rapid progress of robotic technique provides new opportunities for the biomedical and healthcare engineering. For instance, a micro-nano robot allows us to study the fundamental problems at cellular scale owing to its precise positioning and manipulation ability; the medical robot paves a new way for the low invasive and high efficient clinical operation; and rehabilitation robot is able to improve the rehabilitative efficacy of patients. This special issue aims at exhibiting the latest research achievements, findings, and ideas in the field of robotics in biomedical and healthcare engineering, especially focusing on the upper/lower limb rehabilitation, walking assistive robot, telerobotic surgery, and radiosurgery.

Currently, there is an increasing population of patients suffering from limb motor dysfunction, which can be caused by nerve injuries associated with stroke, traumatic brain injury, or multiple sclerosis. Past studies have demonstrated that highly repetitive movement training can result in improved recovery. The robotic-assisted technique is a novel and rapidly expanding technology in upper/lower limb rehabilitation that can enhance the recovery process and facilitate the restoration of physical function by delivering high-dose and high-intensity training. This special issue covers several interesting papers addressing these challenges. X. Tu and coworkers introduced an upper limb rehabilitation robot powered by pneumatic artificial muscles which cooperates with functional electrical stimulation arrays to realize active

reach-to-grasp training for stroke patients. The dynamic models of a pneumatic muscle and functional electrical stimulation-induced muscle are built for reaching training. By using surface electromyography, the subject's active intent can be identified. Finally, grasping and releasing behaviors can be realized by functional electrical stimulation array electrodes. C. Guo and coworkers proposed an impedance-based iterative learning control method to analyze the squatting training of stroke patients in the iterative domain and time domain. Patient's training trajectory can be corrected by integrating the iterative learning control scheme with the value of impedance. In addition, the method can gradually improve the performance of trajectory tracking by learning the past trajectory tracking information and obtain specific training condition of different individuals. The paper demonstrated an effective control methodology in dealing with repeated tracking control problems or periodic disturbance rejection problems. Apart from these works, J. Li and coworkers designed an open-structured treadmill gait trainer for lower limb rehabilitation; T. Sun and coworkers proposed a method for detecting the motion of human lower limbs including all degrees of freedoms via the inertial sensors, which permits analyzing the motion ability according to the rehabilitation needs.

Other biomedical and healthcare robots included in this special issue cover a range of interesting topics, such as walking assistive robot, telerobotic surgery, and radiosurgery. To

improve the walking ability of the elderly, the walker-type rehabilitation robot has become a popular research topic over the last decade. C. Tao and coworkers proposed a hierarchical shared control method of the walking-aid robot for both human motion intention recognition and the obstacle emergency-avoidance method based on the artificial potential field. In the implementation, the human motion intention is obtained from the interaction force measurements of the sensory system composed of force sensing registers and a torque sensor. Meanwhile, a laser-range finder forward is applied to detect the obstacles and try to guide the operator based on the repulsion force calculated by artificial potential field. The robot realizes obstacle avoidance while keeping partially the operators' original walking intention. X. Li and coworkers demonstrated a general framework for robot-assisted surgical simulators for a more robust and resilient robotic surgery. They created a hardware-in-the-loop simulator platform and integrated the simulator with a physics engine and a state-of-the-art path planning algorithm to help surgeons acquire an optimal sense of manipulating the robot instrumental arm. Eventually, they achieved autonomous motion of the surgical robot. For coping with the workspace issue during the application of Linac system during radiosurgery, a specialized robotic system was presented by Y. Noh et al. The design and implementation of the robotic system were elaborated. All of these works showed comparative advantages versus classical approaches and will hold great potential for providing insights on the practical and systematic design of robots that serve for broad applications in biomedical and healthcare engineering.

The objectives of the special issue were reached in terms of advancing the state of the art of robotic techniques and addressing the challenging problems in biomedical and healthcare engineering. Several critical problems in these areas were addressed, and most of the proposed contributions showed very promising results that outperform existing studies. Some of the proposed approaches were also validated from patients' perspectives, which show the applicability of these techniques in realistic environments.

## **Acknowledgments**

We would like to express our thanks to all the authors who submitted their work to this special issue and to all the reviewers who helped us ensure the quality.

*Chengzhi Hu  
Qing Shi  
Lianqing Liu  
Uche Wejinya  
Yasuhisa Hasegawa  
Yajing Shen*

## Research Article

# Hierarchical Shared Control of Cane-Type Walking-Aid Robot

Chunjing Tao,<sup>1</sup> Qingyang Yan,<sup>2</sup> and Yitong Li<sup>3</sup>

<sup>1</sup>Beijing Key Laboratory of Rehabilitation Technical Aids for Old-Age Disability, The National Research Centre for Rehabilitation Technical Aids, Beijing 100176, China

<sup>2</sup>Key Laboratory of Ministry of Education for Image Processing and Intelligent Control, School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China

<sup>3</sup>The 28th Research Institute of China Electronics Technology Group Corporation, Nanjing 210007, China

Correspondence should be addressed to Qingyang Yan; [yanqingyang@hust.edu.cn](mailto:yanqingyang@hust.edu.cn)

Received 5 January 2017; Revised 28 May 2017; Accepted 27 June 2017; Published 13 August 2017

Academic Editor: Chengzhi Hu

Copyright © 2017 Chunjing Tao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A hierarchical shared-control method of the walking-aid robot for both human motion intention recognition and the obstacle emergency-avoidance method based on artificial potential field (APF) is proposed in this paper. The human motion intention is obtained from the interaction force measurements of the sensory system composed of 4 force-sensing registers (FSR) and a torque sensor. Meanwhile, a laser-range finder (LRF) forward is applied to detect the obstacles and try to guide the operator based on the repulsion force calculated by artificial potential field. An obstacle emergency-avoidance method which comprises different control strategies is also assumed according to the different states of obstacles or emergency cases. To ensure the user's safety, the hierarchical shared-control method combines the intention recognition method with the obstacle emergency-avoidance method based on the distance between the walking-aid robot and the obstacles. At last, experiments validate the effectiveness of the proposed hierarchical shared-control method.

## 1. Introduction

With the development of society, the aging of population has caused more and more important social problems [1]. The elderly are faced with the problems such as weakening of physical ability and visual deterioration as they grow older. Thus, the elderly need to improve their walking ability to cope with their basic needs of daily life indeed. Opportunely, many researchers have paid attention to the applications of the robot and medical technology in recent years [2–10]. They manufactured many intelligent rehabilitation aids (e.g., walking-aid robots) to help the elderly to gain the ability of normal walking and developed a series of achievements, such as Kawamoto et al. invented “HAL” [5]. A wearable Power-Assist Locomotor (WPAL) was invented for the lower limb [6]. Kikuchi et al. [7] proposed an intelligently controllable walker (i-walker). Hirata et al. invented a passive intelligent walker called “RT-Walker” [9]. Wakita et al. [10] also invented a cane-type walking-aid robot “i-cane” to help elderly walk and rehabilitate.

In addition, another hot point of the robotics industry is the obstacle avoidance. These investigations generally need various sensors such as ultrasonic sensors, laser sensors, and cameras. Combined with some specific algorithms, the autonomous navigation and obstacle avoidance function of robots can be achieved. Currently, a great deal of researches have been published [11–14]. References [15, 16] proposed the obstacle detection and avoidance methods for the robot with a camera. However, the process of the images will make the computation more complicated, which will also cost more time and is not suitable for walking-aid robots.

Based on aforementioned researches, it can be found that the walking-aid robots have good human-machine interaction interfaces and there are many human motion intention recognition methods which can fully consider the human's subjective intention [17–24]. The admittance control performs well in using walking-aid robots [10]. However, these walking-aid robots usually cannot recognize and rectify the operator's unreasonable or incorrect intentions, which may cause some safety hazard. In comparison with the walking-

aid robots, the obstacle avoidance robots have the function of path planning and can reach the target point safely. Regrettably, the obstacle avoidance robots only have the single function and lack of the human-machine interface.

Considering the advantages and disadvantages of the robot control and the human control, many investigators combined advantages of these two control methods and proposed the concept of shared control. The shared control is defined that a system can share its controller with one or more human beings and one or multiple robotic controllers [25]. In the field of shared control, many researchers gained quite a few achievements [26–30]. Overall, the research on the shared-control robot is still in its infancy. References [31, 32] proposed a shared-control method for the wheel robot with detection of the human intention through the EMG. But devices for obtaining the bioelectricity signals with pins cannot offer comfortable experiences. Meanwhile, those devices are expensive and not convenient to use in daily life without professional staffs around.

In this study, a convenient and cost-effective hierarchical shared-control method of the walking-aid robot based on human motion intention recognition and obstacle emergency-avoidance methods is presented for solving the situation if there are obstacles during the normal walking. It can save effort during obstacle avoidance and keeps part of the operator's original walking intention. The walking-aid robot is introduced in Section 2. The hierarchical shared-control method which considers both the walking-aid and the obstacle avoidance functions in the walking-aid robot is introduced in Section 3 in detail. In particular, the intention recognition algorithms can make the walking-aid robot thoroughly consider the operator's subjective intention and enhance the quality of human-computer interaction. Also, the artificial potential field method used in this paper can plan the path of walking-aid robot to avoid risks caused by the operators' unreasonable intentions. Furthermore, different control strategies are assumed according to the distance between the walking-aid robot and the obstacles. Section 4 talks about the experiments and analysis. Conclusions are made in Section 5. Finally, experiments are conducted in the real environment which proves the effectiveness of the proposed shared-control algorithm.

## 2. Walking-Aid Robot System

**2.1. Architectural Structure of the Walking-Aid Robot.** In this work, the walking-aid robot system is composed of a set of solid support structures, an omni-directional platform, an industrial personal computer (IPC), a laser sensor, a force-sensing device with FSR, and a torque sensor. The actual photo of the walking-aid robot is shown in Figure 1. The omni-directional platform consists of three omni-directional mecanum wheels driven by DC motor. The laser sensor is adopted to detect obstacles around. The operation principle of the force-sensing device will be stated in the following section.

Figure 2 shows the control flowchart of the walking-aid robot. In this system, the interactive force from the operator collected by the laser sensor can be transmitted

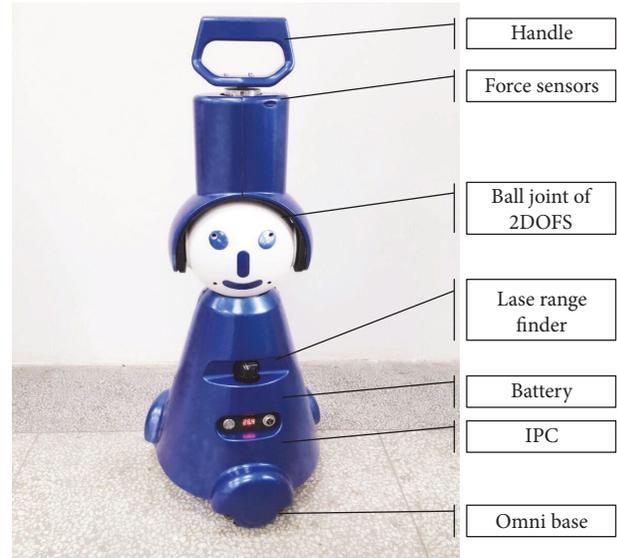


FIGURE 1: The walking-aid robot.

to the IPC. IPC can send control commands based on the collected data to the platform to control the movements of the walking-aid robot.

**2.2. The Force-Sensing Device in Walking-Aid Robot.** Figure 3 shows the force-sensing device in the walking-aid robot in detail. Figure 3(a) shows the structure of the force-sensing device including the handle, the torque sensor, and the FSR for four directions. Figure 3(b) shows the distribution of the four FSR sensors which are pasted to the four sides of the metal rod of the handle, thus making the FSR sensors and the torque sensor precisely detect the magnitude and direction of the force from the operator. Figure 3(c) shows the FSR sensor, which is a one-dimensional variable-resistance pressure sensor. Connecting these FSR sensors and the torque force sensor to the signal conditioning circuits, the force from the operator shall be obtained as the intention force.

**2.3. The Establishment of Coordinate System.** Figure 4 shows the top view of the walking-aid robot architecture. Because the shared-control algorithm is just based on the human motion intention recognition and the local obstructions around the walking-aid robot, a global coordinate system is not necessary in this study. But the coordinate systems of the omni-directional platform, the force-sensing device, and the laser sensor need to be established. In order to simplify the calculations, the coordinate systems of the omni-directional platform, the force-sensing device, and the laser sensor are unified with a fixed local coordinate system  $XOY$ . In the coordinate system  $XOY$ , the front direction of the walking-aid robot is set as the positive  $x$ -axis and the left anterior side is set as the positive  $y$ -axis. Because the FSR sensors are one-dimensional pressure sensors, the data obtained by the force-sensing device is the component of the operator's force in  $X$  or  $Y$  direction. We assume that the force along the positive  $y$ -axis is  $F_1$ . Along the clockwise direction, the force detected by the torque

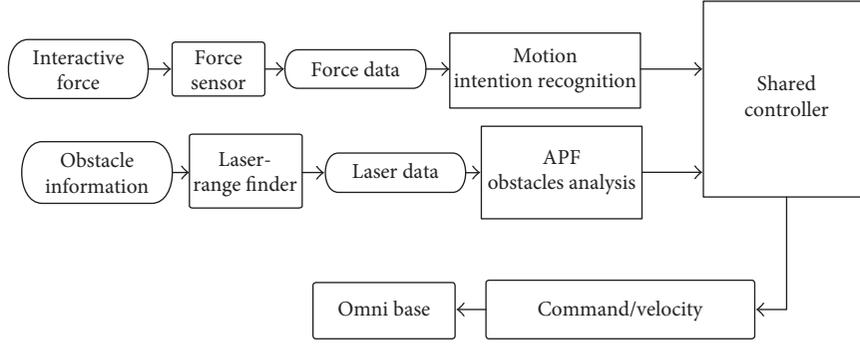


FIGURE 2: Architecture of control system applied in the walking-aid robot.

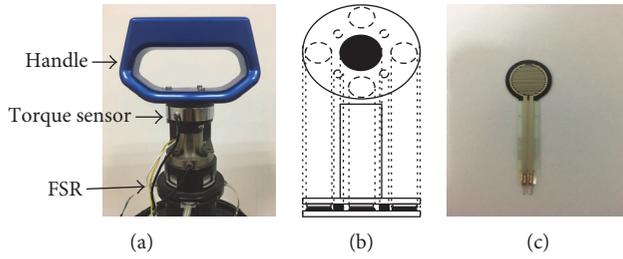


FIGURE 3: Force sensors.

sensor is  $F_0$ . The forces towards four directions acquired by the FSR sensors are  $F_1, F_2, F_3$ , and  $F_4$ , respectively. The laser sensor scans the front obstacles in the counterclockwise direction continuously to acquire the environment information. In the local coordinate system  $XOY$ , the positive  $X$  direction is set as the initial angle  $0^\circ$ , and the angle increases along the counterclockwise direction. In this case, the scanning range of the laser sensor is  $[-90^\circ, 90^\circ]$ .

### 3. Shared-Control Algorithm

**3.1. The Algorithm for Admittance Control.** From the force-sensing device described in 2.2 and the forces  $F_0-F_4$  in 2.3, we can get the components of the operator's intention force along the  $x$ -axis and  $y$ -axis and the rotate direction, respectively. And the operator's intentions are  $F_X, F_Y$ , and  $M_Z$ . Thus, the force components can be obtained by

$$\begin{aligned} F_X &= F_1 - F_3 \\ F_Y &= F_2 - F_4 \\ M_Z &= F_0 R. \end{aligned} \quad (1)$$

In this study, the operator's intention forces are represented by a five-dimensional vector and the operator's intentions are represented by a three-dimensional vector. Then, the intention forces and operator's intentions can be expressed as

$$\begin{aligned} F_S &= [F_0 \ F_1 \ F_2 \ F_3 \ F_4]^T, \\ F_I &= [F_X \ F_Y \ M_Z]^T. \end{aligned} \quad (2)$$

$F_X$  is the intention force along the direction of the  $x$ -axis.  $F_Y$  is the intention force along the direction of the  $y$ -axis, and  $M_X$  is the torque exerted on the walking-aid robots. The positive direction of the torque is counterclockwise direction in this article. Then, the intention forces can be rewritten into

$$F_I = E_{IS} F_S. \quad (3)$$

$E_{IS}$  is a transformation matrix which is defined by

$$E_{IS} = \begin{bmatrix} 0 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & -1 & 0 \\ R & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4)$$

$R$  is half the width of the walking-aid robot in Figure 4.

Once the vector of the operator's intention force  $F_I$  is obtained, the vector  $F_I$  can be converted into a velocity vector  $V$  through the open-loop controlled admittance algorithm. According to the admittance control algorithm, the transfer function can be written as

$$\frac{V(s)}{F_I(s)} = \frac{k}{\tau s + 1}. \quad (5)$$

$k$  is the proportional coefficient. Further, (5) can be taken into the form of continuous time domain model:

$$\tau \frac{dV(t)}{dt} + V(t) = k F_I(t). \quad (6)$$

Discretizing (6), we have

$$\tau \frac{V(i) - V(i-1)}{T} + V(i) = k F_I(i). \quad (7)$$

Finally, (7) can be rewritten in a simplified form as discretizing (6), we have

$$V(i) = k \frac{T}{T + \tau} F_I(i) + \frac{\tau}{T + \tau} V(i-1). \quad (8)$$

**3.2. The Artificial Potential Field Approaches.** The repulsion force based on the artificial potential field approaches is given by the formula

$$F_{ri} = K(r - R_0)^{-n}. \quad (9)$$

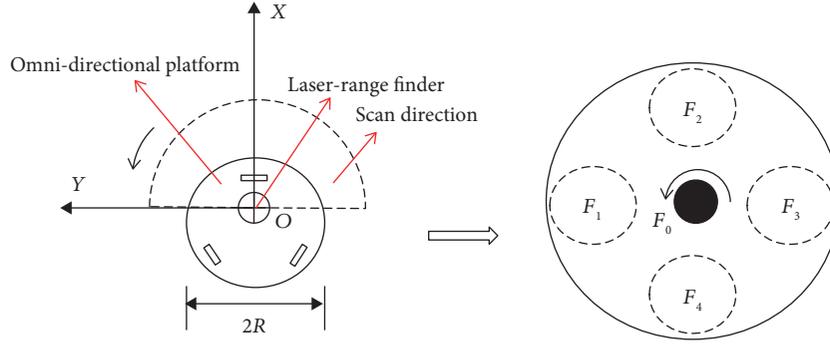


FIGURE 4: Top view of the walking-aid robot system.

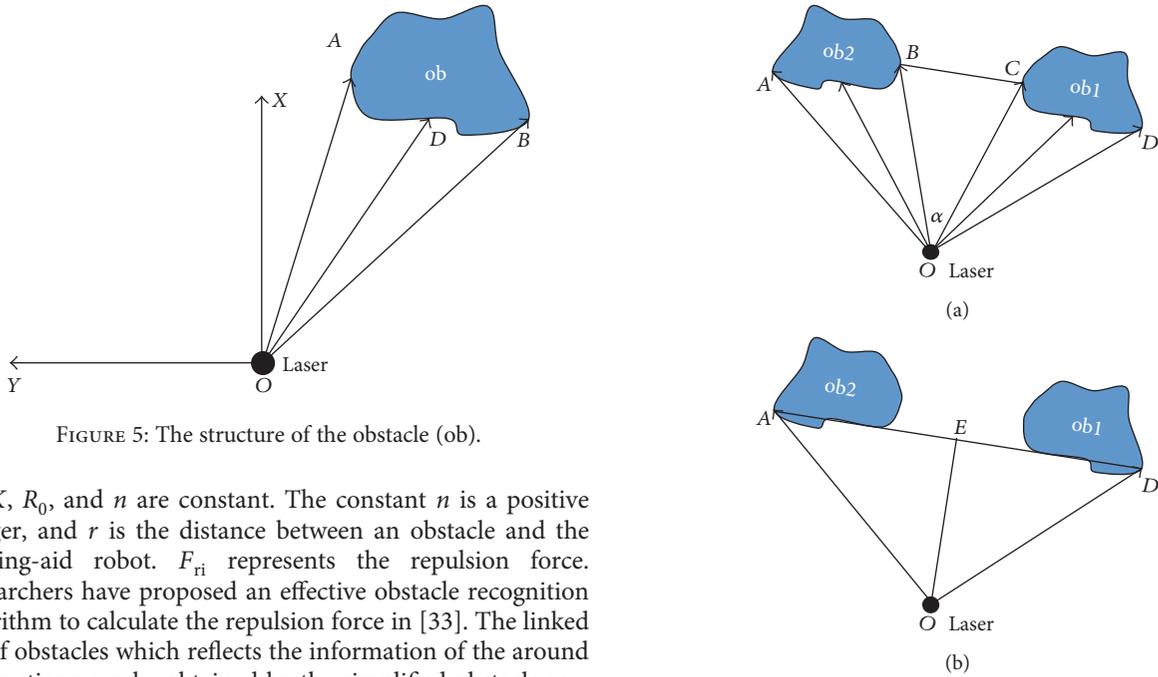


FIGURE 5: The structure of the obstacle (ob).

$K$ ,  $R_0$ , and  $n$  are constant. The constant  $n$  is a positive integer, and  $r$  is the distance between an obstacle and the walking-aid robot.  $F_{ri}$  represents the repulsion force. Researchers have proposed an effective obstacle recognition algorithm to calculate the repulsion force in [33]. The linked list of obstacles which reflects the information of the around obstructions can be obtained by the simplified obstacle recognition algorithm.

Each obstacle in the environment could be expressed in a six-tuple structure [ $rEdge, rAngle, lEdge, lAngle, Force$ , and  $Angle$ ], as shown in Figure 5.

The laser sensor locates at the point  $O$ . The obstacle is  $ob$ .  $rEdge$  represents the distance  $|OB|$  between the obstruction's right edge and the laser sensor.  $rAngle$  is the angle between the  $x$ -axis and the line which connects the right edge of obstruction and the laser sensor. Similarly,  $lEdge$  and  $lAngle$  represent the distance  $|OA|$  and the angle  $\angle AOX$ , respectively.  $Force$  is the magnitude of the repulsion force, and  $Angle$  is the angle between the positive  $x$ -axis and the direction of repulsion force. For each obstacle in this study, the direction of repulsion force is defined by the angle bisector  $OD$  of the angle  $\angle AOB$  which is between the edge of the obstacle and the detection line of laser sensor. Substituting  $r = |OD|$  into (9), the repulsion force caused by the single obstacle  $ob$  can be obtained as

$$F_{ob} = K(|OD| - R_0)^{-n}. \quad (10)$$

FIGURE 6: Diagram of the merging obstacles  $ob1$  and  $ob2$ .

It should be noted that the gap between two obstacles may be so small that the walking-aid robot cannot pass through the gap safely. Since the repulsion force caused by an obstacle is only determined by the relative distance between the obstacle and the walking-aid robot, the accumulative repulsive force caused by the group of obstacles in the same location may be greater than the operator's intention force. In this case, the accumulative repulsion force will hinder the motion of the walking-aid robot. To avoid this case, the obstacles with short distance should be merged. Figure 6 gives a diagram of the merging obstacles.

For the two adjacent obstacles  $ob1$  and  $ob2$  as shown in Figure 6(a), the value of the angle  $\angle BOC$  can be easily obtained based on the given six-tuple structure. According to the law of cosines, the distance  $|BC|$  between  $ob1$  and  $ob2$  is found as

$$|BC| = \sqrt{|OB|^2 + |OC|^2 - 2|OB||OC|\cos \alpha}. \quad (11)$$

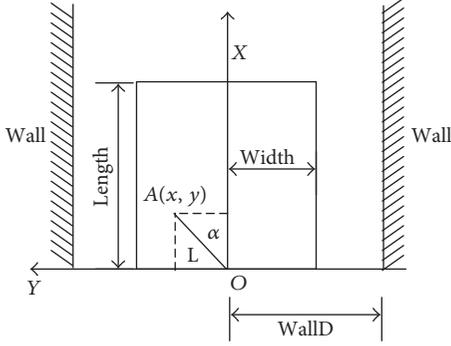


FIGURE 7: The obstacle emergency-avoidance region.

If the distance is smaller than the size of the walking-aid robot, the two obstacles need to be merged, as shown in Figure 6(b). The merged margin of the two obstacles is determined by the right edge of obstacle ob1 and the left edge of obstacle ob2. Then, the distance of the angle bisector is obtained. Based on (9), the repulsion force caused by the merged obstacle can be calculated as

$$F_{ob} = K(|OE| - R_0)^{-n}. \quad (12)$$

The distance  $|OE|$  cannot be directly determined by the laser sensor. The result is that the light beam from the laser sensor (the line extension cord) might just cross the gap between two obstacles. If this happens, the laser sensor cannot detect obstacles and the repulsion force cannot be obtained. The information of merged obstacles is stored in the list, and the resultant of repulsion forces is given by

$$F_r = \sum_{i=0}^n F_{ob(i)}. \quad (13)$$

In practical application, the components of the repulsion forces  $F_{ob}$  along the  $X$  and  $Y$  direction are calculated, respectively, in order to facilitate the fusion with the intention force. The positive and negative direction of the repulsion forces is consistent with the  $x$ -axis and the  $y$ -axis, respectively. The corresponding components can be obtained as

$$\begin{aligned} F_{obX} &= - \sum_{i=0}^n \left| F_{ob(i)} \right| \cos \theta_i, \\ F_{obY} &= - \sum_{i=0}^n \left| F_{ob(i)} \right| \sin \theta_i. \end{aligned} \quad (14)$$

**3.3. The Algorithm of Obstacle Emergency Avoidance.** The short distance between the walking-aid robot and the obstacle may cause the collision. Thus, the algorithm of obstacle emergency avoidance must be taken to ensure the operator's safety and the safety of the robot. Considering the features of the walking-aid robot, an obstacle avoidance strategy is proposed in this study. When the walking-aid robot moves, the obstacle emergency-avoidance region is firstly determined based on the strategy. If the component of the intention force along the  $x$ -axis is negative when there are obstacles in the

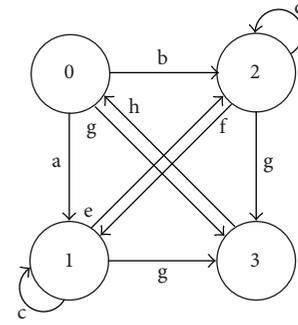


FIGURE 8: The state transition diagram of obstacle emergency-avoidance method.

TABLE 1: The states of the walking-aid robot.

State	Clarification	Action
0	No obstacles	Admittance control with APF
1	Obstacles are detected	Move to the right
2	Obstacles are detected	Move to the left
3	Emergency	Emergency stop

TABLE 2: The state-transition conditions.

State-transition condition	Clarification
a	The obstacles on the right side are less than the ones on the left side.
b	The obstacles on the left side are less than the ones on the right side.
c	There is enough space on the right side to avoid obstacles.
d	There is enough space on the left side to avoid obstacles.
e	There is no enough space on the right side to avoid obstacles.
f	There is no enough space on the left side to avoid obstacles.
g	The emergency cases or no obstacle-avoidance space.
h	The release of the emergency cases.

region, the walking-aid robot will move back to avoid obstacles. If not, the walking-aid robot will move in the lateral way to avoid obstacles. In this study, the selected obstacle emergency-avoidance region is a rectangle while the laser sensor locates at point  $O$  as shown in Figure 7. If a scanning point of the laser sensor is detected in the rectangular area, it means that there are obstacles in the area. The rectangle condition is described as follows:

$$\begin{aligned} |L \cos \alpha| &< \text{length} \\ |L \sin \alpha| &< \text{width}. \end{aligned} \quad (15)$$

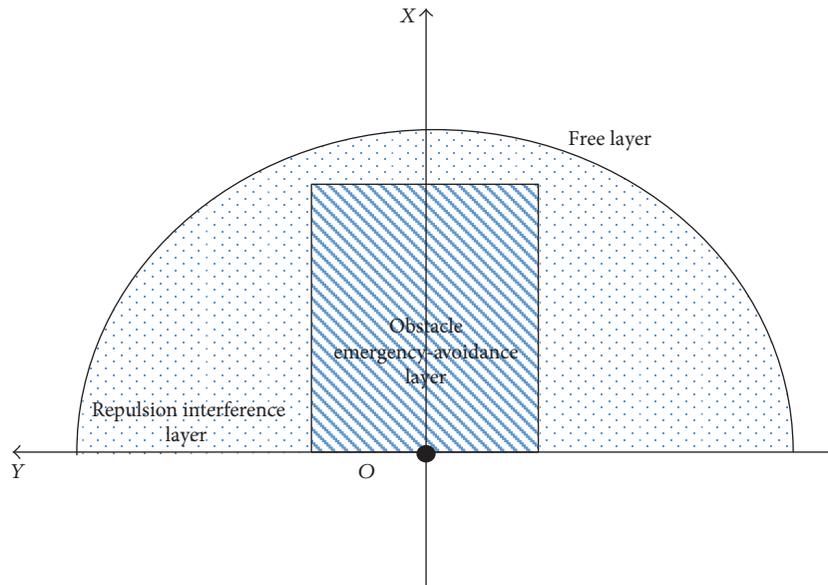


FIGURE 9: The hierarchical layers of the shared-control algorithm.

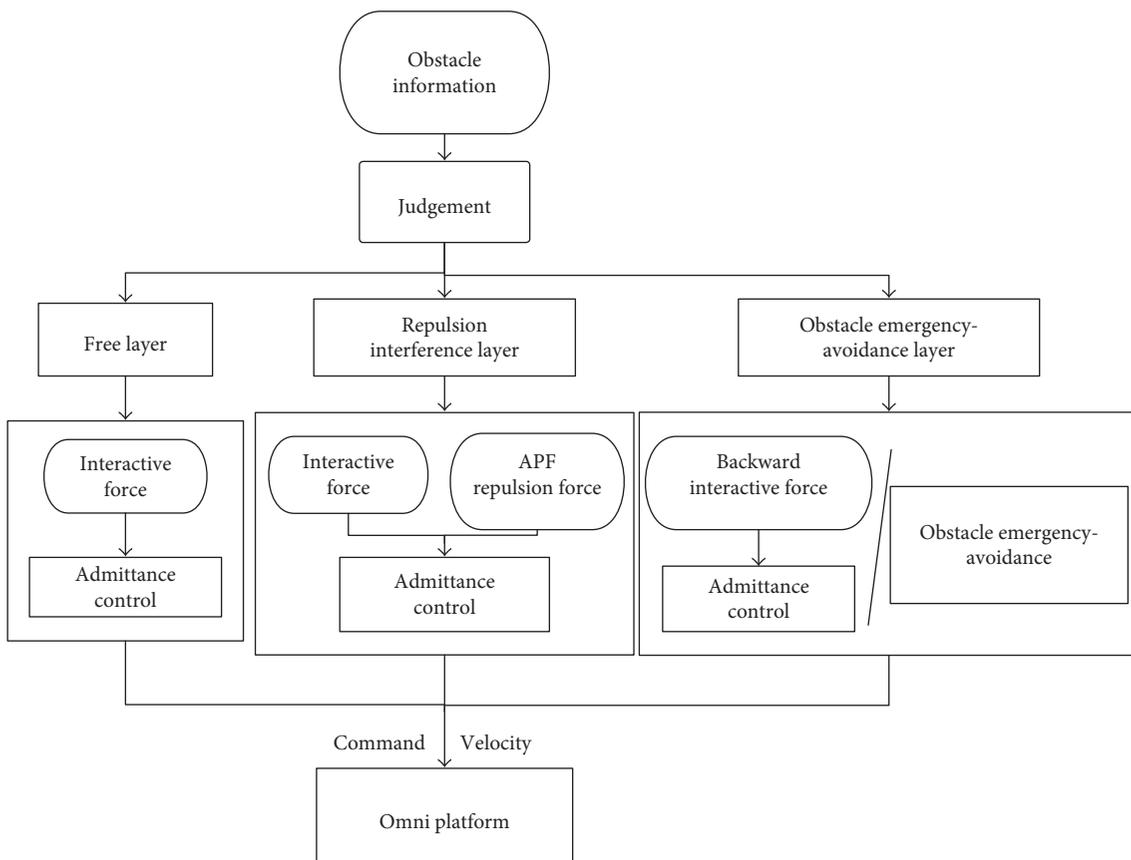


FIGURE 10: The hierarchical shared-control algorithm.

The point  $A(x, y)$  is the detection point of the laser sensor and  $|OA| = L$ . When (15) holds, the rectangle condition is true.

In Figure 7, the “WallD” represents the distance between the walking-aid robot and the lateral wall.

As the first step, the algorithm can make the robot move parallel along the positive or negative direction of the  $y$ -axis to avoid obstacles. For the selection of the robot movement direction, a voting algorithm is used. The robot counts the numbers of the scanning points which satisfy the rectangle

```

In: [rEdge, rAngle, lEdge, lAngle, Force, Angle],  $F_I(k)$ ,  $V(k-1)$ 
Out: State,  $V(k)$ ,  $F_r(k)$ 
1: while there is no emergency accidents do
2:   Determine the position of the obstacles by [rEdge, rAngle, lEdge, lAngle, Force, Angle]
3:   if the obstacle is detected in the Free Layer then
4:     Calculate  $V(k)$  by (8).
5:     The robot moves at  $V(k)$ 
6:     The robot is controlled by the operator
7:   else if the obstacles is detected in the Repulsion Interference Layer then
8:     Calculate  $F_r(k)$  by (13)
9:      $F_I(k) \leftarrow F_I(k) + F_r(k)$ 
10:    Calculate  $V(k)$  by (8).
11:    The robot moves at  $V(k)$ 
12:    The robot and the operator share the controller
13:   else if the obstacles is detected the Emergency Layer then
14:     if  $F_x < 0$  (Backward) then
15:       Calculate  $V(k)$  by (8).
16:       The robot moves back at  $V(k)$ 
17:     else
18:       if there is enough spaces around to avoid obstacles then
19:         State = 0
20:         Calculate  $F_r(k)$  by (13)
21:          $F_I(k) \leftarrow F_I(k) + F_r(k)$ 
22:         Calculate  $V(k)$  by (8).
23:         The robot moves at  $V(k)$  applied with the Admittance Control and the Artificial Potential Field method.
24:       else if the obstacles on the right side are less than ones on the left side then
25:         State = 1
26:         The robot moves to the right at a certain speed.
27:       else if the obstacles on the right side are less than ones on the left side then
28:         State = 2
29:         The robot moves to the left at a certain speed.
30:       else if there are emergency cases or there is no space for avoiding the obstacles then
31:         State = 3
32:         The robot stops immediately.
33:       end if
34:     end if
35:   end if
36: end while

```

ALGORITHM 1: Hierarchical Shared Control.

condition (15) within each side of  $x$ -axis. Then, the side where fewer scanning points satisfying the rectangle condition (15) exist is considered as the enough space for the obstacle avoidance. Thus, it will move to the side where fewer scanning points satisfying the rectangle condition (15) exist. Based on the voting algorithm, the robot will move to the side where fewer scanning points exist. The initial state of the walking-aid robot is also determined by using the voting algorithm. Above all, the proposed obstacle emergency-avoidance method can be described by the state transition diagram, as shown in Figure 8. The numbers 0–3 represent the states of the walking-aid robot as shown in Table 1, and the letters “a–h” represent the state-transition conditions in Table 2.

During the obstacle avoidance, the robot detects the distance “WallD” all the time. Once the distance is less than the safe distance, the robot can change its state of motion immediately. The walking-aid robot can convert among various motion states when there are obstacles in the obstacle avoidance region.

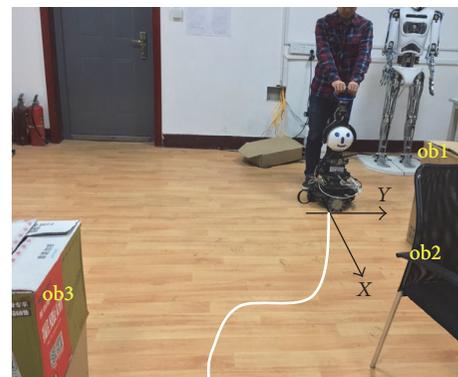


FIGURE 11: The human-robot interaction experiment.

3.4. *The Hierarchical Shared-Control Algorithm.* From the above descriptions, the admittance control algorithm proposed in Section 3.1 can effectively obtain the corresponding

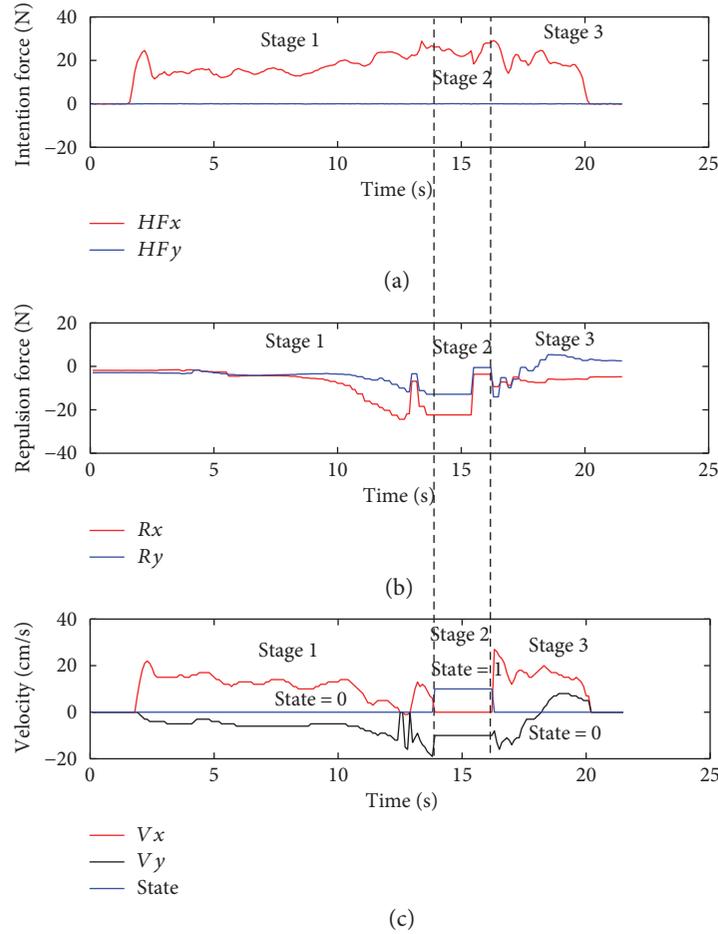


FIGURE 12: The experimental results with subject 1.

velocity of the robot based on the operator's intention force. The admittance control algorithm can also make the operator control smoothly. In Section 2.2, the artificial potential field method synthetically considers the obstacles around the robot and the operator's intention force, which is conducive to make the robot move towards a reasonable direction. The obstacle emergency-avoidance method proposed in Section 3.3 can fully guarantee the safety of the robot and avoid obstacles in the mean time.

Integrating above three algorithms, the shared-control algorithm can be designed easily. According to different distances between the robot and the obstacles in the hierarchical way as shown in Figure 9, the shared-control algorithm can use different control algorithms to control the robot, as shown in Figure 10 and Hierarchical Shared-Control Algorithm.

In the control system, the laser sensor is set as the center and the outermost is set as the **Free Layer**. Without obstacles in the semicircular region shown in Figure 9, the robot is in fully compliance with the operator's intention. The region between the semicircular region and the rectangular region is defined as the **Repulsion Interference Layer**. If any obstacle is detected in this layer, the repulsion force calculated by the artificial potential field method and the operator's intention force control the movement of the robot in the same time. The region within the rectangular region is defined as the

**Obstacle Emergency-Avoidance Layer**. If any obstacle is detected in this layer, the robot will only receive the backward movement instruction from the operator; otherwise, it will follow the algorithm of obstacle emergency-avoidance proposed in Section 3.3 to avoid obstacles.

#### 4. Experiment and Analysis

In this work, the experiments are conducted in the specific indoor environments depicted in Figure 11. In the experiments, constant  $n$  in (9) is selected as 2 to ensure and the repulsion force intensively depends on the distance between the obstacles and the walking-aid robot. Meanwhile, the value of repulsion force should match with the magnitude of the intention force which the sensors measure. According to the AD conversion result of the IPC, the valid range of the intention force is approximately 1–25 N. When the distance  $r = 1.5$  m, the obstacle is very far from the walking-aid robot and its repulsion force is so small that cannot affect the movements of the walking-aid robot. At this time, the repulsion force  $F_{ri} = 1$  N. When the distance  $r = 0.4$  m, the obstacle is very close to the walking-aid robot. At this time, the repulsion force shall roughly be equivalent to the maximum intention force, namely,  $F_{ri} = 25$  N. Substituting these constants into (9), we can obtain the parameters:  $K = 1.313$ ,  $R_0 = 0.354$ .

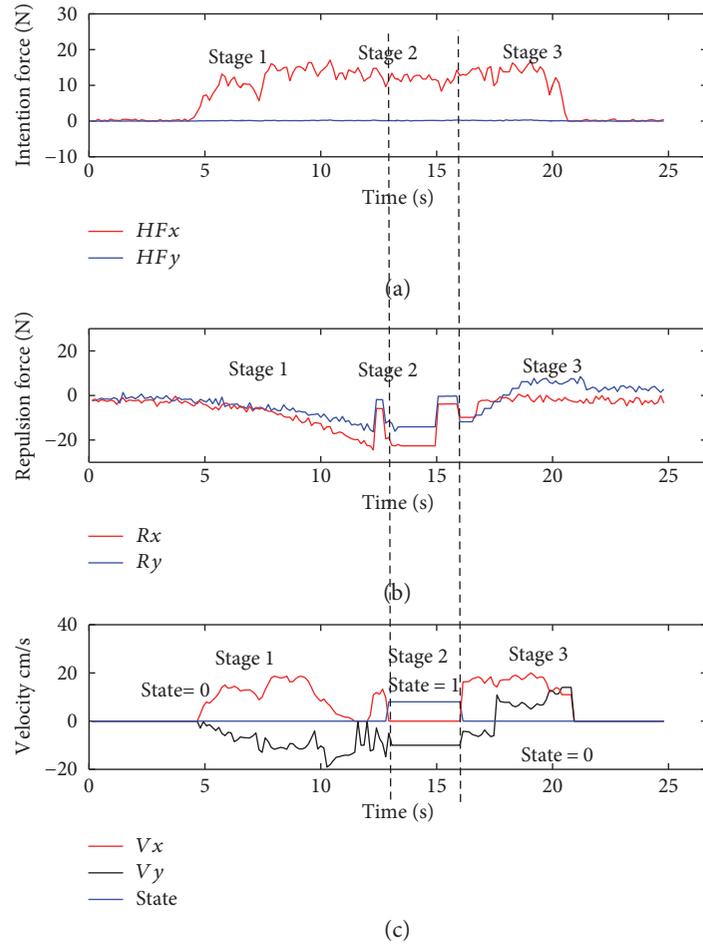


FIGURE 13: The experimental results with subject 2.

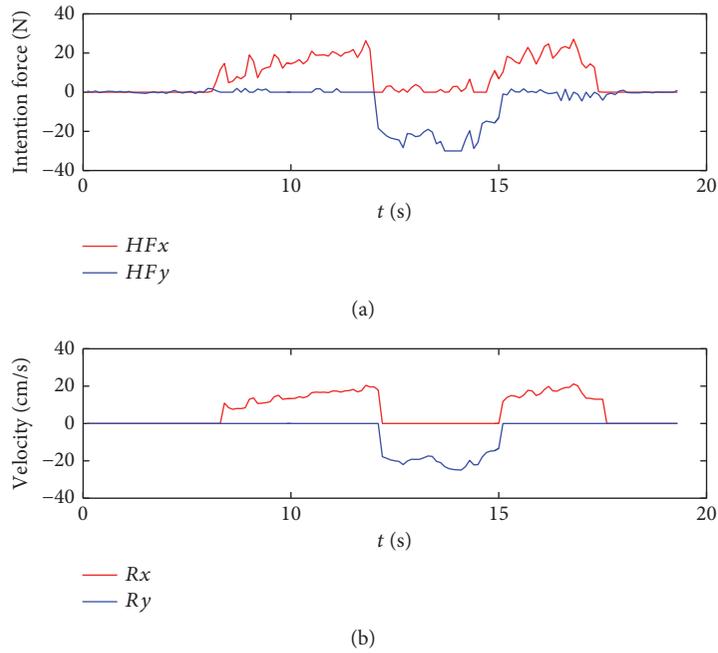


FIGURE 14: The experiment without shared control.

In the experiment, the walking-aid robot is pushed forward. Figure 11 shows the movement trajectory of the walking-aid robot.

The experiment is conducted with a 24-year-old student and a 23-year-old student. The experimental results are shown in Figures 12 and 13. In the figures, the horizontal axis is a time coordinate.  $HF_x$  and  $HF_y$  represent the intention forces along the  $x$ -axis and  $y$ -axis, respectively.  $R_x$  and  $R_y$  represent the repulsion forces along the  $x$ -axis and  $y$ -axis, respectively.  $V_x$  and  $V_y$  represent the velocities of the walking-aid robot along the  $x$ -axis and  $y$ -axis, respectively.

From the above three figures, the experiments can be divided into three stages as below:

**Stage 1.** At the beginning, the walking-aid robot is very far from the obstacles, so the obstacles stay in the **Free Layer**. At this moment, the component of the operators' intention force along the  $x$ -axis is greater than the repulsion force while the operator's intention force along the  $y$ -axis is close to zero. Thus, the robot is mainly controlled by the operators now. Meanwhile, the repulsion force is very small and mainly along the  $y$ -axis, as shown in Figures 12(b) and 13(b). It is indicated that the walking-aid robot moves at a certain speed in the negative direction of the  $y$ -axis. After 10 seconds, the repulsion force obviously increases and the walking-aid robot gradually approaches the obstacle ob2. The obstacle is detected in the **Repulsion Interference Layer**. At the same time, the velocity of the robot along the  $x$ -axis decreases rapidly to zero, while the velocity of the robot along the  $y$ -axis increases with the repulsion force increasing. During this stage, the state of the walking-aid robot is "state 0."

**Stage 2.** At  $t = 13s$ , the operators feel that the velocity of the walking-aid robot decreases; thus, the operators increase the thrust in the direction of the  $x$ -axis in order to make the walking-aid robot move again. At this moment, the obstacle ob2 is detected in the **Obstacle Emergency-Avoidance Layer**. According to the obstacle emergency-avoidance method, it meets the state-transition condition "a" then the state of the walking-aid robot changes from "state 0" to "state 1," resulting that the velocity of the robot along the  $x$ -axis drops rapidly to zero and the robot moves at a fixed velocity along the  $y$ -axis. From Figures 12(b) and 13(b), we can see that the repulsion force remains constant. It indicates that in the "state 1," the operators' intention force cannot affect the velocity of the robot in the stage 2 and the obstacles are not detected in the emergency obstacle avoidance region.

**Stage 3.** Due to the influence of the obstacle ob3, the walking-aid robot moves slowly along the positive direction of the  $y$ -axis according to the obstacle emergency-avoidance method. At  $t = 20s$  in Figure 12 and  $t = 21s$  in Figure 13, the operators stop applying the intention force to the walking-aid robot and the walking-aid robot stops moving.

In the experiment with only the admittance control applied, it cannot reflect the effect of the obstacles in Figure 14. When the operator walks towards the obstacles, the operator have to avoid the obstacles on his own. Compared to the intention force  $HF_y$  in Figures 12 and 13, the intention force  $HF_y$  in Figures 12 and 13 is greater during the obstacle avoidance, which indicates that the shared-control method can save effort.

Over all, the human-robot interaction experiments show the effectiveness of the hierarchical shared control for the walking-aid robot. Based on the hierarchical shared-control algorithm, the robot can successfully help the operator to avoid obstacles and guide the operator to move in a feasible direction, which is really convenient.

## 5. Conclusions

In this work, we proposed a hierarchical control method for the walking-aid robot by combining the human motion intention recognition and the obstacle avoidance methods. It can save effort during obstacle avoidance and keeps the part of the operators' original walking intention. Using this control strategy, the walking-aid robot can autonomously choose different control algorithms to avoid obstacles based on the distance between the walking-aid robot and the obstacles. Also, the emergency obstacle avoidance mechanism is designed to ensure the security of the robot system. The experimental results show that the walking-aid robot can switch among different control algorithms smoothly and guide operators to walk safely.

## Conflicts of Interest

The authors indicated no potential conflicts of interest.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) [Grant no. 61473130], the Applied Basic Research Program of Wuhan [Grant no. 2016010101010014], and in part by the Science Fund for Distinguished Young Scholars of Hubei Province (2015CFA047). The authors would like to gratefully acknowledge these supports.

## References

- [1] K. A. Kline and D. M. Bowdish, "Infection in an aging population," *Current Opinion in Microbiology*, vol. 29, pp. 63–67, 2016.
- [2] C. Hu, F. Aeschlimann, G. Chatzipirpiridis et al., "Spatiotemporally controlled electrodeposition of magnetically driven micromachines based on the inverse opal architecture," *Electrochemistry Communications*, vol. 81, 2017.
- [3] C. Hu, H. Vogler, M. Aellen et al., "High precision, localized proton gradients and fluxes generated by a microelectrode device induce differential growth behaviors of pollen tubes," *Lab on a Chip*, vol. 17, 2017.
- [4] J. Huang, X. Tu, and J. He, "Design and evaluation of the RUPERT wearable upper extremity exoskeleton robot for clinical and in-home therapies," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 926–935, 2016.
- [5] H. Kawamoto, K. Kamibayashi, Y. Nakata et al., "Pilot study of locomotion improvement using hybrid assistive limb in chronic stroke patients," *BMC Neurology*, vol. 13, no. 1, pp. 93–98, 2013.

- [6] S. Tanabe, E. Saitoh, S. Hirano et al., "Design of the Wearable Power-Assist Locomotor (WPAL) for paraplegic gait reconstruction," *Disability and Rehabilitation: Assistive Technology*, vol. 8, no. 1, pp. 84–91, 2013.
- [7] T. Kikuchi, T. Tanaka, K. Anzai, S. Kawakami, M. Hosaka, and K. Niino, "Evaluation of line-tracing controller of intelligently controllable walker," *Advanced Robotics*, vol. 27, no. 7, pp. 493–502, 2013.
- [8] J. Huang, Z. H. Guan, T. Matsuno, T. Fukuda, and K. Sekiyama, "Sliding-mode velocity control of mobile-wheeled inverted-pendulum systems," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 750–758, 2010.
- [9] Y. Hirata, A. Hara, and K. Kosuge, "Motion control of passive intelligent walker using servo brakes," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 981–990, 2007.
- [10] K. Wakita, J. Huang, P. Di, K. Sekiyama, and T. Fukuda, "Human-walking-intention-based motion control of an omnidirectional-type cane robot," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 285–296, 2013.
- [11] M. Khatib and R. Chatila, "An extended potential field approach for mobile robot sensor-based motions," in *International Conference on Intelligent Autonomous Systems*, pp. 490–496, 1995.
- [12] K. Iwatsuka, K. Yamamoto, and K. Kato, "Development of a guide dog system for the blind with character recognition ability," in *First Canadian Conference on Computer and Robot Vision, 2004. Proceedings*, pp. 401–405, London, ON, Canada, 2004, IEEE Computer Society.
- [13] T. Gonnot and J. Saniie, "Integrated machine vision and communication system for blind navigation and guidance," in *2016 IEEE International Conference on Electro Information Technology (EIT)*, pp. 0187–0191, Grand Forks, ND, 2016.
- [14] E. B. Kaiser and M. Lawo, "Wearable navigation system for the visually impaired and blind people," in *IEEE Computer Society, 2012 IEEE/ACIS 11th International Conference on Computer and Information Science*, pp. 230–233, Shanghai, China, 2012.
- [15] C. J. Kim and D. Chwa, "Obstacle avoidance method for wheeled mobile robots using interval type-2 fuzzy neural network," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 3, pp. 677–687, 2015.
- [16] F. Fabrizio and A. D. Luca, "Real-time computation of distance to dynamic obstacles with multiple depth sensors," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 56–63, 2017.
- [17] J. Huang, P. Di, and T. Fukuda, "Motion control of omnidirectional type cane robot based on human intention," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2983–2988, Nice, France, 2009.
- [18] P. Di, Y. Hasegawa, S. Nakagawa et al., "Fall detection and prevention control using walking-aid cane robot," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 625–637, 2016.
- [19] Q. Y. Yan, W. X. Xu, J. Huang, and P. C. Su, "Laser and force sensors based human motion intent estimation algorithm for walking-aid robot," in *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 1858–1863, Shenyang, China, 2015.
- [20] W. Wang, Z. G. Hou, L. Cheng et al., "Toward patients' motion intention recognition: dynamics modeling and identification of iLeg-An LLRR under motion constraints," *IEEE Transactions on Systems Man and Cybernetics Systems*, vol. 46, no. 7, pp. 1–13, 2016.
- [21] K. Khokar, R. Alqasemi, S. Sarkar, K. Reed, and R. Dubey, "A novel telerobotic method for human-in-the-loop assisted grasping based on intention recognition," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4762–4769, Hong Kong, 2014.
- [22] J. H. Han, S. J. Lee, and J. H. Kim, "Behavior hierarchy-based affordance map for recognition of human intention and its application to human' robot interaction," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 5, pp. 1–15, 2016.
- [23] J. Huang, W. Huo, W. Xu, S. Mohammed, and Y. Amirat, "Control of upper-limb power-assist exoskeleton using a human-robot interface based on motion intention recognition," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1257–1270, 2015.
- [24] H. C. Sang, J. M. Lee, S. J. Kim, Y. Hwang, and J. An, "Intention recognition method for sit-to-stand and stand-to-sit from electromyogram signals for overground lower-limb rehabilitation robots," in *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 418–421, Busan, South Korea, 2015.
- [25] T. B. Sheridan, *Telerobotics, Automation and Human Supervisory Control*, The MIT Press, Cambridge, 1992.
- [26] S. S. Nudehi, R. Mukherjee, and M. Ghodoussi, "A shared-control approach to haptic interface design for minimally invasive telesurgical training," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 588–592, 2005.
- [27] F. Chen, P. Di, J. Huang, H. Sasaki, and T. Fukuda, "Evolutionary artificial potential field method based manipulator path planning," in *2009 International Symposium on Micro-NanoMechatronics and Human Science*, pp. 92–97, Nagoya Japan, 2009.
- [28] H. T. Trieu, H. T. Nguyen, and K. Willey, "Shared control strategies for obstacle avoidance tasks in an intelligent wheelchair," in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4254–4257, Vancouver, BC, Canada, 2008.
- [29] W. G. Huh and S. B. Cho, "Optimal partial filters of EEG signals for shared control of vehicle," in *2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pp. 290–293, Fukuoka, Japan, 2015.
- [30] W. X. Xu, J. Huang, Y. J. Wang, C. J. Tao, and L. Cheng, "Reinforcement learning-based shared control for walking-aid robot and its experimental verification," *Advanced Robotics*, vol. 29, no. 22, pp. 1463–1481, 2015.
- [31] S. Sathish, K. Nithyakalyani, S. Vinurajkumar, C. Vijayalakshmi, and J. Sivaraman, "Control of robotic wheel chair using EMG signals for paralysed persons," *Indian Journal of Science and Technology*, vol. 9, no. 1, 2016.
- [32] A. Accogli, L. Grazi, S. Crea et al., "EMG-based detection of user's intentions for human-machine shared control of an assistive upper-limb exoskeleton," in *Wearable Robotics: Challenges and Trends*, pp. 181–185, Springer International Publishing, Cham, 2017.
- [33] W. Hong, Y. T. Tian, Z. Dong, and M. Zhou, "Extracting features from local environment for intelligent robot system," *Robot*, vol. 25, no. 3, 2003.

## Research Article

# Surgeon Training in Telerobotic Surgery via a Hardware-in-the-Loop Simulator

Xiao Li,<sup>1</sup> Homa Alemzadeh,<sup>2</sup> Daniel Chen,<sup>3</sup> Zbigniew Kalbarczyk,<sup>3</sup>  
Ravishankar K. Iyer,<sup>3</sup> and Thenkurussi Kesavadas<sup>4</sup>

<sup>1</sup>Department of Mechanical Engineering, University of Illinois, Urbana, IL 61801, USA

<sup>2</sup>Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22903, USA

<sup>3</sup>Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL 61801, USA

<sup>4</sup>Department of Industrial and Enterprise Systems Engineering, University of Illinois, Urbana, IL 61801, USA

Correspondence should be addressed to Xiao Li; [xiaoli16@illinois.edu](mailto:xiaoli16@illinois.edu)

Received 6 January 2017; Revised 4 April 2017; Accepted 14 May 2017; Published 3 August 2017

Academic Editor: Qing Shi

Copyright © 2017 Xiao Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work presents a software and hardware framework for a telerobotic surgery safety and motor skill training simulator. The aims are at providing trainees a comprehensive simulator for acquiring essential skills to perform telerobotic surgery. Existing commercial robotic surgery simulators lack features for safety training and optimal motion planning, which are critical factors in ensuring patient safety and efficiency in operation. In this work, we propose a hardware-in-the-loop simulator directly introducing these two features. The proposed simulator is built upon the Raven-II™ open source surgical robot, integrated with a physics engine and a safety hazard injection engine. Also, a Fast Marching Tree-based motion planning algorithm is used to help trainee learn the optimal instrument motion patterns. The main contributions of this work are (1) reproducing safety hazards events, related to da Vinci™ system, reported to the FDA MAUDE database, with a novel haptic feedback strategy to provide feedback to the operator when the underlying dynamics differ from the real robot's states so that the operator will be aware and can mitigate the negative impact of the safety-critical events, and (2) using motion planner to generate semi-optimal path in an interactive robotic surgery training environment.

## 1. Introduction

The field of surgical robotics has been rapidly expanding over the last decade [1]. Robot-assisted surgery is the preferred technique for a variety of minimally invasive procedures worldwide. Simulation-based learning and training is now a standard in robotic surgery because healthcare professionals improve performance and reduce errors through comprehensive medical care simulation [2, 3]. Simulation can bridge the gap in learning robotic surgery skills without accidentally harming the patient. For example, LapSim Haptic System™ is a laparoscopic surgery simulator with realistic hardware interface and tactile feedback, which is mainly used for near-field nonteleoperated surgery training [4]. Other commercial surgical simulators on the market such as Mimic's dV-Trainer™ [5] and Simulated Surgical System's RoSS [6]

provide basic motor skills training modules using virtual reality with surgeon console similar to the da Vinci surgical system, to provide life-like simulation and help prepare surgeons. Figure 1 shows these simulators' profiles. A key issue with simulation-based surgical training is the lack of safety-critical incident scenarios in simulation-based curricula, which is critical in bringing this form of surgical education to practice. Current surgical simulators focus on better graphics rendering and curriculum of goal achievement (success in arriving at goals while ignoring intermediate motion patterns) rather than real robot dynamics motion as well as teaching the surgeon optimal motion pattern and path in an obstacle-surrounded environment.

Our previous study of adverse events reported to the U.S. Food and Drug Administration (FDA) Manufacturer and User Facility Device Experience (MAUDE) database showed



FIGURE 1: From left to right: LapSim haptic system, dV-trainer, and RoSS.

that despite significant improvements in robotic surgery technology through the years and broader adoption of the robotic approach, there are ongoing occurrences of safety incidents that negatively impact patients. The number of injury and death events per procedure has stayed relatively constant since 2007, with an average of 83.4 events per 100,000 procedures [7]. Although, these incidents are often caused by accidental malfunctions or technical problems with the robot and steep learning curves, it has also been shown that surgical robots can be subject to malicious cyber-attacks that impact patient safety and progress of surgery [8, 9]. The ability of current robotic surgery technology to automatically mitigate the impact of safety incidents still lags other safety-critical industries, such as commercial aviation. In such industries, great effort has been spent over the years on improving safety practices by providing comprehensive simulation-based training that includes operation in the presence of safety-critical failures [10]. In current robotic surgeon training, the emphasis is on improving surgical skills and not on handling safety-critical events and responding to technical problems. Adverse events or accidental machine failures are rarely used as potential scenarios for safety training of surgical teams.

In this work, we are motivated by the idea of simulating safety hazards [11, 12] during robotic surgery training in order to prepare surgeons for handling safety-critical events. The objective is to develop a hardware-in-the-loop simulator platform that emulates realistic safety hazard scenarios in a virtual environment and provides awareness of the impending hazards to the operator through haptic force feedback. In this work, we use Raven-II [13] surgical robot as the hardware that the operator will be trained with. Previous studies have shown that users trained on the Raven platform can transfer their skills to da Vinci system [14]. We developed a robot-environment interaction model using a physics engine as the robot's nominal state estimator (fault-free run), which runs simultaneously with the Raven-II robot hardware. We also developed a safety hazard injection engine that intentionally and artificially creates adverse events by inserting faults into the robot control system using Software-implemented Fault Injection (SWIFI) [15]. The faults are injected to the control software after the system's automatic safety checks are performed to increase the chance that they cause safety hazards.

The main goal of SWIFI is to validate the effectiveness of fault-tolerance mechanisms by studying system behavior in the presence of simulated faults. Here, we use software-based fault injection techniques to emulate the safety hazards

and enable evaluation of human operator performance and response to safety hazards during simulation-based training. Fault injection and cyber-attacks on the safety-critical systems, such as smart grid [16–18], automotive embedded systems [19], and robotic vehicles [20, 21], have been the subject of many studies. They presented attack scenarios that directly target the physical system, the control commands sent to the physical system, or the sensor measurements received from the physical layer to corrupt the state of controller in the cyber-domain (false data injection attacks). In this work, we use fault injection and target the robot control system to corrupt the control commands in a legitimate manner that is not detectable by the robot's safety mechanisms. In our previous work, we showed that these injections could lead to unexpected and sudden jumps of the robotic arms and negatively impact the robot operation and patient safety in just a couple of milliseconds, making it difficult for both automated mechanisms and human operators to respond in a timely manner. Thus, the detection and response mechanisms in real-time surgical cyber-physical systems should be optimized and deployed in such a way that can mitigate the impact of faulty and malicious commands before they even execute in the physical layer [8].

In [22], authors demonstrated content modification attacks on a bilateral teleoperation system and used Lyapunov-based analysis to conclude that if the sent velocity does not equal to the position's derivative, then there is a static attack (linear modification of the states using time-invariant gains). Their method however might suffer from sensitivity to model accuracy, since all analysis is based on model-based Lyapunov analysis. Providing the user with haptic cues using the haptic force calculated based on the difference between desired and actual position of end-effectors in the slave robot was proposed in [23]. Teleoperation for Raven robot uses Interoperable Telerobotics Protocol [24]—sending incremental rather than absolute motion command from master to slave and a human operator is in the control loop to correct the position errors. In transient phase, there is tracking error between the actual slave robot end-effector position and the desired position; therefore, comparing the two to generate the haptic feedback cannot be done accurately. We address this issue by developing a dynamical model for the physical robot (a virtual robot) as the underlying state monitor. The model and robot receive the same motion command from the master, and we use their state difference to create the haptic feedback.

When training a novice surgeon, he/she can acquire some sense of optimality by observing or sensing (through haptics) the robot's execution of an automated task. The motion can be planned optimally by minimizing certain cost functions. Related work in this area focuses on automating some of the real surgery scenarios in different robot-assisted surgery types. Weede et al. in [25] introduced an autonomous camera system including a prediction of interventions, to provide a long-term prediction of the steps a surgeon will perform in the next few minutes and move the endoscope to an optimal position. Combined with vision techniques, automatic positioning and autonomous retrieval of surgical instruments have been achieved in [26, 27]. Chow et al. in [28] showed

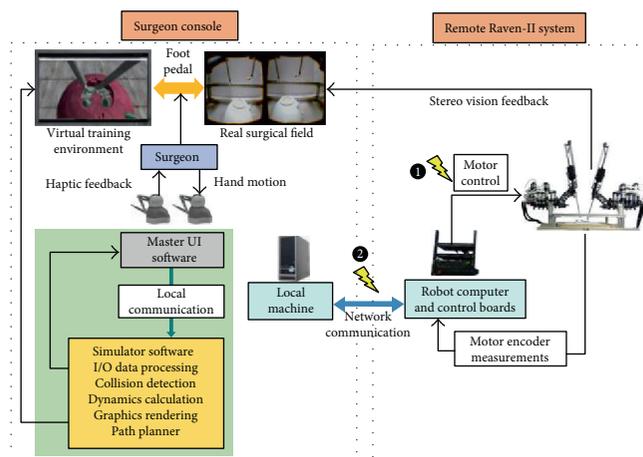


FIGURE 2: Hardware-in-the-loop surgical simulator architecture.

that vision-guided autonomous knot-tying in robotic-assisted surgery has the potential to be faster than human performance. Kehoe et al. in [29] demonstrated the first reliable autonomous robot performance of surgical sub-task, that is, removing tissue fragments using Raven, by generating centralized motion plans through 3D sensing and *trajopt*, a low-level motion planning algorithm based on sequential convex optimization to plan locally optimal, collision-free trajectories simultaneously for both arms. Hu et al. in [30] investigated path planning and semi-automated motion for the scenario of robotic ablation of tumor residues in various shapes using Raven robot, and different metrics were delivered to the surgeon to select candidate path plan. In nonlaparoscopic type of robotic surgery, for example, in needle steering community, efforts have been made in surgical preplanning for the needle type surgical robot [31, 32] and demonstrated in simulation environment [33].

In this work, we present a comprehensive software framework for the telerobotic surgical simulator. The simulator includes a failure scenario generation module which simulates failures during a surgery through fault injections. These failure scenarios can train surgeons to recognize adverse events during a surgery through haptic cues. The optimal trajectory generated by Fast Marching Tree (FMT\*) algorithm designed for Raven-II platform at an interactive rate will also help trainee gain an optimal sense of manipulating the surgical instrument.

## 2. Materials and Methods

**2.1. Simulator Framework.** We design the simulator system based on the Raven-II surgical robot, an open source platform running on top of Robot Operating System (ROS). To develop a surgical simulator with high fidelity in reproducing adverse events, we include the robot hardware in the simulator's execution loop and integrate it with a safety hazard injection engine [8] and a physics engine to simulate the robot dynamics and interaction with environment. The simulator system architecture is shown in Figure 2. Raven-II is a teleoperated surgical robot which uses network

communication between the local machine on surgeon's console side and the remote Raven computer. The simulator runs on the local machine and performs dynamics and collision calculations. Two Phantom Omni devices receive the incremental motion command from the operator and then send the data to both the local machine and the remote machine through UDP/IP. A virtual Raven robot and 3D training environment are displayed on the screen of surgeon's console. The graphics are rendered through C++ OpenGL pipeline with a frequency of 30 Hz, while other calculations, for example, haptic loop and physics engine and network data transmission, using multiple threads are being synchronized and run at 1000 Hz, which is the same as the running frequency of Raven's control loop.

The connection between the Surgeon Console and the Raven-II system is bilateral network communication in our hardware-in-the-loop simulator shown in Figure 2. One direction is for transmitting Omni command data from the local machine to the remote Raven system, while the other direction is for sending the robot state data (joint positions and velocities) back to the local machine using TCP/IP socket connection for reliability and to make comparison with the dynamics calculation results in physics engine thread. The haptic force feedback is provided to the operator if the virtual and real Raven's end-effector trajectories do not match (above a predefined threshold). Since perfect transparency (master device force/torque matching the slave's end-effector force/torque) is not possible and is especially challenging for teleoperators with significant nonlinear dynamics and no force sensors mounted at the robot end-effector, we utilize haptics feature for safety propose, rather than for surgical palpation. Because the haptic device sensor/actuator asymmetries can cause instability and robustness issues, we apply a spring-damper model with appropriate gains and saturations for feedback force calculation.

To simulate the safety hazards in real surgery, we integrate the robot control software with a safety hazard injection engine that strategically inserts faults into the control software at critical junctures during operation [8]. More explicitly, the injected faults corrupt either the Omni commands or the motor control commands sent to the Raven

hardware after the safety checks are done in the Raven software in robot computer, which is indicated in Figure 2. As a result, unexpected robot motion will generate trajectory errors compared with the underlying model dynamics. Then, we show how the operator can gain awareness of the erroneous robot trajectory in the presence of faults through haptic force feedback.

Beyond the capability of basic motor skills training and simulating adverse events, there are additional features in our simulator that can help improve the surgeon's performance in real surgery. For example, in some scenarios, it is preferable to let the surgeon do a virtual trial, rather than manipulating the actual robot all the time. One of the important capabilities of our simulator is to allow the user to disengage from the actual robot to do a trial movement in the simulator's virtual environment and see the outcomes of virtual motion. If the outcome is satisfactory, then the actual robot could be reengaged to track the recorded command trajectory data and move in an autonomous fashion. The user can also use the path planner to specify the target configuration of the robot arm, and then it will automatically generate trajectory waypoints for the robot arm to track. A foot pedal is placed on the surgeon's side to enable switching between the robot teleoperation mode and pure simulation mode (being disengaged from the robot hardware) and also toggling the view between the real surgical field and virtual environment.

**2.2. Robot Dynamics Modeling and Training Scenario.** In our previous work, we simulated dynamics, numerically integrating the equations of motion derived by Euler-Lagrange (E-L) approach in [34]. This approach provides very little freedom for simulating interactions with the environment. In this work, we integrate a physics engine—Open Dynamics Engine (ODE) into the simulator, to simulate dynamic behaviors of the robot and interactions between the robot manipulators and the environment. ODE is an open source, high-performance library, which relies on a Linear Complementary Problem Solver (LCP solver) [35]. In robotics simulation, ODE is being widely used for a variety of applications [36, 37].

In [34], we obtained the equations of motion by E-L and determined the mechanical properties of the links through CAD models. In ODE setup, we directly specify each link's mesh properties and joint properties, so that we can achieve the same robot motion as in [34] (with no collisions). Besides, ODE has the capability of doing collision checking between primitive objects or meshes and using contact friction models to apply contact forces, which gives the possibility of simulating interactions.

For the training scenario, we use the training model which is widely used in the Fundamentals of Robotic Surgery (FRS) organization, the definitive robotic surgical skills education, training, and validated assessment program [38]. The objectives of using the training model can be found in [38]. Specifically, in our simulator prototype, we made a small modification to the original model (extruded cut half of the top lid) so that the Raven robot arm can be inserted into the cavity to perform the tasks, that is, picking up a cube ring



FIGURE 3: Simulated surgeon console and FRS training model—yellow sphere on each arm indicates the fixed remote motion center, and the cavity of the dome represents a human abdomen area; more details of the model can be found in Figure 7 and Figure 8.

initially placed inside the cavity. The operator will manipulate the robot arms to get used to 3D teleoperation by picking up a ring, transferring it from one arm to the other, and navigating the ring along the loops. The abovementioned motor skills training scenarios can be done in a semiautonomous fashion using the Fast Marching Tree- (FMT\*-) based path planner (if only involved one arm motion, dual arm motion case can be split into one arm motion followed by another).

**2.3. FMT\*-Based Path Planner.** The path planner utilizes FMT\* algorithm, in which "\*" indicates optimality to given cost criteria. An on-off foot pedal is used to activate the path planning functionality. In planning mode, the user is required to use the master device controlling the slave robot in the simulation environment to reach a target configuration. In this phase, collision detection is disabled and just pure kinematic motion is performed. Once the path planner got the goal configuration, it will start to compute dynamically feasible trajectory and execute the motion plan for the arm at interactive rates (only a few seconds for planning and executing motion plans, resp.).

The cavity in the FRS training model is used to simulate the human abdomen (Figure 3), where the inside volume is very constraint and, thus, requires very fine motions of the robot end-effectors. It is preferable to make the surgical instruments (robot arms) automatically inserted or retrieved before or after the surgical procedure. Raven-II robot has two instrument arms that are independent of each other in terms of assembly and controls, and each has seven degrees of freedom (DOF) and six rotational joints plus one translational joint. In high-dimensional space, sampling-based path planning algorithms can explore the configuration space effectively by sampling the collision-free configurations according to a probability distribution (in this work, uniform sampling in feasible joint space is used). Rapidly exploring random tree (RRT) and probabilistic roadmap method (PRM) and their variants have become prevalent in robot path planning applications and literature over the past ten years, especially when the RRT\* and PRM\*'s optimality proofs was formally given in [39]. In [39], it is shown that

PRM\* and RRT\* are provably asymptotically optimal, that is, the cost of the returned solution converges almost surely to the optimum. However, building the RRT tree or connecting PRM edges require extensive collision checking. In our case, collision checking between triangle meshes will severely hurt the performance. Recently, a new probabilistic sampling-based planning algorithm called Fast Marching Tree (FMT\*) was introduced [40]. The algorithm is specifically aimed at solving complex motion planning problems in high-dimensional configuration spaces. This algorithm is proven to be asymptotically optimal and is shown to converge to an optimal solution faster than its state-of-the-art counterparts, namely, PRM\* and RRT\*. However, the sacrifice is that it lazily skips collision checks when evaluating the local connections. This lazy collision checking strategy may introduce suboptimal connections, but the crucial property of FMT\* is that such suboptimal connections become vanishingly rare as the number of samples goes to infinity. In both low- and high-dimensional benchmark problems tested in [40], which across a variety of problem instances, ranging in obstacle clutter and in dimensions from 2D to 7D, it is shown that FMT\* outperforms state-of-the-art algorithms such as PRM\* and RRT\*, often by a significant margin. The speedups are particularly prominent in higher dimensions and in scenarios where collision checking is expensive, which is exactly the regime in which sampling-based algorithms excel. In this work, we utilize the advantages of FMT\* algorithm to achieve the goal of motion planning and executing tasks.

**2.3.1. Assumptions.** In designing the path planner using FMT\* algorithm, we make the following two assumptions:

- (1) Decoupling between control and joint motion. The Raven-II robot uses cable-driven mechanisms, so the joint motion of the instrument arms are not only affected by one DC motor. The closer to the end-effector, the more complicated coupling motion would involve. In our ODE simulation environment, we do not model the cable coupling behavior, and we assume each actuator will control one joint motion only.
- (2) Fixed opening angle for the grasper. Each grasper consists of two jaws and, thus, has two DOFs. We can think of jaws as two independent DOFs or considering them as one part, and then it has one DOF as the center line of the grasper, another DOF would be the opening angle of the grasper, and these two representations are kinematically equivalent. In some cases, when the grasper is holding some object (e.g., holding the cube ring in our training scenario), we do want to keep the opening angle unchanged. So in our path planner, we only consider six DOFs, instead of seven.

**2.3.2. Problem Statement.** As mentioned above, in this work, we consider motion planning problem for one arm, either left arm or right arm, because the case of dual arm planning

problem can be treated as moving the arms one by one. This simplification will reduce the computation cost tremendously. The motion planning problem of Raven-II surgical robot in a surgical training environment can be stated as follows:

- (i) *Inputs:* surgical environment  $\Psi$  and Raven-II robot  $\mathcal{R}$  described by mesh file objects, initial robot configuration  $\theta_{\text{init}}$  for the moving arm (collision-free) when the path planning is enabled, goal robot configuration  $\theta_{\text{goal}}$  for the same arm (collision-free) specified by the user,  $n$  number of collision-free configurations, and  $r_n$  connection radius
- (ii) *Output:* a feasible collision-free motion plan consisting waypoints  $\Theta$  for one arm trajectory and each waypoint is a 6-dimensional vector including 6 joint positions.

Note that motion between two intermediate waypoints is dynamically feasible and free of collision. And if no path exists between start and end configurations, the algorithm terminates in finite time.

**2.3.3. FMT\* Algorithm.** We consider this planning problem for one robot arm. The planning objective is to find a dynamically feasible collision-free path while minimizing the overall cost function:

$$J(\sigma^*) = \min \left\{ \eta_1 \cdot c(\sigma) + \eta_2 \cdot \int_0^T |\alpha(t)|^2 dt, \sigma \text{ is feasible} \right\}, \quad (1)$$

in which  $\sigma$  is a feasible path in joint space,  $c(\sigma)$  is the arc length of  $\sigma$  with respect to Euclidean metric,  $\alpha(t)$  is the end-effector's linear acceleration, and  $\eta_1, \eta_2$  are two user-defined coefficients to weight the effects of path length and velocity variations. The FMT\* algorithm is outlined in Algorithm 1.

The description of the functions (e.g., SampleFree, Near, and Save) in the FMT\* algorithm are described in [40]. In surgical planning, we should make sure that when connecting two waypoints on the path, no collision happens and the end-effector's velocity is smooth, that is, there is no jerky motion on the robot joints. So for evaluating the connectedness of two waypoints, rather than simply do linear or nonlinear interpolations as the kinematics level test, we integrate the robot dynamics with ODE and use it as the prediction to test whether by moving from one configuration to another as collision happens. This is the most expensive part of our implementation.

Similar to RRT\* and PRM\*, FMT\* also requires to explicitly specify radius when considering neighboring samples to achieve asymptotic optimality, which is given by the (3) in [40]. So we can write

$$r_n = \gamma \left( \frac{\log(n)}{n} \right)^{1/d}, \quad (2)$$

for some positive  $\gamma$ . In our implementation, we normalized all joint positions in the range of  $[0, 1]$  to do uniform

```

1. if  $\theta_{goal} \in X_{free}$ 
2.  $V \leftarrow \{\theta_{init}\} \cup \text{SampleFree}(n)$ ;  $E \leftarrow \emptyset$ 
3.  $V_{unvisited} \leftarrow V \setminus \{\theta_{init}\}$ ;  $V_{open} \leftarrow \{\theta_{init}\}$ ,  $V_{closed} \leftarrow \emptyset$ 
4.  $z \leftarrow \theta_{init}$ 
5.  $N_z \leftarrow \text{Near}(V \setminus \{z\}, z, r_n)$ 
6.  $\text{Save}(N_z, z)$ 
7. while  $z \neq \theta_{goal}$  do
8.    $V_{open,new} \leftarrow \emptyset$ 
9.    $\Theta_{near} = N_z \cap V_{unvisited}$ 
10.  for  $\theta \in \Theta_{near}$  do
11.     $N_\theta \leftarrow \text{Near}(V \setminus \{\theta\}, \theta, r_n)$ 
12.     $\text{Save}(N_\theta, \theta)$ 
13.     $\Psi_{near} \leftarrow \Theta_{near} \cap V_{open}$ 
14.     $\psi_{min} \leftarrow \arg \min_{\psi \in \Psi_{near}} \{J(\psi) + \text{Cost}(\psi, \theta)\}$ 
15.    if  $\text{CollisionFree}(\psi_{min}, \theta)$  then
16.       $E \leftarrow E \cup \{(\psi_{min}, \theta)\}$ 
17.       $V_{open,new} \leftarrow V_{open,new} \cup \{\theta\}$ 
18.       $V_{unvisited} \leftarrow V_{unvisited} \setminus \{\theta\}$ 
19.       $J(\theta) = J(\psi_{min}) + \text{Cost}(\psi_{min}, \theta)$ 
20.    end if
21.  end for
22.   $V_{open} \leftarrow (V_{open} \cup V_{open,new}) \setminus \{\theta\}$ 
23.   $V_{closed} \leftarrow V_{closed} \cup \{\theta\}$ 
24.  if  $V_{open} = \emptyset$  then
25.    return Failure
26.  end if
27.   $\theta = \arg \min_{\psi \in V_{open}} \{J(\psi)\}$ 
28. end while
29. return Path( $\theta, T = (V_{open} \cup V_{closed}, E)$ )
30. end if

```

ALGORITHM 1: FMT\*.

sampling. Then compute a conservative bound of  $\gamma$  using Monte Carlo simulation in order to find the  $d$ -dimensional Lebesgue measure of collision-free configurations with respect to all possible configurations.

Before trying to obtain collision-free samples, a decision-making module will determine which arm is supposed to move and call the corresponding FMT\*, because the two arms are slightly different in terms of transformations and kinematics chain [41].

Under this path planning with robot dynamics framework, joint controls at each time step can also be obtained as a byproduct of doing collision checking when trying to connect two samples. After a feasible path is computed, we can either apply the joint controls explicitly at each time step or through feedback control to execute the plan as a trajectory following problem. Since the first option is open loop, the error could accumulate over time. So, we will use PD controllers for each joint to track the desired trajectory in joint space when executing the motion plan autonomously in the simulator and send the joint positions to the Raven computer via internet if real robot motion is also needed to perform.

**2.4. Spring-Damper Model for Haptic Force Feedback.** In this section, we present the haptic force feedback mechanism. The use of haptic devices in teleoperated surgical robots

has the potential of providing both cutaneous (tactile) and kinesthetic (force) information during exploration or manipulation of an object or environment. To the best of the authors' knowledge, even the latest commercial surgical system (da Vinci Xi) does not have haptic feedback feature. In robotic surgery, haptic feedback is useful in teleoperated palpation [42, 43]. Beyond this application, we expect that haptic feedback also can provide extra but crucial information to the operator about the status of the system when some uncertain events happen and before the errors are accumulated to some degree that the system is taken to an emergency stop. For human perception, our haptic rendering loop in the simulator also runs at 1000 Hz, otherwise, the user may perceive force discontinuities and a loss in fidelity [44].

We send the ROS published joint states in the Raven computer to the simulator through the network. From the physics engine (ODE) thread, we extract the joint velocities. We compute the end-effector velocities by using spatial manipulator Jacobian transformation:

$$[\nu \ \omega]^T = J\dot{\theta}. \quad (3)$$

The end-effector position is computed through the forward kinematics chain for both the robot and the model using the joint positions, as shown in

$$p = f(\theta), \quad (4)$$

where  $f$  indicates the forward kinematics chain of the robot [41]. Then, the haptic force provided to the operator is given by

$$F = \begin{cases} K_p \|p_{\text{model}} - p_{\text{robot}}\| + K_d \|v_{\text{model}} - v_{\text{robot}}\| \\ 0.05, \text{ if } \|p_{\text{model}} - p_{\text{robot}}\| > \text{tol}. \end{cases} \quad (5)$$

And the force direction applied to the haptic device is given by

$$d = \frac{p_{\text{model}} - p_{\text{robot}}}{\|p_{\text{model}} - p_{\text{robot}}\|}. \quad (6)$$

In this setup, if an adverse scenario happens, or the robot moves in an unexpected way, the haptic device will provide haptic cues to the operator. This provides awareness of impeding hazards, enabling the operator to take action or correct the robot behavior based on the internal model of the simulator.

**2.5. Safety Hazard Injection.** The Safety Hazard Injection engine in our simulator uses software-based fault injection techniques to recreate safety hazards observed during real surgical procedures. This enables evaluation of surgeon performance and response to safety hazards and prepares them for the best response actions to take in case of incidents.

Based on our preliminary review of almost 1500 accident reports on the da Vinci surgical system from the FDA MAUDE database, we identified three categories of common safety hazard scenarios as shown in Table 1. We simulate these scenarios by injecting faults into the Raven control software during the training scenarios. The possible causes

TABLE 1: Three common safety hazard scenarios, with corresponding examples from real incidents reported to the FDA MAUDE database.

Safety hazard scenario (outcome)	Unsafe control action example	Possible causal factors (accidental failures)	Raven-II simulation Target software module	Impact (clinical scenarios for safety training) [example]
System temporarily unavailable ( <i>recoverable system error</i> )	A user command is provided but not followed by the robot.	Improper operator actions or console control malfunctions	Network-layer thread (network_layer)	Restart the system [MAUDE 3293519] Troubleshoot error contact manufacturer
System permanently unavailable ( <i>nonrecoverable system error</i> )	A motor command is provided by the robot control, but it is not followed by the motors.	Sensor (encoder) failure	Control thread (get_USB_packet)	Convert the procedure [MAUDE 2663924] Reschedule [MAUDE 3275500] Report to manufacturer
Unintended movement of robotic arms ( <i>sudden jump</i> )	A command is provided by the robot control to motors while the calculated next position is at large distance (big jump) from current position.	Actuator failures	Control thread (put_USB_packet)	Puncture of artery [MAUDE 1590517] Bleeding of uterine tube [MAUDE 2120175]

of hazards may include accidental faults in robotic hardware or software, unintentional human operator errors, or intentional malicious attacks to the control system of the robot. For each safety hazard, Table 1 shows the potential accidental causes (column 3) and impact on patients and surgical team (column 5) based on representative examples from the real incidents reported to the FDA MAUDE database. The patient impacts represent clinical scenarios and response actions on which the robotic surgeons should be trained on.

The Safety Hazard Injection Engine consists of customized modules for (a) retrieving hazard scenarios, (b) generating software fault injection campaign and selecting fault injection strategy, (c) conducting fault injection experiments, and (d) logging and collecting data in an automated fashion [12]. The *Injection Controller* is responsible for starting, stopping, and automating the fault injection campaign. In a normal campaign execution, a Safety Hazard Scenario Library constructed based on the analysis of adverse events is accessed to retrieve the list of desired hazard scenarios. Then causal factors leading to each desired hazard scenario are simulated by selecting the fault injection parameters. Each hazard scenario includes a possible unsafe control action and a list of potential causal factors. An example unsafe control action would be a motor command is provided by the control software when there is a mismatch between the software state and hardware state of the robot. Faulty communication between software and hardware (e.g., through USB) is an example causal factor that might lead to such unsafe control action (see the third example in Table 1). Based on the causal factors involved in each hazard scenario, the analysis of Raven source code and software/hardware architecture, the *Fault Injection Strategies* module retrieves information on software functions which can most likely mimic the causal factors leading to the safety hazard as well as the key variables in those functions and their normal operating ranges. This information is translated to the parameters to be used by the fault injectors for simulating potential causal factors. The fault injection parameters include the location in the software function, the trigger or condition under which the fault should be injected, and the target variables to be modified by the injection (see column 4 of Table 1). Finally, the appropriate software-implemented *Fault Injectors* and the robot software are executed to conduct a fault injection experiment during a training scenario. At the end of each injection run, the injection parameters and data are collected for further analysis. For a more detailed description of Safety Hazard Injection Engine, refer to our previous work [12].

### 3. Results and Discussion

We present the experimental evaluations of the proposed hardware-in-the-loop simulator in this section. There are mainly two parts: (1) simulating safety hazards and their detection and (2) motion planning in a training environment with FRS model.

#### 3.1. Fault-Free and Contact-Free Run for Model Verification.

In this work, the robot dynamics is modeled for all 7 DOF, in

ODE environment, compared to only 3 DOF were modeled in [45]. Our dynamic model is from joint torque to joint states, ignoring the motor dynamics and cable tensions. One reason for doing this is that the cable coupling introduces uncertainties in the model and nonuniformity of cable tension behaviors. In our experiment setting, we tighten the cables as tight as possible before the testing. Another reason is that the system is running at a 1000 Hz frequency, which gives very little margin for heavy computation and introducing even a small time delay that will cause system instability. Although the dynamics calculation is done on the Surgeon Console machine rather than the Raven computer, we still do not want to violate the timing constraints in each control loop.

The joint torque vector  $\tau$  is the controller output based on desired joint position obtained through inverse kinematics and current joint position. PD controllers are used for joints 1, 2, 4, 5, 6, and 7, and PID is used for joint 3, which is the tool insertion translational joint. A set of manually tuned PID gains make the system closely track the desired joint positions while keeping the joint torque/force  $\tau$  within certain bounds. This means that the model is behaving like the robot rather than a system which has low damping and is fast enough to track reference signal. Figure 4 shows the trajectories for arbitrary motions provided by the operator (black), the internal physics engine (dynamic model) calculations (red), and the real robot trajectory (blue) for the first five joints on the left robot arm (the right arm is identical to the left arm in terms of modeling and control). Through forward kinematics chain, one can obtain the end-effector position error. Figure 4 shows the different portions of the trajectories (separated with dashed lines) corresponding to different teleoperation scaling factors, respectively, ranging from 0.05 to 0.2 with the spacing interval of 0.05. With larger motion scaling factors, the error also increases, because the modeling error for joints 4 and 5 are more sensitive to the scaling factor. These results verify the accuracy of the modeling in ODE environment, compared to the real robot trajectory.

**3.2. Fault Injection to Robot Software.** Many of the hazard scenarios shown in Table 1 may cause unexpected instrument movements and sudden jumps. In this section, we use the safety hazard injection engine to trigger the faulty commands at network layer and software hardware communication layer as in [8]. More specifically, to simulate the resulting safety hazard scenarios, we corrupt the motor commands sent to the robot hardware and the Omni commands sent to the Raven computer, as indicated by numbers in Figure 2.

In the experimental setup, we disabled the collision checking in ODE, that is, we did not consider the case where hard collision happens and will affect the robot dynamics too much. We inject periodic faults to (i) the first (shoulder) joint of the robot (which has stronger cable in the Raven-II) and (ii) the motion command data in network layer transmitting from the local machine to Raven computer, while the simulator receives the original “clean” Omni input. In the second case (in Figure 2), it is obvious that receiving the corrupted

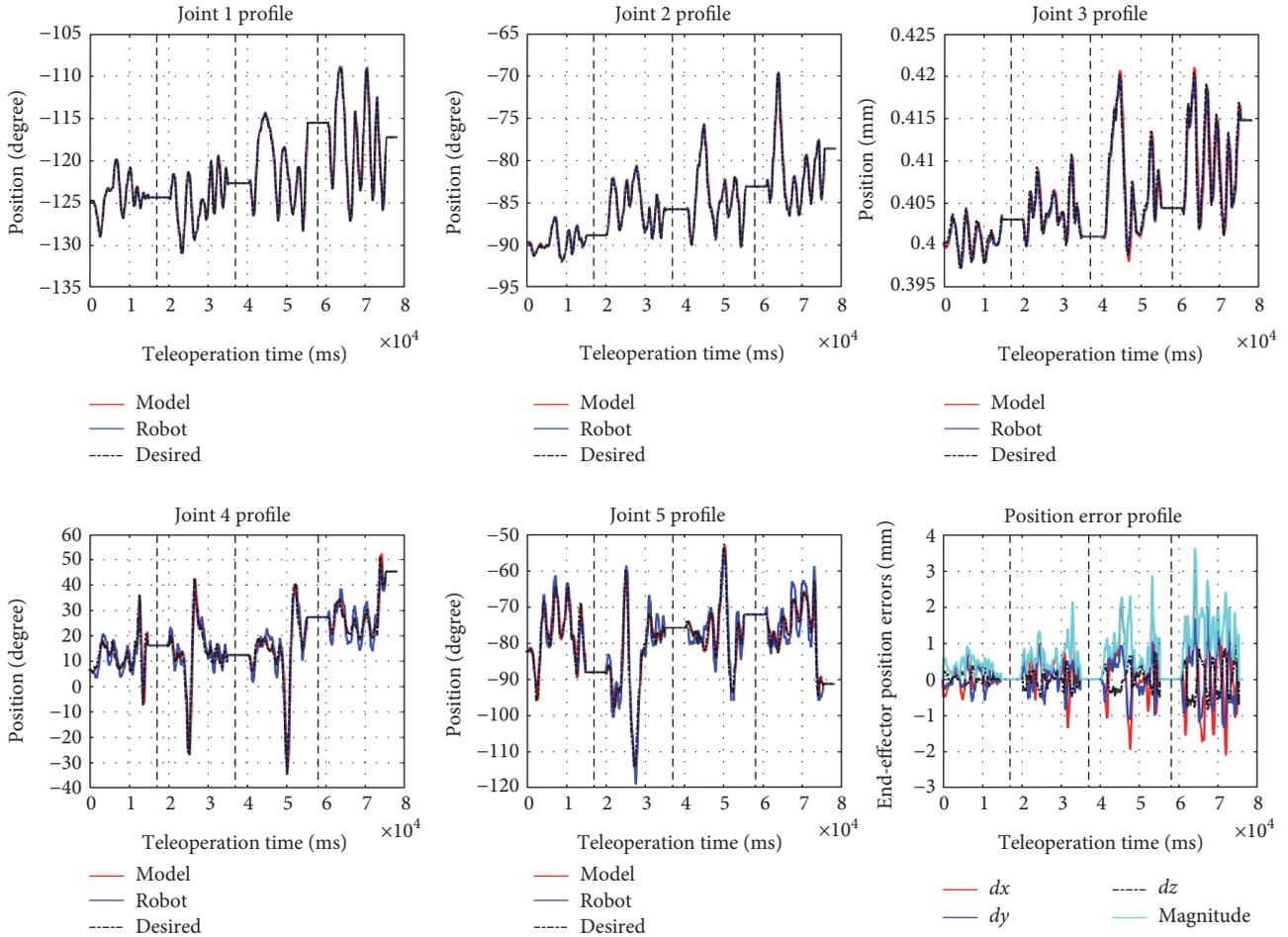


FIGURE 4: Comparison of the model and robot running data (up to 5 joints) and end-effector position error of ( $2.43 \pm 1.72$ ).

desired state data, the robot will follow the incorrect trajectory and end up deviating from the trajectory expected by the operator. From the fault-free run result as shown in Figure 4, we set the threshold of triggering the haptic force feedback when the end-effector positions between the model and the robot deviate more than 3 mm. In this section, we mainly focus on simulating and analyzing the resulting adverse events in the first case (i.e., injection of periodic faults to the motor commands).

**3.2.1. Simulation of Sudden Jump.** In robotic surgery, many reported adverse events can be classified as unexpected joint motion in a small time interval, that is, sudden jump (the third scenario in Table 1). Although the causality can be many to one, we are able to reproduce this kind of adverse events and expose the surgeon during training phase by using our hardware-in-the-loop simulator incorporated with the safety hazard injection engine. We use haptic force feedback to provide information to the operator immediately, so that they can respond to the adverse events as quickly as possible, by emergency actions such as release the foot pedal to disengage the robot and triggering motor breaks (to avoid patient injuries). To simulate robot jump, during the teleoperation running mode, we inject constant motor command (can be

zero or nonzero but within the valid range of motor's DAC command) to the shoulder joint at a specified time period. The underlying reason for the jump is the accumulation of position errors because the controller has to generate large torque commands to track the desired position once the robot goes back to the nominal run.

Figure 5 shows the result of our hardware-in-the-loop simulator running with the fault injector. In this scenario, every 8 second after pedal down (teleoperation mode), the safety hazard injection engine corrupts the motor command sent to the shoulder joint and keeps the fault active for 300 cycles (300 ms). One can observe that the sudden jump behavior happened in joint 1 profile in Figure 5. The sudden jumps can happen many times (in this experiment, 4 times), while the operator may not notice them since the duration is quite short (a few milliseconds). Such abrupt jumps if only happen a few times during the procedure, they will leave no impression to the operator and he may even think that it is his own mistake. However, the sudden movements/jumps may happen due to hardware problems (see Table 1). The robot has the safety mechanisms to monitor the robot status and detect such faults, but in our fault injection experiments, we demonstrated the robot can jump frequently without triggering the robot's safety mechanisms [8, 12] (e.g., the robot

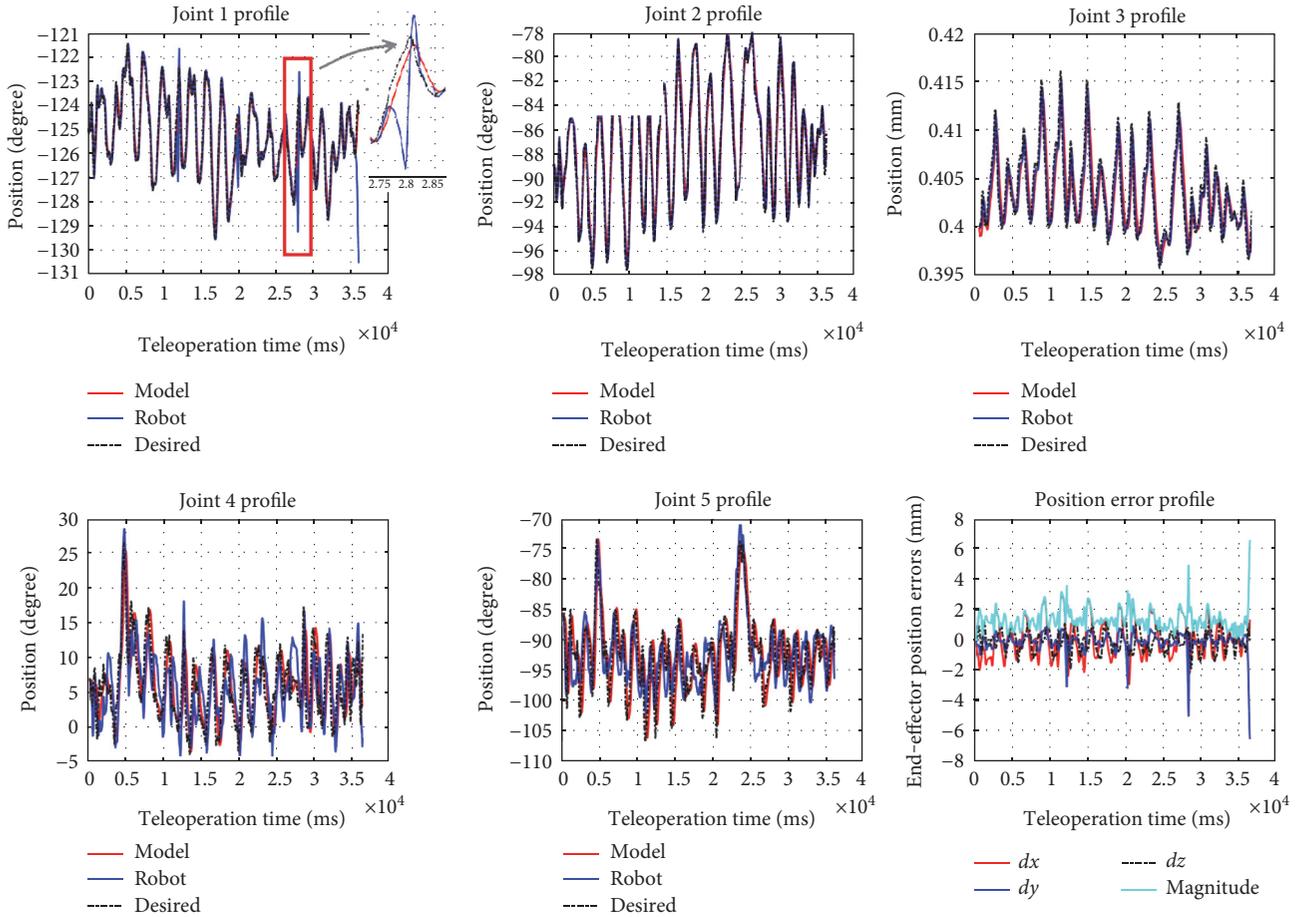


FIGURE 5: Robot and model trajectories during fault injection are enabled (with teleoperation scaling factor of 0.1).

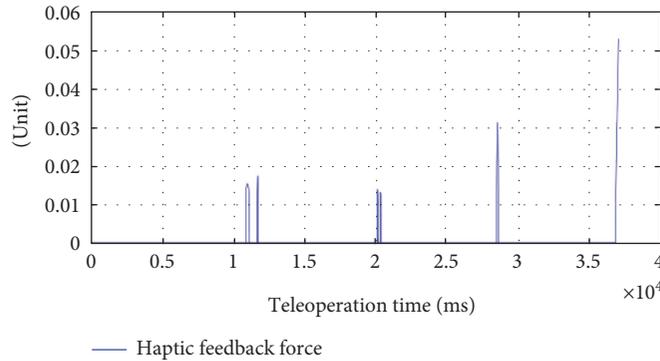


FIGURE 6: Haptic force feedback on the Omni device during fault injection.

stopped at the last jump due to the computed motor control is beyond the limit).

Figure 6 shows the magnitude of the haptic force feedback provided to the Omni device using (5). The results show that we captured the adverse events exactly at the times the fault injection was performed and provided the feedback to the user in time. The haptic force is being saturated in a range that it will not interfere the normal teleoperation due to the passivity and high damping of the human operator. When a surgeon faces such a scenario in real surgery, possible

mitigation strategies include slow down the motions or release the pedal to disengage the master and slave and call the technical help in the hospital (see the last column of Table 1).

3.3. *Test Results on Path Planner.* In this section, we evaluate the performance of the proposed Raven-II simulator with the integration of physics engine and the path planner. We performed all tests locally, that is, the simulator software and the Omni client software are running on the same Windows

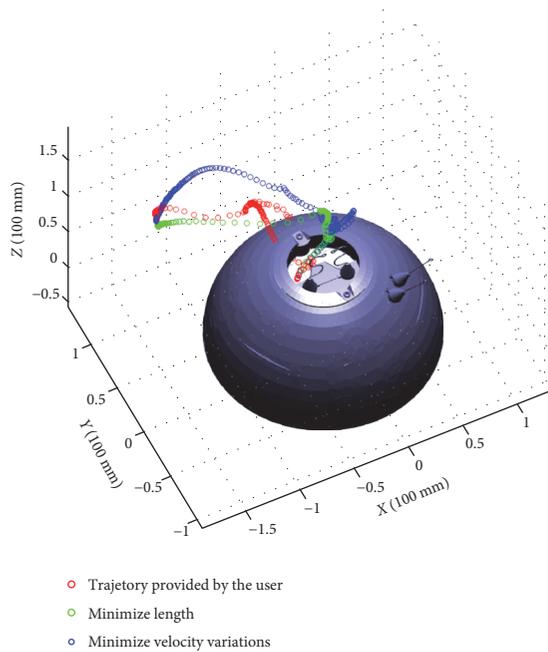


FIGURE 7: Three end-effector trajectories representing user's movement and two different optimization criteria.

7 machine, with Intel Core i7 CPU @ 3.50 GHz, to avoid the time delay caused by long distance communication over the network. And for all tests, the step size used in ODE is 0.001 second.

Initially, the robot arm is placed deeply inside the FRS training module. The test scenario is to achieve the motion sequences of retrieving the robot arm first from the surgical field and then moving forward to reach the cube ring. The opening area of the dome does not coincide with the pivot point (indicated by the yellow sphere in Figure 3) of the robot arm; thus, it is very easy to collide with the dome or loops. The user was asked to activate the planning mode, which means he can directly manipulate the arm to reach the target by any means with collision checking disabled. Once he arrived at the target configuration, the planner will record this configuration and start to run the planning algorithm. We evaluate the performance with respect to different objective functions, the number of collision-free samples, and the algorithm execution time.

Choosing 500 collision-free samples for the path planner, Figure 7 shows the end-effector (mass center of the two jaws) trajectories of user's movement when specifying the target configuration (red), minimizing 6-dimensional path length (green), and minimizing end-effector linear velocity variations (blue), respectively. They are the path tracking results rather than the waypoints for the planning results. We record the waypoints on every other loop time of ODE simulation loop so that they will be shown more clearly in the figure. The density of waypoints indicates the velocity. The sparser means the velocity is larger while more dense waypoints means the velocity is relatively slow. In our test scenario, it is difficult for the novice user

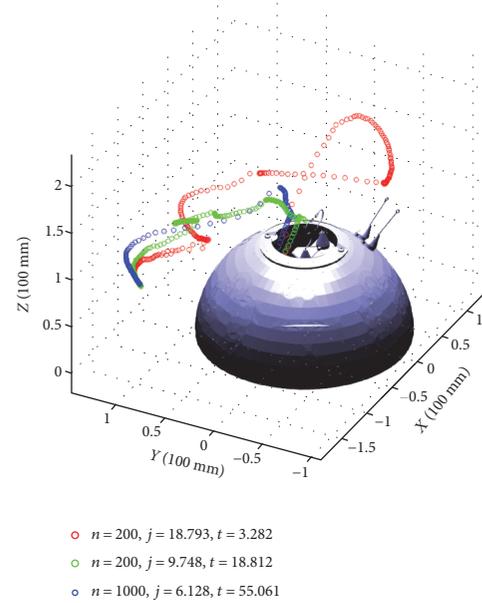


FIGURE 8: Three end-effector trajectories representing returned by FMT\* using different number of samples.

TABLE 2: Performance evaluation result.

Sample number	Planning time (s)	Path tracking time (s)	Returned reference waypoints number	Cost	Success rate (%)
200	3.676	4.705	13.5	21.9	90
500	15.935	4.277	14.3	11.1	85
1000	47.708	3.909	11.1	6.5	85
2000	216.373	3.688	11.1	5.7	75

to figure out a collision-free path to reach the cube ring's location while manipulating the robot arm. So in our interactive path planning setting, collision is allowed just simply for the user to quickly reach the target configuration. By choosing  $\eta_1 = 1$  and  $\eta_2 = 0$ , we got a path with smaller length (planner returns 36 waypoints), while by choosing  $\eta_1 = 0$  and  $\eta_2 = 1$ , we got a path with smaller velocity variations (planner returns 54 waypoints).

Next, we compare the planning performance of using a different number of collision-free samples and verify that the algorithm can converge to the optimum with respect to the cost function. In this test scenario, we choose  $\eta_1 = 0.1$  and  $\eta_2 = 0.5$ . As shown in Figure 8, we observe that as the number of samples increases, the cost becomes smaller and the algorithm running time becomes longer due to heavier computation in the "for" loop of the algorithm.

Table 2 shows the algorithm performance with  $\eta_1 = 0.1$  and  $\eta_2 = 0.5$  and autonomous tracking performance. Each test scenario runs the planning algorithm and autonomous tracking 20 times to get the average result. In

Table 2, “returned reference waypoints” means the number of waypoints returned by FMT\*, and going from one position to another is dynamically feasible in the planning phase. However, as the number of samples increases, the path gets shorter, and thus in path tracking, it would be closer to the obstacle (wall of the dome) and result in lower success rate (more chances to collide). One way to resolve this issue is to include an additional term in cost function as maximizing the obstacle clearance distance. In this training setup, the planning time and the execution time are acceptable if using appropriate number of samples to start the algorithm.

#### 4. Conclusions

We have demonstrated a general framework for robot-assisted surgical simulators for a more robust and resilient robotic surgery. With the goal of providing high-fidelity surgical training in simulation, we created a hardware-in-the-loop simulator platform. We integrate the simulator with a physics engine and a state-of-the-art path planning algorithm to help surgeon acquire an optimal sense of manipulating the robot instrumental arm and achieve autonomous motion of the surgical robot. We integrated a safety hazard injection engine integrated with the simulator software to reproduce safety hazards happened in real surgery. A haptic force feedback mechanism was designed to provide surgeon an extra modality of information about the robot status when unexpected motion happens. Delivering the safety alarm to the surgeon by haptics is an efficient way of capturing such occurrences but will need additional human factor studies.

The future work includes providing haptic feedback to guide the operator moving along the preplanned optimal path to perform training tasks, for example, use of gripper to grasp a ring. Since the current teleoperation mechanism uses incremental motion of the master and maps to the end-effector of the slave robot to resolve the two different workspaces of master and slave (thus using the pedal to disengage/engage and reconfigure the master), we can provide visual or audio cues to the operator once it approaches the workspace boundary of the haptic device. With haptic force guidance, we believe we can further reduce the need for supervisory by the expert surgeon during training.

#### Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

#### Acknowledgments

This work was partially supported by the National Science Foundation under Award nos. CNS 13-14891 and CNS 15-45069 and a grant through the JUMP-ARCHES (Applied Research for Community Health through Engineering and Simulation) program for addressing safety and reliability of surgical robots. This project was carried out at the Health Care Engineering Systems Center at Illinois.

#### References

- [1] G. Spinoglio, *Robotic Surgery: Current Applications and New Trends*, Springer, 2015.
- [2] M. Passiment, H. Sacks, and G. Huang, *Medical Simulation in Medical Education: Results of an AAMC Survey*, 2011.
- [3] Y. Okuda, E. O. Bryson, and e. al., “The utility of simulation in medical education: what is the evidence?” *Mount Sinai Journal of Medicine*, vol. 76, no. 4, pp. 330–343, 2009.
- [4] <http://surgalscience.com/systems/lapsim/haptic-system/>.
- [5] <http://www.mimicsimulation.com/products/dv-trainer/>.
- [6] <http://www.simulatedsurgicals.com/ross.html>.
- [7] H. Alemzadeh, J. Raman, N. Leveson, Z. Kalbarczyk, and R. K. Iyer, “Adverse events in robotic surgery: a retrospective study of 14 years of FDA data,” *PLoS One*, vol. 11, no. 4, pp. 1–20, 2014.
- [8] H. Alemzadeh, D. Chen, X. Li, T. Kesavadas, Z. T. Kalbarczyk, and R. K. Iyer, “Targeted attacks on teleoperated surgical robots: dynamic model-based detection and mitigation,” in *The 46 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 395–406, 2016.
- [9] T. Bonaci, J. Herron, T. Yusuf, J. Yan, T. Kohno, and H. J. Chizeck, “To Make a Robot Secure: An Experimental Analysis of Cyber Security Threats against Teleoperated Surgical Robots,” 2015, <http://arxiv.org/abs/1504.04339>.
- [10] F. F. Bilotta, S. M. Werner, S. D. Bergese, and G. Rosa, “Impact and implementation of simulation-based training for safety,” *Scientific World Journal*, vol. 2013, Article ID 652956, 6 pages, 2013.
- [11] H. Alemzadeh, D. Chen, Z. Kalbarczyk et al., “A software framework for simulation of safety hazards in robotic surgical systems,” in *Special Issue on Medical Cyber Physical Systems Workshop*, vol. 12, p. 4, 2015.
- [12] H. Alemzadeh, D. Chen, A. Lewis et al., “Systems-theoretic safety assessment of telerobotic surgical systems,” in *the 34th International Conference on Computer Safety, Reliability, and Security (SAFECOMP)*, pp. 213–227, 2015.
- [13] <http://applieddexterity.com/>.
- [14] D. Glassman, L. White, A. Lewis et al., “Raven surgical robot training in preparation for da Vinci,” *Medicine Meets Virtual Reality*, vol. 21, pp. 135–141, 2014.
- [15] M.-C. Hsueh, T. K. Tsai, and R. K. Iyer, “Fault injection techniques and tools,” *Computer*, vol. 30, no. 4, pp. 75–82, 1997.
- [16] Y. Mo, T. H.-J. Kim, K. Brancik et al., “Cyber-physical security of a smart grid infrastructure,” in *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, 2012.
- [17] M. LeMay and C. A. Gunter, “Cumulative attestation kernels for embedded systems,” in *Proceedings of the 14th European Conference on Research in Computer Security*, pp. 655–670, 2009.
- [18] H. Lin, A. Slagell, Z. Kalbarczyk, P. Sauer, and R. Iyer, “Runtime semantic security analysis to detect and mitigate control-related attacks in power grids,” in *IEEE Transactions on Smart Grid*, 2016.
- [19] K. Koscher, A. Czeskis, F. Roesner et al., “Experimental security analysis of a modern automobile,” in *IEEE Symposium on Security and Privacy (SP)*, pp. 447–462, 2010.
- [20] T. P. Vuong, G. Loukas, and D. Gan, “Performance evaluation of cyber-physical intrusion detection on a robotic vehicle,” in *IEEE International Conference on Computer and Information*

- Technology; Ubiquitous Computing and Communications; Dependable*, 2015.
- [21] T. P. Vuong, G. Loukas, D. Gan, and A. Bezemskij, "Decision tree-based detection of denial of service and command injection attacks on robotic vehicles," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015.
- [22] Y. Dong, N. Gupta, and N. Chopra, "On content modification attacks in bilateral teleoperation systems," in *American Control Conference (ACC)*, 2016.
- [23] Z. Ju, C. Yang, Z. Li, L. Cheng, and H. Ma, "Teleoperation of humanoid baxter robot using haptic feedback," in *International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, 2014.
- [24] H. H. King, B. Hannaford, K. W. Kwok et al., "Plugfest 2009: global interoperability in telerobotics and telemedicine," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [25] O. Weede, H. Mönnich, B. Müller, and H. Wörn, "An intelligent and autonomous endoscopic guidance system for minimally invasive surgery," in *IEEE International Conference on Robotics and Automation*, 2011.
- [26] A. A. Navarro, A. Hernansanz, E. A. Villarraga, X. Giralt, and J. Aranda, "Automatic positioning of surgical instruments in minimally invasive robotic surgery through vision-based motion analysis," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.
- [27] A. Krupa, J. Gangloff, M. de Mathelin et al., "Autonomous retrieval and positioning of surgical instruments in robotized laparoscopic surgery using visual servoing and laser pointers," in *IEEE International Conference on Robotics & Automation*, 2002.
- [28] D.-L. Chow, R. C. Jackson, M. C. Cavusoglu, and W. Newman, "A novel vision guided knot-tying method for autonomous robotic surgery," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2014.
- [29] B. Kehoe, G. Kahn, J. Mahler et al., "Autonomous multilateral debridement with the Raven surgical robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [30] D. Hu, Y. G. B. Hannaford, and E. J. Seibel, "Path planning for semi-automated simulated robotic neurosurgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.
- [31] J. Xu, V. Duindam, R. Alterovitz, and K. Goldberg, "Motion planning for steerable needles in 3D environments with obstacles using rapidly-exploring random trees and back-chaining," in *International Conference on Automation Science and Engineering (CASE)*, 2008.
- [32] S. Patil, J. Burgner, R. J. W. III, and R. Alterovitz, "Needle steering in 3-D via rapid replanning," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 853–864, 2014.
- [33] N. Chentanez, R. Alterovitz, D. Ritchie et al., "Interactive simulation of surgical needle insertion and steering," *ACM Transactions on Graphics (Proceedings SIGGRAPH)*, vol. 28, no. 3, pp. 88:1–88:10, 2009.
- [34] X. Li, H. Alemzadeh, D. Chen, Z. Kalbarczyk, R. K. Iyer, and T. Kesavadas, "A hardware-in-the-loop simulator for safety training in robotic surgery," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016)*, 2016.
- [35] <http://www.ode.org/>.
- [36] J. M. Hsu and S. C. Peters, "Extending open dynamics engine for the DARPA virtual robotics challenge," in *Simulation, Modeling, and Programming for Autonomous Robots*, 2014.
- [37] S. Peters and J. Hsu, "Comparison of Rigid Body Dynamic Simulators for Robotic Simulation in Gazebo," 2014.
- [38] <http://frsurgery.org/training-model/>.
- [39] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [40] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: a fast marching sampling-based method for optimal motion planning in many dimensions," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 883–921, 2015.
- [41] *Kinematic Analysis of the Raven-II Research Surgical Robot Platform (REV: 9-Mar-2015)*.
- [42] L. N. Verner and A. M. Okamura, "Force & torque feedback vs force only feedback," in *EuroHaptics Conference*, 2009.
- [43] J. C. Gwilliam, M. Mahvash, B. Vagvolgyi, A. Vacharat, D. D. Yuh, and A. M. Okamura, "Effects of haptic and graphical force feedback on teleoperated palpation," in *IEEE International Conference on Robotics and Automation*, 2009.
- [44] *OpenHaptics Toolkit Programmers Guide*.
- [45] M. Haghhighipناه, Y. Li, M. Miyasaka, and B. Hannaford, "Improving position precision of a servo-controlled elastic cable driven surgical robot using unscented Kalman filter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

## Research Article

# Iterative Learning Impedance for Lower Limb Rehabilitation Robot

Chenhui Guo,<sup>1</sup> Shuai Guo,<sup>1</sup> Jiancheng Ji,<sup>1</sup> and Fengfeng Xi<sup>2</sup>

<sup>1</sup>*School of Mechatronic Engineering and Automation, Shanghai University, Shanghai, China*

<sup>2</sup>*Department of Aerospace Engineering, Ryerson University, Toronto, ON, Canada*

Correspondence should be addressed to Shuai Guo; [guoshuai@shu.edu.cn](mailto:guoshuai@shu.edu.cn)

Received 3 March 2017; Revised 10 May 2017; Accepted 2 July 2017; Published 1 August 2017

Academic Editor: Lianqing Liu

Copyright © 2017 Chenhui Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper discusses the problem of squatting training of stroke patients. The main idea is to correct the patient's training trajectory through an iterative learning control (ILC) method. To obtain better rehabilitation effect, a patient will typically be required to practice a reference posture for many times, while most of active training methods can hardly keep the patients training with correct posture. Instead of the conventional ILC strategy, an impedance-based iterative learning method is proposed to regulate the impedance value dynamically and smartly which will help patients correct their posture gradually and perform better. To facilitate impedance-based ILC, we propose two objectives. The first objective is to find the suitable values of impedance based on the ILC scheme. The second objective is to search the moderate learning convergence speed and robustness in the iterative domain. The simulation and experimental results demonstrate that the performance of trajectory tracking will be improved greatly via the proposed algorithm.

## 1. Introduction

Stroke has been the third major cause of permanent disability or death around the world. According to statistics, in China, the incidence of stroke is 1.82%; up to 10.36 million adults, over 40 years of ages, suffer stroke [1]. Retraining movement is a critical part of recovery for the stroke patients, and walking and other standing dynamic balance activities are typically very high on the list of goals for the patients [2]. To make stroke patients regain motion ability and release the burden of physical therapists, many lower limb rehabilitation robots are developed. Among them, Lokomat is a typical representative, which is a kind of external skeletal type lower limb rehabilitation robot, taking hybrid force-position control strategy, is patient-driven, and enables the patient to accomplish free walking movement [3]. The KineAssist is a wheeled mobile robot for gait and balance training, which allows patients to walk freely and provides balance assistance and weight support [4, 5]. Both of them are highly praised and have achieved great results in the rehabilitation field. Gait training is an important section of lower rehabilitation

robot, which attracts a lot of researcher's attention. On the opposite, although squatting training is an indispensable section, it is rarely mentioned. Instead of focusing on gait training, we put emphasis on squatting training in this paper, necessary for improving hip and thigh power production [6]. The underlying problem is that it is hard to adapt to different patients and ensure the accuracy of tracking and the safety and comfort of patients and, meanwhile, give necessary assistance to patients.

During rehabilitation training, the robots have to have direct interaction with humans and safety is a critical concern. To ensure the robot's safe interaction with an unknown environment, a form of impedance control was first raised by Hogan in 1985, based on the idea that neither position nor force should be controlled, but rather the dynamic relation between the two [7]. A patient-driven training strategy requires interactive robot-patient control and is mostly achieved by the use of impedance control. Various researches on impedance control for rehabilitation robots have been studied [8–10]. By adjusting the impedance of the robotic rehabilitation devices, the behavior of the robot can be

adjusted from very stiff to very compliant. The robot behavior can be made more compliant if the patient is slightly impaired so that the patient can contribute more voluntary effort in the robotic training process. Similarly, the robot behavior can be made stiffer if the patient is unable to achieve the required degree of motion during the robotic training process [11]. Using impedance control is helpful to improve the security and comfort of human-robot interaction, a strategy widely proved.

Iterative learning control (ILC) has become one of the most effective control methodologies in dealing with repeated tracking control problems or periodic disturbance rejection problems [12]. The notion of ILC is that the performance of a system that executes the same task multiple times can be improved by learning from previous execution (trials) [13]. Meanwhile, many ILC algorithms have been proposed to design a purely feedforward action depending solely on the previous control performance [14–17]. The iterative learning controller, a feedforward controller, generates an improved tracking signal over a specific trajectory utilizing past control results to the plant. Since modeling errors are unavoidable, the real ILC system may violate its convergence condition, although the ILC satisfies the condition for a nominal plant model [18, 19]. In practice, a robust control is usually imported, along with the ILC, for system robustness enhancement and better tracking performance [20, 21]. Ahead of the application of ILC, a feedback controller is typically implemented to act as a prestabilizer which will ensure the closed-loop stability and suppress exogenous disturbance by learning from previous iterations, while the iterative learning controller provides improved tracking performance over a specific trajectory utilizing past control results. In the past decade, ILC is introduced into rehabilitation field owing to its learning characteristic. Due to the characteristic of gait motions, Joonbum and Masayoshi proposed a gait rehabilitation strategy that the assistive torque in the current stride is calculated based on the information from the previous strides, inspired by an iterative learning algorithm [22]. RUPERT, an exoskeleton robot used for assisting rehabilitation of arm functions, which has a closed-loop controller combining a PID-based feedback controller and an iterative learning controller based on a feedforward controller, is designed to assist in repetitive therapy tasks related to activities of daily living [23]. Recently, Freeman et al. developed an FES-based upper limb rehabilitation system which can adjust FES signals according to subject's tracking performance through ILC [24]. Joonbum and Masayoshi developed wearable lower limb rehabilitation robots for gait training which will provide smart assistive torque for patients with the help of ILC. Both RUPERT and Freeman et al.'s robots put emphasis on upper limb rehabilitation and combined a feedback controller with ILC to make the system stable.

In this paper, an impedance-based ILC method is proposed and analyzed for the squatting training of stroke patients in the iterative domain and time domain. The method is to correct patient's training trajectory by integrating the ILC scheme with the value of impedance. Specifically, the correction of the training trajectory input for the rehabilitation robot controller is derived by learning the proper



FIGURE 1: Overview of the rehabilitation system.

impedance value via ILC. By learning the past trajectory tracking information, the proposed ILC method is able to gradually improve the performance of trajectory tracking, and specific training condition of different individuals can be obtained. The convergence and effectiveness of the proposed methods are validated through the results of case studies via simulations and the experiments.

## 2. Lower Limb Rehabilitation Robot System Description

**2.1. Hardware Description.** As shown in Figure 1, the robot system is mainly composed of omnidirectional mobile chassis (OMC) and body weight support (BWS) system. The OMC consists of two passive wheels and two active wheels, the former ones are installed in the front of the robot which can move in all directions and the latter ones are laid out in the back of the robot. Each active wheel is driven by two independent servo motors, one for driving and the other one is used for steering. Encoders are mounted on each motor shaft to record the rotate angle through which the position of the robot can be calculated. Actually, patients would be divided into different groups taking into account their disability level. The disability level of patients suffering from stroke impairments varies from person to person and also for the same patient during the course of rehabilitation. Patients in high disability level may unable to stand up, not to speak of training, under normal gravity environment. Hence, the BWS system is designed to provide 0–100% body weight support, which can relieve the burden of patient's legs. Furthermore, the BWS system contains 3-DOF (marked as ①, ②, and ③ in Figure 1) to satisfy the demands of free walking and each DOF can be restricted by the locking mechanism. The BWS system is installed on the guide rail, driven by an independent servo motor, which makes it to have one more translational DOF in vertical direction.

At the end of the BWS system, two six-axis force/torque sensors, manufactured by ADI, are installed to record the

human-robot interaction force. Forces can be measured of up to 200 N applied in the horizontal plane with a resolution of 0.0122 N. The human subjects will be required to wear a specially designed belt which is strapped on the force sensors tightly. In this case, the end-effector of the robot and human subject is almost overlapped, so we hypothesized that the end-effector of the patient is the same to the robot's. The patient's task is to repeat squatting according to a given trajectory; in addition, the monitor will provide visual feedback of the tracking performance during the training. In the tracking task, the robot's job is to give the patient certain support, to apply assistance during the tracking task, and to move the patient's position when necessary.

**2.2. Modeling and Linearization.** As shown in Figure 2, it is the geometry of the human and robotic system. During the tracking task, the robot needs to compensate the position of X-direction and Z-direction when the subject squats. Since the motion in Y-direction is unnecessary in this task, the redundant DOF of BWS system are restricted by means of the locking mechanism. Meanwhile, the position of the subject's foot is taken as the origin of the coordinate system, because it is almost fixed on the ground during the training. It is assumed that the subject interacts with the robot by applying a vector of forces and torques at point Q, the interaction point, where the forces and torques are measured by two force sensors.

The combined human-robot interaction dynamic model can be described as

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + G(q) = \tau_d + \tau_h, \quad (1)$$

where  $q$ ,  $\dot{q}$ , and  $\ddot{q}$  represent the vectors of robot's position, velocity, and acceleration, respectively.  $M$  is the system mass matrix,  $V$  is a Coriolis matrix, and  $G$  is the gravitational matrix. The matrix  $\tau_h$  is the vector of interaction torque of the human-robot system. The matrix  $\tau_d$  represents the vector of impedance, the dynamic relationship between human and robot, used to guide the subject's limbs on reference trajectory.

It is well known that accurate robot motion control requires complex nonlinear controllers [25]. Considering the complexity and uncertainty of modeling, it is difficult to design the controllers. To facilitate the design of controller, we linearized the nonlinear human-robot system and described it as state-space representation [26]. The general state-space representation of a linear system is given as follows:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx, \end{aligned} \quad (2)$$

where the term  $A = -M^{-1}V$ , the term  $B = -M^{-1}$ , and the term  $C = \text{diag}(1 \ 1)$ . The input term  $u(t)$  represents the torque signal, and the output term  $y(t)$  represents the velocity vector of interaction point Q.

**2.3. Trajectory Planning.** As there is no specific reference trajectory for tracking, an experiment is conducted by a healthy subject to reveal the law of squatting trajectory. In this

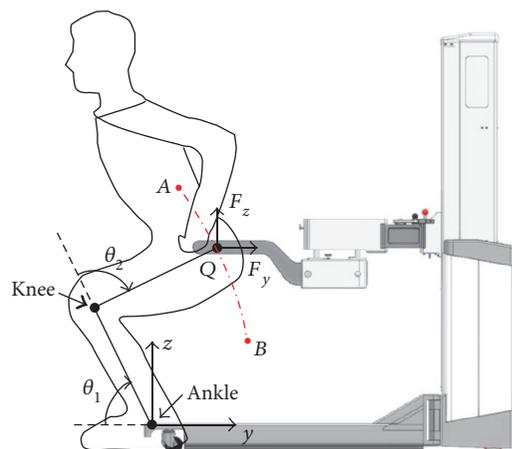


FIGURE 2: The geometry of the dual human and robotic system.

procedure, the robot is set to following mode to respond to the subject's motion, and encoders' data is stored in real time. As shown in Figure 3, it is a diagram of squatting trajectory generated by recorded position parameters. Although the displayed trajectories shown in Figure 3(a) look confusing, all of the trajectories are inside the safety zone. Different from other trajectory tracking tasks, squatting training is very possible to cause falling down or injury which indicates the given trajectory for human subjects should not cross the safety zone. To have a good knowledge of the variation of each training trajectory, three dashes are picked from Figure 3(a). We can see that the three dashes shown in Figure 3(b) are irregular because the test subject's motion is optional and unrestrained. To make the tracking task easy, the reference trajectory should be smooth and simple. So the idea is that the given trajectory is generated according to the test data; meanwhile, position deviation within a certain range is tolerated which will guarantee the compliance during training.

As shown in Figure 4, it is a diagram of the human subject's lower limbs. Points A and B represent two extreme positions of training.  $L_1$ , the distance from joint ankle to joint knee, is the length of shank, and  $L_2$  is the length of thigh. The human joint angle vector is  $\theta = [\theta_1 \ \theta_2]^T$ , where  $\theta_1$  and  $\theta_2$  are the joint angles of ankle and knee, respectively. It is found that the training trajectory of subjects varies from person to person while the variation of joint angles is similar. Therefore, the kinematic formulation of squatting trajectory is given as

$$q_r = f(\theta_r, l_r), \quad (3)$$

where  $q_r$  is a vector that represents the reference training trajectory,  $\theta_r$  is a vector of the reference joint variables, and  $l_r$  is a vector of the kinematic parameters. Given that  $l_r$  can be measured in advance, the trajectory of the end-effector is mainly determined by  $\theta_r$ .

The disability level of stroke patients varies from person to person; some patients may be impaired seriously, while the others are impaired slightly. Therefore, taking into account the different disability levels of patients, the time

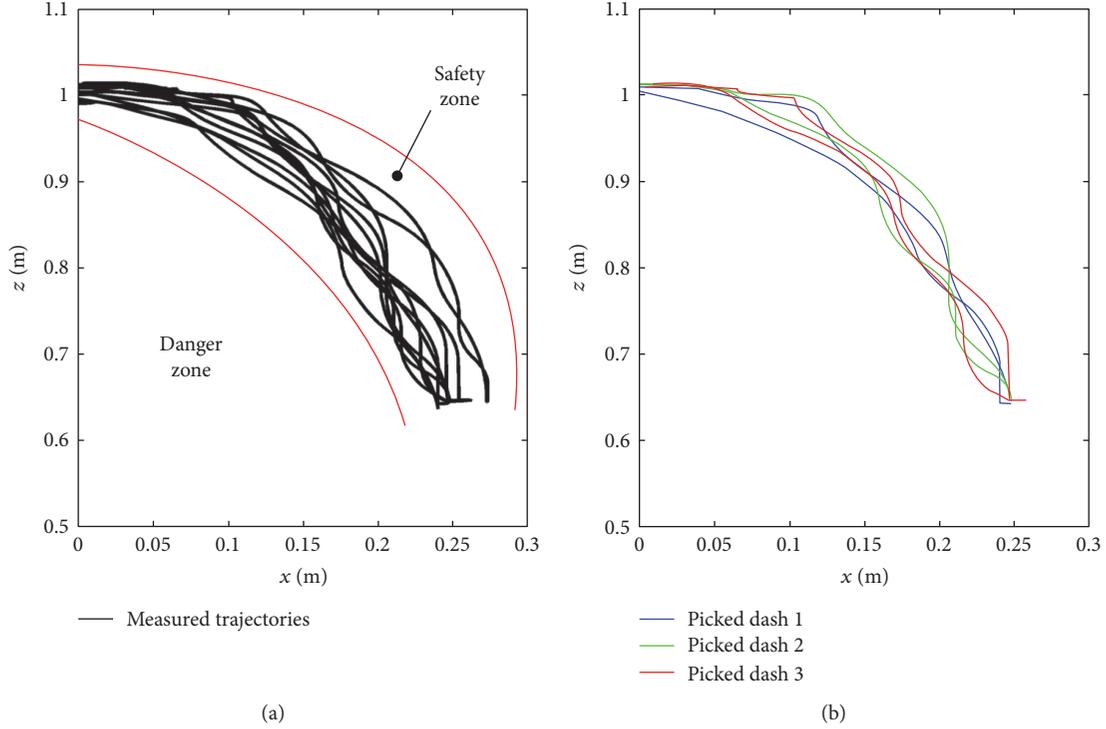


FIGURE 3: (a) The measured trajectories over 10 times. (b) The squatting trajectories at the first (blue dash), second (green dash), and third (red dash) tracking.

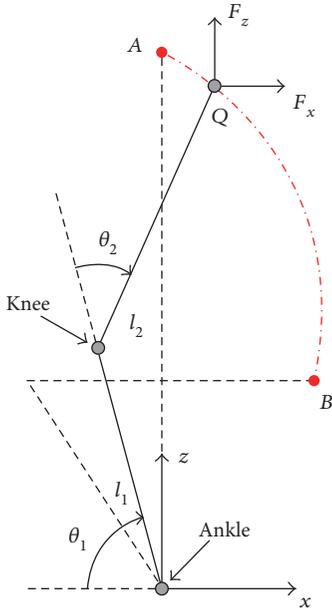


FIGURE 4: Squatting trajectory in the vertical plane.

taken,  $T$ , to travel along the given trajectory, takes a value between 5 and 15 s. Besides, subjects will often be required to track the same trajectory over 20 times. The main idea of this paper is to correct the subjects' posture according to the reference trajectory and make the training process very compliant and comfortable by modifying the term  $\tau_d$  with iterative learning method.

### 3. Impedance-Based Iterative Learning Control

**3.1. Impedance Control.** A widely used impedance model is given as

$$\tau_d = K_k(q_r - q) + K_B(\dot{q}_r - \dot{q}) + K_M(\ddot{q}_r - \ddot{q}). \quad (4)$$

The terms  $K_K = K_K I$ ,  $K_B = K_B I$ , and  $K_M = K_M I$  are the gain matrices, where  $I$  is the identity matrix. If we hope the subject moves freely along the trajectory, the gain  $K_k$  is often set as 0 and the values of  $K_B$  and  $K_M$  are assumed to be positive values to create a natural feel. We can let the robot moves the patient's position along predefined trajectories with setting  $K_k = K_k I$  with the scalar  $K_k > 0$ . The higher the gain  $K_k$  is, the stiffer the system is and vice versa. Therefore, the stiff and compliant characteristic is mainly determined by the gain  $K_k$ .

As mentioned above, the robot's job is to give subjects help when necessary and guarantee a safe interaction. How the impedance gain is set is a tough task. Adaptive impedance control of the rehabilitation robot is a well-established method to modify the robotic assistance in gait training based on the concept of setting the robotic impedance high (low compliance) if little effort or participation is detected and vice versa [11]. However, high impedance will increase the robotic assistance in order to guide the subject's limbs on reference trajectory which will make the subjects uncomfortable and more likely to fall down. Different from adaptive impedance control method, ILC can reduce position error gradually. The object of this paper is to develop a trajectory

corrector that can correct the subject's posture through ILC and improve the subject's tracking performance.

**3.2. Iterative Learning Impedance.** Before the discussion of our method, a brief introduction to ILC is presented. As shown in Figure 5, the formulation of ILC is given as follows. Consider the following linear discrete time-invariant system, that is,

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t), \end{aligned} \quad (5)$$

where  $t$  is the time index,  $x(t) \in R^n$ ,  $u(t) \in R^r$ , and  $y(t) \in R^r$  represent the state, control input, and output, respectively. The vectors  $A$ ,  $B$ , and  $C$  are matrices with corresponding dimensions. The control target is to find a suitable input  $u_i$  (denoted by  $u_r$ ) which produces  $y_i$  that precisely follows a reference trajectory  $y_r$ . The integrator, or  $I$  term, is rarely used for learning function because ILC has a natural integrator action from one trial to the next [13]. Therefore, the PD-type learning law can be given as

$$u_{i+1}(t) = u_i(t) + k_p e_i(t) + k_d \dot{e}_i(t), \quad (6)$$

where  $k_p$  is the proportional gain,  $k_d$  is the derivative gain, and  $e_i = y_r - y_i$  is the tracking error. From (4) and (6), the impedance learning law can be given as

$$\tau_{d_{i+1}}(t) = \tau_{d_i}(t) + \Gamma(\dot{e}_i(t) - Re_i(t)), \quad (7)$$

where  $\Gamma$  and  $R$  are the learning gain matrix.

**3.3. Robustness Analysis.** The robustness will be discussed as follows. In our case, the human subject will return to the starting point at the end of the tracking cycle so the initial state condition remains the same at each iteration. Then, the output trajectory can be estimated in terms of the desired output trajectory and the initial state error.

**Theorem 1:** *The initial condition at each iteration is always the same; that is,  $x_i(0) = x_0$ .*

$$\text{If} \quad 0 < \|I - CB\Gamma\|_{\infty} \leq \rho < 1, \quad (8)$$

the update law (6) ensures that

$$\lim_{i \rightarrow \infty} y_i(t) = y_d(t) + e^{Rt} C \{x_0 - x_d(0)\}, \quad (9)$$

where  $\tau_d$  is simplified as  $y$ .

*Proof.* Let  $u_a(t)$  be a control input

$$y_d(t) + e^{Rt} C \{x_0 - x_d(0)\} = Ce^{At} x_0 + C \int_0^t e^{A(t-\theta)} Bu_a(\theta) d\theta. \quad (10)$$

The problem is equivalent to prove  $\lim_{i \rightarrow \infty} u_i(t) = u_a(t)$ .

We define

$$\delta u_i(t) = u_a(t) - u_i(t). \quad (11)$$

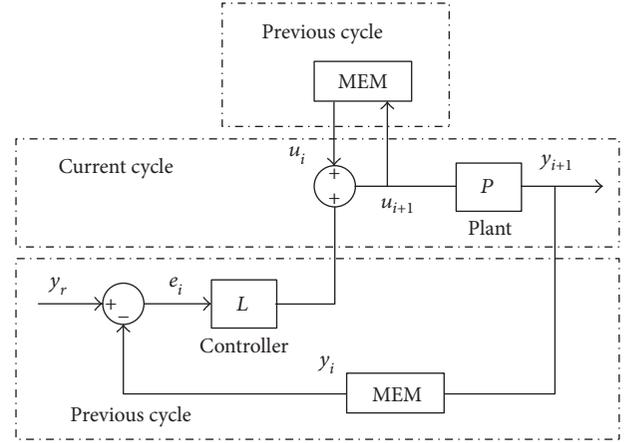


FIGURE 5: Schematic block of ILC.

The main idea of the proof is to show that  $\|\delta u_{i+1}(t)\|_{\lambda} \leq \rho_0 \|\delta u_i(t)\|_{\lambda}$ , where  $0 \leq \rho_0 < 1$ .

From (4) and (9), we obtain

$$\begin{aligned} \delta u_{i+1}(t) &= u_a(t) - u_i(t) - \Gamma(\dot{y}_d(t) - \dot{y}_i(t)) \\ &= (I_r - CB\Gamma)\delta u_i(t) - \Gamma(CA - RC) \int_0^t e^{A(t-\theta)} B \delta u_i(\theta) d\theta. \end{aligned} \quad (12)$$

Taking the norm  $\|\cdot\|_{\infty}$  on both sides of (11), we have

$$\begin{aligned} \|\delta u_{i+1}(t)\|_{\infty} &\leq \|I_r - CB\Gamma\|_{\infty} \cdot \|\delta u_i(t)\|_{\infty} \\ &\quad + \|\Gamma(CA - RC)\|_{\infty} \int_0^t \|e^{A(t-\theta)}\|_{\infty} \|B\|_{\infty} \|\delta u_i(\theta)\|_{\infty} d\theta \\ &= \rho \|\delta u_i(t)\|_{\infty} + h \int_0^t e^{a(t-\theta)} \|\delta u_i(\theta)\|_{\infty} d\theta, \end{aligned} \quad (13)$$

where  $h \triangleq \|\Gamma(CA - RC)\|_{\infty} \cdot \|B\|_{\infty}$ , and  $a \triangleq \|A\|_{\infty}$ .

By multiplying both sides of (12) by  $e^{-\lambda t}$  and taking the norm  $\|\cdot\|_{\lambda}$ ,

$$\begin{aligned} \|\delta u_{i+1}(t)\|_{\lambda} &\leq \max_{0 \leq t \leq T} e^{-\lambda t} \|\delta u_{i+1}(t)\|_{\infty} \\ &\leq \rho \|\delta u_i(t)\|_{\lambda} + h \max_{0 \leq t \leq T} \int_0^t e^{(a-\lambda)(t-\theta)} \max_{0 \leq t \leq T} e^{-\lambda \theta} \|\delta u_i(t)\|_{\infty} d\theta \\ &= \left( \rho + h \frac{1 - e^{(a-\lambda)T}}{\lambda - a} \right) \|\delta u_i(t)\|_{\lambda}. \end{aligned} \quad (14)$$

Assume that  $0 \leq \rho < 1$ , it is possible to choose  $\lambda$  sufficiently large to have

$$\rho_0 = \rho + h \frac{1 - e^{(a-\lambda)T}}{\lambda - a} < 1. \quad (15)$$

Thus,

$$\lim_{i \rightarrow \infty} \|\delta u_i(t)\|_{\lambda} = 0. \quad (16)$$

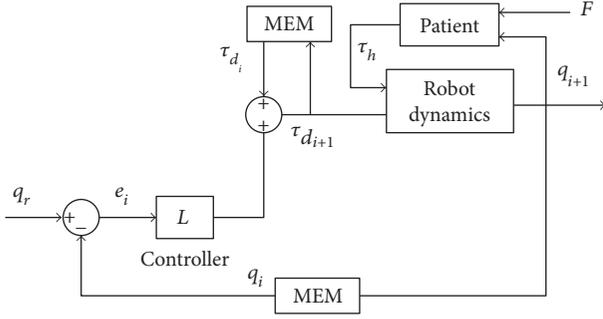


FIGURE 6: Schematic block of the proposed scheme.

TABLE 1: Geometric parameters of human subject.

Gender	Age (years)	Leg length (m)	Thigh length (m)	Height (m)	Body weight (kg)
Male	22	0.42	0.51	1.71	72

According to the definition of the norm  $\|\cdot\|_\lambda$ , these convergence are uniform on  $t \in [0, T]$ . Therefore,  $\lim_{i \rightarrow \infty} u_i(t) = u_a(t)$  uniformly on  $[0, T]$ .

From (10), (11), (12), (13), (14), (15), and (16), we have  $\lim_{i \rightarrow \infty} y_i(t) = y_d(t) + e^{Rt} C \{x_0 - x_d(0)\}$ .

Note that, in the proof, the initial state condition of each iteration remains the same. Therefore, the resulting output trajectory can be exactly estimated by the design gain  $R$  and the initial state error  $x_0 - x_d(0)$ . From (9), if  $R = 0$ , the converged output trajectory follows the desired trajectory with the offset of the initial error, and if  $R$  is chosen such that  $\lambda(R) < 0$ , the learned control input enables the system to possess an asymptotic tracking capability even in the face of nonzero initial error.

#### 4. Simulation and Experiment Results

As shown in Figure 6, it is the schematic block of the proposed scheme. The position controller in the overall iterative learning impedance scheme generates the impedance based on the trajectory tracking errors, but does not consider the contribution of human subjects' active force. The ILC block is used to store the input impedance signal in the previous run which will be used in the next run after modified. Thus, the impedance of the robot will, in turn, increase or decrease where the human subject deviates or not. Under the circumstances, the human subjects will move freely in the preliminary stage; then, with the help of the controller, they will get help when they deviate from the predefined trajectory. The more mistakes, the more impedance and vice versa.

The dynamic model is applied in the simulation, and the geometric parameters of human subject are shown in Table 1. The simulation is conducted to validate the proposed algorithm on a MATLAB R2014a with a simulation toolbox. Then, the experiments are complemented on the rehabilitation robot for validating the proposed algorithm.

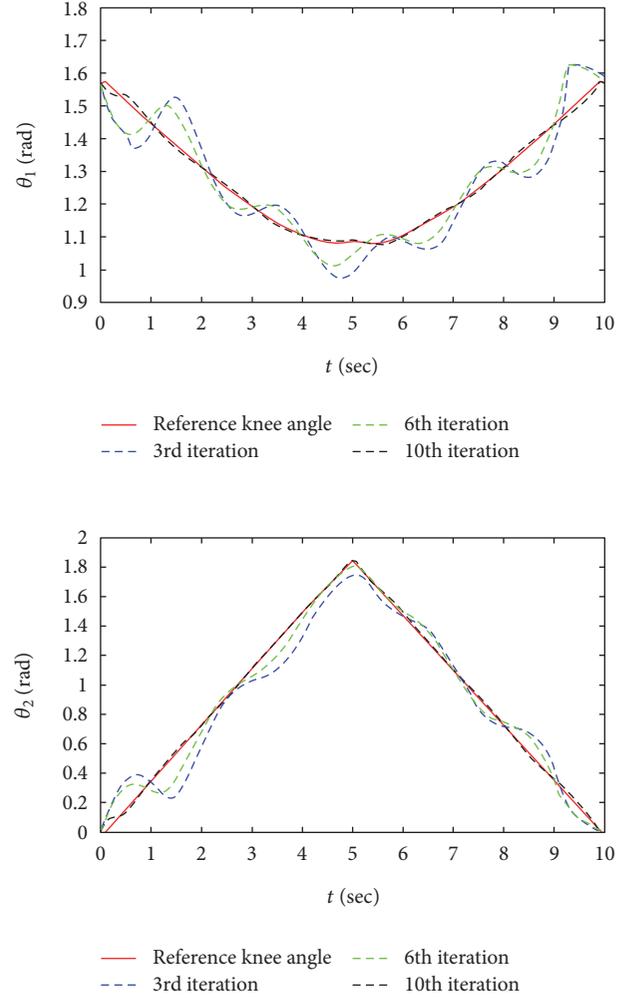


FIGURE 7: The variation of joint angle in simulation.

4.1. Simulation Studies. The parameters of the human-robot system in simulation are

$$A = \begin{bmatrix} -0.1695 & 0 \\ 0 & -0.1 \end{bmatrix}, \quad (17)$$

$$B = \begin{bmatrix} -0.0477 & 0 \\ 0 & 3 \end{bmatrix}.$$

The learning gains of the impedance controller are given as follows:

$$\Gamma = \begin{bmatrix} -10 & 0 \\ 0 & 0.5 \end{bmatrix} \quad (18)$$

$$R = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix},$$

where the term  $\Gamma$  and the term  $R$  satisfy the convergence condition (8).

In the simulation, the impedance gains are set to make the system compliant and the convergence of the proposed algorithm is verified. Since it is hard to predict the subject's active force, the active force is taken as a repeated

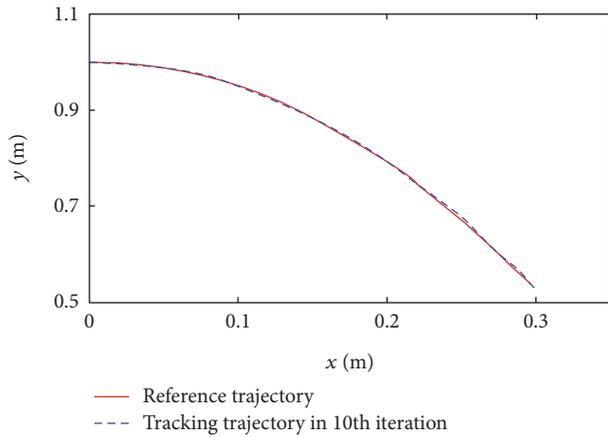


FIGURE 8: The performance of tracking trajectory in simulation.

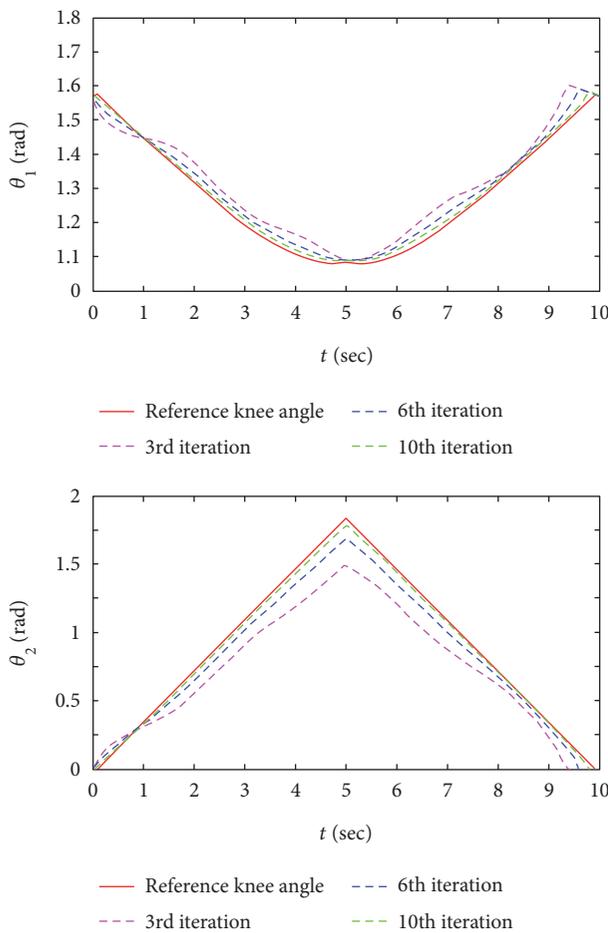


FIGURE 9: The variation of joint angle in experiment.

disturbance noise  $0.5\sin 3.14t$  in order to verify the effectiveness of the proposed control scheme in an actual work environment. The simulation results are shown in Figures 7 and 8, and the potential problems are discussed in the following.

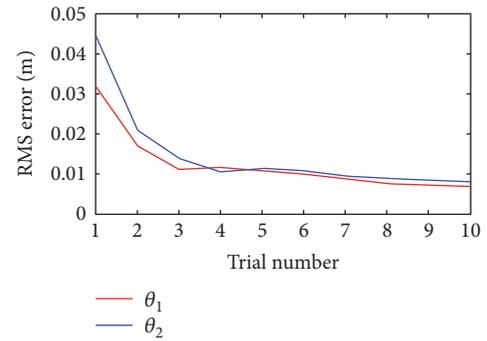


FIGURE 10: RMS error joint angle.

As shown in Figure 7, it is a diagram of the variation of joint angle calculated by the reverse solution of (1). Although the comparison between the desired trajectory and tracking trajectory reflects the quality of the training directly, the tracking performance of joint angle is more important which can provide us more training details especially for doctors. The blue dash, green dash, and black dash represent the variation of joint angle in different iterations. We can learn that the proposed method shows the ability of rejecting a repeating disturbance.

As shown in Figure 8, the tracking performance of reference trajectory and tracking trajectory is compared. We can learn that the tracking errors converge to zero with the increase of iteration.

**4.2. Experimental Results.** As mentioned above, the convergence condition and robustness of the proposed algorithm are proved. Further, we apply the proposed algorithm on the lower limb rehabilitation robot to verify its actual performance. In experiment, the test subject was required to track the reference trajectory 10 times whose geometric parameters are shown in Table 1. Body weight support was not used during the experiments as the test subject was not suffering from stroke or neurologic impairments. Experiments with healthy subject were conducted to evaluate if the iterative learning impedance scheme could modify the robotic assistance based on the past tracking information.

As shown in Figure 9, it is a diagram of the variation of joint angle in the experiment. During the first several trials, the test subject's tracking performance of joint angle is not good which indicates that his posture is not correct. With the increase of iteration, the variation of test subject's joint angle approximately approaches to the reference one, which indicates that the subject's training performance is gradually improved under the assistance of the robot.

As shown in Figure 10, it is a diagram of the root mean square (RMS) error corresponding to the proposed method which converges to approximately 8 mm. Considering that the squatting training is a relatively tough task for the subject, error within 15 mm is acceptable. It can be seen that the error reduces rapidly in the first several training cycles which indicates the improvement in tracking accuracy that the PD-type ILC schemes can provide.

As shown in Figure 11, the reference trajectory and tracking trajectory of the test subject are compared indicating that

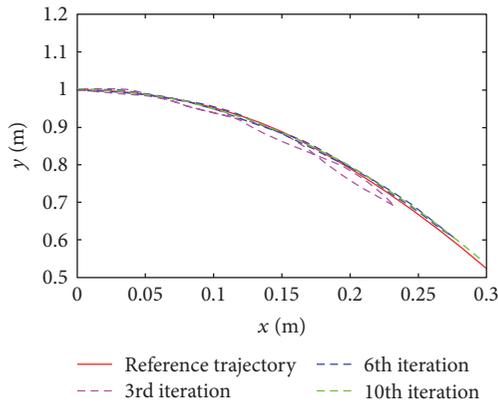


FIGURE 11: The trajectory tracking performance in experiment.

the test subject can almost track the reference trajectory accurately in the 10th iteration.

## 5. Conclusion

In the paper, the method of iterative learning impedance has been proposed to support training assistance and correct the patient's posture with the use of the lower limb rehabilitation robot. The convergence condition of the proposed algorithm is given, and the robustness to the parameter variables is analyzed. The simulation and experimental results show that compliancy and assistance have been achieved with the proposed iterative learning method. The most of current rehabilitation robot shows the same problem that it can hardly have both compliancy and robotic assistance. Although many robotic orthosis using impedance control can make the subject track the reference trajectory accurately, many subjects are easy to get into a situation that their limbs are driven by the robot passively. The main idea proposed in this paper introduced a learning impedance method, which can give subjects certain support where they underperform and make them train voluntarily in most training period. The system will enhance the impedance where the subjects deviate by the learning of last operation information. Considering the different geometric parameters of subjects, we proposed a method to generate the training trajectory easily with the absence of any training experiments. Although ILC algorithms have moved beyond these relatively simple structure types and now encompass as wide range of plant models and control law structures, the approach taken here was to apply ILC laws with the simplest structure which could meet the necessary performance requirement. Future works will be extended to the use of different structure types of ILC algorithms such as considering the whole past operation information or taking current iteration structure which will learn the current iteration error by introducing a feedback controller.

## Conflicts of Interest

The authors declare no competing interests.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China, Grant no. 61573234, and the Shanghai Municipal Science and Technology Commission, Grant nos. 15441900802 and 16111100802.

## References

- [1] J. Hu, Z. G. Hou, Y. X. Chen, F. Zhang, and W. Q. Wang, "Lower limb rehabilitation robots and interactive control methods," *Acta Automatica Sinica*, vol. 40, no. 11, pp. 2377–2390, 2014.
- [2] J. L. Patton, D. Brown, E. Lewis et al., "Motility evaluation of a novel overground functional mobility tool for post stroke rehabilitation," in *2007 IEEE 10th International Conference on Rehabilitation Robotics*, pp. 1049–1054, Noordwijk, 2007.
- [3] M. Bernhardt, M. Frey, G. Colombo, and R. Riener, "Hybrid force-position control yields cooperative behaviour of the rehabilitation robot LOKOMAT," in *9th International Conference on Rehabilitation Robotics, 2005 (ICORR 2005)*, pp. 536–539, Chicago, IL, USA, 2005.
- [4] K. Adams, J. P. O'Shea, K. L. O'Shea, and M. Climstein, "The effect of six weeks of squat, plyometric and squat-plyometric training on power production," *The Journal of Strength & Conditioning Research*, vol. 6, no. 1, pp. 36–41, 1992.
- [5] M. Peshkin, D. A. Brown, J. J. Santos-Munné et al., "KineAssist: a robotic overground gait and balance training device," in *9th International Conference on Rehabilitation Robotics, 2005 (ICORR 2005)*, pp. 241–246, Chicago, IL, USA, 2005.
- [6] N. J. Dahlkvist, P. Mayo, and B. B. Seedhom, "Forces during squatting and rising from a deep squat," *Engineering in Medicine*, vol. 11, no. 2, pp. 69–76, 1982.
- [7] N. Hogan, "Impedance control: an approach to manipulation: part II—implementation," *Journal of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 8–16, 1985.
- [8] Y. Yang, L. Wang, J. Tong, and L. Zhang, "Arm rehabilitation robot impedance control and experimentation," in *2006 IEEE International Conference on Robotics and Biomimetics*, pp. 914–918, Kunming, 2006.
- [9] J. F. Veneman, R. Kruidhof, E. E. Hekman, R. Ekkelenkamp, E. H. AsseldonkVan, and H. KooijVan Der, "Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 3, pp. 379–386, 2007.
- [10] J. A. Blaya and H. Herr, "Adaptive control of a variable-impedance ankle-foot orthosis to assist drop-foot gait," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 12, no. 1, pp. 24–31, 2004.
- [11] S. Hussain, S. Q. Xie, and P. K. Jamwal, "Adaptive impedance control of a robotic orthosis for gait rehabilitation," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1025–1034, 2013.
- [12] J. H. Lee, S. Natarajan, and K. S. Lee, "A model-based predictive control approach to repetitive control of continuous processes with periodic operations," *Journal of Process Control*, vol. 11, no. 2, pp. 195–207, 2001.
- [13] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control," *IEEE Control Systems*, vol. 26, no. 3, pp. 96–114, 2006.

- [14] D. H. Owens and K. Feng, "Parameter optimization in iterative learning control," *International Journal of Control*, vol. 76, no. 11, pp. 1059–1069, 2003.
- [15] K. Abidi and J.-X. Xu, "Iterative learning control for sampled-data systems: from theory to practice," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 7, pp. 3002–3015, 2011.
- [16] G. Pipeleers and K. L. Moore, "Unified analysis of iterative learning and repetitive controllers in trial domain," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 953–963, 2014.
- [17] D. Q. Huang, J. X. Xu, V. Venkataramanan, and T. C. T. Huynh, "High performance tracking of piezoelectric positioning stage using current cycle iterative learning control with gain scheduling," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 2, pp. 1085–1098, 2014.
- [18] C. J. Chien, "A discrete iterative learning control for a class of nonlinear time-varying systems," *IEEE Transactions on Automatic Control*, vol. 43, no. 5, pp. 748–752, 1998.
- [19] K.-H. Park, "An average operator-based PD-type iterative learning control for variable initial state error," *IEEE Transactions on Automatic Control*, vol. 50, no. 6, pp. 865–869, 2005.
- [20] J. X. Xu and J. Xu, "On iterative learning from different tracking tasks in the presence of time-varying uncertainties," *IEEE Transactions on Systems, Man and Cybernetics Part B, Cybernetics*, vol. 34, no. 1, pp. 589–597, 2004.
- [21] C. J. Chien and A. Tayebi, "Further results on adaptive iterative learning control of robot manipulators," *Automatica*, vol. 44, no. 3, pp. 830–837, 2008.
- [22] J. Bae and M. Tomizuka, "A gait rehabilitation strategy inspired by an iterative learning algorithm," *Mechatronics*, vol. 22, no. 2, pp. 213–221, 2012.
- [23] S. Balasubramanian, R. Wei, M. Perez et al., "RUPERT: an exoskeleton robot for assisting rehabilitation of arm functions," in *2008 Virtual Rehabilitation*, pp. 163–167, Vancouver, BC, 2008.
- [24] C. T. Freeman, E. Rogers, A. M. Hughes, J. H. Burrige, and K. L. Meadmore, "Iterative learning control in health care: electrical stimulation and robotic-assisted upper-limb stroke rehabilitation," *IEEE Control Systems*, vol. 32, no. 1, pp. 18–43, 2012.
- [25] T. J. Tarn, A. K. Bejczy, A. Isidori, and Y. Chen, "Nonlinear feedback in robot arm control," in *The 23rd IEEE Conference on Decision and Control*, pp. 736–751, Las Vegas, Nevada, USA, 1984.
- [26] K. Zhang, Y. X. Wu, X. Q. Lu, and X. Jin, "Dynamic modeling for differentially steered welding mobile robot," *Journal of Mechanical Engineering*, vol. 44, no. 11, pp. 116–120, 2008.

## Research Article

# Development of a Prototype Robotic System for Radiosurgery with Upper Hemispherical Workspace

**Sun Young Noh, Kyungmin Jeong, Yong-chil Seo, Chang-hoi Kim, Jongwon Park, Yoo Rark Choi, Sung Uk Lee, Yeong-Geol Bae, and Seungho Kim**

*Korea Atomic Energy Research Institute, Daejeon, Republic of Korea*

Correspondence should be addressed to Sun Young Noh; [synoh@kaeri.re.kr](mailto:synoh@kaeri.re.kr)

Received 20 January 2017; Revised 6 May 2017; Accepted 30 May 2017; Published 24 July 2017

Academic Editor: Qing Shi

Copyright © 2017 Sun Young Noh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper introduces a specialized robotic system under development for radiosurgery using a small-sized linear accelerator. The robotic system is a 5-DOF manipulator that can be installed above a patient to make an upper hemispherical workspace centered in a target point. In order to determine the optimal lengths of the link, we consider the requirements for the workspace of a linear accelerator for radiosurgery. A more suitable kinematic structure than conventional industrial manipulators is proposed, and the kinematic analysis is also provided. A graphic simulator is implemented and used for dynamic analysis. Based on those results, a prototype manipulator and its control system are under development.

## 1. Introduction

The aim of stereotactic radiosurgery systems such as Linac, Gamma Knife, and CyberKnife is to destroy tumor tissue while preserving adjacent normal tissue using ionizing radiation rather than excision with a blade [1, 2]. Thus, they should be able to accurately focus radiation beams from as many different angles as possible to converge on one target tumor [3, 4].

A Gamma Knife is used for treating not only brain tumors but also vascular and functional pathologies. And it typically contains 201 Co-60 radiation sources, each placed in a circular array, and emits gamma ray through a target point in the patient's brain. Thus, the Gamma Knife has over 200 beam delivery angles [5, 6].

Compared with the Gamma Knife, a Linac system uses X-rays generated from a linear accelerator, and the accelerator is mechanically rotated around the patient, in a full or partial circle to change the delivery angle aiming a target point. And the couch where the patient is lying can also be moved in small linear or angular steps. Thus, the combination of the movements of the gantry and of the couch can make more beam delivery angles. But they are confined

within an approximately 2-dimensional space [7, 8]. A CyberKnife system also uses a linear accelerator as a radiation source, but relatively small and light-weighted. It uses a 6-DOF industrial robot manipulator to move the X-ray source to aim arbitrary points from different angles [9, 10]. So, CyberKnife systems are recognized as one of the most versatile radiosurgery systems having a broad range of the beam delivery angle in a 3-dimensional space [11, 12].

But even though the degree of freedom of the manipulator used in a CyberKnife system is 6, 5-DOF is sufficient to aim an arbitrary 3-dimensional point from an arbitrary delivery angle because the change in the rolling angle of the beam about its axis has no effects [13, 14].

And besides, whereas the required radius of workspace for the radiosurgery is about 1 meter and the maximum moving speed of the end-effector is less than 1 m/s, the maximum radius of the workspace in the CyberKnife system is approximately 3 meters and its moving speed is also higher than required. That means that the performance of the joint-driving motors of industrial robot manipulators in the market is usually higher than required, and the driving motors are oversized. It may cause the system to be heavy-weighted and make its cost high. Furthermore, in cases when

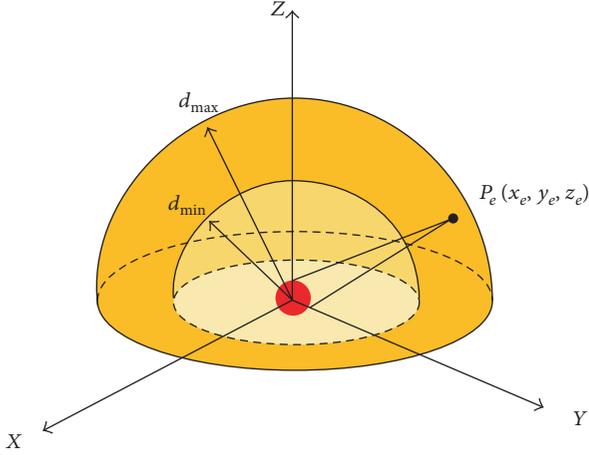


FIGURE 1: Workspace for radiosurgery.

the robot manipulator is installed on the floor beside the couch, there are limitations for increasing the range of the beam delivery angle.

So, this paper introduces a specialized robotic system under development for radiosurgery using a small-sized linear accelerator. The robotic system is a 5-DOF manipulator installed above a patient to make an upper hemispherical workspace centered in a target point.

In Section 2, the basic structure and the design process for determining the lengths of its links are described. Section 3 provides the kinematic analysis of the manipulator, and Section 4 explains a graphic simulation for dynamic analysis and its results. The manufactured manipulator under development and the structure of control system are shown in Section 5.

## 2. Design of Lengths of Robot Arm Links

As mentioned above section, the manipulator for radiosurgery needs 5-DOF joints in minimum to deliver a linear accelerator, so the lengths of the links should be optimized for its operation. And also, the manipulator workspace takes into consideration the position of the patient and is designed to avoid contact with the patient. This is achieved by creation of a safety zone around the patient and the treatment couch.

In Figure 1, the center of a tumor tissue to be treated is assumed to be located at the origin of xyz coordinates. The  $x$ - and  $y$ -axes define a horizontal plane, and the  $z$ -axis is vertical.

Let  $(x_e, y_e, z_e)$  refer to the position of X-ray source inside the linear accelerator.

If SAD (source-to-axis distance) is between  $d_{\min}$  and  $d_{\max}$ , and the workspace is limited to the upper hemisphere, then the required workspace where the end point should reach at is

$$\begin{aligned} d_{\min} &\leq \sqrt{x_e^2 + y_e^2 + z_e^2} \leq d_{\max}, \\ z_e &\geq 0. \end{aligned} \quad (1)$$

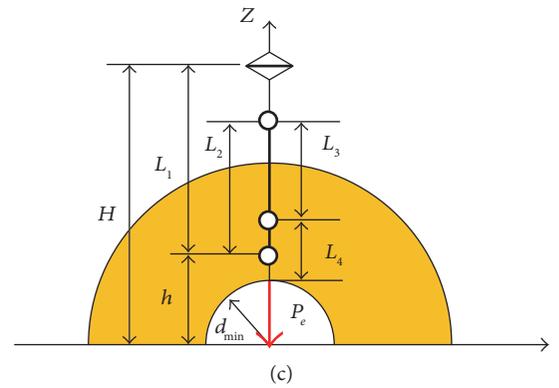
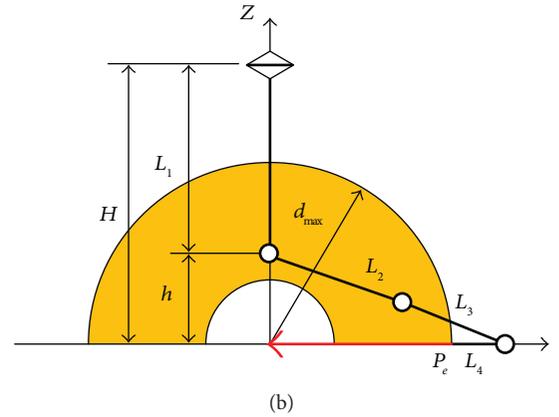
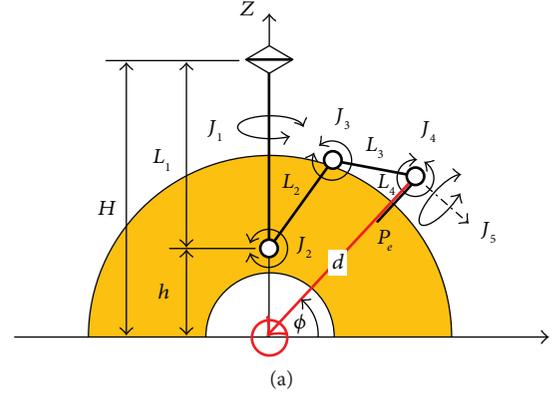


FIGURE 2: Proposed configuration of robot.

TABLE 1: Optimal length.

$H$	$L_2$	$L_3$
0.5	0.8	0.5
0.6	0.78	0.58
0.7	0.75	0.65
<b>0.8</b>	<b>0.73</b>	<b>0.73</b>
0.9	0.7	0.8
1.0	0.69	0.89
1.1	.67	0.97

At any point in the workspace, the accelerator should direct radiation beams to a small region around the origin.

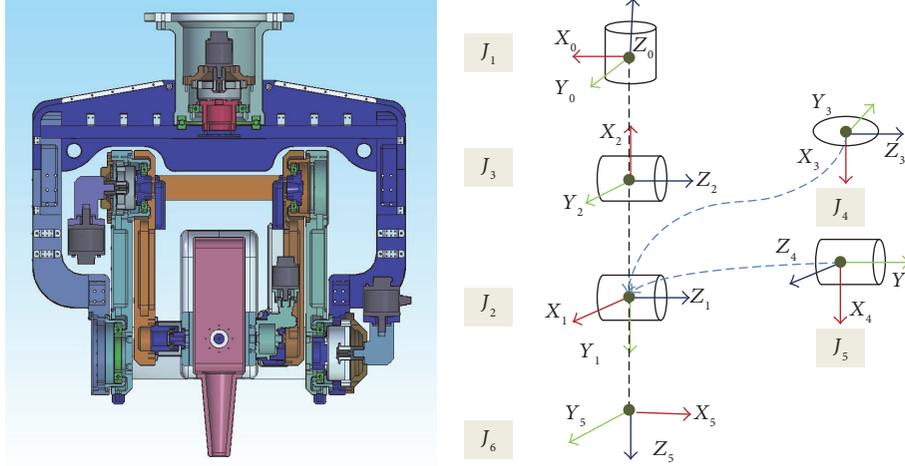


FIGURE 3: D-H parameters layout for manipulator.

TABLE 2: D-H parameters.

Joint number ( $i$ )	$a_{i-1}$ (m)	$\alpha_{i-1}$ [deg]	$d_i$ [m]	$\theta_i$ [deg]
1	0	-90	-1.604	90
2	0.8	0	0	-90
3	0.73	0	0	180
4	0	90	0	0
5	0	-90	0	90
6	0	180	-0.567	0

In our study, a serial manipulator with revolute joints is considered as shown in Figure 2(a).

The joint 1 ( $J_1$ ) is aligned with the  $z$ -axis and defines a central sagittal plane, and by using joints 2 ( $J_2$ ), 3 ( $J_3$ ), and 4 ( $J_4$ ) parallel to each other, the accelerator is directed to the origin at any point in that plane. Both joints 4 and 5 ( $J_5$ ) are used to direct the beam to the neighboring region.

The center of joint 2 is shifted above the origin with  $h$  to avoid collision with a patient.

One extreme constraint with the lengths of the links is the case when  $\phi = -\pi/2$ ,  $d = d_{\max}$  where the angle  $\phi$  is defined as shown in Figure 2(a), and the reach of the arm is maximum as shown in Figure 2(b). To access such a posture in its full reach, the lengths of the links satisfy the following equation:

$$L_2 + L_3 = \sqrt{h^2 + (d_{\max} + L_4)^2}. \quad (2)$$

Another case is when  $\phi = \pi/2$ ,  $d = d_{\min} \rightarrow \phi = -\pi$ ,  $d = d_{\min}$ . In that case, the following equation satisfies the posture:

$$h + L_2 - L_3 - L_4 = d_{\min}. \quad (3)$$

The link's lengths  $L_2, L_3, L_4$  satisfy (2) and (3), and the required workspace of the arm is obtained.

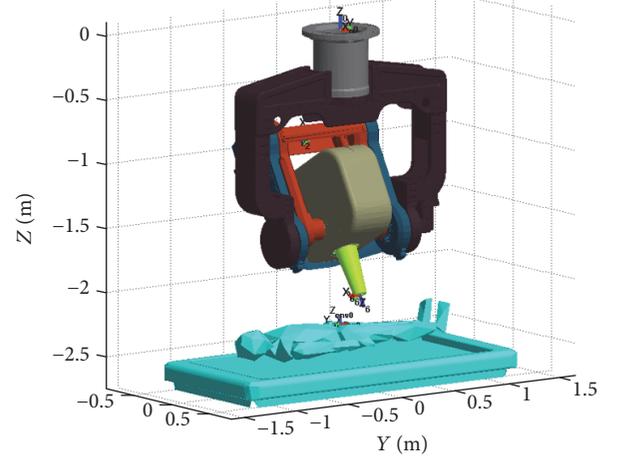


FIGURE 4: The 3D graphic simulation environment.

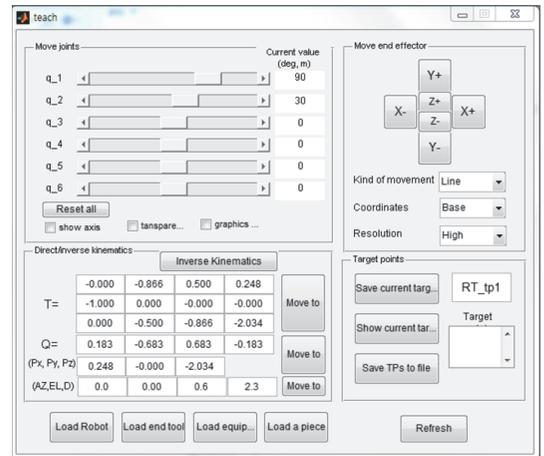


FIGURE 5: The control panel window.

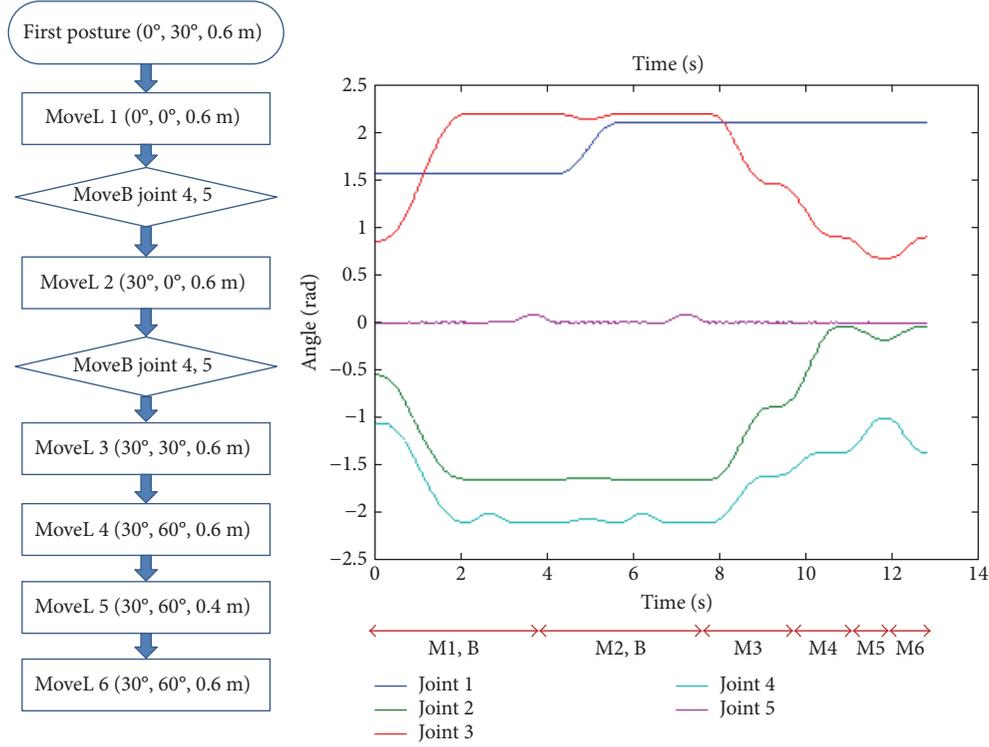


FIGURE 6: An example trajectory six postures.

When  $d_{\min}$ ,  $d_{\max}$ ,  $L_4$  are given,  $L_2$  and  $L_3$  can be determined as follows:

$$\begin{aligned} L_2 &= \frac{1}{2} \left( \sqrt{h^2 + (d_{\max} + L_4)^2} - h + L_4 + d_{\min} \right), \\ L_3 &= \frac{1}{2} \left( \sqrt{h^2 + (d_{\max} + L_4)^2} + h - L_4 - d_{\min} \right). \end{aligned} \quad (4)$$

Table 1 shows  $L_2$  and  $L_3$  determined from (4) when  $d_{\min} = 0.6m$ ,  $d_{\max} = 1.0m$ ,  $L_4 = 0.1m$  optimal length according to  $H$ .

### 3. Kinematic Analysis

This section provides analysis for forward and inversed kinematics of the radiosurgery manipulator based on the proposed structure and the lengths of links.

**3.1. Forward Kinematic Analysis.** Figure 3 shows the outline of the manipulator and its coordinates in D-H convention. The base coordinate system is located on the mounting plate of the manipulator, and the origin of the coordinate systems 3 and 4 is coincident with each other. The joints 2 and 3 are supported by two links in both sides to reduce deflections. And the dummy joint 6 is not really actuated but added to follow conventional kinematic analysis procedure for 6-DOF manipulators.

Table 2 provides the D-H parameters in mm for length and in degree for angle.

Using the D-H parameters, homogeneous transform matrices are given by

$${}^i T_{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & a_{(i-1)} \\ \sin \theta_i \cos \alpha_{(i-1)} & \cos \theta_i \cos \alpha_{(i-1)} & -\sin \alpha_{(i-1)} & -\sin \alpha_{(i-1)} d_i \\ \sin \theta_i \sin \alpha_{(i-1)} & \cos \theta_i \sin \alpha_{(i-1)} & \cos \alpha_{(i-1)} & \cos \alpha_{(i-1)} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

By multiplying all homogeneous transform matrices sequentially from the left to the right, the final homogeneous matrix is obtained as follows:

$${}^1 T_0 {}^2 T_1 {}^3 T_2 {}^4 T_3 {}^5 T_4 {}^6 T_5 = {}^6 T_0 = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_x \end{bmatrix}. \quad (6)$$

**3.2. Inverse Kinematic Analysis.** To calculate joint angles from a given target position and orientation, the central position of the accelerator, the origin of the coordinate system 3 is derived at first by the following equation:

$$P_m = [P_x \ P_y \ P_z]' - L_6 \times W(L_6 = \|\theta_6\|). \quad (7)$$

Like a usually 6-DOF manipulator, there are 4 sets of solution for  $\theta_1 \sim \theta_3$  in maximum depending to its configurations such as the left or the right arm, elbow up or down.

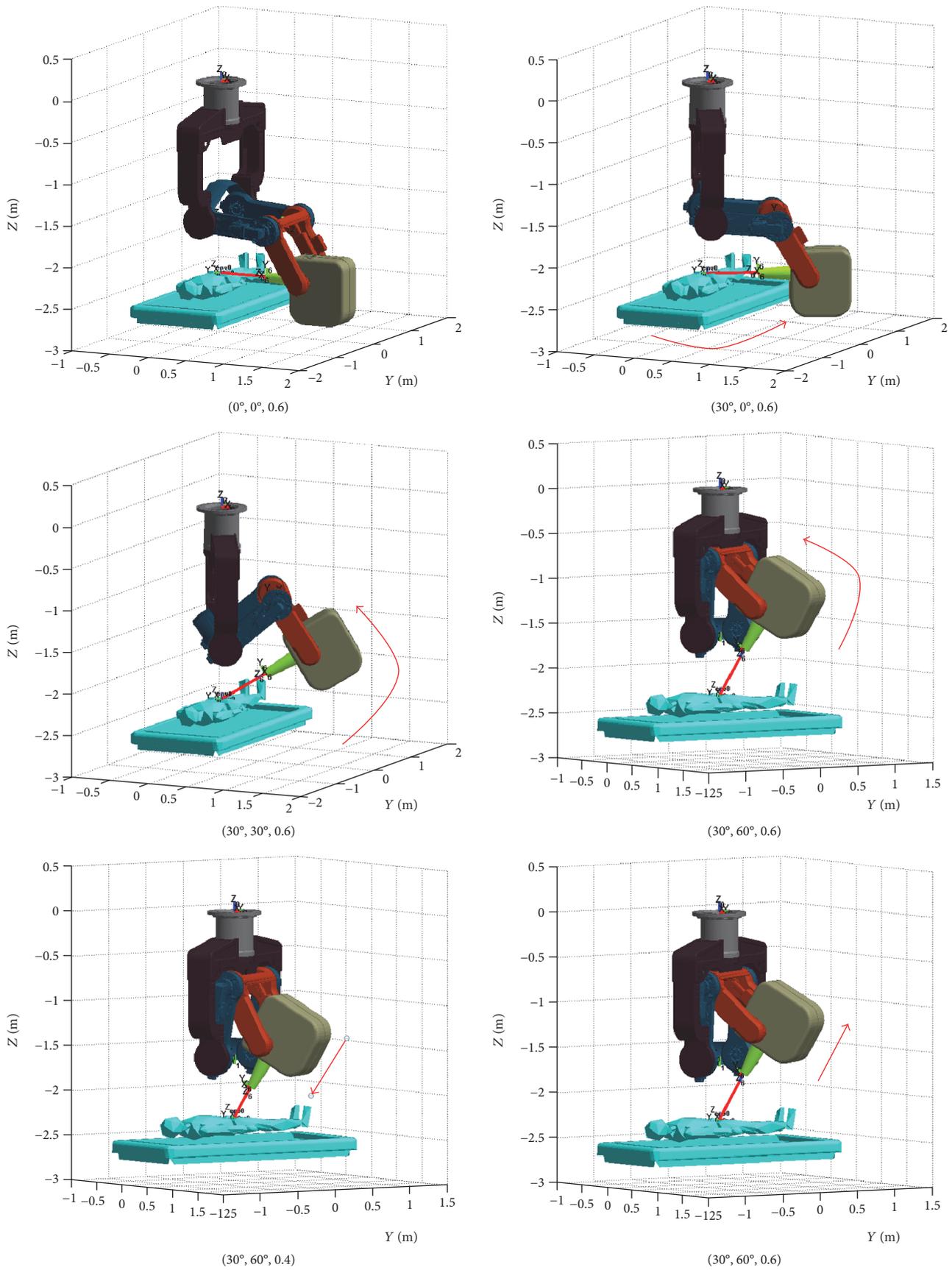


FIGURE 7: The six postures in the 3D graphic environments.

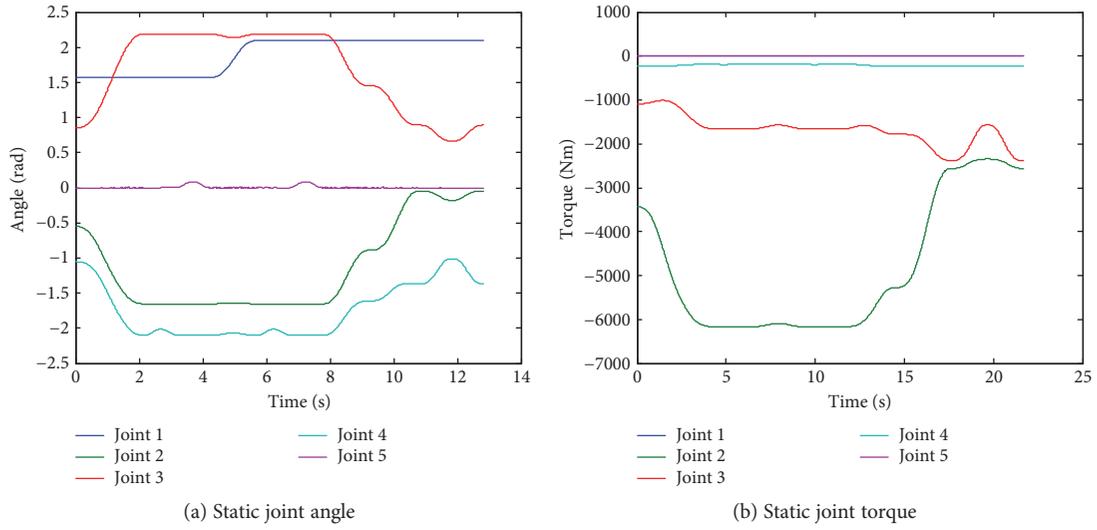


FIGURE 8: Joint angle and static torque.

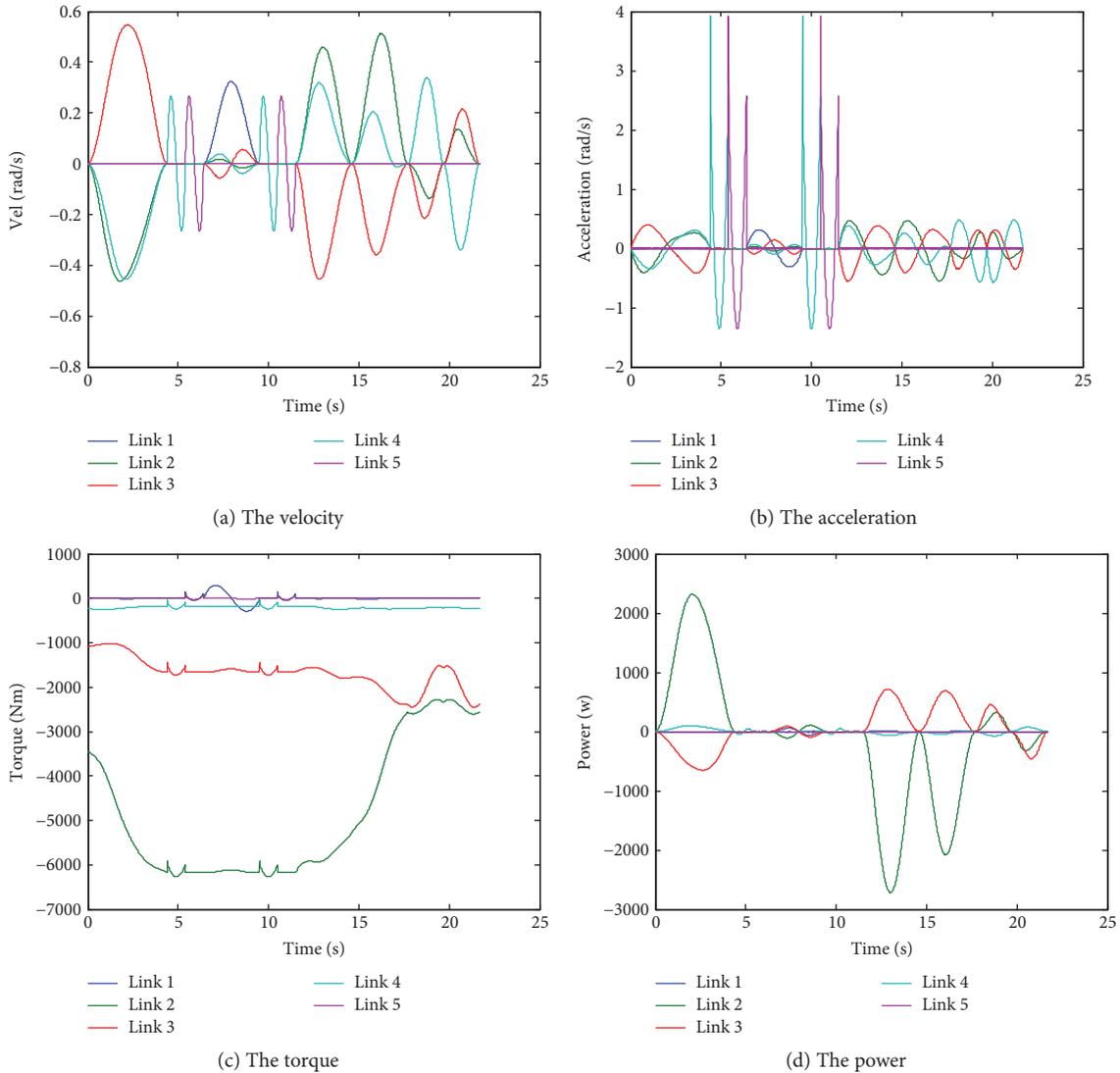


FIGURE 9: Simulation results.

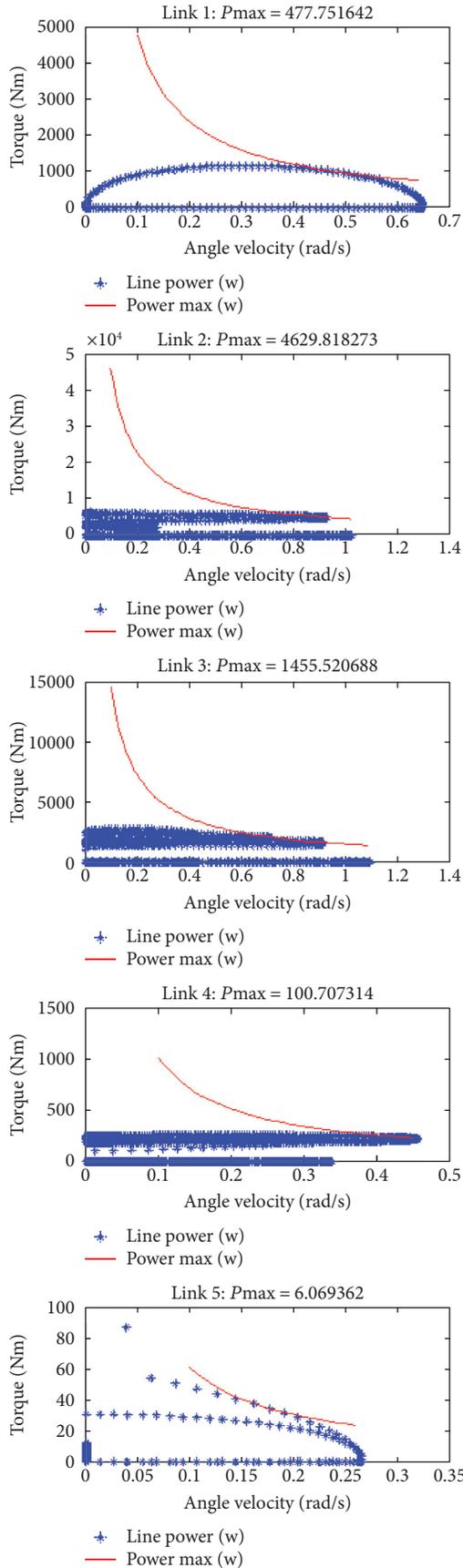


FIGURE 10: S velocity-torque diagram.

TABLE 3: Maximum torque and power.

	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5
Max. torque	289.61	6271	2446	250.60	87.17
Max. power	59.41	<b>2318</b>	718.32	100.7	6.06

There are two sets of  $\theta_4$  and  $\theta_5$  that are determined to satisfy the given orientation for each set of  $\theta_1 \sim \theta_3$ .

Thus, 8 sets of solution can be found if the given end point is inside the workspace. Among them, the joint angle limit criteria are applied to screen the solutions and the solution that is nearest from the current joint configuration is selected as the final solution.

A high continuous path can make smooth and efficient movements. To improve the continuity of the path, two successive points are interpolated with spline curves that consist of 5th-order polynomial equation and satisfy given points, velocities, and accelerations as follows:

$$S(t) = a + bt + ct^2 + dt^3 + et^4 + ft^5. \quad (8)$$

#### 4. Graphic Simulator and Dynamic Analysis

In this study, a MATLAB-based open source code, ARTE (A Robotics Toolbox for Education) [15], is used and modified to simulate the motion of the radiosurgery manipulator under development in 3-dimensional graphic virtual environments.

To input geometric model of the manipulator to the simulator, 3D CAD models are converted to STL files in ASCII format.

And such dynamic parameters as masses, mass moments of inertia, and centers of masses for each link are also extracted from the 3D CAD files to enable dynamic simulation capability implemented in the toolbox.

Figure 4 shows the manipulator and a patient lying on a couch in the 3D graphic simulation environment.

Figure 5 shows the control panel window that displays current joint angles and target inputs to move the manipulator. It is modified from the teach program packaged in ARTE.

To command the manipulator to move, the azimuth angle, the elevation angle, and the distant from the origin are directed by operators. The desired target postures can be saved to text files for later dynamic simulations.

As an example of trajectory, six postures are given as shown in Figure 6 using a function named MoveL() and the joints 4 and 5 are rotated slightly about  $5^\circ$  in some posture using a function named MoveB() that independently moves the joints without moving all the other joints.

Figure 7 shows the six postures in the 3D graphic environments, and Figure 8 shows the static joint torques.

As a dynamic simulation, the linear velocity of the end point is given 0.1 m/s, and the velocity, the acceleration, the torque, and the power of each joint are provided in Figure 9.



FIGURE 11: Prototype manipulator.

Figure 10 shows the joint velocity, the joint torque, and the power in velocity-torque diagram for each joint to help motor selection for performing the simulated motion.

As shown in Table 3, the maximum torque and power for the joint 2 is 6271 Nm and 2317 W.

## 5. Manufactured Robot and Control System

The motors and gearboxes are selected based on the dynamic simulation results, and the manipulator has been manufactured as shown in Figure 11.

As for controlling motors, MR-J4-B series servo amplifiers manufactured by Mitsubishi electric company are selected. They have an interface to a battery-backup optical encoder, and so there is no need to start initialization process at every power-up stage.

Those of 5 servo amplifiers are linked through SSCNET optical fibers to MR-MC 241 multiaxis position control board mounted in a PCI slot of a desktop computer operated in MS Windows 7. And also, we consider two types of safety features of the robotic system when danger exists on machines and systems they have to be immediately shut down, in order to protect people, machines, and systems. At first, stoppers are attached to the side of the joint 2 of the manipulator. It can be sure to take preventive method to the manipulator against falling to patients when danger exists on machines. Second, an emergency stop button and a signal emergency stop button in control system are designed to happen only in emergency situations. Pressing the emergency stop button, brakes, is activated in the manipulator axis, and the robot stops. The motor power turns off at this timing.

Figure 12 shows the test GUI programmed in MS Visual C++ environment. This program has included several functions implemented in the simulator.

## 6. Conclusion

This paper introduces a radiosurgery manipulator system that has 5-DOF and a more suitable workspace for its operation. This is achieved by creation of a safety zone around the patient and the treatment couch. In order to design the manipulator, kinematic and dynamic analyses are conducted

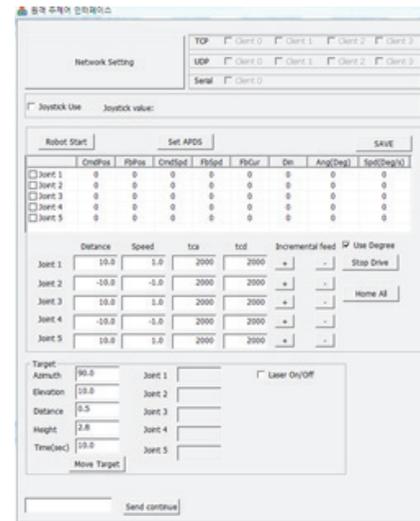


FIGURE 12: Graphical use interface.

and implemented in a 3D graphic simulator based on one of MATLAB-based open source robotics toolboxes, ARTE. Resulted from some motion simulations for the manipulator, a prototype manipulator is manufactured, and its control system is implemented.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

The research for this paper was supported by the National Research Foundation of Korea (NRF) (NRF-2013M2A8A1051065).

## References

- [1] D. B. Camarillo, T. M. Krummel, and J. K. Salisbury, "Robotic technology in surgery: past, present, and future," *The American Journal of Surgery*, vol. 188, no. 4, pp. 2–15, 2004.
- [2] M. Kushnirsky, V. Patil, and M. Schulder, "The history of stereotactic radiosurgery," *Practice of Stereotactic Radiosurgery*, pp. 3–10, 2015.
- [3] X. Y. Cedric and G. Tang, "Intensity-modulated arc therapy: principles, technologies and clinical implementation," *Physics in Medicine and Biology*, vol. 56, no. 5, pp. R31–R51, 2011.
- [4] M. Matinfar, E. Ford, I. Iordachita, and J. Wong, "Image-guided small animal radiation research platform: calibration of treatment beam alignment," *Physics in Medicine and Biology*, vol. 54, no. 4, pp. 892–905, 2009.
- [5] D. L. d. Lunsford, J. Flickinger, G. Lindner, and A. Maitz, "Stereotactic radiosurgery of the brain using the first United States 201 cobalt-60 source gamma knife," *Neurosurgery*, vol. 24, no. 2, pp. 151–159, 1989.
- [6] M. Levivier, T. Gevaert, and L. Negretti, "Gamma Knife, CyberKnife, TomoTherapy: gadgets or useful tools?" *Current Opinion in Neurology*, vol. 24, no. 6, pp. 616–625, 2011.

- [7] T. C. Zhu and K. K. H. Wang, "Linear accelerators (LINAC)," *Encyclopedia of Radiation Oncology*, pp. 437–450, 2013.
- [8] L. A. Dawson and D. A. Jaffray, "Advances in image-guided radiation therapy," *Journal of Clinical Oncology*, vol. 25, no. 8, pp. 938–946, 2007.
- [9] W. P. Leemans, R. W. Schoenlein, P. Volfbeyn, and A. H. Chin, "X-ray based subpicosecond electron bunch characterization using 90 Thomson scattering," *Physical Review*, vol. 77, no. 20, p. 4182, 1996.
- [10] I. C. Gibbs, S. D. Chang, and J. R. Adler, "Cyber-Knife radiosurgery experience at Stanford university," *Robotic Radiosurgery*, 2005.
- [11] "Accuray, equipment specifications-Cyberknife (brochure style)," <http://www.cyberknifelatin.com/pdf/brochure-tecnico.pdf>.
- [12] D. Olender, W. Kilby, and R. A. Sc, "Robotic whole body stereotactic radiosurgery: clinical advantages of the CyberKnife integrated system," *Medical Robotics and Computer*, vol. 1, no. 2, pp. 28–39, 2005.
- [13] W. Kilby, J. R. Dooley, G. Kuduvalli, and S. Sayeh, "The CyberKnife® robotic radiosurgery system in 2010," *Current Opinion in Neurology*, vol. 9, no. 5, pp. 433–452, 2010.
- [14] M. Levivier, T. Gevaert, and L. Negretti, "Gamma Knife, CyberKnife, TomoTherapy: gadgets or useful tools?" *Technology in Cancer*, vol. 24, no. 6, pp. 616–625, 2011.
- [15] A. G. Aparicio, *A Robotics Toolbox for Education*. Copyright (c), 2012, [http://arvc.umh.es/arte/index\\_en.html](http://arvc.umh.es/arte/index_en.html).

## Research Article

# Inertial Sensor-Based Motion Analysis of Lower Limbs for Rehabilitation Treatments

Tongyang Sun,<sup>1</sup> Hua Li,<sup>2</sup> Quanquan Liu,<sup>1,3</sup> Lihong Duan,<sup>3</sup> Meng Li,<sup>3</sup> Chunbao Wang,<sup>1,3,4</sup> Qihong Liu,<sup>1</sup> Weiguang Li,<sup>1</sup> Wanfeng Shang,<sup>3</sup> Zhengzhi Wu,<sup>3</sup> and Yulong Wang<sup>2</sup>

<sup>1</sup>School of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou, Guangdong, China

<sup>2</sup>The Second People's Hospital of Shenzhen, Shenzhen, Guangdong, China

<sup>3</sup>Shenzhen Institute of Geriatrics, Shenzhen, Guangdong, China

<sup>4</sup>School of Mechanical Engineering, Guangxi University of Science and Technology, Liuzhou, Guangxi, China

Correspondence should be addressed to Chunbao Wang; [chunbaowang@163.com](mailto:chunbaowang@163.com) and Zhengzhi Wu; [szwzz001@163.com](mailto:szwzz001@163.com)

Received 2 March 2017; Accepted 9 May 2017; Published 5 July 2017

Academic Editor: Chengzhi Hu

Copyright © 2017 Tongyang Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The hemiplegic rehabilitation state diagnosing performed by therapists can be biased due to their subjective experience, which may deteriorate the rehabilitation effect. In order to improve this situation, a quantitative evaluation is proposed. Though many motion analysis systems are available, they are too complicated for practical application by therapists. In this paper, a method for detecting the motion of human lower limbs including all degrees of freedom (DOFs) via the inertial sensors is proposed, which permits analyzing the patient's motion ability. This method is applicable to arbitrary walking directions and tracks of persons under study, and its results are unbiased, as compared to therapist qualitative estimations. Using the simplified mathematical model of a human body, the rotation angles for each lower limb joint are calculated from the input signals acquired by the inertial sensors. Finally, the rotation angle versus joint displacement curves are constructed, and the estimated values of joint motion angle and motion ability are obtained. The experimental verification of the proposed motion detection and analysis method was performed, which proved that it can efficiently detect the differences between motion behaviors of disabled and healthy persons and provide a reliable quantitative evaluation of the rehabilitation state.

## 1. Introduction

Nowadays, the aging problem becomes a very topical and complicated social challenge [1, 2]. A high incidence and recurrence rate among aged people is exhibited by such disease as hemiplegia, which implies paralysis of one side of the body usually caused by a brain lesion, such as a tumor, or by stroke syndrome. The number of hemiplegia cases is increasing quickly, and a large share of survivors after stroke become disabled (about 70%) and even severely disabled (about 40%) [3].

The correct diagnosis of the motion disorders is critical for prescribing an effective treatment, but the judgment of therapists is based on their experience of therapists and,

thus, is somewhat subjective. The unbiased representation of the patient state is the basic requirement for developing the best treatment matching this state and reducing the rehabilitation period.

Motion detection and parametric description are the main components of the integral evaluation system. Multiple detecting methods have been developed in the last decades, including WB-4 [4], Vicon [5], OPTOTRAK of Northern Digital [6], STAGE of Organic Motion [7], Kinect of Microsoft [8], Liberty 240/8 of Polhemus [9], NDI [10], and HX17 of Hexamite [11]. In view of such factors as the rehabilitation environment complexity, mechanical tracking having a complex calibration, optical sensors being interfered by therapists, low accuracy of the acoustic

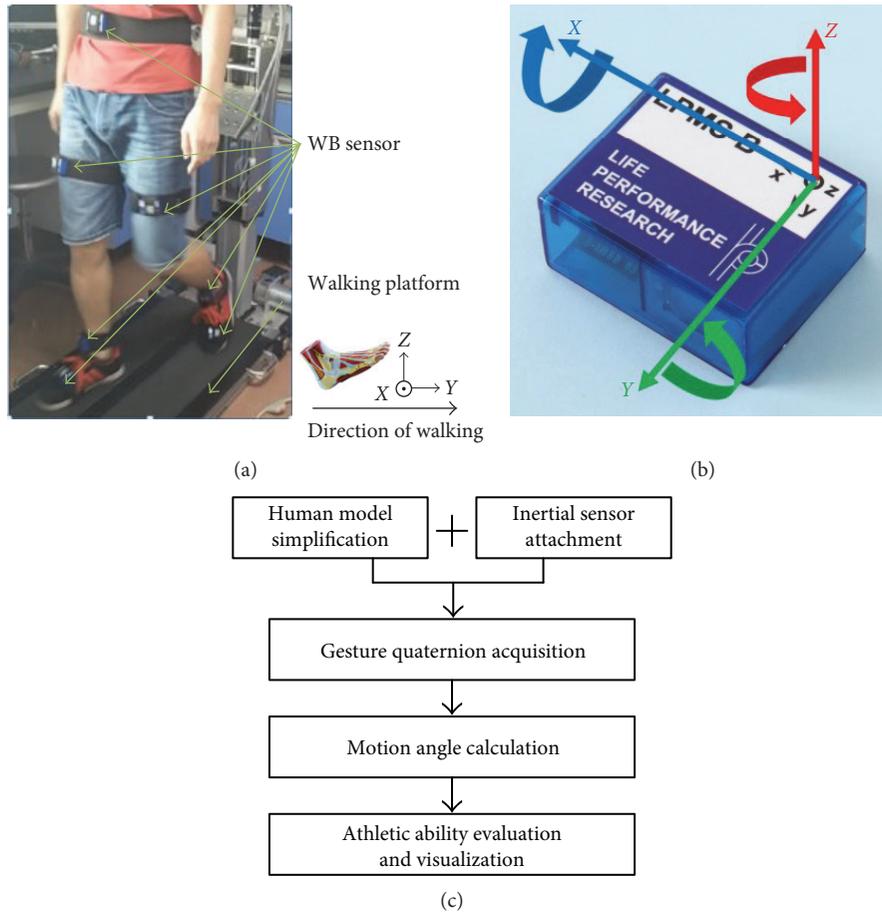


FIGURE 1: (a) Walking platform. (b) WB sensor. (c) System flowchart.

tracking, and electromagnetic tracking vulnerability to metal interference, many motion detection methods fail to meet the detection requirements.

The inertial sensing technology is a relatively innovative motion tracking system with high performance and large measurement range, which utilizes easy wearable portable inertial measurement units (IMUs). Recently, inertial sensor has been used to detect and evaluate human motion: whole body motion [12, 13], scapula calibration [14], lie-to-stand transfer [15], and gait analysis [16–20]. The studies using inertial sensors measure movement time; calculate joint or inclination angles, walking speed, step or stride length, and segment position relative to other position; and detect gait event timings. However, there are certain limitations on the applications, which impede more effective diagnosing and training as to the simplifying or resolving of the joint motion and motor function evaluation according to the clinical requirement of the therapist diagnosing.

This paper aims to detect the motion of human lower limbs via the inertial sensors, which permits analyzing the motion ability according to clinical rehabilitation needs. This method is applicable to arbitrary walking directions and tracks of persons under study, and its results are unbiased, as compared to therapist qualitative estimations. Using the simplified mathematical model of a human body, the rotation angles for each lower limb joint DOF are calculated from

the input signals acquired by the inertial sensors via the respective gesture quaternion. Finally, the rotation angle versus joint displacement curves are constructed, and the estimated values of joint motion angle and motion ability are obtained. The experimental verification of the proposed motion detection and analysis method was performed, which proved that it can efficiently detect the differences between motion behaviors of disabled and healthy persons and provide a reliable quantitative evaluation of the rehabilitation state.

The rest of the paper is organized as follows. Section 2 gives a detailed description of the proposed system configuration. Section 3 presents the experimental results. The performance and potential improvements of the proposed system are discussed in Section 4.

## 2. Methods

**2.1. System Overview.** The system comprises a platform [21] for rehabilitation training (with weight support device and pelvic fixation device) as shown in Figure 1(a), seven WB sensors depicted in Figure 1(b) and a laptop with bespoke data processing software and a graphic user interface (GUI) developed in C++ Builder. The system aimed at acquiring the studied participant's gait kinematics indicated by hip knee and ankle angle of each joint's degree of freedom

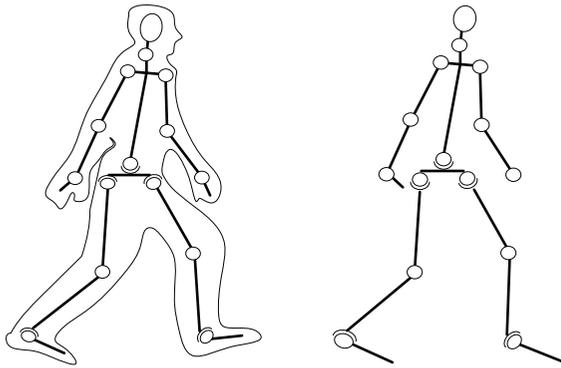


FIGURE 2: The simplified human motion model.

(DOF) and evaluating the athletic ability from the motion angle phase. This is achieved by acquiring the gesture quaternion of the studied participant from the inertial sensors attached to his/her waist, thigh, crus, and foot when he/she walks on the platform. As shown in Figure 1(c), the system procedure includes human model simplification and inertial sensor attachment, gesture quaternion acquisition, motion angle calculation, and athletic ability evaluation and visualization. Each of these acquisition and processing steps is described in the following.

**2.2. Human Model Simplification and Inertial Sensor Attachment.** Human body skeleton has a very complex structure, which must be simplified, in order to achieve the real-time analysis of human motion, using special models of human body motion models. Thus, the essence of the stick figure model is that it reduces the human body motion to that of human skeleton bones, so that various parts of a human body are approximated by the straight lines. For example, the stick figure model proposed by Chen and Lee [22] contains 17 sections and 14 connection points to represent the head, torso, and limbs. Since the different models have different motion mathematical relationships, different models will lead different results by the same acquired data. The simplified human motion model used herein is depicted in Figure 2.

The human skeleton model used in this study regards a human skeleton as a rigid rod and reduces a knee to a uniaxial joint according to the clinical diagnosing requirement. The DOFs of hip and ankle are three, and the DOF of knee is only one.

Within the framework of the applied inertial sensing technology, which was briefly discussed in Section 1, the life performance research motion sensor (LPMS) was selected as the motion sensor. The LP-Research Motion Sensor Bluetooth version (LPMS-B) is a miniature wireless IMU/attitude and heading reference system (AHRS). The unit is very versatile, performing accurate, high-speed orientation, and displacement measurements. By the use of three different MEMS sensors (3-axis gyroscope, 3-axis accelerometer, and 3-axis magnetometer), drift-free, high-speed orientation data about all three axes is achieved. The LPMS-B communicates with a host system via a Bluetooth connection. The LPMS sensor measures the orientation difference between the fixed sensor and global reference coordinate systems.

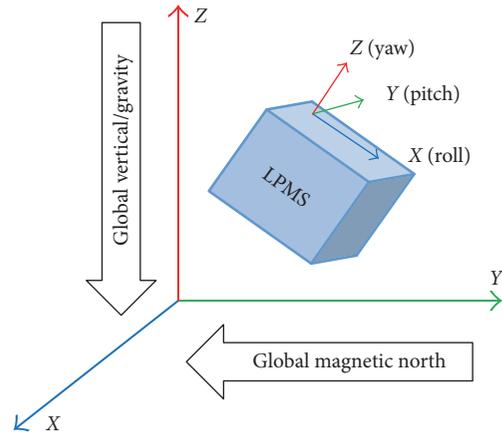


FIGURE 3: Global and local sensor coordinates.

$$Q_{\text{sensor}} = q_{\text{difference}} Q_{\text{global}} q_{\text{difference}}^{-1}, \quad (1)$$

where  $Q_{\text{sensor}}$  is the fixed sensor coordinate system,  $Q_{\text{global}}$  is the global coordinate system, and  $q_{\text{difference}}$  is the orientation difference between  $Q_{\text{global}}$  and  $Q_{\text{sensor}}$ .

The local and global reference coordinate systems used are defined as right-handed Cartesian coordinate systems, where  $X$  is positive when pointing to the magnetic West,  $Y$  is positive when pointing to the magnetic South, and  $Z$  is positive when pointing up (in the opposite direction to gravity vector). The axial orientation of LPMS-B and the relationship between the local sensor coordinate system and global coordinates are shown in Figure 3.

**2.3. Gesture Quaternion Acquisition.** This section includes the definition of quaternion and discusses its acquisition procedure.

There are various ways of representing orientation, and the use of Euler angles is one of them. The Euler angles are used to represent roll, pitch, and yaw of a body. There is one constraint when using the Euler angles for this purpose: this representation has singularities at pitch angles of  $\pm 90^\circ$ . Quaternions avoid these singularities by having a fourth element. The addition of this element is at the constraint of being a unit length. The concept of quaternion has been introduced by Hamilton in 1843 [23]. A special subset of the quaternion space, denoted by  $\mathbf{IH}$ , is defined that when  $\|q\| = 1$ , then  $q$  is called a unit norm quaternion, and the unit quaternion space is denoted as  $\mathbf{IH}_1$ . This particular subset is of special interest, since it provides the characterization of orientation trajectories. The temporal orientation trajectories are studied in the unit norm quaternion space  $\mathbf{IH}_1 \subset \mathbf{IH}$  [24]. Any general three-dimensional rotation can be transformed into a unit norm quaternion  $q \in \mathbf{IH}_1$ .

In this study, the rotation quaternion is expressed as  $q = (w, x, y, z)^T$ , where  $w$  is the cosine of the rotation semiangle, while  $x, y, z$  are the multiplication X, Y, Z coordinates of the rotation axis and the sine of the rotation semiangle, whereas  $w^2 + x^2 + y^2 + z^2 = 1$ .

In order to acquire quaternions via the gesture quaternion recording technique, seven sensors are attached by

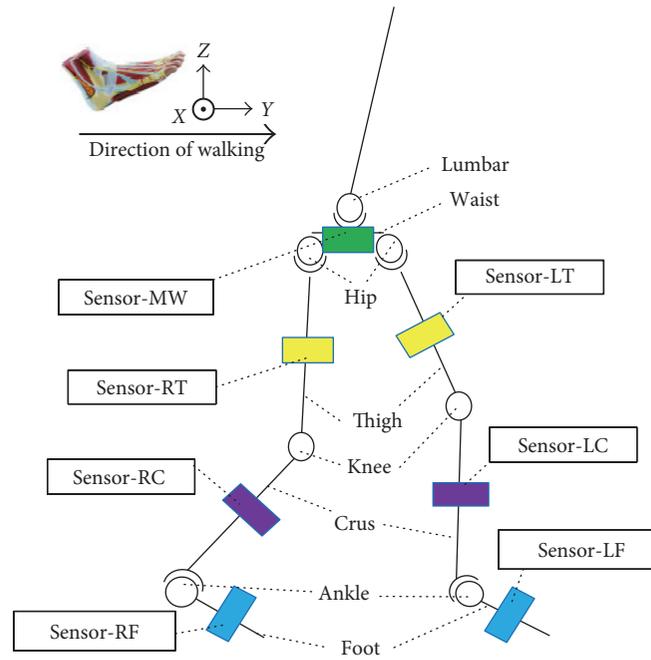


FIGURE 4: Location of attached sensors.

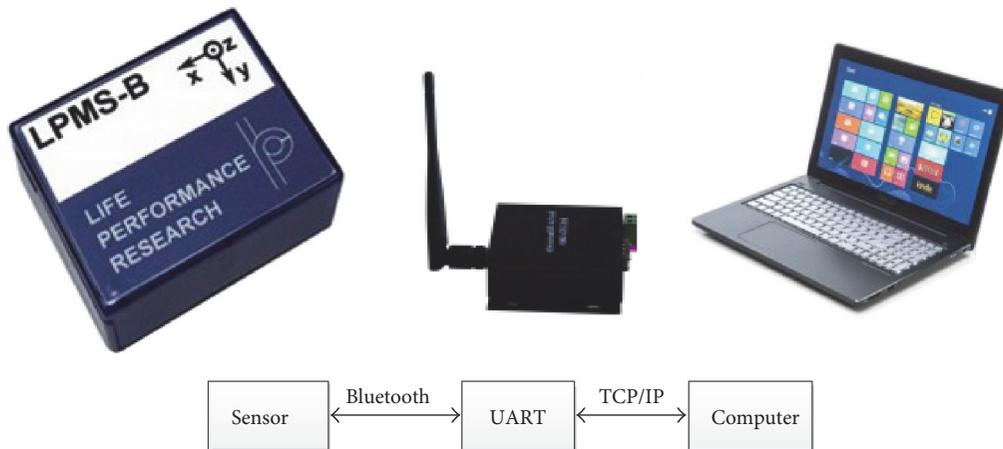


FIGURE 5: Hardware communication.

Velcro straps to the waist, thigh, crus, and foot on both legs of the studied participant in the walking platform. The location and the orientation of sensors are shown in Figure 4, where the following designations are used: MW corresponds to the waist-attached sensor, while LT/LC/LF and RT/RC/RF are sensors attached to the left and right thigh/crus/foot, respectively.

The hardware communication between sensors and PC is depicted in Figure 5. Here, the transceiver communicates with sensors via Bluetooth, obtains the sensor code key and MAC address, and then converts the latter to an IP address and port. The PC is connected with UART through the network communication and, thus, communicates with sensors attached to the specified parts of human body and acquires the gesture quaternion from them in the real-time scale.

**2.4. Motion Angle Calculation.** This section includes two parts: (1) mathematical modeling and (2) motion angle calculation according to the mathematical model and gesture quaternion.

#### (1) Mathematical Modeling

It is necessary to elaborate the appropriate diagnostic criteria for the human lower limb joints, in particular, the left and right hips, knees, and ankles (six joints in total), which have different kinematics, motion functions, and range. A lot of efforts have been made to meet the needs of medical diagnosis and convenient modeling and measurement, resulting in the application of six coordinate systems corresponding to each joint and accounting for their structural and functional specifics.

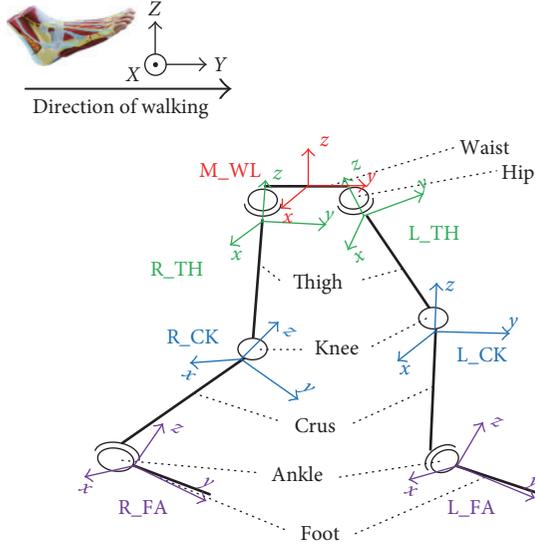


FIGURE 6: The mathematical model of the human lower limbs.

From the conventional medical standpoint, the hip athletic ability is assessed from the posture between the waist and thigh, the knee ability assessment is based on the relationship of thigh and crus, and the ankle is related to the crus and foot. In this paper, the analysis of the abilities of human lower limb is based on the same method as that used by the conventional medical approach. The mathematical model for lower limb motion detection is presented in Figure 6, where seven local coordinate systems are used, including the simplified waist level, left/right hip, left/right knee, and left/right ankle ones related to the gesture quaternion data acquired from the respective sensors, whose location is specified in Figure 4.

The human hip joint motion can be reduced to 3 DOFs: flexion-extension, exhibition-adduction, and internal-external rotation motion. Normally, the hip motion is compensatory. The range of hip motion is an important parameter for the human motion ability analysis. To analyze the ability of the hip, normally, three orthogonal axes are selected as the basic ones. Figure 7 shows the simplified human structure with the coordinates. Here, taking the same roll of human body, the pitch axis is defined between the femoral ends, the row axis coincides with the femoral bone axis, the waist coordinates are depicted by M\_WL, and the respective coordinate axes  $x$ ,  $y$ , and  $z$  are drawn in red. The coordinate systems of the left and right thighs are depicted as L\_TH and R\_TH, respectively.

The knee joint plays a critical role in the human walking process. Its rotation angle range is 0 to 135 degrees. For the knee joint, the bending of the knee during walking is referred to as flexion/extension, while its rotation about the other two axes (abduction/adduction and internal/external rotation) is generally quite small for this joint. Therefore, the knee motion during walking can be reduced to one flexion-extension DOF. The original coordinates are matched with the femoral bone, human roll axis, and the other axis following the right-hand rule, as is shown in Figure 8, where the left and right crus coordinates are L\_CK and R\_CK, respectively.

Ankles are critical for balance-keeping in the walking process by realizing such foot actions like dorsiflexion, plantar flexion, abduction, adduction, and various ever-sions. As shown in Figure 9, the original coordinates coincide with the crus bone, human roll axis, and the other axis following the right-hand rule. The left and right ankle coordinates L\_CK and R\_CK, respectively, are acquired during motion together with the left and right feet ones—L\_FA and R\_FA, respectively.

Thus, the hip joint has three rotation DOFs about the  $x$ -,  $y$ -, and  $z$ -axes, the knee joint has one rotation DOF about  $x$ -axis, and the ankle joint has three rotation DOFs about the  $x$ -,  $y$ -, and  $z$ -axes. Each DOF can be reduced to a single quaternion angle value measured via two coordinate systems. So each lower limb motion is described by 7 angular measurements, which implies that 14 ones are required for the motion description of both lower limbs. The angle calculation method is as follows.

## (2) Motion Angle Calculation

A mathematical model of joint is shown in Figure 10, where SK1 and SK2 stand for two bones connected by the joint  $J$ , and their coordinate systems are CS1 and CS2.

Two quaternions are acquired from sensors adjacent to the joint  $J$ . The SK1 gesture quaternion is defined as  $q_{JSK_n}$  and that of K2 as  $q_{JSK_{n+1}}$ . Then, quaternion data have to be calibrated by their initial quaternion values.

$$\begin{aligned} q_{J_n} &= q_{JSK_n} * q_{JSK0_n}^{-1} \\ q_{J_{n+1}} &= q_{JSK_{n+1}} * q_{JSK0_{n+1}}^{-1}, \end{aligned} \quad (2)$$

where  $q_{JSK0_n}$  and  $q_{JSK0_{n+1}}$  are the initial quaternions when the patient/participant under study stands still, while  $q_{J_n}$  and  $q_{J_{n+1}}$  are the respective quaternions after calibration.

Then, a conversion quaternion  $q_{JC}$  can be calculated by the following equation:

$$q_{J_{n+1}} * q_{J_n}^{-1}. \quad (3)$$

Defining the unit vector about  $x$ -,  $y$ -, and  $z$ -axes of the coordinate system by  $V_{JCU}$ , the unit vectors after conversion  $V_{JCU'_x}$ ,  $V_{JCU'_y}$ , and  $V_{JCU'_z}$  can be calculated via

$$V_{JCU}' = q_{JC} * V_{JCU} * q_{JC}^{-1}, \quad (4)$$

where

$$V_{JCU} = [V_{JCU_x}, V_{JCU_y}, V_{JCU_z}], \quad (5)$$

$$V_{JCU} = [V_{JCU_x}, V_{JCU_y}, V_{JCU_z}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

$V_{JCU_x}$ ,  $V_{JCU_y}$ , and  $V_{JCU_z}$  are the unit vectors about the three axes.

Next, a vector projected by the unit vector rotated by conversion quaternion can be calculated via

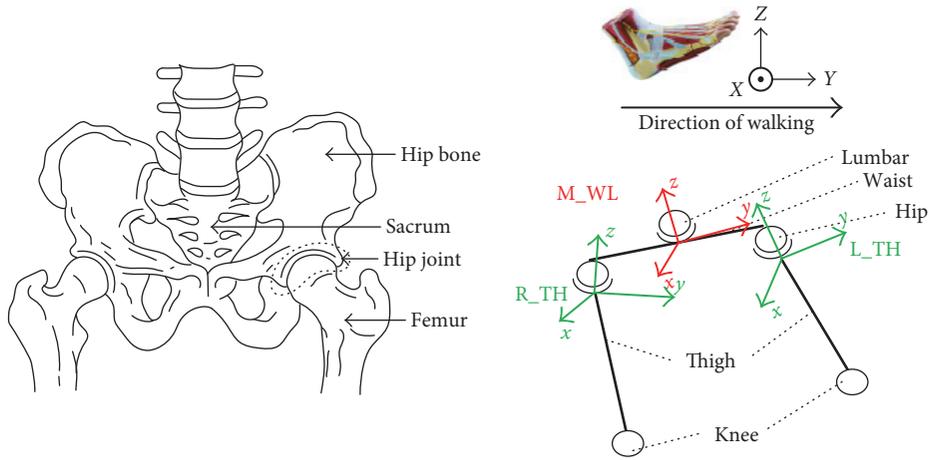


FIGURE 7: The physical structure and the coordinate system of the hip joint.

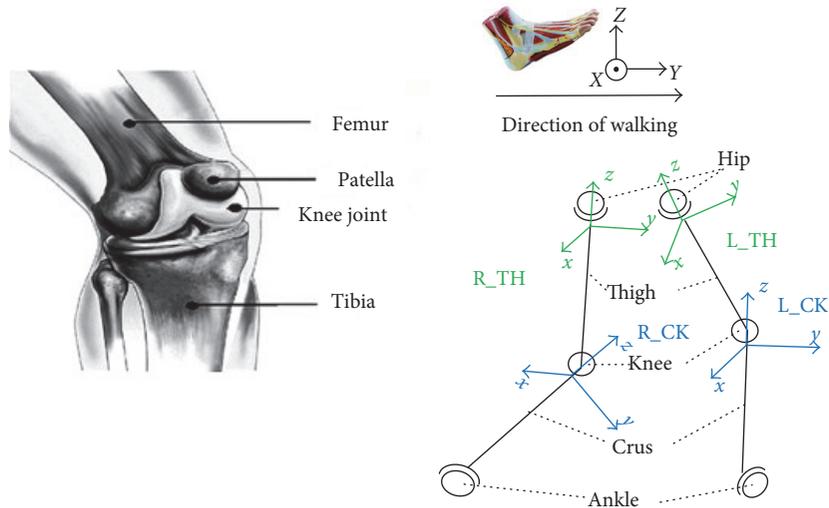


FIGURE 8: The physical structure and the coordinate system of the knee joint.

$$P_{JC_{TM}} = M_{P_{TM}} * V_{JCU}' * M_{CTM}, \quad (7)$$

where TM corresponds to the  $x$ -,  $y$ -, and  $z$ -axes and  $P_{JC_x}$ ,  $P_{JC_y}$ , and  $P_{JC_z}$  are vectors projected on YOZ, XOZ, and XOY planes, respectively; and

$$\begin{aligned} M_{P_x} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ M_{P_y} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ M_{P_z} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \end{aligned} \quad (8)$$

$$\begin{bmatrix} M_{C_x} & M_{C_y} & M_{C_z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad (9)$$

They are used to operate the matrix by simple row and column transformations.

The rotation angle about the coordinate axis can be calculated via

$$\theta_{J_{TM}} = \text{atan2}_{TM} \langle P_{JC_{TM}}, M_{CTM} \rangle, \quad (10)$$

where  $\theta_{J_{TM}}$  is the rotation angle about the TM axis. Function  $\text{atan2}_x(A, B)$  returns the argument of plural  $(y_B + z_A i)$  of two three-dimensional coordinates  $A(x_A, y_A, z_A)$  and  $B(x_B, y_B, z_B)$ ;  $\text{atan2}_y(A, B)$  returns the argument of plural  $(z_B + x_A i)$  of two three-dimensional coordinates  $A(x_A, y_A, z_A)$  and  $B(x_B, y_B, z_B)$ ; and  $\text{atan2}_z(A, B)$  returns the argument of plural  $(x_B + y_A i)$  of two three-dimensional coordinates  $A(x_A, y_A, z_A)$  and  $B(x_B, y_B, z_B)$ .

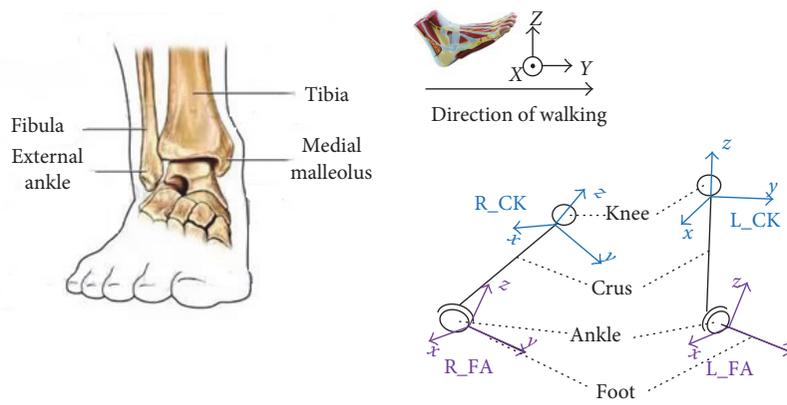


FIGURE 9: The physical structure and the coordinate system of the ankle joint.

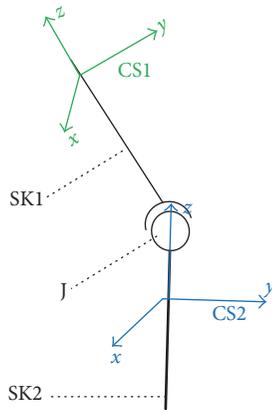


FIGURE 10: Mathematical model of the joint.

The angular speed and acceleration can be calculated by taking derivative of the rotation angle.

$$\begin{aligned} \omega_{J_{TM}} &= \frac{d\theta_{J_{TM}}}{dt} \\ \alpha_{J_{TM}} &= \frac{d\omega_{J_{TM}}}{dt}, \end{aligned} \quad (11)$$

where  $\omega_{J_{TM}}$  is the rotation angular speed about the TM axis and  $\alpha_{J_{TM}}$  is the rotation angular acceleration about the TM axis.

The joint  $J$  includes left hip (LP), left knee (LK), left ankle (LA), right hip (RH), right knee (RK), and right ankle (RA). Here,  $n$  is the number of sensors ranging from 1 to 7.

**2.5. Athletic Ability Evaluation and Visualization.** The joint angle graph describes the relationship between the angular variations of each joint in the gait cycle for the total gait cycle phase [25, 26]. The patients with hemiplegia exhibit the unilateral lower limb symptoms, such as inability of exercising one lower limb or unilateral handicap obstructing their harmonious motion, which are reflected in their lower limb rotation angle-angular speed curves. The difference and correlation between the above curves constructed for different joints of healthy and disabled participants can reflect the

degree of the patient illness, if any. By calculating the residual sum of squares for the motion data of the lower limb target side, as compared to those measured for the other side, the related ability degrees of healthy and disabled participants can be estimated.

$$R_J = \sum_{i=1}^7 \rho_i R_i, \quad (12)$$

where  $R_J$  is the rehabilitation evaluation of joint,  $R_i$  is the rehabilitation degree of each DOF of joint, and  $\rho_i$  is the adjustment coefficient of  $R_i$ . Since there are 7 motion angles,  $\rho = [0.3 \ 0.15 \ 0.15 \ 0.28 \ 0.06 \ 0.03 \ 0.03]$ .

The joint motion angular speed is a key parameter to reflect the performance of joint athletic ability, and the rotation angle-angular speed curve can provide a judgment for the joint motion characteristics.

$$E_J = \sum_{i=1}^7 \frac{A_{Di}}{A_{Hi}}, \quad (13)$$

where  $E_J$  is the rehabilitation state estimate: the closer to 1, the better rehabilitation state.  $A_{Di}$  and  $A_{Hi}$  are the track areas of lower limb rotation angle-angular speed curves of disabled and healthy participants, respectively.

### 3. Results

A series of walking experiments have been conducted to get the motion data of disabled and healthy participants. The disabled participant, 43 years old, has a serious movement dysfunction on his left side since the sequel of cerebrovascular disease. He suffers from strephenopodia on the ill side. His knee joint cannot bend normally, and he has to use his waist muscle making up the hip joint instead of the ankle flexion, so that he can lift toe off the ground and complete the step. The motion angle curves of disabled and healthy participants' hips knees and ankles are shown in Figure 11. The evaluation of disabled participant's joint motion angles is made via Equation (12), where  $R_J = 0.2964$ . The rotation angle versus angular speed curves of disabled and healthy participants' lower limbs are shown in Figure 12.

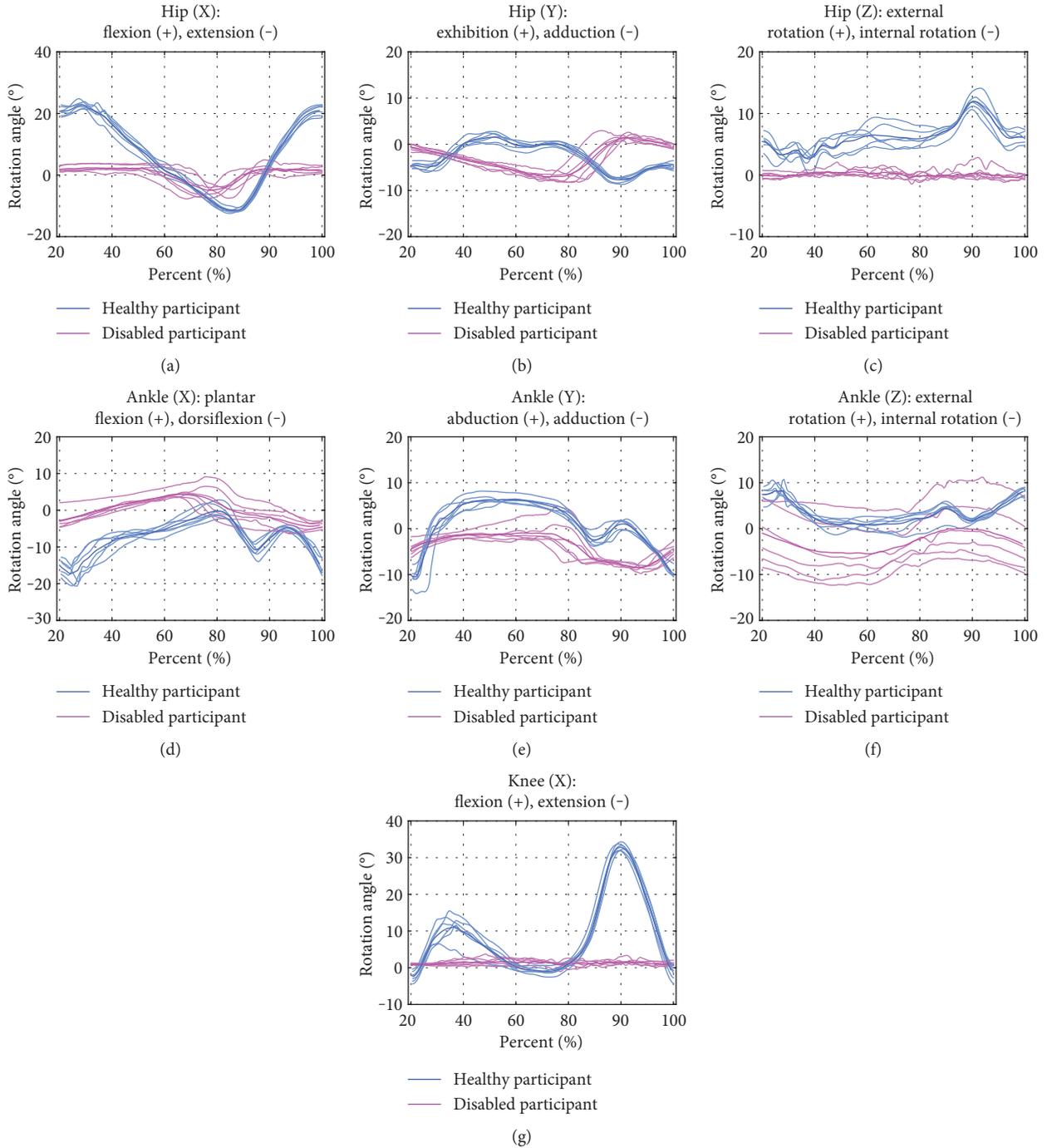


FIGURE 11: Rotation angle curves of joints' motion: 3 DOFs for hip (a, b, c), 3DOFs for ankle (d, e, f), and 1 DOF for knee (g). Here and in Figure 12, curves of healthy and disabled participants are shown in blue and red, respectively.

The evaluation of rehabilitation state of the disabled participant is performed via Equation (13), where  $E_J = 0.1951$ . The obtained value of  $R_J$  implies the large differences between the disabled and healthy participant's lower limb athletic abilities.

The disabled participant has a serious motion dysfunction, as compared with the healthy one, as is shown in Figure 11 and indicated by the obtained value of  $R_J = 0.2964$ . The disabled participant's hip, knee, and ankle

DOFs about the  $x$ -axis are limited and cannot accomplish the whole gait cycle properly, while the hip DOF about the  $z$ -axis is close to zero, while the ankle DOF about the  $z$ -axis exhibits a constant difference from that of the healthy participant.

The resulting disabled participant rehabilitation state estimate is not quite optimistic as is shown in Figure 12 and indicated by the obtained value of  $E_J = 0.1951$ . The athletic ability of joints, such as hip joint about the  $x$ - and  $y$ -axes,

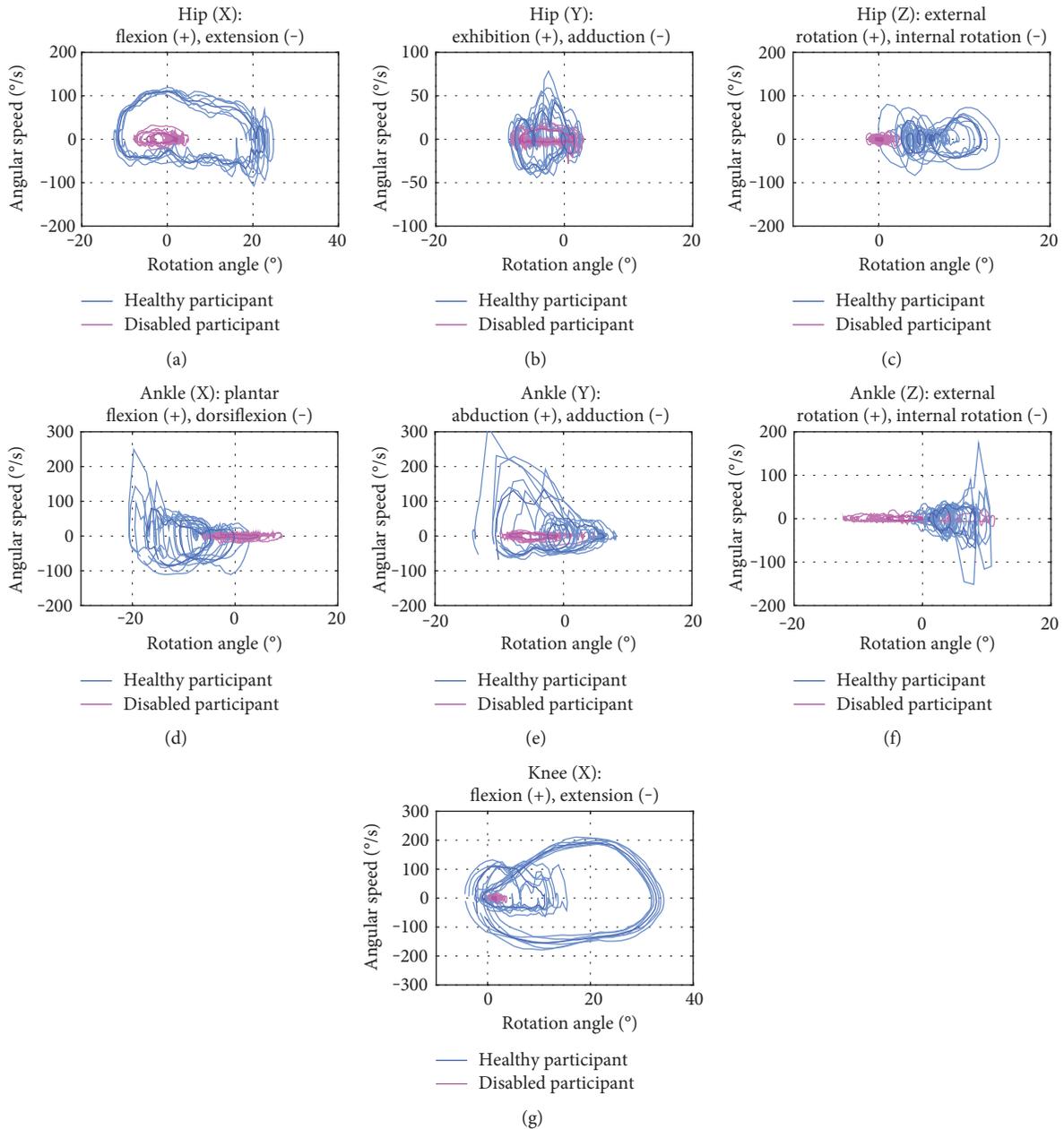


FIGURE 12: Rotation angle-angular speed curves of joints: 3 DOFs for hip (a, b, c), 3 DOFs for ankle (d, e, f), and 1 DOF for knee (g).

knee joint about the  $x$ -axis, and ankle joint about the  $x$ - and  $y$ -axes, is so deteriorated that a long rehabilitation period is required. The athletic ability of both hip and ankle joints about the  $z$ -axis exhibits a serious morbidity, insofar as the track area of rotation angle versus angular speed curves is far from that of the healthy participant.

The experimental results obtained imply that the proposed inertial sensor-based method of the lower limb motion analysis is quite practical, reliable, and applicable for rehabilitation state evaluation. The gesture quaternion of lower limbs based on inertial sensors can be converted to angles. The joint rotation angle can be calculated using the simplified lower limb motion model. Finally, the rotation angle versus angular speed curves for the hips, knees, and ankles are

constructed using the proposed algorithm, and the analysis of joint motion angles and athletic ability is provided, in order to evaluate the rehabilitation state.

#### 4. Conclusions and Future Work

Gait motion analysis plays an important role in the patient state evaluation. In this paper, a method for detecting the motion of human lower limbs including all degrees of freedom via the inertial sensors is proposed, which permits analyzing the motion ability according to the rehabilitation needs. This method is applicable to arbitrary walking directions and tracks of persons under study, and its results are unbiased, as compared to therapist qualitative estimations.

Using the simplified mathematical model of a human body, the rotation angles for each lower limb joint are calculated from the input signals acquired by the inertial sensors via the respective gesture quaternion. Finally, the rotation angle versus joint displacement curves are constructed, and the estimated values of joint motion angle and motion ability are obtained. The experimental verification of the proposed motion detection and analysis method was performed, which proved that it can efficiently detect the differences between motion behaviors of disabled and healthy persons and provide a reliable quantitative evaluation of the rehabilitation state.

As a future work, the proposed model refinement and more fine calibration of the experimental setup for minimization of motion detection errors are envisaged, in order to improve the method effectiveness and functionality. Next, the applied lower limb motion detection method can be integrated into the rehabilitation robot control system, realizing intelligent detection and evaluation. Eventually, the rehabilitation robots can be elaborated, which would provide the automatic adjustment of training parameters based on the particular patient status. Upon incorporation of the above features into the system, experiments will be arranged among hemiplegic patients to verify the feasibility and efficiency of the motion detection, robot control, and rehabilitation evaluation systems. Further development of this research is expected to have a significant influence on the motion detection, rehabilitation evaluation, and medical rehabilitation robot domains.

## Conflicts of Interest

There is no conflict of interests in this paper.

## Authors' Contributions

Tongyang Sun, Hua Li, and Quanquan Liu contributed equally to this work.

## Acknowledgments

The authors are thankful to the financial support from Science and Technology Foundation of Guangdong (nos. 2016A020220001 and 2014A020225004), Returned Overseas Business Foundation of Shenzhen (no. 2016001), Technology Research Foundation of Basic Research Project of Shenzhen (nos. JCYJ20170306170851910, JCYJ20160428110654601, and JCYJ20160428110354308), Research Foundation of Health and Family Planning Commission of Shenzhen Municipality (no. 201601054), Research Foundation of Beijing Advanced Innovation Center for Intelligent Robots and Systems (no. 2016IRS12), and Medical Research Foundation of Guangdong (no. A2017250).

## References

[1] *Aging Population Development Trend Forecasting Research Reports of China*, Members of the National Council on Aging office, 2006.

- [2] *State Department China Aging Development "Twelve Five" Plan*, vol. 2, 2011.
- [3] J. Rong, *Practical Hemiplegia Rehabilitation Technical Illustration*, People's Medical Publishing House, 2005.
- [4] S. Cosentino, K. Petersen, Z. Lin et al., "Natural human-robot musical interaction: understanding the music conductor gestures by using the WB-4 inertial measurement system," *Advanced Robotics*, vol. 28, no. 11, pp. 781–792, 2014.
- [5] L. D. Duffell, N. Hope, and A. H. McGregor, "Comparison of kinematic and kinetic parameters calculated using a cluster-based model and Vicon's plug-in gait," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 228, no. 2, pp. 206–210, 2014.
- [6] J. Zhou and Y. Q. Yu, "Coordination control of dual-arm modular robot based on position feedback using Optotrak3020," *Industrial Robot*, vol. 38, no. 2, pp. 172–185, 2011.
- [7] Organic Motion Inc., <http://www.Organicmotion.com>.
- [8] A. Pfister, A. M. West, S. Bronner, and J. A. Noah, "Comparative abilities of Microsoft Kinect and Vicon 3D motion capture for gait analysis," *Journal of Medical Engineering and Technology*, vol. 38, no. 5, pp. 274–280, 2014.
- [9] Polhemus Inc., <http://www.polhemus.com>.
- [10] "Aurora electromagnetic measurement system," <http://www.ndigital.com/medical/aurora.php>.
- [11] "Ultrasonic industrial positioning systems," <http://www.hexamite.com>.
- [12] X. Robert-Lachaine, H. Mecheri, C. Larue, and A. Plamondon, "Accuracy and repeatability of single-pose calibration of inertial measurement units for whole-body motion analysis," *Gait & Posture*, vol. 54, pp. 80–86, 2017.
- [13] X. Robert-Lachaine, H. Mecheri, C. Larue, and A. Plamondon, "Validation of inertial measurement units with an optoelectronic system for whole-body motion analysis," *Medical & Biological Engineering & Computing*, pp. 1–11, 2016.
- [14] J. C. van den Noort, S. H. Wiertsema, K. M. C. Hekman, C. P. Schönhuth, J. Dekker, and J. Harlaar, "Measurement of scapular dyskinesia using wireless inertial and magnetic sensors: importance of scapula calibration," *Journal of Biomechanics*, vol. 48, no. 12, pp. 3460–3468, 2015.
- [15] L. Schwickert, R. Boos, J. Klenk, A. Bourke, C. Becker, and W. Zijlstra, "Inertial sensor based analysis of lie-to-stand transfers in younger and older adults," *Sensors*, vol. 16, no. 8, p. 1277, 2016.
- [16] T. Sun, Q. Liu, W. Li et al., "Hip, knee and ankle motion angle detection based on inertial sensor," in *2016 IEEE International Conference on Information and Automation (ICIA)*, pp. 1612–1617, 2016.
- [17] Y. Gao, Z. Jiang, W. Ni et al., "A novel gait detection algorithm based on wireless inertial sensors," in *CMBEBIH 2017*, pp. 300–304, Springer, Singapore, 2017.
- [18] C. Tunca, N. Pehlivan, N. Ak, B. Arnrich, G. Salur, and C. Ersoy, "Inertial sensor-based robust gait analysis in non-hospital settings for neurological disorders," *Sensors*, vol. 17, no. 4, p. 825, 2017.
- [19] C. Nüesch, E. Roos, G. Pagenstert, and A. Mündermann, "Measuring joint kinematics of treadmill walking and running: comparison between an inertial sensor based system and a camera-based system," *Journal of Biomechanics*, vol. 57, pp. 32–38, 2017.

- [20] J. Kodama and T. Watanabe, "Examination of inertial sensor-based estimation methods of lower limb joint moments and ground reaction force: results for squat and sit-to-stand movements in the sagittal plane," *Sensors*, vol. 16, no. 8, p. 1209, 2016.
- [21] T. Watanabe, *Gait Phase Based Force Control Method for Body Weight Support Gait Rehabilitation from Orthopedic Surgery [D]*, Doctor thesis of Waseda University, Tokyo, 2012, (in Japanese).
- [22] Z. Chen and H. J. Lee, "Knowledge-guided visual perception of 3D human gait from a single image sequence," *IEEE Transactions on Systems Man Cybernetics*, vol. 22, no. 2, pp. 336–342, 1992.
- [23] D. Lian, "The real transformations of quaternion vector and matrix," *Journal of XiaMen University*, vol. 42, no. 6, 2003.
- [24] J. B. Kuipers, *Quaternions and Rotations Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*, Princeton University Press, 1999.
- [25] A. G. Schache, P. Blanch, and D. Rath, "Three-dimensional angular kinematics of the lumbar spine and pelvis during running," *Human Movement Science*, vol. 21, pp. 273–293, 2002.
- [26] J. P. Hallorana, A. J. Petrelab, and P. J. Rullkoettera, "Explicit finite element modeling of total knee replacement mechanics," *Journal of Biomechanics*, vol. 38, pp. 323–331, 2005.

## Research Article

# Upper Limb Rehabilitation Robot Powered by PAMs Cooperates with FES Arrays to Realize Reach-to-Grasp Trainings

Xikai Tu,<sup>1</sup> Hualin Han,<sup>1</sup> Jian Huang,<sup>2</sup> Jian Li,<sup>3</sup> Chen Su,<sup>1</sup> Xiaobo Jiang,<sup>1</sup> and Jiping He<sup>4,5</sup>

<sup>1</sup>School of Industrial Design, Hubei University of Technology, Wuhan 430068, China

<sup>2</sup>School of Automation, Huazhong University of Science and Technology, Wuhan 430074, China

<sup>3</sup>School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China

<sup>4</sup>Advanced Innovation Center for Intelligent Robots and Systems, Beijing Institute of Technology, Beijing 100081, China

<sup>5</sup>Arizona State University, Tempe, AZ 85287, USA

Correspondence should be addressed to Jian Huang; [huang\\_jan@mail.hust.edu.cn](mailto:huang_jan@mail.hust.edu.cn) and Jiping He; [jiping.he@asu.edu](mailto:jiping.he@asu.edu)

Received 30 December 2016; Accepted 1 March 2017; Published 15 June 2017

Academic Editor: Chengzhi Hu

Copyright © 2017 Xikai Tu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The reach-to-grasp activities play an important role in our daily lives. The developed RUPERT for stroke patients with high stiffness in arm flexor muscles is a low-cost lightweight portable exoskeleton rehabilitation robot whose joints are unidirectionally actuated by pneumatic artificial muscles (PAMs). In order to expand the useful range of RUPERT especially for patients with flaccid paralysis, functional electrical stimulation (FES) is taken to activate paralyzed arm muscles. As both the exoskeleton robot driven by PAMs and the neuromuscular skeletal system under FES possess the highly nonlinear and time-varying characteristics, iterative learning control (ILC) is studied and is taken to control this newly designed hybrid rehabilitation system for reaching trainings. Hand function rehabilitation refers to grasping. Because of tiny finger muscles, grasping and releasing are realized by FES array electrodes and matrix scan method. By using the surface electromyography (EMG) technique, the subject's active intent is identified. The upper limb rehabilitation robot powered by PAMs cooperates with FES arrays to realize active reach-to-grasp trainings, which was verified through experiments.

## 1. Introduction

Nowadays, the population of patients with limb motor dysfunction is increasing, which is caused by nerve injuries associated with stroke, traumatic brain injury, or multiple sclerosis. Particularly, the prevalence rate of stroke in China is increasing rapidly. Stroke survivors with various degrees of motor dysfunction not only endure inconvenience of the daily lives but also feel great psychological pressure, in addition to economic burden on the family and society. Many types of rehabilitation robots have been developed to assist rehabilitation in individuals with stroke [1–9]. In order to help stroke patients to receive intensive rehabilitation trainings as much as possible, cost-efficient portable rehabilitation equipment used in the community or home should be developed for patients after discharge, which would be a major improvement of limb rehabilitation.

Many stroke rehabilitation experiments show a positive role in using FES for recovery of motor function. FES is a method for activating sensory-motor systems by delivering electrical charge in the form of bursts of electrical pulses. By surface electrodes, FES stimulates motor or sensory nerves of muscles and facilitates motor rehabilitation and function reconstruction. Wu et al. [10] adopted a hybrid method of combining bilateral arm training with FES in patients post-stroke to improve hand function, and a linear guide platform with FES feedback control is developed to execute the training of bilateral reaching movements. A robotic workstation for stroke rehabilitation of upper extremity using FES is developed by Freeman et al. [11]. They use voluntary control with the addition of electrical stimulation applied to muscles in the impaired shoulder and arm. FES can also realize the inhibition of abnormal reflexes and induce active movements [12]. Freeman et al. [13–15] in the University of Southampton have developed a portable upper limb exoskeleton system

for reaching rehabilitation trainings. The rehabilitation system is composed of a FES stimulator and the passive Armeo Spring [16]. It uses the spring force to compensate for the gravity of the patient's upper limb and uses FES to activate paralyzed muscles to produce driving power. Rehabilitation training is a kind of continuous training with a certain intensity, but it will not be continuous due to muscle fatigue caused by electrical stimulation. The power of this system is all generated from FES stimulating the muscle, so training time and intensity of training will depend on whether the muscle state is fatigue or not.

RUPERT, a portable upper limb exoskeleton rehabilitation robot, is developed by Arizona State University. The system with five degrees of freedom (DOFs) is activated by low-cost PAMs and controlled by adaptive sensory feedback control algorithms for smooth- and safe-guarded movements during the task-oriented training. The unique features of the proposed robotic system are that it (1) is anchored on each user's trunk and aligned at the shoulder of the trained arm; (2) generates unidirectional assistive pulling force in each joint to encourage active participation of the user during each movement; (3) provides gravity compensation only if the user is too weak; and (4) evaluates the effectiveness of therapy by performance analysis, which includes kinematic criteria and users' effort. The new design makes the proposed robotic device portable for the user and can be used in various positions (sitting or standing) and different locations. By using less actuators, the weight and cost of the robot have been significantly reduced. The therapeutic benefits of the robot are not limited since antigravity tasks can be carried out by the motion control system which adapts to specific gravity compensation. In comparison with our previous studies [6, 17-18], this research mainly focuses on the safety and feasibility of our latest robotic arm, which has one more DOF of humeral internal/external rotation for enlarging the reaching space. More stroke patients were enrolled, and further biometric analyses were performed including clinical laboratory therapy sessions and in-home therapy sessions for the purpose of enabling frequent training at home. Despite all this, RUPERT with one way actuator of the joint is not suitable for stroke patients with weak muscles in the flaccid paralysis period. In order to expand the range of RUPERT rehabilitation application including reaching exercises for ordinary patients with flaccid paralysis, FES is used to activate paralyzed muscles. FES induced muscle force, and a pneumatic muscle pull force is a new kind of combination actuation, which can produce muscle torque and compensate the drawbacks of RUPERT. They cooperate together and realize the robotic joint two-way movement. Our proposed hybrid system in this research can allow patients to receive more lasting endurance rehabilitation trainings than the system developed by C.T. Freeman et al. As both the exoskeleton robot driven by PAMs and neuromuscular skeletal system under FES possess the highly nonlinear and time-varying characteristics, which add control difficulty to the hybrid dynamic system, ILC is studied and taken to control this newly designed hybrid rehabilitation system to realize repetitive task trainings. The transfer of ILC to rehabilitation is based on the patient making

repeated attempts to complete a task, such as reaching out over a table top to an object.

The ability of grasping and releasing the object plays an important role in our daily lives. Most patients with stroke suffer from the hand dysfunction, the symptoms of which are that finger flexor muscle tone is high and patients cannot open their own hands actively. The hand rehabilitation includes two kinds of intelligent strategies: robot-assisted and FES. Heo et al. summarized the existing multi-DOF hand rehabilitation exoskeletons [19]. The number of the hand's joints is up to 22 DOF, which will make the mechanical and electrical designs of the hand exoskeleton very complicated. As stroke patients' hands show the phenomenon of abnormal nerve reflex, the rehabilitation exoskeleton is hard for patients to wear, even patients experiencing the secondary damage in the process of putting it on. FES can activate paralyzed muscles to produce joint movement through stimulus pulses conducted by use of surface-adhesive electrodes, but the precise finger joint movement by FES is not realized and therefore it is difficult to produce FES-induced functional grasping and releasing. The reasons are that the shapes of ordinary self-adhesive electrodes are bigger and their stimulus selectivity is not enough and yet it can activate many finger muscles at the same time. Westerveld et al. [20] invented an artificial way to paste small pad electrodes above the motor points of finger flexors and extensors and implemented the hand grasping and releasing with the help of FES and model predict control. The shortcomings of this method are that it takes much time of the therapist to place small electrodes accurately on the corresponding motor points. Malešević et al. [21] developed a  $4 \times 4$  electrode array, which can achieve the intelligent trainings of hand grasping and releasing by virtue of FES, but it cannot implement reach-to-grasp trainings. In order to realize reach to grasp trainings, Westerveld et al. added a 3-DOF end-effector rehabilitation robot to the FES system with pad electrode [22]. They used robot and FES to achieve reaching and grasping, respectively [23], but this end-effector rehabilitation system guided the patient's upper limb movement only through his/her hand but the shoulder, elbow, and wrist joints could not be rehabilitated individually. RUPERT upper limb rehabilitation exoskeleton integrated with FES can overcome the above shortcomings by the search algorithm. The surface electrode array is composed of many small electrodes arranged in matrix  $4 \times 6$  form, which can realize each finger selective stimulation. This setup can solve the time-consuming problem of self-adhesive electrode placement while RUPERT can realize the multijoint coordination trainings.

Active training involves motion intention recognition, and now these sensors most widely used include two categories: electromechanical and bioelectrical. Electromechanical sensors mainly include position and force/torque ones, and this kind of sensors has electromagnetic mechanical time delay, especially expensive high-performance multiaxis force/torque sensors. Bioelectric sensors collect biological signals such as ECG, EMG, and EEG, and the time delay of this kind is shorter than the previous kind. EMG occurs 20–30 milliseconds ahead of muscles producing joint

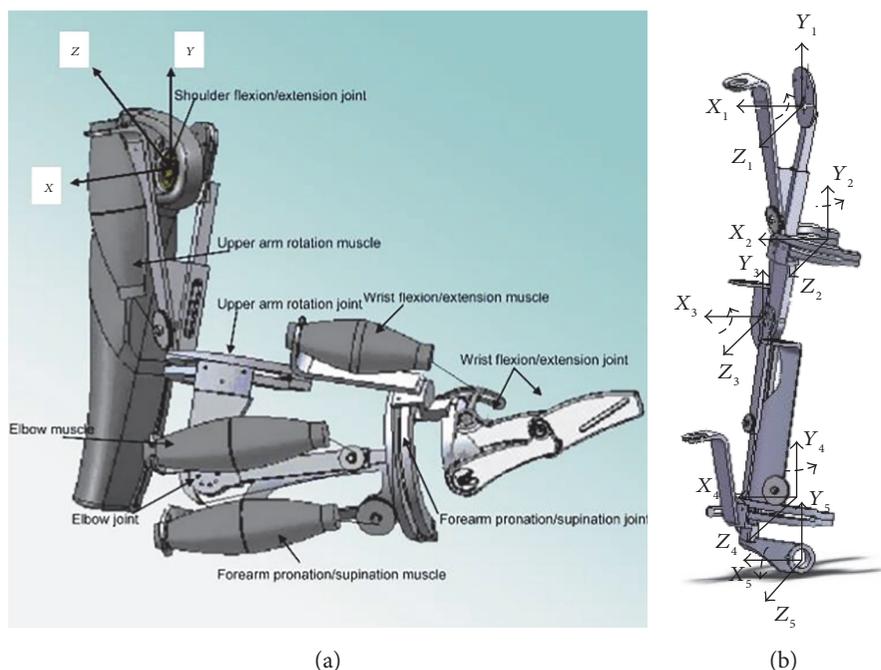


FIGURE 1: (a) The assembly drawing of the 5-DOF upper limb exoskeleton of the RUPERT and (b) the coordinate system diagram of RUPERT [6].

movement. Many rehabilitation devices use surface EMG to extract biological information as a way of identifying human motion intention. It is also used in the intention-based FES to actively activate the muscles to produce movement, but surface EMG is buried in stimulus artifact and induced muscle response (M wave). Comb filter and blanking window methods are used to extract intention information for the active intent.

In this paper, it introduces that upper limb rehabilitation robot powered by PAMs cooperates with FES arrays to realize active reach-to-grasp trainings for stroke patients. In Section 2, the dynamic models of a pneumatic muscle and FES-induced muscle are built for reaching trainings. In Section 3, the subject's active intent is identified using EMG and grasping and releasing are realized by FES array electrodes. Section 4 introduces the ILC control strategy and its practical application to reach-to-grasp trainings by virtue of robot and FES. Section 5 reports the experimental results of PAMs in cooperation with FES arrays to realize active reach-to-grasp trainings. Conclusion and future work are shown in Section 6.

## 2. Dynamic Models of PAMs and FES Muscle for Reaching Trainings

*2.1. The Modeling and Identification of RUPERT.* The RUPERT upper limb rehabilitation robot has 5 DOFs: shoulder flexion/extension, humeral internal/external rotation, elbow flexion/extension, forearm pronation/supination, and wrist flexion/extension shown in the Figure 1(a). Figure 1(b) shows the mechanical design of the RUPERT robot. For each DOF, a pneumatic muscle is used as a uni-directional actuator to generate a joint pulling force. This

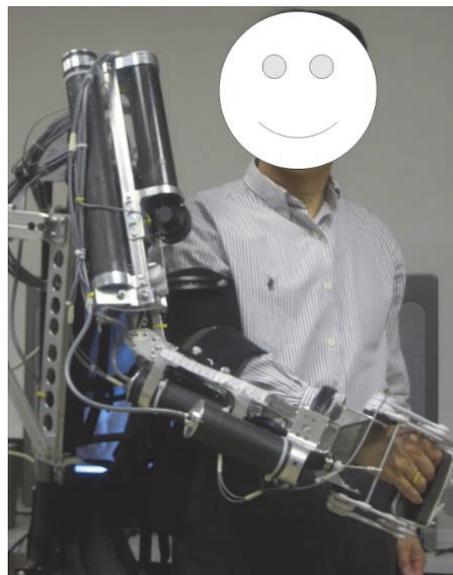


FIGURE 2: The diagram of the subject wearing the 5-DOF RUPERT [6].

accords with the stroke patients' symptoms and that means the muscle is in a condition of high muscular tension while flexor muscles will produce involuntary contraction. As FES can stimulate paralyzed muscles to move against a PM-driving direction, RUPERT can achieve two-way joint movement with the help of FES, which enables more patients to use RUPERT to do rehabilitation trainings in different recovering phases. The depiction of a subject wearing the RUPERT robot is shown in Figure 2.

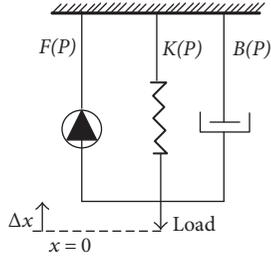


FIGURE 3: Three-element model of pneumatic muscle.

$$\overline{M}(\theta)\ddot{\theta} + \overline{B}(\dot{\theta})\dot{\theta} + \overline{K}(\theta)\theta + \overline{G}(\theta) + \Delta\overline{\tau} = \overline{\tau}_p - \overline{\tau}_{\text{FES}}. \quad (1)$$

The hybrid dynamic system of RUPERT exoskeleton and FES neuromuscular model is shown in (1) and in which  $\theta, \dot{\theta}, \ddot{\theta} \in R^5$  are joint angle, angular velocity, and angular acceleration of 5-DOF RUPERT, respectively.  $\overline{M}(\theta) \in R^{5 \times 5}$  is symmetric positive definite inertia matrix, and  $\overline{B}(\dot{\theta})$ ,  $\overline{K}(\theta) \in R^{5 \times 5}$  are damping matrix and stiffness matrix, respectively.  $\overline{G}(\theta) \in R^5$  is the gravity moment. Based on the previous work [17, 18, 24, 25], it is known that  $\overline{M}(\theta)$ ,  $\overline{B}(\dot{\theta})$ , and  $\overline{K}(\theta)$  are diagonal matrix and  $\overline{B}(\dot{\theta})$  and  $\overline{K}(\theta)$  are set to a constant value shown in (2). The values of  $\overline{M}(\theta)$  and  $\overline{G}(\theta)$  vary according to the different subjects by use of the specific calculation method according to the literature [17].  $\Delta\overline{\tau}$  is torque generated by patients' muscle forces and other disturbances.  $\overline{\tau}_p$  is the torque generated by pneumatic muscle, and  $\overline{\tau}_{\text{FES}}$  is torque generated by neuromuscular electrical stimulation. The dynamic models of pneumatic muscle play an important role in this hybrid combination. So this chapter introduces the modeling and identification of pneumatic muscle.

$$\begin{aligned} \overline{B}(\dot{\theta}) &= \text{diag}([0.01, 0.015, 0.03, 0.02, 0.02]) \\ \overline{K}(\theta) &= \text{diag}([0.005, 0.02, 0.01, 0.005, 0.01]). \end{aligned} \quad (2)$$

By virtue of the physical model, the controlling of the pneumatic system would become more complex, as some parameters are not easy to be detected. After linear simplification, damping-related items are omitted, which are not conducive to the relatively rapid real-time control of the lower limb gait. The phenomenon model is taken from the external observation, which is usually represented by a mass-stiffness-damping dynamic system. This kind of model is also called the three-element model of pneumatic muscle, and its equations can be described as shown in the following:

$$\begin{aligned} M(\ddot{x} + g) + B(P)\dot{x} + K(P)x &= F(P) \\ B(P) &= B_0 + B_1P \\ K(P) &= K_0 + K_1P \\ F(P) &= F_0 + F_1P \\ B(P) &= B_{0i} + B_{1i}P \quad (\text{inflation}) \\ B(P) &= B_{0d} + B_{1d}P \quad (\text{deflation}). \end{aligned} \quad (3)$$

Three elements include inertia  $M$ , damping  $B$ , and stiffness  $K$  shown in Figure 3. The only input control variable is pneumatic pressure  $P$ , and  $F(P)$  is active contraction force. Damping  $B$  has different values according to the process of inflation and deflation, respectively.  $B(P)\dot{x}$  is the viscous force impeding the pneumatic muscle movement itself, and  $K(P)x$  is the spring force impeding the pneumatic muscle shortening.  $M(\ddot{x} + g)$  is the driving force for the load, of which  $M$  is the load mass,  $g$  is the gravity acceleration, and  $x$  is the axial contraction length of the pneumatic muscle.  $x = 0$  is marked as the initial position of the pneumatic muscle in a completely bleeding state. The contraction coefficient and stiffness coefficient are obtained by using different pressures and least square method (LSM) through the static force balance experiments. Damping system is obtained through the static disturbance experiment, and for their specific identification process, please refer to the literature [26]. Pneumatic muscle experimental platform is shown in Figure 4 while the identification results are shown in Table 1.

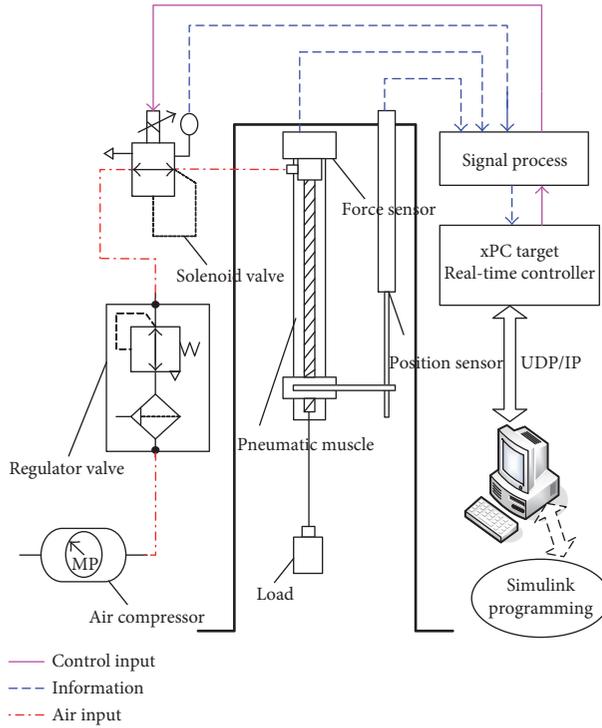
**2.2. Modeling and Identification of FES Muscle.** Neuromuscular electrical stimulation models are widely used in various fields of researches, which can explore the characteristics of isometric and nonisometric contraction of muscles. In the case of nonisometric contraction, the force produced by the muscle is not only related to the length of contraction, but also the rate of muscle contraction. If the muscle lies in the condition of equal length changes such as elongation or shortening in the case of isometric contraction, the muscle produces the maximum contraction force and then the maximum muscle contraction force will be reduced. Hill model is the most commonly used model for muscle modeling, by use of mass-spring-damping to describe the dynamic behavior of muscle. Durfee model [27] is expanded on the basis of the Hill muscle model.

$u(k)$  is the input variable of electrical stimulation signal, and  $k$  is the  $k$ th sampling.  $f(u(k))$  is the "static nonlinear" function of the discrete-time Hammerstein model shown in Figure 5, and linear dynamic function is  $G(q^{-1})$ .  $q^{-1}$  is delay factor, and  $m$  and  $n$  are the poles and zeroes of the transfer function  $G(q^{-1})$ , respectively.  $d$  is the sample number of time delay.  $v(k)$  is disturbance, and  $y(k)$  is the output of the neuromuscular electrical stimulation of muscle force or torque. Nonlinear function  $f(u(k))$  is the cubic spline function;  $u_1, u_2, u_3, \dots, u_l$  are cubic spline interpolation points shown in the following [28]:

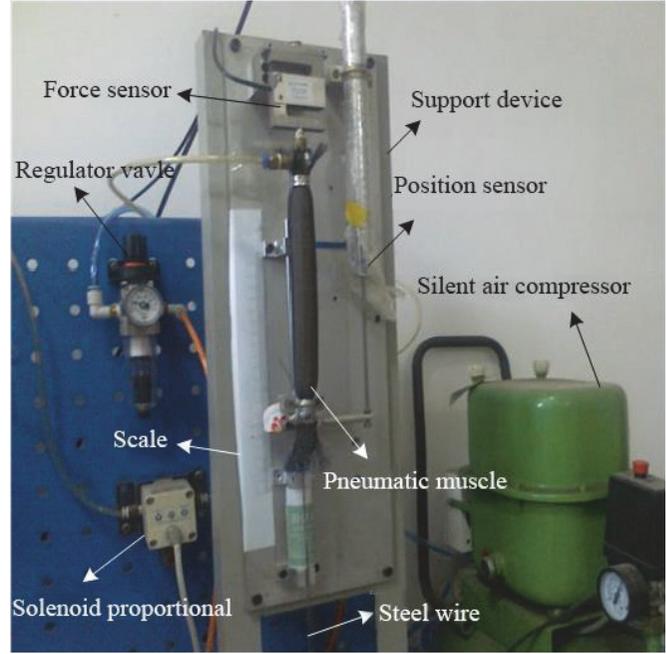
$$G(q^{-1}) = \frac{B(q^{-1})}{A(q^{-1})} = \frac{q^{-d}(b_0 + b_1q^{-1} + \dots + b_nq^{-n})}{1 + a_1q^{-1} + \dots + a_mq^{-m}} \quad (4)$$

$$\theta_G = \begin{bmatrix} \theta_a \\ \theta_b \end{bmatrix} = [a_1 \dots a_l \quad b_0 \quad b_1 \dots b_n]^T \quad (5)$$

$$\begin{aligned} f(u(k)) &= \sum_{i=1}^{l-2} c_i u(k) - u_{i+1}(k)^3 + c_{l-1} + c_l u(k) + c_{l+1} u^2(k) \\ &\quad + c_{l+2} u^3(k), \quad u_{\min} = u_1 < u_2 < u_3 < \dots < u_l = u_{\max} \end{aligned} \quad (6)$$



(a)



(b)

FIGURE 4: The platform for modeling the characteristics of pneumatic muscle, (a) the pneumatic circuit and control platform, and (b) the experimental device for modeling pneumatic muscle.

TABLE 1: Three-element model of pneumatic muscle model identification parameters.

Coefficient	Contraction		Stiffness		Damping (inflation)		Damping (deflation)	
	$F_0$	$F$	$K_0$	$K_1$	$B_0$	$B_1$	$B_0$	$B_1$
PAM								
Shoulder flexion	269.5	1.71	8.65	0.0505	1.31	0.008	0.68	0.0009
Humeral rotation	130.3	0.98	6.11	0.0295	0.88	0.005	0.48	0.0006
Elbow extension	160.7	1.23	6.56	0.0341	0.98	0.006	0.53	0.0007
Forearm pronation	120.3	0.86	5.48	0.0265	0.76	0.004	0.42	0.0005
Wrist extension	115.1	0.81	5.73	0.0243	0.73	0.004	0.39	0.0005

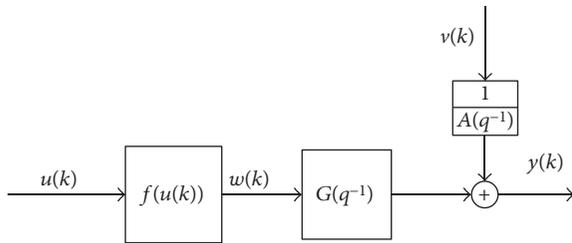


FIGURE 5: Discrete-time Hammerstein model.

$$\theta_f = [c_1 \ c_2 \ \dots \ c_{i+2}]^T \quad (7)$$

$$\theta = \begin{bmatrix} \theta_G \\ \theta_f \end{bmatrix} \quad (8)$$

$$\|v\|_2^2 = \sum_{k=1}^N v^2(k). \quad (9)$$

Therefore, (9) can be regarded as the problem of least square shown in (10) to (11). Hammerstein structure non-linear function and linear system parameter identification are taken by the use of an iterative algorithm using the ARX model.

$$\arg \min_{\theta_f} \left\| Y_f(y, \hat{\theta}_a) - \Phi_f(u, \hat{\theta}_b) \theta_f \right\|_2 \quad (10)$$

$$\hat{\theta}_G = \Phi_G(u, y, \hat{\theta}_f)^T \Phi_G(u, y, \hat{\theta}_f)^{-1} \Phi_G(u, y, \hat{\theta}_f)^T Y'. \quad (11)$$

The surface electrical stimulator (Hasomed, Rehasim2) is a constant current source with eight stimulus channels.

TABLE 2: The identification value of neuromuscular electrical stimulation.

Parameter	Corresponding parameters of muscle model								
	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\beta_5$	$a_1$	$a_2$	$b_0$	$b_1$
Value	$2.41 \times 10^{-8}$	-0.0294	0.0023	$-6.98 \times 10^{-6}$	$1.56 \times 10^{-8}$	-1.21	0.117	0.1	0

Real-time control can be realized based on the ScienceMode2 communication protocol by the use of RS232 serial port with Simulink xPC Target.

The process of generating test data is called TR (triangular ramp). The value of pulse width is linear from 0 to  $350\mu s$  and then back to 0 value. Its range has been distributed while the stimulus frequency is 20 Hz, and the stimulus amplitude is 20 mA. Nonlinear (6) of this proposed electrical stimulation model is rewritten as (12). Neuromuscular electrical stimulation dynamic contraction process can be expressed with the two order system [28], so the parameter values of linear expression (4) are  $d=1, n=1, m=2$ , respectively. The electrical stimulation model parameters of the biceps are shown in Table 2, and the functional equation of identification is shown in (13) to (14). Consider

$$f(u) = \beta_1 u - 150^3 + \beta_2 + \beta_3 u + \beta_4 u^2 + \beta_5 u^3 \quad (12)$$

$$f(u) = -0.0294 + 0.0023u - 6.98 \times 10^{-6}u^2 + 1.56 \times 10^{-8}u^3 + 2.41 \times 10^{-8}u - 150^3 \quad (13)$$

$$G(q^{-1}) = \frac{q^{-1}(1 - 0.364q^{-1})}{1 - 1.21q^{-1} + 0.117q^{-2}}. \quad (14)$$

### 3. EMG Triggered Active Grasping and Releasing Trainings

**3.1. Surface Array Electrodes Used in the Grasping and Releasing.** Surface finger muscles related with grasping and releasing include flexor pollicis longus (FPL), extensor digitorum communis (EDC), and thumb thenar muscle. Grasping is generally divided into two categories: power grasping and precision grip. To test the effectiveness of grasping and releasing by using the exoskeleton and FES, "power grasping" under electrical stimulation is relatively simple and taken as the research paradigm. To be simplified, flexor pollicis longus (FPL) and extensor digitorum communis (EDC) are selected as the stimulus objects shown in Figure 6. 42% of hand movement only needs four fingers, including the index finger, middle finger, ring finger, and little finger. Metacarpal phalangeal (MCP) is the metacarpophalangeal joint, and proximal interphalangeal (PIP) is the proximal interphalangeal joint.  $\phi_{kM}$  is the angle of the palm and finger joint while  $\phi_{kP}$  is the proximal interphalangeal joint angle, and  $k=1, 2, 3, 4$  represent the index finger, middle finger, ring finger, and little finger, respectively, as shown in Figure 7.  $4 \times 6$  array electrodes in Figure 8 were used while stimulation system adopted German Rehasim2, which used the Omron G3MB solid-state relay with switch frequency 5 kHz. MCP and PIP joint angles were measured by using Cyberglove.

$$\text{RMS} = \sqrt{\frac{1}{8} \sum_{k=1}^4 [(\phi_{dM}(t) - \phi_{kM}(t))^2 + (\phi_{dP}(t) - \phi_{kP}(t))^2]}. \quad (15)$$

The desired hand gesture was realized by matrix scanning method applied to array electrodes. The joint angle errors were kept within the plus or minus 3 degrees. Electrical stimulation frequency was 20 Hz, and stimulating pulse width 350 was constant. Stimulating pulse width 0–15 mA was regarded as the stimulus variable, and  $u$  was the electrical stimulation amplitude. Root mean square error RMS in (15) was taken as a performance optimization goal. When RMS was the smallest, the corresponding combination of electrode array targets was selected by trials and errors. The range of stimulus amplitude was from 2 mA to 15 mA while electrode array target number was from  $n=2$  to  $n=12$ , as shown in Figure 9.

**3.2. Real-Time Intention Extraction of Surface EMG under FES.** Intention-based EMG can be taken as the trigger signal of robot and FES, but it is contaminated by FES stimulus artifact in this research. A strategy was developed that real-time intention surface EMG was extracted from FES stimulus signals. Surface EMG sensor was using the model SX230 of Biometrics Corporation. Simulink xPC target system was used, as shown in Figure 10.

$$y(k) = \frac{x(k) - x(k - N_s)}{\sqrt{2}}. \quad (16)$$

Electrical stimulation artifact was detected using Simulink Comb filter [29], shown in (16) to (17).  $x(k)$  is the  $k$ th sample of the original signal, and  $N_s$  is the sampling number of two adjacent stimulus intervals.  $\sqrt{2}$  is the energy matching coefficient, and  $y(k)$  is the filter EMG. In order to make the active intention EMG suitable for trigger control, it needs to be normalized.  $\text{EMG}_f$  is the actual electrical amplitude while  $\text{EMG}_0$  is the envelope line.  $\text{EMG}_{\text{Max}}$  is the envelope line in the condition of muscle isometric contraction.  $\alpha$  represents the active intent coordination coefficient, and its arrange is [0-1]. 0 means no intention output while 1 stands for maximum active power.

$$\alpha = \frac{\text{EMG}_f - \text{EMG}_0}{\text{EMG}_{\text{Max}} - \text{EMG}_0}. \quad (17)$$

### 4. Iterative Learning Control

Rehabilitation training is a kind of repetitive training. Body state of patients will improve with an increase in the number

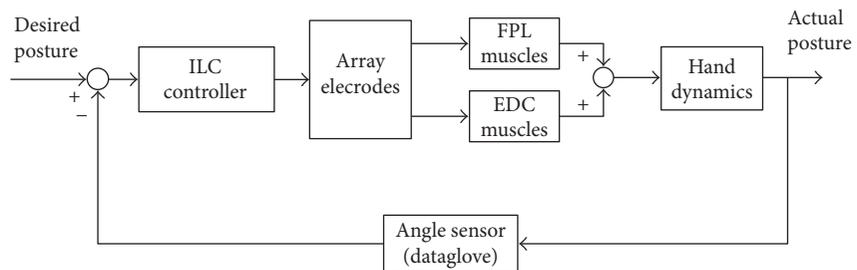
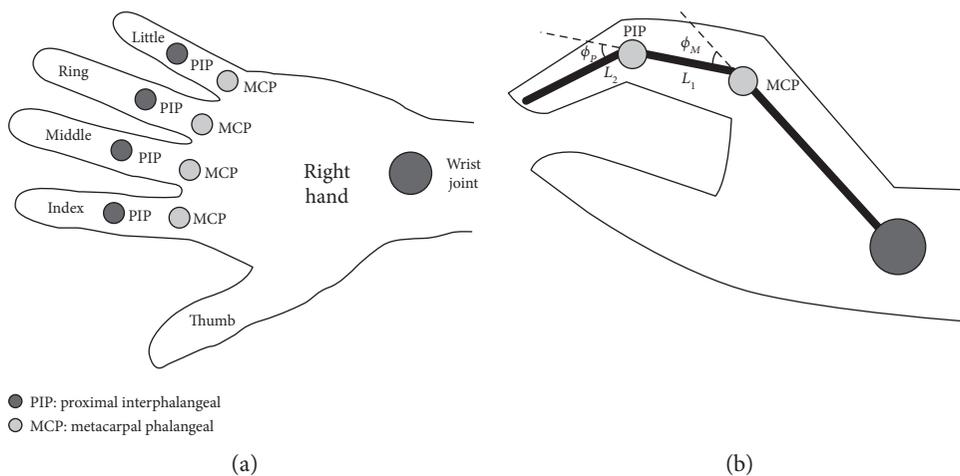
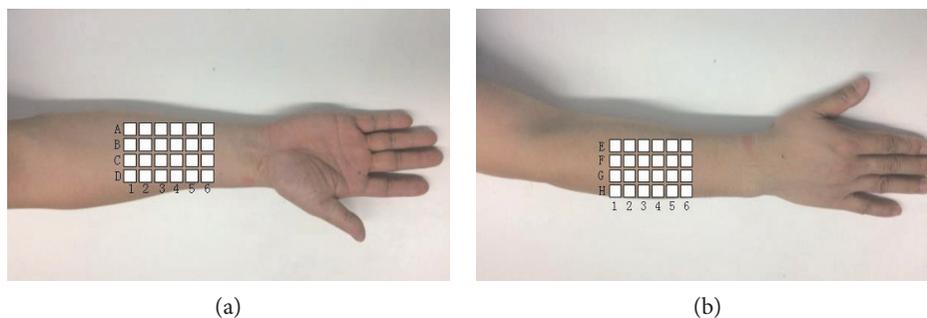


FIGURE 6: Control diagram of FES array electrodes to realize hand grasping and releasing.

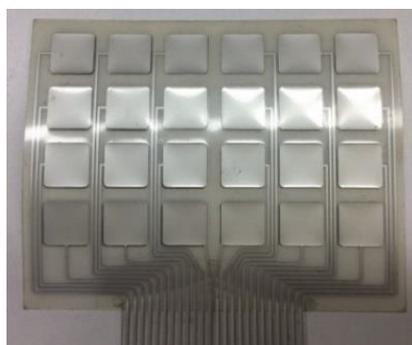


(a) (b)

FIGURE 7: Schematic diagram of finger joint and angle.



(a) (b)



(c)

FIGURE 8: (a) Array electrode for FPL, (b) array electrode for EDC, and (c) array electrode prototype.

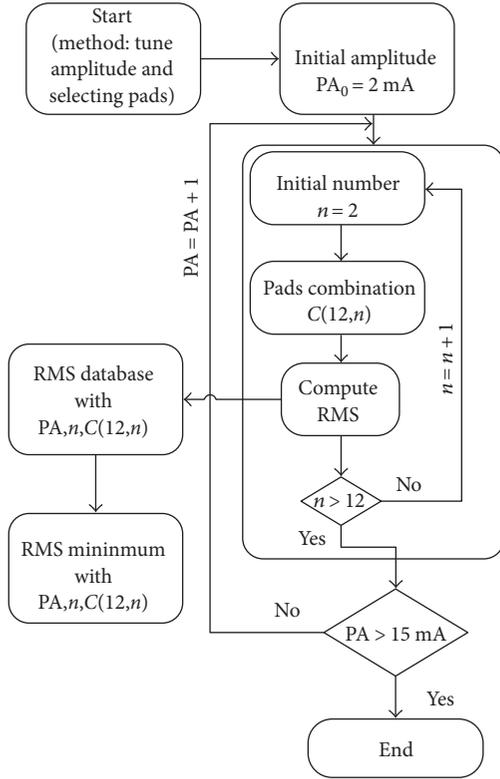


FIGURE 9: Block diagram of array electrode matrix scanning method.



FIGURE 10: The picture of intention EMG extraction under electrical stimulation.

of training while the auxiliary level of robot and electrical stimulation will be reduced. In addition, it becomes more difficult to control the exoskeleton system because of its highly nonlinear and time-varying characteristics, which are caused in the existence of nonlinear actuators including PMAs and FES. Iterative learning control improves the dynamic system control performance by use of the previous errors and control inputs, which is consistent with the process of rehabilitation training. Patients try to complete the appointed tasks with the help of RUPERT and FES, and the desired trajectory and the actual trajectory will produce movement error. After the one training finishes, the robot returns to the initial position and the motion error information can be used as a prior knowledge of the next training.

Iterative learning control is in line with this repetitive training mode shown in Figure 11.

With an increase of assisted rehabilitation trainings, patients' upper limb motor function will improve gradually, so the contribution of patients' active muscular force will increase and the assistance of RUPERT and FES will reduce. Newton's iterative learning control (ILC Newton) is applied in this hybrid rehabilitation system [30, 31].  $k$  is the number of iterations, and  $\theta_d$ ,  $\theta_k$ ,  $e_k$ , and  $u_k$  are the expected angle, the actual angle, the angle error, and the control input of the  $k$  times iterations, respectively, shown in the following:

$$x_k(p+1) = f(x_k(p), u_k(p)) = Ax_k(p) + Bu_k(p)$$

$$\theta_k(p) = h(x_k(p)) = Cx_k(p), x_k(0) = x_0$$

$$\theta_k = [\theta_k^T(0) \theta_k^T(1) \dots \theta_k^T(T)]^T$$

$$u_k = [u_k^T(0) u_k^T(1) \dots u_k^T(T)]^T$$

$$\theta_d = [\theta_d^T(0) \theta_d^T(1) \dots \theta_d^T(T)]^T$$

$$u_{k+1} = u_k + Le_k$$

$$e_k = \theta_d - \theta_k$$

$$\lim_{k \rightarrow \infty} \|e_k\| = 0, \lim_{k \rightarrow \infty} \|u_k - u_d\| = 0$$

$$\theta(0) = Cx(0) = g_0(x(0))$$

$$\theta(1) = Cx(1) = C(Ax(0) + Bf(u(0)))$$

$$= g_1(x(0), u(0))$$

$$\theta(2) = Cx(2) = C(Ax(1) + Bf(u(1)))$$

$$= CA(Ax(0) + Bf(u(0))) + CBf(u(1))$$

$$= g_2(x(0), u(0), u(1))$$

$$\vdots$$

$$\theta(N-1) = Cx(N-1) = C(Ax(N-2) + Bf(u(N-2)))$$

$$= g_{N-1}(x(0), u(0), u(1), \dots, u(N-2))$$

$$\theta = g(\cdot) = [g_0(\cdot), g_1(\cdot), g_2(\cdot), \dots, g_{N-1}(\cdot)]^T.$$

(18)

Newton method is well known for searching approximate real roots of nonlinear functions through successive approximation, and the specific process is to develop the real valued function  $\theta_d - g(u_k)$  by using the Taylor series. By selecting a few terms of the Taylor approximation series of real valued functions and using iterative method for solving  $\theta_d - g(u_k) = 0$ , approximate roots are gotten. Given a function  $\theta_d - g(u_k)$  and its derivative  $g'(u_k)$ , the root of the iterative estimation is shown in the following equations.  $u_k$  is the iteration estimates of  $k$  times, and  $u_{k+1}$  is the iteration estimates of the  $k+1$  times. Newton method converges fast, because the mathematical expression of its convergence rate is the second order rather than linear, and the premise is the existence of inverse  $g'(u_k)$  that means the existence of  $g'(u_k)^{-1}$ .

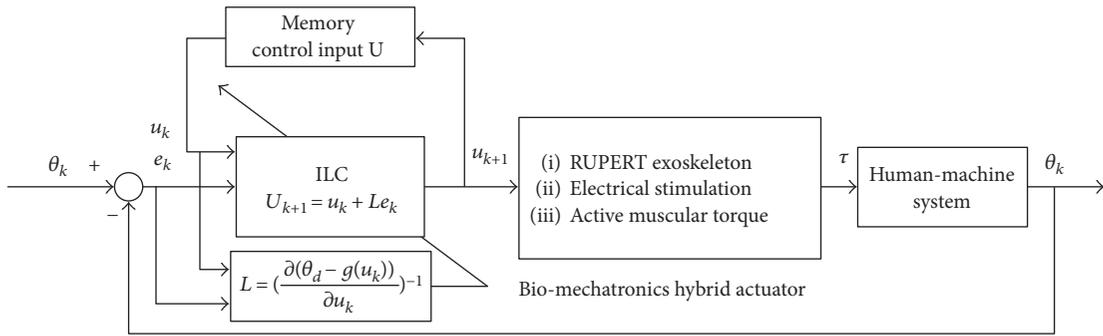


FIGURE 11: Block diagram of iterative learning control for upper limb rehabilitation system.

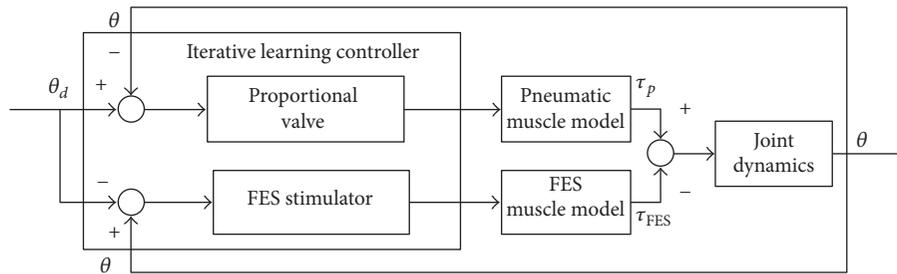


FIGURE 12: Control block diagram of RUPERT and FES hybrid actuators.

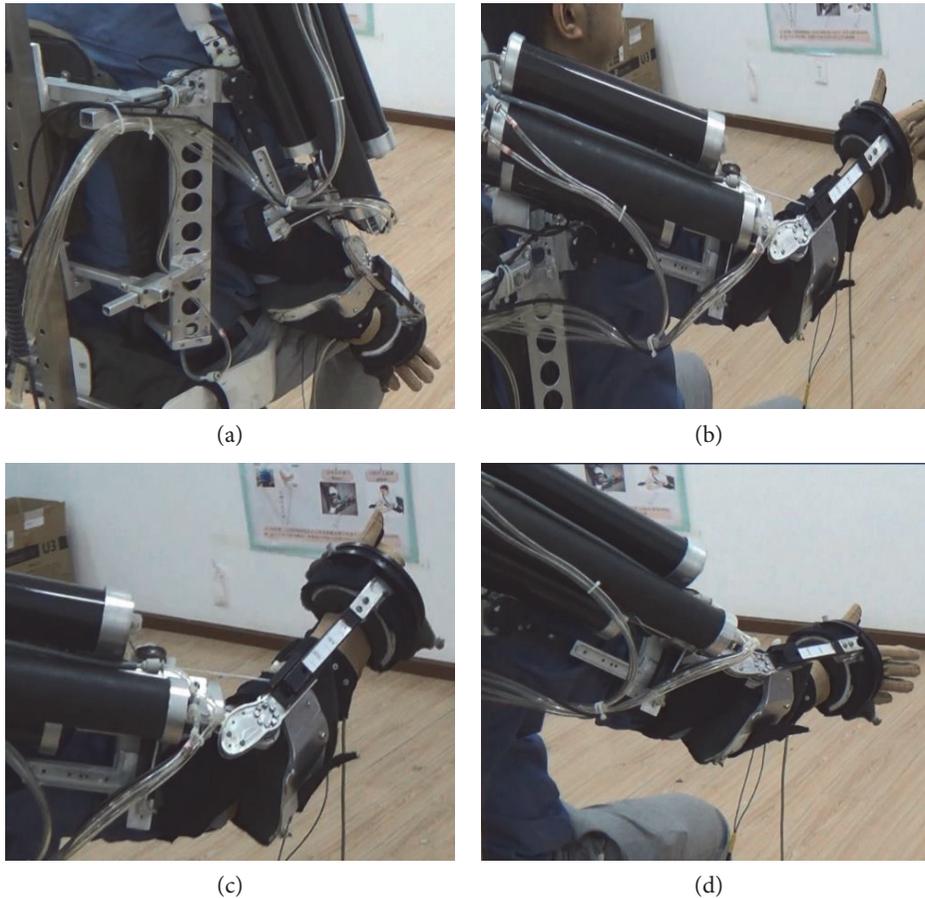


FIGURE 13: (a) Stretch and grip preparation stages, (b) stretching (ascending process), accompanied by hand release, (c) to achieve the intended goal, hand grip, and (d) stretch (descent process).

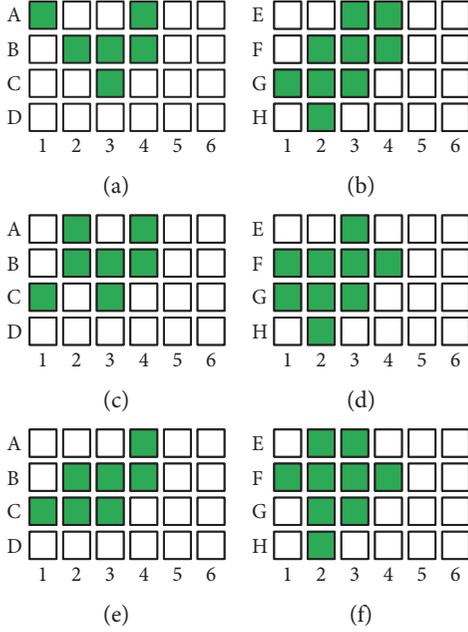


FIGURE 14: (a) Hand grasping under FES array electrode for subject 1, (b) hand releasing under FES array electrode for subject 1, (c) hand grasping under FES array electrode for subject 2, (d) hand releasing under FES array electrode for subject 2, (e) hand grasping under FES array electrode for subject 3, and (f) hand releasing under FES array electrode for subject 3.

$$\begin{aligned}
 u_{k+1} &= u_k - \frac{\theta_d - g(u_k)}{g'(u_k)} \\
 L &= \left( \frac{\partial(\theta_d - g(u_k))}{\partial u_k} \right)^{-1} = -g'(u_k)^{-1} \\
 u_{k+1} &= u_k - g'(u_k)^{-1} e_k \\
 \Delta u_{k+1} &= u_{k+1} - u_k \\
 e_k &= -g'(u_k) \cdot u_{k+1}.
 \end{aligned} \quad (19)$$

## 5. Experiments of Reach-to-Grasp

When the angle error  $\theta_d - \theta$  of elbow joint is more than zero, the error value is set to  $\theta_{FES}$ . When the angle error  $\theta_d - \theta$  of elbow joint is less than zero, the error value is set to  $\theta_p$  which shown in (20). The elbow joint motion is controlled by the proportional valve and FES by use of iterative learning controller as shown in Figure 12.

$$\begin{bmatrix} \theta_p \\ \theta_{FES} \end{bmatrix} = \begin{bmatrix} -1 \\ +1 \end{bmatrix} [\theta_d - \theta]. \quad (20)$$

$B_{\text{elbow}}$ ,  $K_{\text{elbow}}$ , and  $M_{\text{elbow}}$  are damping coefficient, stiffness coefficient, and inertia coefficient of elbow joint, respectively.  $G_{\text{elbow}}$  is the gravity moment of elbow joint, and  $\Delta\tau$  is the active muscular torque and other bounded disturbances.  $\tau_p$  and  $\tau_{FES}$  are the torques the pneumatic muscle and FES produce in (21), (22), and (23), respectively.  $F_p$ ,  $K_p$ , and  $B_p$  are the contraction coefficient, stiffness coefficient, and

damping coefficient of pneumatic muscle, respectively.  $x_p$  is the pneumatic muscle contraction length, and  $r$  is the radius of elbow joint.  $B_p$  and  $B_{FES}$  are the torque coefficients of the pneumatic muscle and FES, respectively.  $F_{l,v}(\theta, \dot{\theta})$  is the effect of the elbow angle and angular velocity on the FES-induced torque shown in (24).  $M_{\text{elbow}}$ ,  $B_{\text{elbow}}$ , and  $K_{\text{elbow}}$  are 0.02, 0.03, and 0.01, respectively. In order to facilitate the calculation,  $\Delta\tau$  is set to zero and the linearization of  $G_{\text{elbow}}$  is equal to  $0.02\theta$ . Equation (25) is the expression combination of elbow joint drivers. The sampling time  $T_s$  is 1 millisecond.  $u_p$  and  $u_{FES}$  are the control inputs of pneumatic muscle and FES, respectively, both of which use P-type iterative learning control to update the input. In the process of iterative learning control, when the angle error is within 2 degrees, the iterative process is stopped.

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -K_{\text{elbow}} & -B_{\text{elbow}} \\ M_{\text{elbow}} & M_{\text{elbow}} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ -G_{\text{elbow}} + \Delta\tau \\ M_{\text{elbow}} \end{bmatrix} \quad (21)$$

$$+ \begin{bmatrix} 0 \\ B_i \\ M_{\text{elbow}} \end{bmatrix} u_i, \quad i = p, \text{ FES}$$

$$\tau_p = B_p u_p, \quad \tau_{FES} = B_{FES} u_{FES} \quad (22)$$

$$\tau_p = (F_p(u_p) - K_p(u_p)x_p - B_p(u_p)\dot{x}_p)r \quad (23)$$

$$\begin{aligned}
 \tau_{FES} &= (\beta_1 u_{FES} - 150^3 + \beta_2 + \beta_3 u_{FES} + \beta_4 u_{FES}^2 + \beta_5 u_{FES}^3) \\
 &\times \frac{q^{-1}(b_0 + b_1 q^{-1})}{1 + a_1 q^{-1} + a_2 q^{-2}} \times F_{l,v}(\theta, \dot{\theta})
 \end{aligned} \quad (24)$$

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.48 & -1.5 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ B_i u_i \\ 0.02 \end{bmatrix} \quad i = p, \text{ FES}. \quad (25)$$

This experiment uses the Simulink xPC target real-time control platform, communicating with the surface electrical stimulator Rehasim2 through RS232 serial port to achieve the real-time control. The sampling frequency of xPC target real-time system is 20 kHz, and a PCI-6229 NI acquisition card can output 4-channel DA and control 4-way electromagnetic proportional valves. Pressure signal, force sensor signal, and absolute angle sensors are feedback to the real-time system through the PCI-6229 AD acquisition card. Incremental angle sensors are feedback to the control system through NI PCI-6602. Shoulder joint and elbow joint target angles were  $\theta_d(t) = [60^\circ \ 45^\circ]$ . The task time was  $T = 10$  seconds, and RMS was less than 2 degrees.

After the approval of the ethics committee of Huazhong University of Science and Technology, three healthy subjects were recruited to the treadmill-based exoskeleton gait training experiments (subject 1, male, 32 years old; subject 2, female, 34 years old; and subject 3, male, 29 years old). Before the trainings began, three

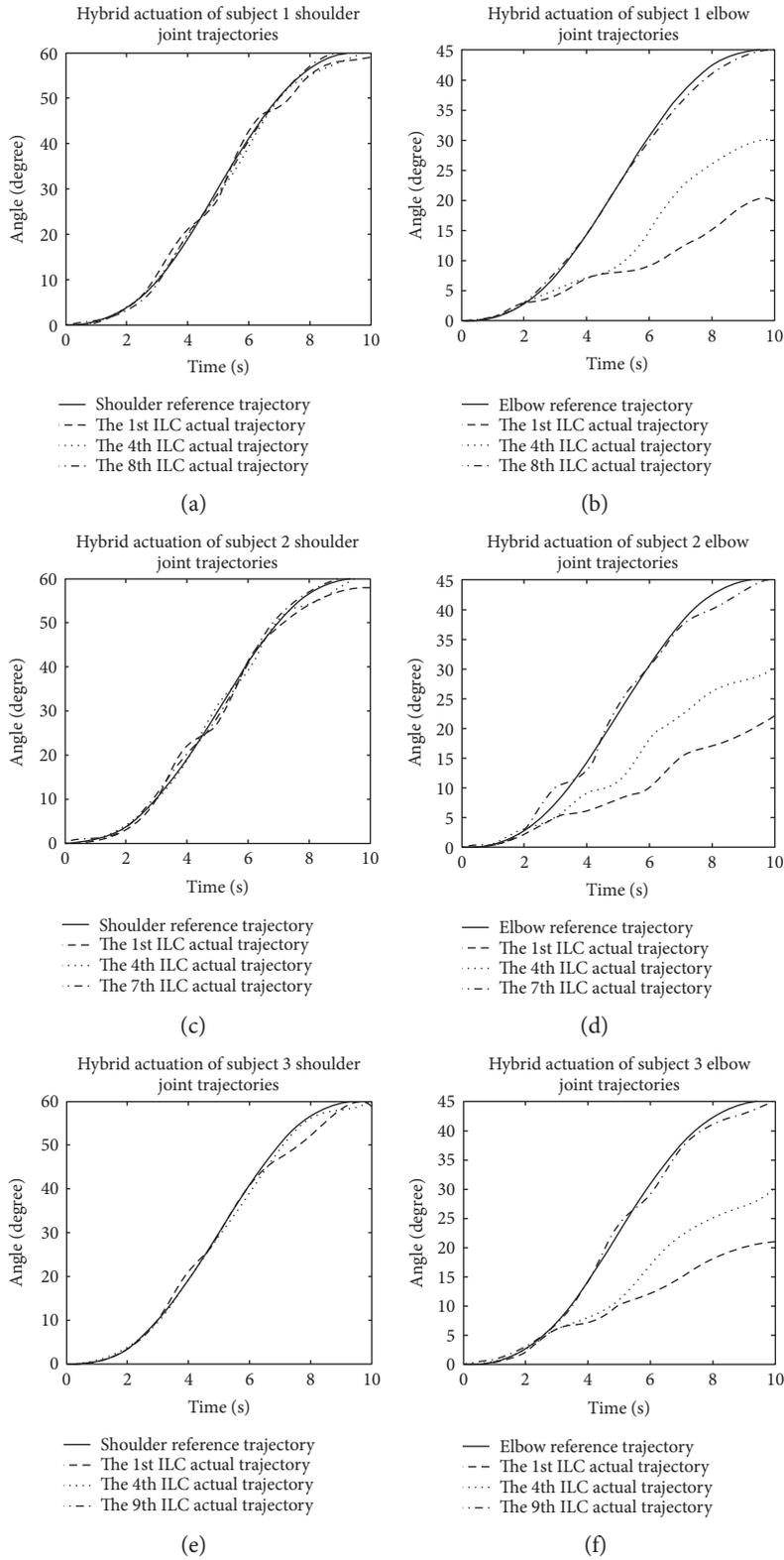


FIGURE 15: Joint trajectory tracking diagrams of grasping and releasing under FES, (a) shoulder joint for subject 1, (b) elbow joint for subject 1, (c) shoulder joint for subject 2, (d) elbow joint for subject 2, (e) shoulder joint for subject 3, and (f) elbow joint for subject 3.

subjects were informed of the experimental content and purpose shown in Figure 13. In the process of robot and FES-assisted trainings, the subjects were asked to relax as

much as possible. The task time is  $T$  for grasping, and active coordination intention parameter is  $\alpha$ . In order to reduce the experimental difficulty, the external/internal

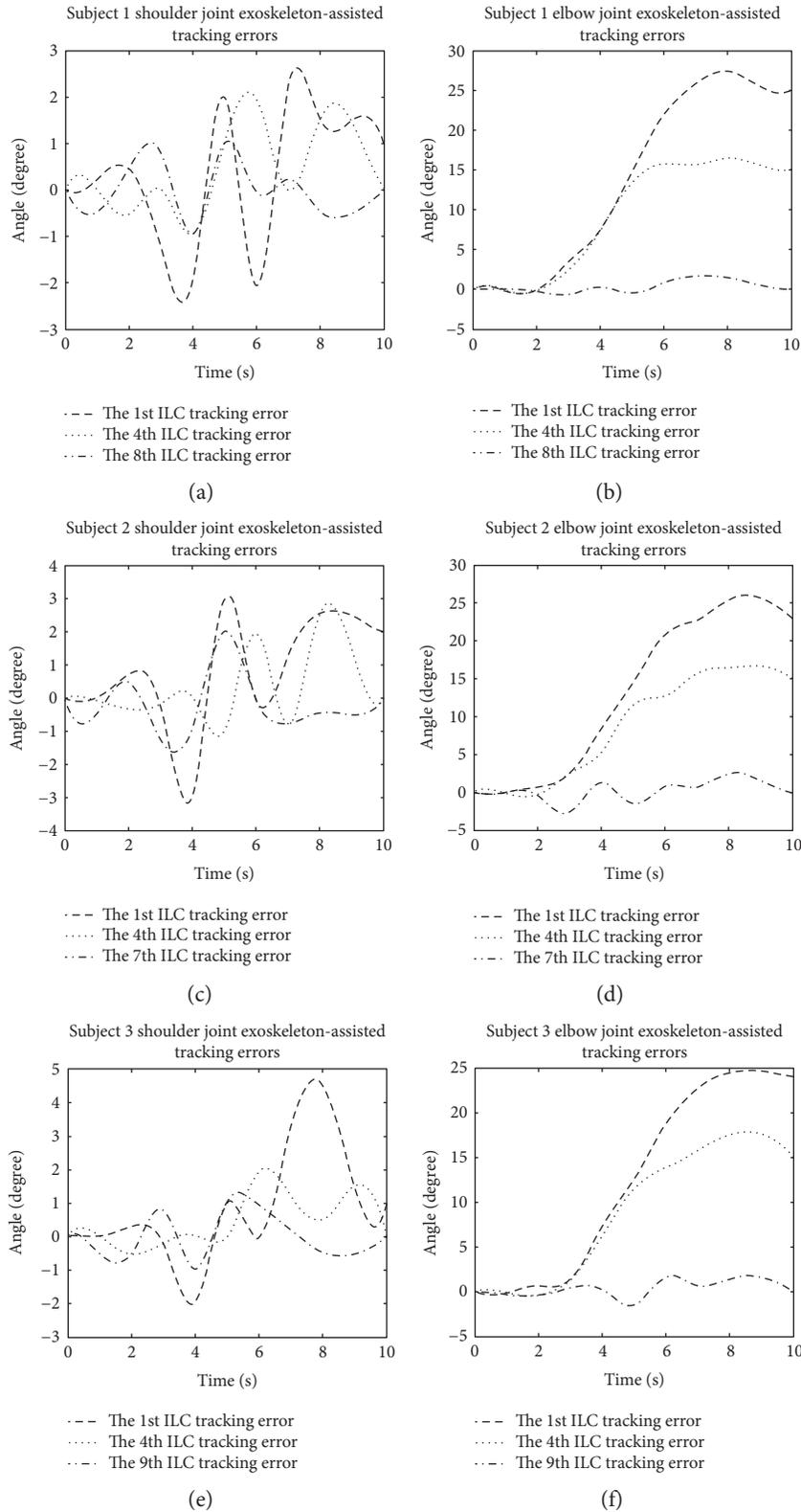


FIGURE 16: Joint trajectory tracking error diagrams of grasping and releasing under FES, (a) shoulder joint for subject 1, (b) elbow joint for subject 1, (c) shoulder joint for subject 2, (d) elbow joint for subject 2, (e) shoulder joint for subject 3, and (f) elbow joint for subject 3.

rotation was fixed to 30 degrees, and only the shoulder and elbow joints were executed. When  $\alpha$  is less than 0.3, it means hand releasing. When  $\alpha$  is more than 0.3, it means hand

grasping. Stimulus intensity is calculated according to (26). For the above three subjects,  $k_1$  and  $k_2$  were set as 14 and 43, 13 and 43, 17 and 47, respectively.

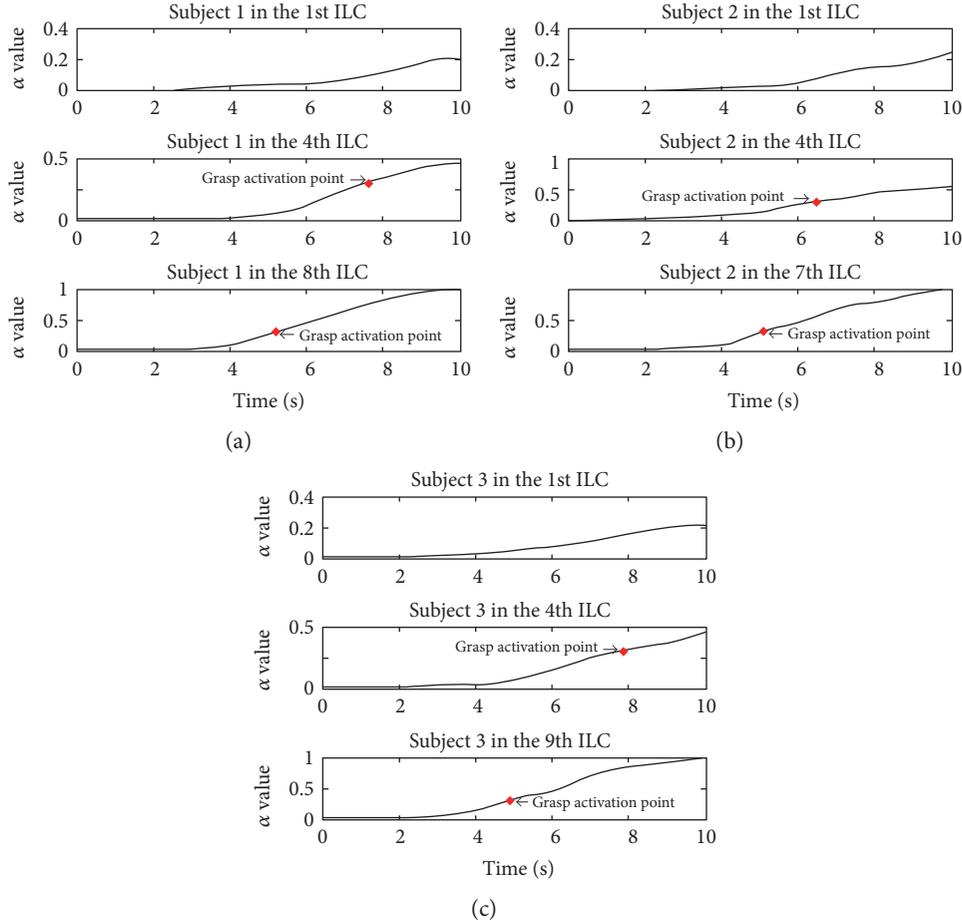


FIGURE 17: (a) The first subject extended grip coordination training, (b) the second subject extended grip coordination training, and (c) the third subject extended grip coordination training.

$$\begin{aligned} PA_{FPL} &= k_1 \times (\alpha - 0.3), & \alpha > 0.3 \\ PA_{EDC} &= k_2 \times \alpha, & \alpha < 0.3. \end{aligned} \quad (26)$$

## 6. Experimental Results

For the same desired hand posture, mapping targets of the three subjects' grasping and releasing under array electrical stimulation are different, shown in Figure 14. Grasping targets of subject 1 are a total of 6 targets with stimulation current amplitude of  $I = 10$  mA, A1, A4, B2, B3, B4, and C2, respectively. Releasing targets of subject 1 are a total of 9 targets with stimulation current amplitude of  $I = 13$  mA, E3, E4, F2, F3, F4, G1, G2, G3, and H2, respectively. Grasping targets of subject 2 are a total of 7 targets with stimulation current amplitude of  $I = 9$  mA, A2, A4, B2, B3, B4, C1, and C3, respectively. Releasing targets of subject 2 are a total of 9 targets with stimulation current amplitude of  $I = 13$  mA, E3, F1, F2, F3, F4, G1, G2, G3, and H2, respectively. Grasping targets of subject 3 are a total of 7 targets with stimulation current amplitude of  $I = 12$  mA, A4, B2, B3, B4, C1, C2, and C3, respectively. Releasing targets of subject 3 are a total of 9 targets with stimulation current amplitude of  $I = 14$  mA, E2, E3, F1,

F2, F3, F4, G2, G3, and H2, respectively. About hand grasping and releasing experiments, it can be concluded that each subject's stimulus threshold currents are not the same. These differences are caused by several aspects, including arm morphology, locations of array electrode placement, neuromuscular activation depth, and so on.

From Figure 15, it shows that the tracking errors of shoulder and elbow joint movement become smaller and smaller with the increasing of times. For the same error performance indicator, the number of iterations of three subject is 7 times, 8 times, and 9 times, respectively. This difference in the number of iterations may be caused by individual variations of the subjects.

Figure 16 shows the angle errors of trajectory tracking in the process of iterations. The tracking errors of shoulder joint have not changed obviously. The elbow joint errors greatly vary between the first time and the last time, and finally the error gradually decreases.

As is shown in Figure 17, for all the three subjects in the first iteration of the training process, intention-based grasping task is not activated, which indicates that it may be related to the adaptability of experiments. When it is in the fourth iteration of the training process, grasping intention is detected and grasping task of each subject is activated,

but the duration that each subject spent was different. The last iteration is compared with the fourth iteration, which indicates that grasping movement is activated in advance for each subject. This adaptation to intention-based rehabilitation training can help patients to actively participate in trainings and promote the motor function rehabilitation.

## 7. Conclusions and Future Work

In this research, it is presented that upper limb rehabilitation robot powered by PAMs cooperates with FES arrays to realize active reach-to-grasp trainings. FES is taken to activate paralyzed muscles and achieve two-way joint movement targeted for reaching trainings. Modeling of PMA and neuromuscular system under FES and ILC methods is used. The array electrode by virtue of matrix scanning method can solve the problem of the traditional self-adhesive electrode which is time consuming in searching optimum stimulation target. Intention-based FES actively activates the muscles to produce movement. The experimental results validated the effectiveness of this hybrid rehabilitation of robot and FES to realize active reach-to-grasp trainings.

In consideration of their own characteristics of stroke subjects, our proposed integrative strategy is using RUPERT exoskeleton with FES electrically evoked paralyzed ankle muscles to realize reach-to-grasp trainings, which is a promising approach to alleviate the size and mechanical complexity of the robot, thereby the cost of the rehabilitation robot. The future research is discussed for design principle of how to take advantage of each technique in developing a more functional effective hybrid FES and robot-assisted system for upper limb rehabilitation trainings.

## Conflicts of Interest

The authors declare no competing interests.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61233015 and Grant 61473130 and by the “Green Industry Technology Leading Program Project” of Hubei University of Technology (Grant no. ZZTS2017008).

## References

- [1] Y. Koumpouros, “A systematic review on existing measures for the subjective assessment of rehabilitation and assistive robot devices,” *Journal of Healthcare Engineering*, vol. 2016, Article ID 1048964, p. 10, 2016.
- [2] H. I. Krebs, B. T. Volpe, D. Williams et al., “Robot-aided neurorehabilitation: a robot for wrist rehabilitation,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 3, pp. 327–335, 2007.
- [3] R. Colombo, I. Sterpi, A. Mazzone, C. Delconte, G. Minuco, and F. Pisano, “Measuring changes of movement dynamics during robot-aided neurorehabilitation of stroke patients,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 1, pp. 75–85, 2010.
- [4] R. Riener, L. Lünenburger, I. C. Maier, G. Colombo, and V. Dietz, “Locomotor training in subjects with Sensori-motor deficits: an overview of the robotic gait Orthosis Lokomat,” *Journal of Healthcare Engineering*, vol. 1, no. 2, pp. 197–216, 2010.
- [5] S. Hesse, A. Waldner, J. Mehrholz, C. Tomelleri, M. Pohl, and C. Werner, “Combined transcranial direct current stimulation and robot-assisted arm training in subacute stroke patients an exploratory, randomized multicenter trial,” *Neurorehabilitation and Neural Repair*, vol. 25, no. 9, pp. 838–846, 2011.
- [6] J. Huang, X. Tu, and J. He, “Design and evaluation of the RUPERT wearable upper extremity exoskeleton robot for clinical and in-home therapies,” *IEEE Transactions on Systems, man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 926–935, 2016.
- [7] J. Huang, W. Huo, W. Xu, S. Mohammed, and Y. Amirat, “Control of upper-limb power-assist exoskeleton using a human-robot interface based on motion intention recognition,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1257–1270, 2015.
- [8] K. Wakita, J. Huang, P. Di, K. Sekiyama, and T. Fukuda, “Human-walking-intention-based motion control of an omnidirectional-type cane robot,” *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 285–296, 2013.
- [9] J. Huang, X. Yu, Y. Wang, and X. Xiao, “An integrated wireless wearable sensor system for posture recognition and indoor localization,” *Sensors*, vol. 16, no. 11, p. 1825, 2016.
- [10] F.-C. Wu, Y.-T. Lin, T.-S. Kuo, J.-J. Luh, and J.-S. Lai, “Clinical effects of combined bilateral arm training with functional electrical stimulation in patients with stroke,” in *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1–7, Zurich, 2011.
- [11] C. Freeman, A.-M. Hughes, J. Burridge, P. Chappell, P. Lewin, and E. Rogers, “A robotic workstation for stroke rehabilitation of the upper extremity using FES,” *Medical Engineering & Physics*, vol. 31, no. 3, pp. 364–373, 2009.
- [12] R.-Y. Wang, “Neuromodulation of effects of upper limb motor function and shoulder range of motion by functional electric stimulation (FES),” *Acta Neurochirurgica. Supplement*, vol. 97, part 1, pp. 381–385, 2007.
- [13] C. T. Freeman, Z. Cai, E. Rogers, and P. L. Lewin, “Iterative learning control for multiple point-to-point tracking application,” *IEEE Transactions on Control Systems Technology*, vol. 19, no. 3, pp. 590–600, 2011.
- [14] D. H. Owens, C. T. Freeman, and B. Chu, “Multivariable norm optimal iterative learning control with auxiliary optimisation,” *International Journal of Control*, vol. 86, no. 6, pp. 1026–1045, 2013.
- [15] C. T. Freeman, “Upper limb electrical stimulation using input-output linearization and iterative learning control,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1546–1554, 2015.
- [16] D. Gijbels, I. Lamers, L. Kerkhofs, G. Alders, E. Knippenberg, and P. Feys, “The Armeo spring as training tool to improve upper limb functionality in multiple sclerosis: a pilot study,” *Journal of Neuroengineering and Rehabilitation*, vol. 8, no. 1, p. 5, 2011.
- [17] T. G. Sugar, J. He, E. J. Koenen et al., “Design and control of RUPERT: a device for robotic upper extremity repetitive therapy,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 3, pp. 336–346, 2007.

- [18] S. Balasubramanian and J. He, "Adaptive control of a wearable exoskeleton for upper-extremity neurorehabilitation," *Applied Bionics and Biomechanics*, vol. 9, no. 1, pp. 99–115, 2012.
- [19] P. Heo, G. M. Gu, S.-j. Lee, K. Rhee, and J. Kim, "Current hand exoskeleton technologies for rehabilitation and assistive engineering," *International Journal of Precision Engineering and Manufacturing*, vol. 13, no. 5, pp. 807–824, 2012.
- [20] A. J. Westerveld, A. C. Schouten, P. H. Veltink, and H. van der Kooij, "Selectivity and resolution of surface electrical stimulation for grasp and release," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 1, pp. 94–101, 2012.
- [21] N. M. Malešević, L. Z. P. Maneski, V. Ilić et al., "A multi-pad electrode based functional electrical stimulation system for restoration of grasp," *Journal of Neuroengineering and Rehabilitation*, vol. 9, no. 1, p. 66, 2012.
- [22] A. J. Westerveld, B. J. Aalderink, W. Hagedoorn, M. Buijze, A. C. Schouten, and H. van der Kooij, "A damper driven robotic end-point manipulator for functional rehabilitation exercises after stroke," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 10, pp. 2646–2654, 2014.
- [23] F. Resquín, A. C. Gómez, J. Gonzalez-Vargas et al., "Hybrid robotic systems for upper limb rehabilitation after stroke: a review," *Medical Engineering & Physics*, vol. 38, no. 11, pp. 1279–1288, 2016.
- [24] J. He, E. Koeneman, R. Schultz et al., "RUPERT: a device for robotic upper extremity repetitive therapy," in *Proceedings of the 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6844–6847, Shanghai, China, September 1–4, 2005.
- [25] J. Huang, Z. Guan, T. Matsuno, T. Fukuda, and K. Sekiyama, "Sliding mode velocity control of mobile wheeled inverted pendulum systems," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 750–758, 2010.
- [26] D. Reynolds, D. Repperger, C. Phillips, and G. Bandry, "Modeling the dynamic characteristics of pneumatic muscle," *Annals of Biomedical Engineering*, vol. 31, no. 3, pp. 310–317, 2003.
- [27] W. K. Durfee and K. I. Palmer, "Estimation of force-activation, force-length, and force-velocity properties in isolated, electrically stimulated muscle," *IEEE Transactions on Biomedical Engineering*, vol. 41, no. 3, pp. 205–216, 1994.
- [28] F. Le, I. Markovskiy, C. T. Freeman, and E. Rogers, "Identification of electrically stimulated muscle models of stroke patients," *Control Engineering Practice*, vol. 18, no. 4, pp. 396–407, 2010.
- [29] C. Frigo, M. Ferrarin, W. Frasson, E. Pavan, and R. Thorsen, "EMG signals detection and processing for on-line control of functional electrical stimulation," *Journal of Electromyography and Kinesiology*, vol. 10, no. 5, pp. 351–360, 2000.
- [30] T. Lin, D. Owens, and J. Hätönen, "Newton method based iterative learning control for discrete non-linear systems," *International Journal of Control*, vol. 79, no. 10, pp. 1263–1276, 2006.
- [31] C. T. Freeman, "Constrained point-to-point iterative learning control with experimental verification," *Control Engineering Practice*, vol. 20, no. 5, pp. 489–498, 2012.

## Research Article

# An Open-Structure Treadmill Gait Trainer: From Research to Application

Jian Li,<sup>1,2</sup> Diansheng Chen,<sup>2</sup> and Yubo Fan<sup>1,3</sup>

<sup>1</sup>Beijing Key Laboratory of Rehabilitation Technical Aids for Old-Age and Disability and Key Laboratory of Rehabilitation Aids Technology and System of the Ministry of Civil Affairs and Engineering Research Center for Rehabilitation Aids of the Ministry of Civil Affairs, National Research Center for Rehabilitation Technical Aids, Beijing 100176, China

<sup>2</sup>Robotic Institute, Beihang University, Beijing 100191, China

<sup>3</sup>Key Laboratory for Biomechanics and Mechanobiology of Ministry of Education, School of Biological Science and Medical Engineering, Beihang University, Beijing 100191, China

Correspondence should be addressed to Yubo Fan; yubofan@buaa.edu.cn and Diansheng Chen; chends@163.com

Received 22 December 2016; Accepted 26 February 2017; Published 15 June 2017

Academic Editor: Yajing Shen

Copyright © 2017 Jian Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Lower limb rehabilitation robots are designed to enhance gait function in individuals with motor impairments. Although numerous rehabilitation robots have been developed, only few of these robots have been used in practical health care, particularly in China. The objective of this study is to construct a lower limb rehabilitation robot and bridge the gap between research and application. Open structure to facilitate practical application was created for the whole robot. Three typical movement patterns of a single leg were adopted in designing the exoskeletons, and force models for patient training were established and analyzed under three different conditions, respectively, and then a control system and security strategy were introduced. After establishing the robot, a preliminary experiment on the actual use of a prototype by patients was conducted to validate the functionality of the robot. The experiment showed that different patients and stages displayed different performances, and results on the trend variations across patients and across stages confirmed the validity of the robot and suggested that the design may lead to a system that could be successful in the treatment of patients with walking disorders in China. Furthermore, this study could provide a reference for a similar application design.

## 1. Introduction

Lower limb rehabilitation robots have been actively investigated for the past two decades [1, 2], with a number of studies showing the significance of these robots in the rehabilitation of patients with lower limb dysfunctions [3], such as stroke, central nervous system disorders, spinal cord injuries [4], and cerebrovascular diseases [5]. Since the last century, many scientists and research institutes in different countries have developed various rehabilitation robots [6, 7]. However, many of these rehabilitation robots were confined in research laboratories and were not used in practical health care, particularly in China. According to Díaz et al. [8], these robot systems can be classified into five types: (i) treadmill gait trainers, (ii) foot-plate-based gait trainers, (iii) overground

gait trainers, (iv) stationary gait trainers, and (v) ankle rehabilitation systems. The only commercialized lower limb rehabilitation robots were Lokomat [9], LokoHelp, ReoAmbulator, Gangtrainer GT [10], ReWalk [11, 12], and HAL [13]. Generally, the following questions had yet to be answered: What limits the application of rehabilitation robots? What is the gap between robots and patients? Previous studies always found the problems in technology, but progress in solving these problems had been minimal. Although technical realization is the basis for the robot using, it is not the only element for a truly viable product, in which availability and convenience of design are indispensable to the success of a product in practice.

Unquestionably, China has the biggest population and elderly population as well as the most number of patients

with lower limb dysfunctions among the countries in the world. Currently, only few individuals in the country can afford long-term treatments and expensive fees of rehabilitation robots which were almost imported from foreign countries [14]. Furthermore, the payment schemes in China further hinder the purchase of these robots [15, 16]. At present, a rehabilitation robot was not included in the medicare, and only disabled soldiers, praise persons, and work-injury persons can share some fees by the medicare. Although researchers could establish a technique for rehabilitation robots, application feasibility of these robots remains uncertain. Therefore, development of a low-cost and easy-to-use lower limb rehabilitation robot is imperative in China. Accordingly, Tsinghua University [17], the Harbin Institute of Technology, Shanghai Jiao Tong University, and Zhejiang University, among others [18], have tried to address this need and developed more than 20 models of rehabilitation robots. However, these robots also have not come into application because of many reasons [19]. There are also lots of problems existing on the application. For example, some robots used heavy blocks and pulleys for the design of a suspension device, and these components inevitably increased the scale of the robot; some robot's exoskeletons were installed with handrails and other components that would hinder the patients to wear or remove the exoskeletons [20, 21]; and some robot's adjusting devices were complicated, such as armrests and exoskeletons of Lokomat could be adjusted, respectively, including width and height [9], but the adjustments were time consuming in application; in addition, the considerably long preparation time for robots often led to an off-putting feeling to the patients [1]. Therefore, exoskeletons that are easy and quick to wear and remove and those that allow for comfortable and healthy rehabilitation conditions should be developed [2].

Accordingly, the objective of this study is to introduce and design an open-structure and applicable treadmill gait trainer with features such as a simple structure and control scheme, low cost, and a reliable security. The design was developed with the key aims of providing good application that is both simple and convenient. In the next sections, important mechanism design and control scheme of the robot are presented; this will be followed by a description of prototype testing and analyses of the outcomes.

## 2. Materials and Methods

In order to meet rehabilitation needs of people with lower limb dysfunction in China and provide good application for rehabilitation robot, an open-structure treadmill gait trainer with features of easy-to-use structure, high security, and cost-effectiveness was aimed to be created. The robot was schemed to solve practical problems for patients, such as difficulty on wearing exoskeleton and long preparation time.

**2.1. Device Description.** As shown in Figure 1, this open-structure concept was influenced by looking at problems of existing related robots, thinking about why related robots often use parallel rods (relatively closed structure) as armrests and why the exoskeletons were fixed, which would

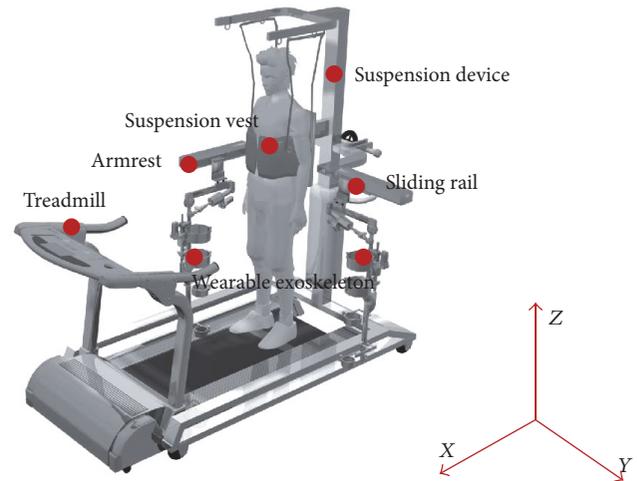


FIGURE 1: Diagram of an open-structure design for the robot and key components.

prevent patients from wearing or removing exoskeletons and getting on and off treadmills. In this study, each armrest could open and rotate  $90^\circ$  to allow a patient to easily move his or her wheelchair up and down the treadmill and wear the suspension vest. Under the armrests, there were two sliding rails. The exoskeleton can slide back and forth; as a result, it was easy to wear. The core structure of the exoskeleton was planned to manufacture with magnesium alloy AZ61, which is the lightest in the practical metals. Besides, the width and height of armrests and exoskeletons could be adjusted, respectively. It was considered that this principle of open structure could be applied to make a patient's training convenient and less-time preparation. The robot (treadmill gait trainer) consists of a wearable exoskeleton, suspension vest, suspension device, treadmill, and so on. Among them, a wearable exoskeleton, suspension device, and control system were the most important parts, which were described as follows:

**2.1.1. Wearable Exoskeleton.** Human locomotion is a rhythmic task: periodic, coordinated, and balanced, which is divided into stance phase and swing phase. And the lower limb usually displays complicated three-dimensional movement on sagittal plane, coronal plane, and horizontal plane in daily life. However, in rehabilitation training, gait on the coronal plane is usually disregarded [22]. In order to analyze hip and knee motion and provide real-time optical data for the exoskeleton design, Vicon motion capture system (Oxford Metrics Limited Company, UK) was used in this study. As shown in Figure 2(a), it was found that the movement of a single leg from stance to walk follows three typical patterns. In the initial state, the lower limb keeps in a straight line. Assuming the hip joint was the origin of the coordinate system, the angles of the hip and knee were  $90^\circ$  and  $180^\circ$ , respectively. In the forward-swing state, the angle of the hip becomes larger than  $90^\circ$ , and the knee angle becomes smaller than  $180^\circ$ . In the rear-swing state, both the hip and knee angles become smaller [23].

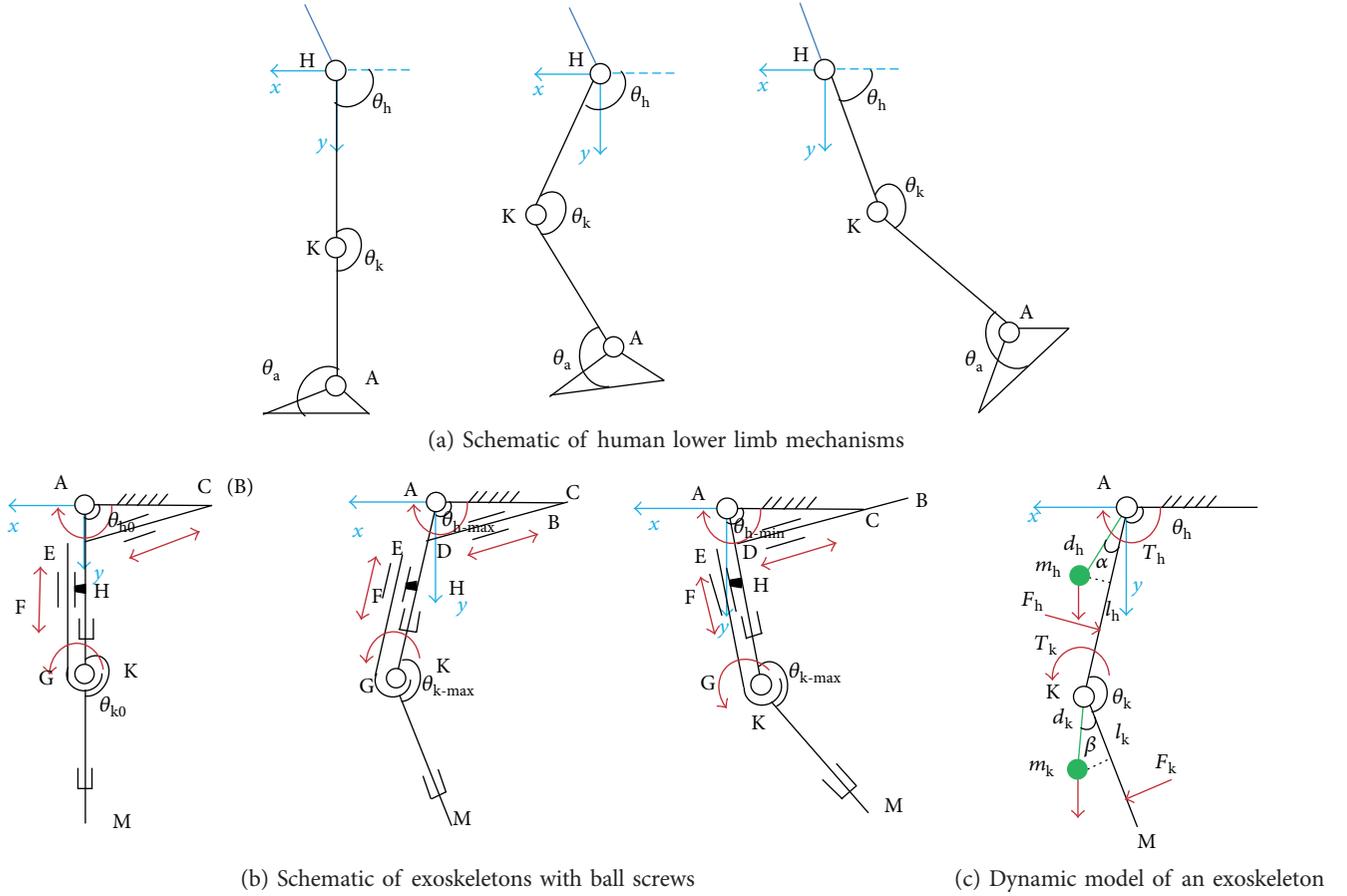


FIGURE 2: Analysis and design of exoskeleton: hip joint was the origin of the coordinate system, forward direction was the  $x$ -axis, and downward direction was the  $y$ -axis.

On the basis of the above analysis [24], a schematic of the proposed exoskeleton was conceived in Figure 2(b), in which “A” was the hip joint, “K” was the knee joint, and “M” was the ankle joint. Similarly, coordinates of “A”, “K”, and “M” were  $(0, 0, 0)$ ,  $(x_K, y_K, 0)$ , and  $(x_M, y_M, 0)$ , respectively, and could be expressed as

$$\begin{aligned} x_K &= l_{AK} \cos(180^\circ - \theta_h) \\ y_K &= l_{AK} \sin(180^\circ - \theta_h) \\ Z_K &= 0. \end{aligned} \quad (1)$$

$$\begin{aligned} x_M &= l_{AK} \cos(180^\circ - \theta_h) - l_{KM} \sin(270^\circ - \theta_k - \theta_h) \\ y_M &= l_{AK} \sin(180^\circ - \theta_h) + l_{KM} \cos(270^\circ - \theta_k - \theta_h) \\ Z_M &= 0. \end{aligned} \quad (2)$$

In the initial state,  $\theta_h$  was a right angle, whereas  $\theta_k$  was a straight angle; thus, the coordinates of “K” and “M” could be simplified to

$$\begin{aligned} x_K &= 0 \\ y_K &= l_{AK} \\ Z_K &= 0. \end{aligned} \quad (3)$$

$$\begin{aligned} x_M &= 0 \\ y_M &= l_{AK} + l_{KM} \\ Z_M &= 0. \end{aligned} \quad (4)$$

In the forward-swing state, coordinates of “K” and “M” could also be expressed by (1) and (2). Given that  $\theta_h$  was an obtuse angle, so

$$\cos(180^\circ - \theta_h) > 0, \quad (5)$$

and when

$$l_{AK} \cos(180^\circ - \theta_h) > l_{KM} \sin(270^\circ - \theta_k - \theta_h), \quad (6)$$

then

$$x_M > 0. \quad (7)$$

Both “K” and “M” were on the left side of the  $y$ -axis, but when

$$l_{AK} \cos(180^\circ - \theta_h) < l_{KM} \sin(270^\circ - \theta_k - \theta_h), \quad (8)$$

it would display

$$x_M < 0. \quad (9)$$

Perhaps, “K” was on the left side of the  $y$ -axis, but “M” was on the right side.

In the rear-swing state, given that  $\theta_h$  was an acute angle, so

$$\cos(180^\circ - \theta_h) < 0, \quad (10)$$

and

$$x_M < 0. \quad (11)$$

Therefore, specific structures of the exoskeleton were designed with two sides, including the right exoskeleton and the left exoskeleton, which were consist with Figure 2(b). Each side had two joints for simulating the extension and flexion of the human hip and knee, and ball screws were used to create the structure to ensure safety. The lengths of thigh rods and crus rods could be adjusted to fit different individual needs. Moreover, weight of exoskeleton was kept to a minimum.

Furthermore, high-output torque ratios of drive mechanisms are crucial to actual use. Therefore, a dynamic model (see Figure 2(c)) [25, 26] was initially developed based on Lagrangian, where coordinates of the mass centers for the hip ( $m_h$ ) and knee ( $m_k$ ) were as follows:

$$\begin{aligned} x_{m_h} &= d_h \cos(180^\circ - \theta_h - \alpha) \\ y_{m_h} &= d_h \sin(180^\circ - \theta_h - \alpha) \\ Z_{m_h} &= 0, \end{aligned} \quad (12)$$

$$\begin{aligned} x_{m_k} &= l_{AK} \cos(180^\circ - \theta_h) - d_k \sin(270^\circ - \theta_k - \theta_h - \beta) \\ y_{m_k} &= l_{AK} \sin(180^\circ - \theta_h) + d_k \cos(270^\circ - \theta_k - \theta_h - \beta) \\ z_{m_k} &= 0. \end{aligned} \quad (13)$$

Therefore, the velocities of  $m_h$  and  $m_k$  were

$$\begin{aligned} \dot{x}_{m_h} &= d_h \sin(180^\circ - \theta_h - \alpha) \dot{\theta}_h \\ \dot{y}_{m_h} &= -d_h \cos(180^\circ - \theta_h - \alpha) \dot{\theta}_h \\ \dot{z}_{m_h} &= 0 \end{aligned} \quad (14)$$

and

$$\begin{aligned} \dot{x}_{m_k} &= l_{AK} \sin(180^\circ - \theta_h) \dot{\theta}_h - d_k (\dot{\theta}_k + \dot{\theta}_h) \cos(270^\circ - \theta_k - \theta_h - \beta) \\ \dot{y}_{m_k} &= -l_{AK} \cos(180^\circ - \theta_h) \dot{\theta}_h + d_k (\dot{\theta}_k + \dot{\theta}_h) \sin(270^\circ - \theta_k - \theta_h - \beta) \\ \dot{z}_{m_k} &= 0, \end{aligned} \quad (15)$$

and the heights of  $m_h$  and  $m_k$  were

$$\begin{aligned} h_{m_k} &= l_{AK} \sin(180^\circ - \theta_h) + l_{KM} \cos(270^\circ - \theta_k - \theta_h) \\ &\quad - d_h \sin(180^\circ - \theta_h - \alpha) \\ h_{m_h} &= l_{KM} \cos(270^\circ - \theta_k - \theta_h) - d_k \cos(270^\circ - \theta_k - \theta_h - \beta). \end{aligned} \quad (16)$$

Given that the mass of the hip and knee consists of

exoskeleton mass, user mass, actuator mass, and other component mass, total mass could be calculated as [27]

$$\begin{aligned} m_h &= m_{\text{exo } h} + m_{\text{uerh } h} + m_{\text{act } h} + m_{\text{oth } h} \\ m_k &= m_{\text{exo } k} + m_{\text{uerh } k} + m_{\text{act } k} + m_{\text{oth } k}, \end{aligned} \quad (17)$$

where  $m_{\text{exo } h}$  was the mass of the hip exoskeleton,  $m_{\text{uerh } h}$  was the mass of the user's hip,  $m_{\text{act } h}$  was the mass of the hip actuators, and  $m_{\text{oth } h}$  was the mass of the other hip components, such as fastening tape. Similarly,  $m_{\text{exo } k}$ ,  $m_{\text{uerh } k}$ ,  $m_{\text{act } k}$ , and  $m_{\text{oth } k}$  were the mass of the knee exoskeleton, user's knee, knee actuators, and other knee components, respectively. Therefore, according to the Lagrangian formulations,

$$E_k = E_{k1} + E_{k2} = \frac{1}{2} m_h q_1^2 + \frac{1}{2} m_k q_2^2 \quad (18)$$

and

$$\begin{aligned} q_1^2 &= \dot{x}_{m_h}^2 + \dot{y}_{m_h}^2 \\ q_2^2 &= \dot{x}_{m_k}^2 + \dot{y}_{m_k}^2. \end{aligned} \quad (19)$$

Meanwhile,

$$E_p = E_{p1} + E_{p2} = m_h g h_{m_h} + m_k g h_{m_k} \quad (20)$$

and

$$L = E_k - E_p. \quad (21)$$

The hip joint torque and knee joint torque were

$$\begin{aligned} T_h = f_h &= \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} \\ T_k = f_k &= \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2}. \end{aligned} \quad (22)$$

Taking into account the actual situation, an additional coefficient was considered to ensure the reliability of the design in order to avoid unexpectedly large loads. These considerations were expressed in (16).

$$\begin{aligned} T_{h\text{-max}} &= c_h T_h \\ T_{k\text{-max}} &= c_k T_k, \end{aligned} \quad (23)$$

where  $c_h$  and  $c_k$  were the coefficients for hip joint and knee joint and  $c_h > 1$  and  $c_k > 1$ . Moreover, motor torque for the hip and knee also could be calculated by [28]

$$\begin{aligned} T_{h\text{-mor}} &= \frac{T_{h\text{-max}} \Delta L_{h\text{-screw}}}{2\pi r_{h\text{-bet}} d_h} \\ T_{k\text{-mor}} &= \frac{T_{k\text{-max}} \Delta L_{k\text{-screw}}}{2\pi r_{k\text{-bet}} d_k}, \end{aligned} \quad (24)$$

where  $T_{h\text{-mor}}$  and  $T_{k\text{-mor}}$  were the motor torques for the hip joint and knee joint, respectively;  $r_{h\text{-bet}}$  and  $r_{k\text{-bet}}$  were the ratios between the diameters of belt wheels, respectively, connected to the ball screws and the DC motors; and  $\Delta L_{h\text{-screw}}$  and  $\Delta L_{k\text{-screw}}$  were the leads of the ball screws for the hip joint and knee joint, respectively. Thus, hip and knee motors could

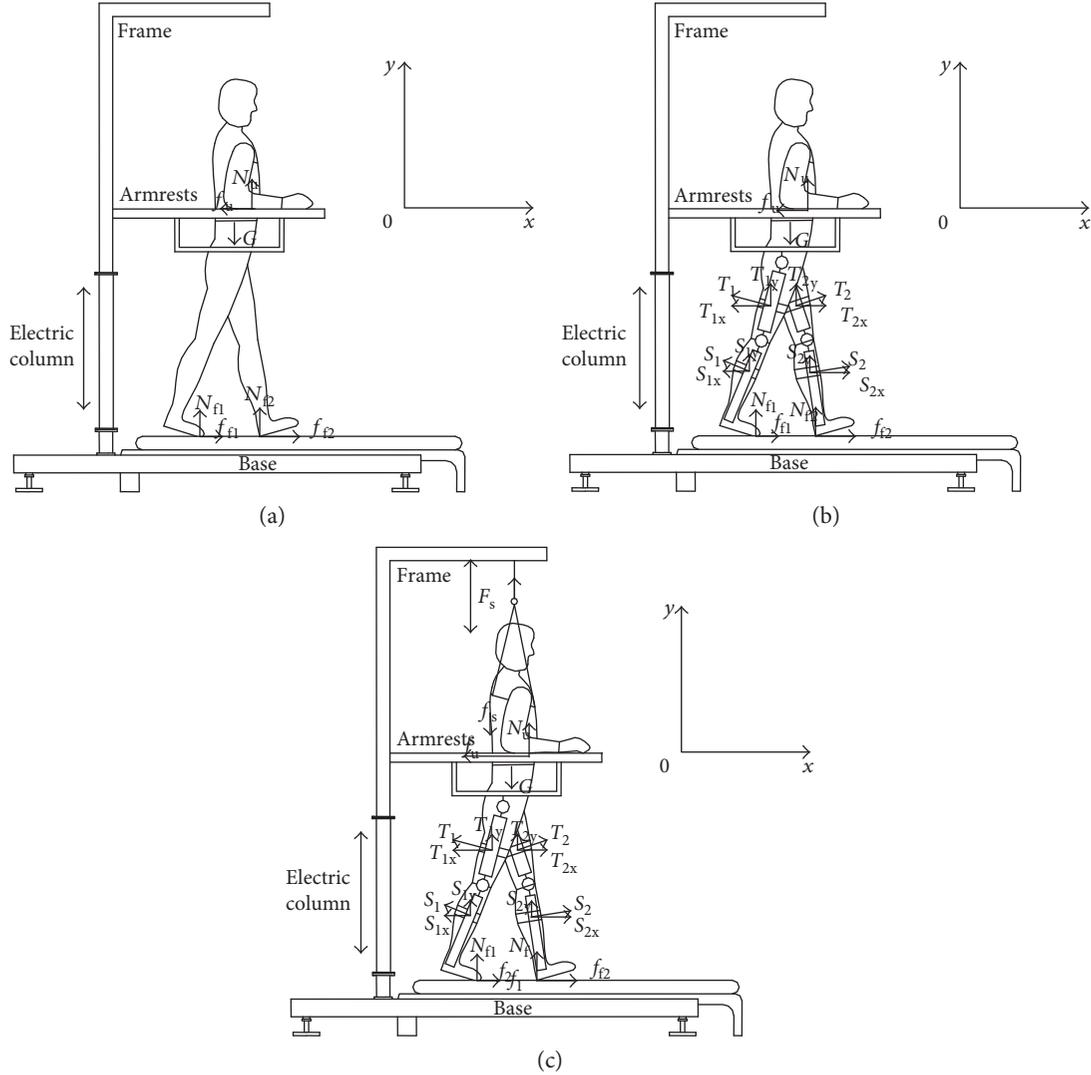


FIGURE 3: Force model for patient training. (a) Walking without exoskeletons and suspension device. (b) Walking with exoskeletons but without suspension device. (c) Walking with both exoskeletons and suspension device [32].

be selected based on the values of  $T_{h-max}$  and  $T_{k-max}$ . In addition, the output torque of the joint was determined by the force along the ball screws and the distance between the center of the joints' rotation and ball screws. Therefore, when transmission loss and mechanism mass were not considered, ratios of output torque for the hip joint and the knee joint could be formulated as follows [29]:

$$\begin{aligned} r_h &= \frac{T_{h-max}}{T_{h-mor}} = \frac{2\pi r_{h-bet} d_h}{\Delta L_{h-screw}} \\ r_k &= \frac{T_{k-max}}{T_{k-mor}} = \frac{2\pi r_{k-bet} d_k}{\Delta L_{k-screw}} \end{aligned} \quad (25)$$

Here,  $r_h$  and  $r_k$  denoted the output torque ratios for the hip and knee joint, respectively, and  $d_h$  and  $d_k$  were the distances between the joints and ball screws, respectively.

Both distances varied with the angles of the associated joint ( $\theta_h$  and  $\theta_k$ ). Furthermore, (25) suggested that a higher output torque ratio could be obtained for the DC motor by increasing  $r_{k-bet}$ ,  $r_{h-bet}$ ,  $d_k$ , and  $d_h$  [30]. All of the above detailed adjustments were beneficial and important for the application design.

**2.1.2. Suspension Device.** Generally, there are three situations that a patient normally encounters with treadmill-gait training. As illustrated in Figure 3, the sagittal axis and vertical axis of the body were defined as the  $x$  direction and  $y$  direction, respectively. When  $\sum F_x = 0$  ( $F_x$  was the horizontal force),  $\sum F_y = 0$  ( $F_y$  was the vertical force), and  $\sum M = 0$  ( $M$  was the torque), the following equations were derived [31].

When the patient walked on the treadmill without exoskeletons and suspension device, the governing equations were

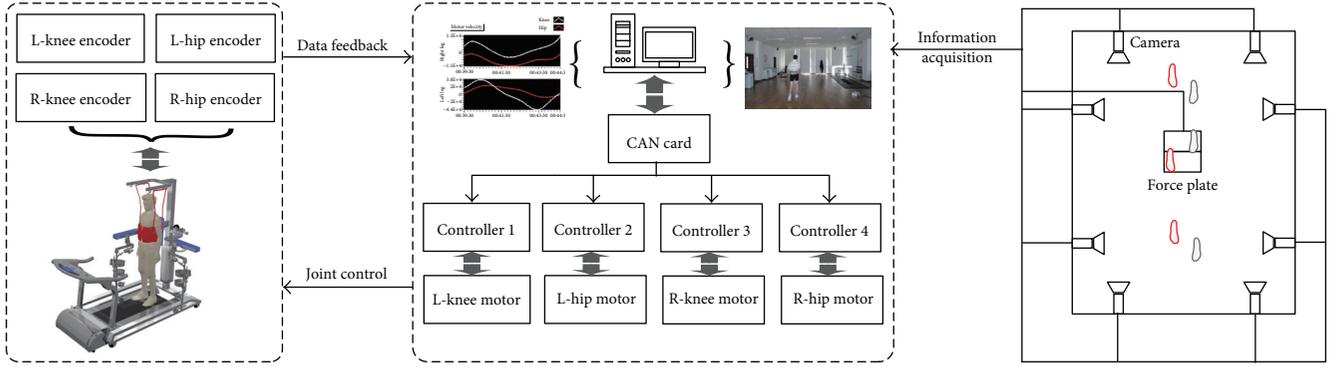


FIGURE 4: Architecture of control system and logic.

$$\begin{aligned}
 2f_u &= f_{f1} + f_{f2} \\
 2N_u + N_{f1} + N_{f2} &= G \\
 N_{f1}l_{Nf1} &= N_{f2}l_{Nf2} + N_u l_{Nu} + f_{f1}l_{f1} + f_{f2}l_{f2} + f_u l_{fu}.
 \end{aligned} \quad (26)$$

When the patient walked on the treadmill with exoskeletons but without suspension device, the governing equations were

$$\begin{aligned}
 T_{2x} + S_{2x} + f_{f1} + f_{f2} &= T_{1x} + S_{1x} + 2f_u \\
 2N_u + T_{1y} + T_{2y} + S_{1y} + S_{2y} + N_{f1} + N_{f2} &= G \\
 T_1 l_{T1} + S_1 l_{s1} + N_{f1} l_{Nf1} &= T_2 l_{T2} + S_2 l_{s2} + N_{f2} l_{Nf2} + N_u l_{Nu} \\
 &+ f_{f1} l_{f1} + f_{f2} l_{f2} + f_u l_{fu}.
 \end{aligned} \quad (27)$$

When the patient walked on the treadmill with both exoskeletons and suspension device, the governing equations were

$$\begin{aligned}
 T_{2x} + S_{2x} + f_{f1} + f_{f2} &= T_{1x} + S_{1x} + 2f_u \\
 F_s + 2N_u + T_{1y} + T_{2y} + S_{1y} + S_{2y} + N_{f1} + N_{f2} &= G + f_s \\
 T_1 l_{T1} + S_1 l_{s1} + N_{f1} l_{Nf1} &= T_2 l_{T2} + S_2 l_{s2} + N_{f2} l_{Nf2} + N_u l_{Nu} \\
 &+ f_{f1} l_{f1} + f_{f2} l_{f2} + f_u l_{fu}.
 \end{aligned} \quad (28)$$

In (26), (27), and (28),  $F_s$  was the lifting force provided by the suspension device to the body;  $f_s$  was the friction force provided by the suspension vest to the body;  $N_u$  and  $f_u$  were the support force and horizontal friction force on the arm of the patient;  $T_{1x}$ ,  $T_{2x}$ ,  $T_{1y}$ , and  $T_{2y}$  were the component forces exerted by the exoskeletons on the thigh in the  $x$  and  $y$  directions, respectively;  $S_{1x}$ ,  $S_{2x}$ ,  $S_{1y}$ , and  $S_{2y}$  were the component forces exerted by the exoskeletons on the crus in the  $x$  and  $y$  directions, respectively; and  $f_{f1}$ ,  $f_{f2}$ ,  $N_{f1}$ , and  $N_{f2}$  were the normal forces and static friction forces for the two feet.

As shown in Figure 3(a) and suggested by (26), when a patient walked on the treadmill without exoskeletons and a suspension device, entire body weight was concentrated on the legs and arms, and the balance of the body was entirely controlled by the hands and feet. If the legs lose support capacity, both arms of the patient would be subjected to discomfort feeling because of a larger support force and

balance force. When the patient walked on the treadmill with exoskeletons but not including a suspension device, as indicated in Figure 3(b) and (27), the exoskeleton could exert a certain amount of support force to the body, thereby reducing support force from the legs. However, support force from the arms and the balance force were not significantly reduced. When the patient walked on the treadmill both with the exoskeletons and suspension device, as indicated in Figure 3(c) and (28), drastic changes occur. The suspension device provided lifting force  $F_s$  to the body, and both forces on the legs and arms were reduced; as a result, the patient would feel comfortable. However,  $F_s$  could also introduce  $f_s$ . When  $F_s$  increases, the forces on the legs and arms were reduced, but  $f_s$  increased; consequently, the suspension vest would slide into the armpit. Given that numerous blood vessels, nerves, and lymph nodes are located in the armpit region, prolonged impingement in this region could cause the upper limb to feel numb. At the same time,  $T_{1x}$ ,  $T_{2x}$ ,  $S_{1x}$ ,  $S_{2x}$ ,  $N_u$ ,  $f_u$ ,  $T_{1y}$ ,  $T_{2y}$ ,  $S_{1y}$ , and  $S_{2y}$  also had direct influences on body comfort. Nevertheless, the forces on the arm were reduced; thus, the influences of  $N_u$  and  $f_u$  were negligible, and  $T_{1x}$ ,  $T_{2x}$ ,  $S_{1x}$ ,  $S_{2x}$ ,  $T_{1y}$ ,  $T_{2y}$ ,  $S_{1y}$ , and  $S_{2y}$  were relatively stable for one person with fastening tapes. If the exoskeletons were sufficiently light and the degree of tightness between the lower limbs and exoskeletons was appropriate, the patient would be comfortable during training because of good coordination [33].

**2.1.3. Security Strategy.** As shown in Figure 4, a closed-loop control scheme was designed [34], the standard information on the hip and knee were obtained with Vicon motion capture system and stored in a computer. The industrial computer served as the master controlling the four motors on both sides of the exoskeletons via a controller area network bus; subsequently, the encoders relayed the position, angle, velocity, and torque of the motor to the computer [35]. Besides, different human-machine parameters (50th percentile and 5th percentile) and training modes (time mode and step mode) were considered and designed to avoid joints exceeding the setting angles and incurring secondary damage.

Figure 5 shows that certain considerations were emphasized in the mechanical design and control design for practical application [36]. Firstly, ball screws were used at the hip and

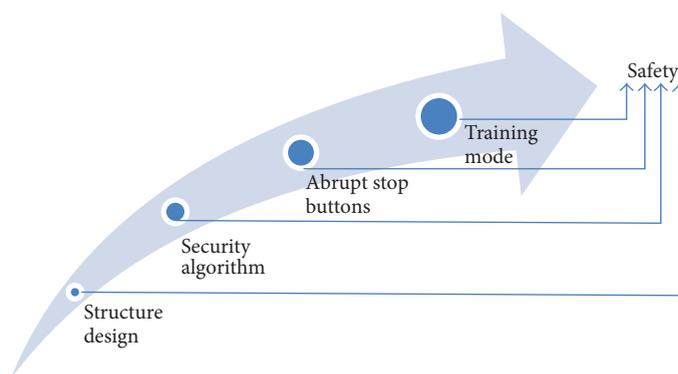


FIGURE 5: Security strategy for the robot.



FIGURE 6: Prototype of the robot.

knee joints, and the itinerary of the ball screws were accurately calculated to limit the rotation range and prevent reverse rotation. This safeguard comprised the last line of defense against mishaps, providing protection even when other security protocols failed to take effect. Secondly, as an intelligent protection, a security algorithm for the control system was established. When the encoders detected that the position, angle, velocity, and torque values of the joints exceeded the scope of protection, drive system would be interrupted, offering the fastest security control. Thirdly, an abrupt stop button was installed on the training platform, and rehabilitation physicians could immediately cut off the power supply. Finally, time mode was designed based on the time setting, and the robot could be stop by clicking the “STOP” icon on the computer screen. All the aforementioned four security protocols were independent. Thus, the robot could not possibly cause or incur accidental damages even when some of the protocols were invalid, unless all four security protocols were breached at the same time. However, this situation was extremely rare.

**2.2. Prototype Testing.** After finalizing the detailed design and analysis of a treadmill gait trainer, a prototype (Figure 6) was processed and assembled [37, 38], whose parameters were listed in Table 1.

TABLE 1: Parameters of robot.

Name	Parameter
Degree of freedom of exoskeletons	4 DOF
Adjustment range suspension height	0–500 mm
Adjustable speed of suspension height	5 mm/s
Thrust of lift column	3000 N
Height range of patient	160–185 cm
Adjustment range of armrest width	500–800 mm
Angle of armrest opening	0–90°
Treadmill width	800 mm
Width of running surface	450 mm
Treadmill height	290 mm
Treadmill speed	≤0.3 mph
Hip motor	Maxon RE50, 200 watts, 24 V, 5680 rpm, 10.8 A
Knee motor	Maxon RE40, 150 watts, 24 V, 6930 rpm, 5.77 A

In order to validate the functionality of the robot, a prototype experiment was conducted, which involved four patients (males aged at 38, 42, 30, and 46, resp.; mean age 39; mean weight 62 kg; mean height 163 cm). It was noted that only patients with a clear mind but with different degrees of walking disorders were selected for the experiment. All ethical issues with the experiment was in accordance with the local ethical regulations and approved by the institutional review boards, and all of the subjects consensually participated. After signing a protocol, they were trained twice every day for ten days completely free of charge under the care of two doctors. Each person was lifted until the feet would leave the treadmill, and every training session lasted 20 minutes.

### 3. Results and Discussion

After ten days of training, all of the patients found that the robot prototype was acceptable. Patients and doctors alike provided many recommendations, which were very useful for the future improvement. Moreover, sizeable amounts of data were obtained, such as joint position, speed, and torque,

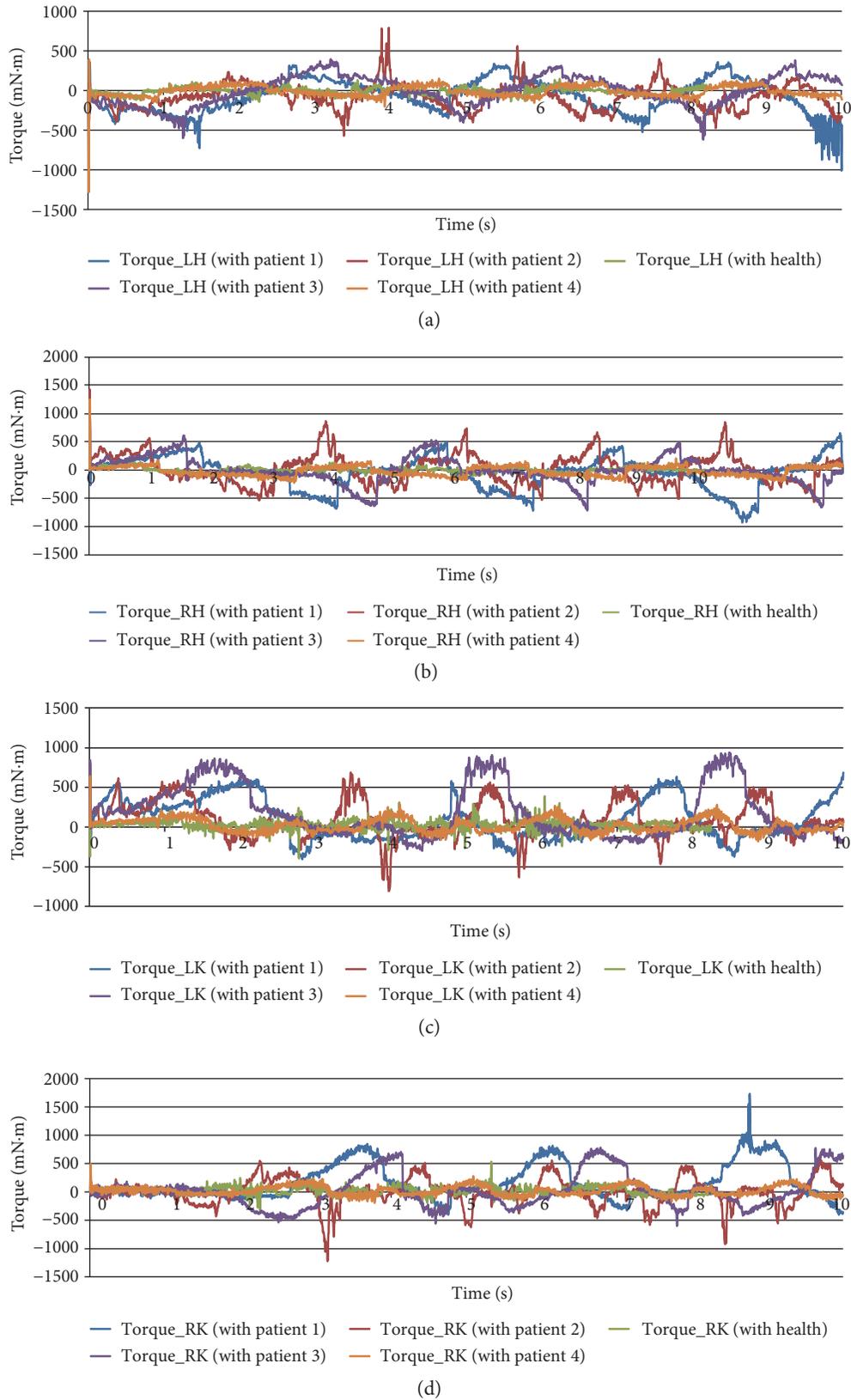


FIGURE 7: Joint torques on different patients. (a) Left hip. (b) Right hip. (c) Left knee. (d) Right knee.

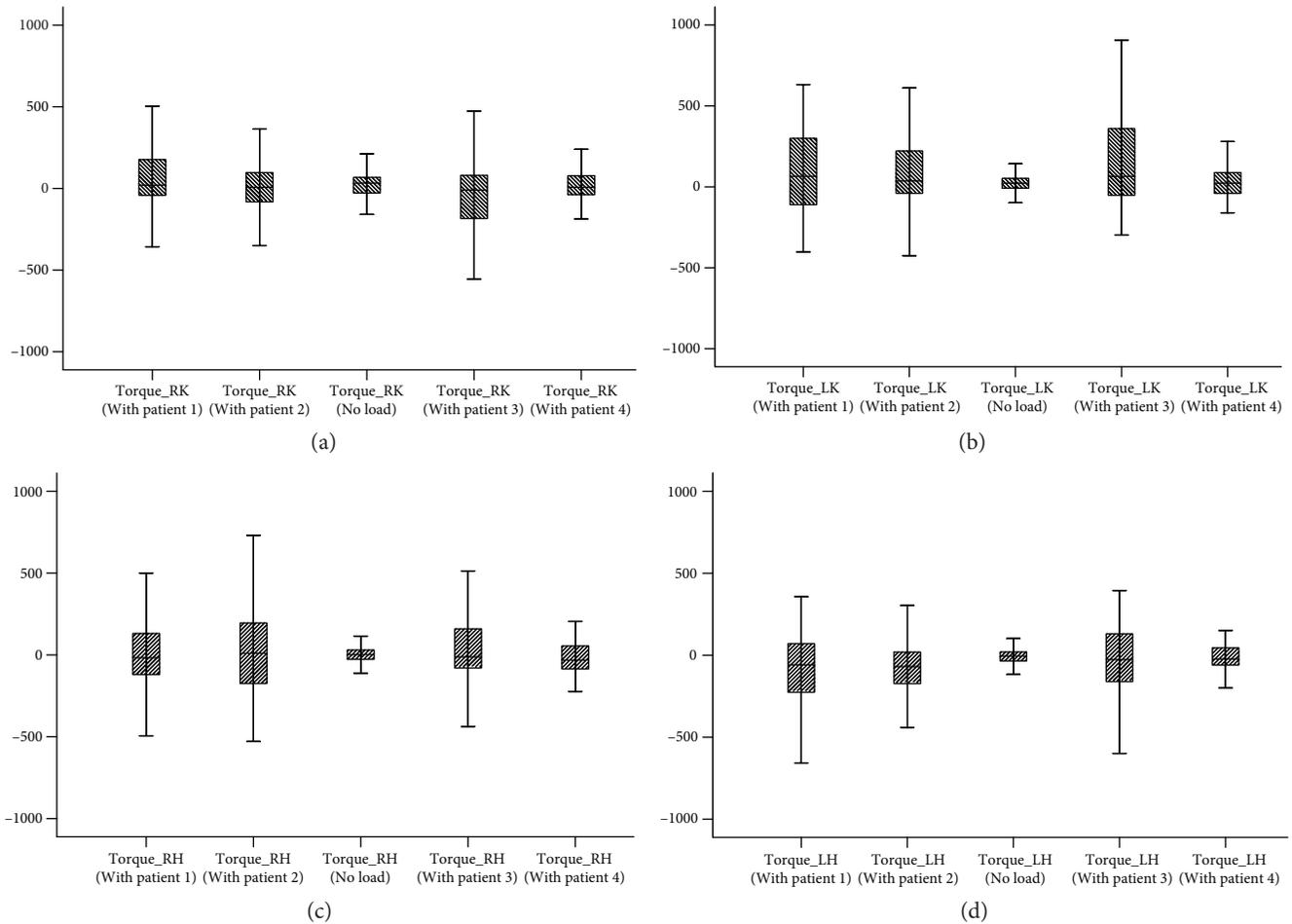


FIGURE 8: Statistical analysis of torque on different patients. (a) Left hip. (b) Right hip. (c) Left knee. (d) Right knee.

which could evaluate the availability directly, in particular joint torque.

**3.1. Analysis on Different Patients.** As shown in Figure 7, in the case of the same degree of weight loss, joint torques and the time of gait cycle varied across patients because of differences in body sizes, pathological conditions, and other factors, but variation trends were similar and consistent with each other. This phenomenon showed that the robot was sensitive to differences among patients. Moreover, torque requirements of joint motors showed a certain degree of diversity, and the maximum torque values for the hip joint (Figures 7(a) and 7(b)) of patient 1 (P1), patient 2 (P2), and patient 3 (P3) could be higher than that of patient 4 (P4) and demonstrated large fluctuations. By contrast with knee torques (Figures 7(c) and 7(d)), the maximum torque in the torque values of P4 were always minimal in the four patients, whose fluctuations values were relatively stable. However, all of the data on the left hip, right hip, left knee, and right knee of P4 showed remarkable bigger values of no load. Nevertheless, some unusual data on the patients are obtained. On the basis of engineering experience, it could suspect that these deviations are due to the interaction between human and robot. When the human was scared or nervous, the machine may demonstrate poor trackability; as

a result, outliers appear. This was in line with the actual situation.

In addition, statistical analysis of joint torques was conducted to further understand the joint torque requirements of different patients [39]. The pictures (Figure 8) illustrated the maximum, minimum, median, and 25th and 75th percentiles of the motor torque for the hip joints and that for the knee joints. For the left hip (Figure 8(a)), P4 demonstrated the most concentrated torque which had the smallest value, whereas P1 and P3 showed the most dispersed torque and the largest value. For the right hip (Figure 8(b)), P4 again presented the most concentrated torque and the smallest value, whereas P2 presented the most dispersed torque and the largest value. A comparison of the other torques are found in Figures 7(c) and 7(d), which show that P3 always had the most dispersed torque and the largest value. However, P4 was just the opposite. Based on the analysis of Figure 7 and the above results, it could be inferred that the illness of P4 would be lighter, whose lower limb may lose part of walking function but not completely, and P3 may had a serious problem. Finally, it was found that these results were consistent with the pathological conditions, in which P4 could walk slowly but unstably and short term and P4 was unable to walk completely by himself. This finding confirmed that the robot was credible in practice, and

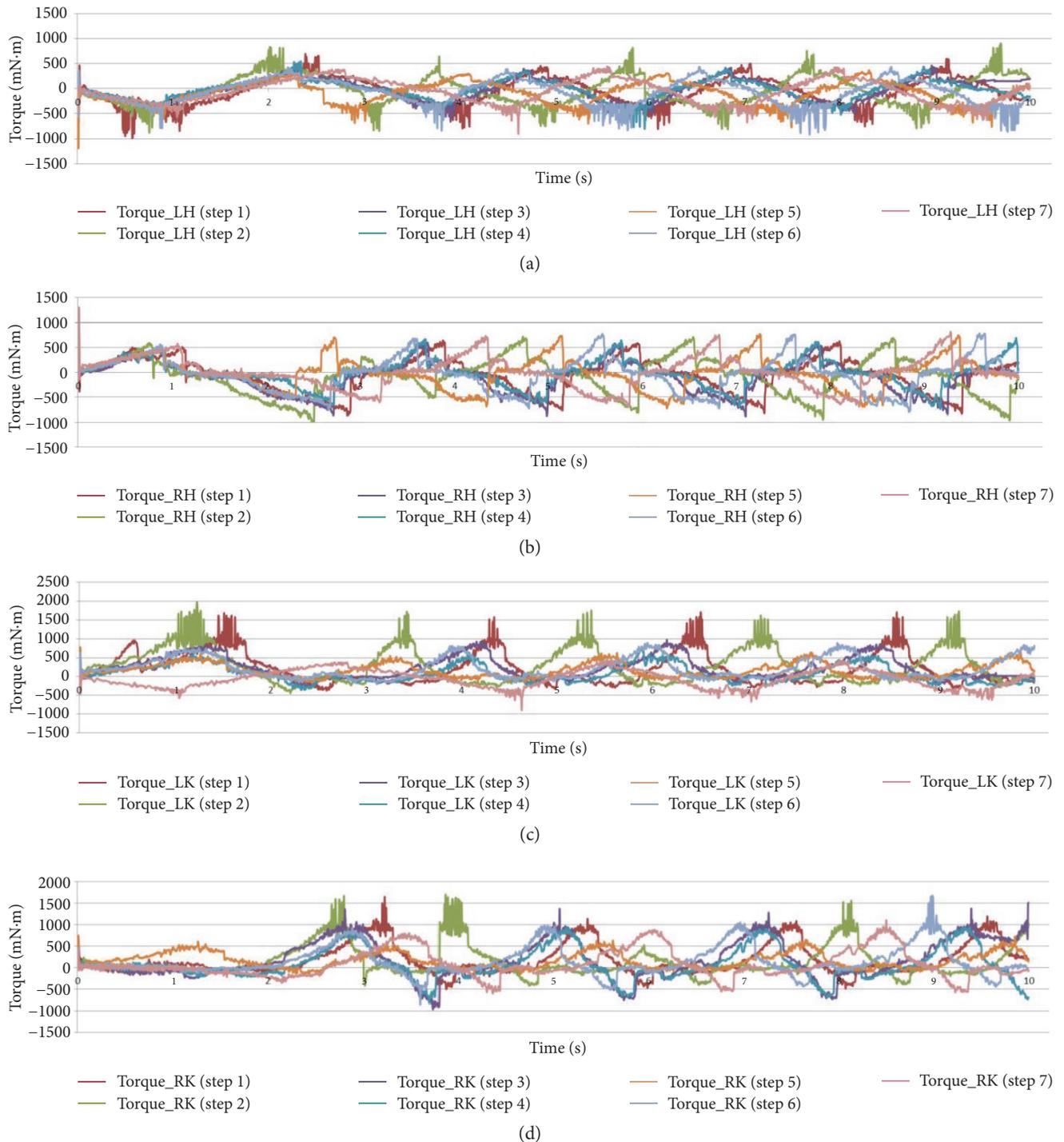


FIGURE 9: Experimental data of one patient at different stages. (a) Left hip. (b) Right hip. (c) Left knee. (d) Right knee.

performance of the patient with it was consistent with the actual conditions.

**3.2. Analysis on Different Stages.** Furthermore, different stages for one patient were showed in Figure 9 [40]. Torque variations for a patient at different stages were different but had very similar trends. However, in stages 1 and 2, numerous noise points existed, causing the overall curves to fluctuate repeatedly (except for Torque\_RH). The noise points were

understandable because the patients were involuntarily nervous and worried at the beginning of the training. Thus, the coordination between the patient and robot was very poor. However, once the patient was familiar with the rehabilitation robot, noise points disappear, as reflected in stages 5, 6, and 7. This finding confirmed that it would be necessary for patients to spend some time to familiarize with the machine before the formal training. In the meantime, the special torque\_RH displayed different performances from

other joints; the reasons could be ascribed to the open-chain structure of the human lower-limb system. Generally, the hip joint acts before the knee joint during human walking. If one side of the lower limb loses its capacity to walk, the hip joint on this side would be useless and unconscious. Thus, the leg and exoskeleton would demonstrate good conformity passively, and it could be surmised that the left leg of the patient was dysfunctional. Upon checking the pathology of the patient, the suspicion had been confirmed. In addition, peak values of both hip and knee torques showed declining trend gradually, and a comparison of the curves showed that the change curves became smoother from the first stage to the last stage. Good trackability brought by the pretraining may be the main reason, but validity of the robot also should be recognized. Therefore, the data on the patient could be used as an index to evaluate the motor ability of the patient, which could help therapists to determine the required diagnostic and therapeutic processes for a patient.

#### 4. Conclusions

In response, few lower limb rehabilitation robots had been used in practical health care and huge numbers of patients with lower limb dysfunction in China. An open-structure and applicable treadmill gait trainer (robot) was devised, and key components were analyzed and introduced. Then, a functional prototype was developed, and preliminary experiment on the actual use of the prototype by patients was conducted to validate the functionality of the robot. The experiment showed that different patients and stages demonstrated different performances, and results on the trend variations across patients and across stages suggested that the design may lead to a system that could be successful in the treatment of patients with walking disorder in China. Meanwhile, cost of the robot may be reduced because of localization and independent property rights. This study may provide a reference for similar application design. However, further improvement and therapeutic effects need to be evaluated further.

#### Conflicts of Interest

None of the authors has any conflict of interest.

#### Acknowledgments

This work was supported in part by the China Ministry of Civil Affairs Welfare Lottery Public Welfare Project, China National Key Research and Development Plan Project (nos. 2016YFB1101100, 2016YFB1101102, 2016YFB1101101, and 2016YFC1102203), NSFC (nos. 11421202 and 11572029), China Central-level Public Research Institutes Fundamental Special Research (1181500900101), and China National Science and Technology Pillar Program (2009BAI71B03). The authors give special thanks to Can-jun Yang (Zhejiang University) and Lin-hong Ji (Tsinghua University) and thank the help from Xiu-feng Zhang, Guo-xin Pan, and Run Ji.

#### References

- [1] W. Meng, Q. Liu, Z. Zhou, Q. Ai, B. Sheng, and S. S. Xie, "Recent development of mechanisms and control strategies for robot-assisted lower limb rehabilitation," *Mechatronics*, vol. 31, pp. 132–145, 2015.
- [2] Z. Qian and Z. Bi, "Recent development of rehabilitation robots," *Advances in Mechanical Engineering*, vol. 7, no. 2, pp. 1–10, 2015.
- [3] A. Nilsson, K. S. Vreede, V. Häglund, H. Kawamoto, Y. Sankai, and J. Borg, "Gait training early after stroke with a new exoskeleton – the hybrid assistive limb: a study of safety and feasibility frequently," *Journal of Neuro Engineering and Rehabilitation*, vol. 11, no. 92, pp. 1–10, 2014.
- [4] A. J. del-Ama, Á. Gil-Agudo, E. Bravo-Esteban, S. Pérez-Nombela, J. L. Pons, and J. C. Moreno, "Hybrid therapy of walking with kinesio overground robot for persons with incomplete spinal cord injury: a feasibility study," *Robotics and Autonomous Systems*, vol. 73, no. C, pp. 44–58, 2015.
- [5] K. Kubo, T. Miyoshi, A. Kanai, and K. Terashima, "Gait rehabilitation device in central nervous system disease: a review," *Journal of Robotics*, vol. 2011, 15 pages, 2011.
- [6] J. Pransky, "The Pransky interview: Russ Angold, co-founder and president of Ekso™ Labs," *Industrial Robot: An International Journal*, vol. 4, no. 4, pp. 329–334, 2014.
- [7] K. Wakita, J. Huang, P. Di, K. Sekiyama, and T. Fukuda, "Human Walking Intention Based Motion Control of Omnidirectional Type Cane Robot," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 285–296, 2013.
- [8] I. Díaz, J. J. Gil, and E. Sánchez, "Lower-limb robotic rehabilitation: literature review and challenges," *Journal of Robotics*, vol. 2011, 12 pages, 2011.
- [9] S. Jezernik, G. Colombo, T. Keller, H. Frueh, and M. Morari, "Robotic orthosis Lokomat: a rehabilitation and research tool," *Journal of International Neuropsychological Society*, vol. 6, no. 2, pp. 108–115, 2003.
- [10] T. Yan, M. Cempini, C. M. Oddo, and N. Vitiello, "Review of assistive strategies in powered lower-limb orthoses and exoskeletons," *Robotics and Autonomous Systems*, vol. 64, pp. 120–136, 2015.
- [11] S. H. Kim, S. K. Agrawal, and J. P. Scholz, "Robot assisted gait training with active leg exoskeleton (ALEX)," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 17, no. 1, pp. 2–8, 2009.
- [12] A. Esquenazi, M. Talaty, A. Packel, and M. Saulino, "The ReWalk powered exoskeleton to restore ambulatory function to individuals with thoracic-level motor-complete spinal cord injury," *American Journal of Physical Medicine & Rehabilitation*, vol. 91, no. 11, pp. 911–921, 2012.
- [13] H. Kawamoto, K. Kamibayashi, Y. Nakata et al., "Pilot study of locomotion improvement using hybrid assistive limb in chronic stroke patients," *BMC Neurology*, vol. 13, no. 141, pp. 1–8, 2013.
- [14] China Pharmaceutical Network, "Analysis on the current situation and future development of medical robot," 2016, February 2017, <http://www.robot-china.com/news/201601/25/30872.html>.
- [15] M. Feng-ling, Z. Hua, and T. Jing, "Analysis and counter measures of rehabilitation assistive device development in China," in *The 8th Beijing International Forum on Rehabilitation*, Beijing, 19–21 September, 2014.

- [16] T. Wei-feng, "Analysis on the Market of Domestic Rehabilitation Robot," 2016, February 2017, <http://www.wtoutiao.com/p/2b8Kv4g.html>.
- [17] J. Sui, Y. Liu, R. Yang, and L. Ji, "A multiposture locomotor training device with force-field control," *Advances in Mechanical Engineering*, vol. 6, pp. 1–10, 2014.
- [18] S. Hussain, "Robot assisted treadmill training: mechanisms and training strategies," *Medical Engineering and Physics*, vol. 33, no. 5, pp. 527–533, 2011.
- [19] S. Yu, C. Han, and I. Cho, "Design considerations of a lower limb exoskeleton system to assist walking and load-carrying of infantry soldiers," *Applied Bionics and Biomechanics*, vol. 11, no. 3, pp. 119–134, 2014.
- [20] X. Liu, X. Zang, Y. Zhu, Y. Liu, and J. Zhao, "System overview and walking dynamics of a passive dynamic walking robot with flat feet," *Advances in Mechanical Engineering*, vol. 7, no. 12, pp. 1–10, 2015.
- [21] S. Srivastava, P.-C. Kao, S. H. Kim et al., "Assist-as-needed robot-aided gait training improves walking function in individuals following stroke," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 6, pp. 956–963, 2015.
- [22] D. S. Chen, M. Ning, B. G. Zhang, and G. Yang, "An improvement to the reciprocating gait orthosis for aiding paraplegic patients in walking," *Science China Technological Sciences*, vol. 58, no. 4, pp. 727–737, 2015.
- [23] T. Zhiyong, X. Xiaodong, and P. Zhongcai, "Trajectory planning and mechanic's analysis of lower limb rehabilitation robot," *Bio-Medical Materials and Engineering*, vol. 26, supplement 1, pp. 347–355, 2015.
- [24] P. Stegall, K. Winfree, D. Zanotto, and S. K. Agrawal, "Rehabilitation exoskeleton design: exploring the effect of the anterior lunge degree of freedom," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 838–846, 2013.
- [25] D. Pan, F. Gao, and Y. Miao, "Dynamic research and analyses of a novel exoskeleton walking with humanoid gaits," *Proceedings of the Institution of Mechanical Engineers - Part C: Journal of Mechanical Engineering Science*, vol. 228, no. 9, pp. 1501–1511, 2014.
- [26] Z. Guo, H. Yu, and Y. H. Yin, "Developing a mobile lower limb robotic exoskeleton for gait rehabilitation," *Journal of Medical Devices*, vol. 8, no. 4, pp. 1–6, 2014.
- [27] C. Krishnan, R. Ranganathan, Y. Y. Dhaher, and W. Z. Rymer, "A pilot study on the feasibility of robot-aided leg motor training to facilitate active participation," *PLoS One*, vol. 8, no. 10, pp. 1–8, 2013.
- [28] W. Yang, X. F. Zhang, and C. J. Yang, "Design of a lower extremity exoskeleton based on 5-bar human machine model," *Journal of Zhejiang University (Engineering Science)*, vol. 48, no. 3, pp. 430–435, 2014.
- [29] D. Sanz-Merodio, M. C. Soto, J. C. Arevalo, and E. Garcia, "Generation and control of adaptive gaits in lower-limb exoskeletons for motion assistance," *Advanced Robotics*, vol. 28, no. 5, pp. 329–338, 2014.
- [30] W. Wang, Z.-G. Hou, L. Tong, F. Zhang, Y. Chen, and M. Tan, "A novel leg orthosis for lower limb rehabilitation robots of the sitting/lying type," *Mechanism and Machine Theory*, vol. 74, no. 6, pp. 337–353, 2014.
- [31] S. L. Nogueira, A. A. G. Siqueira, R. S. Inoue, and M. H. Terra, "Markov jump linear systems-based position estimation for lower limb exoskeletons," *Sensors*, vol. 14, no. 1, pp. 1835–1849, 2014.
- [32] Q. Lu, J. Liang, B. Qiao, and O. Ma, "A new active body weight support system capable of virtually offloading partial body mass," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 11–20, 2013.
- [33] S. Hussain, S. Q. Xie, P. K. Jamwal, and J. Parsons, "An intrinsically compliant robotic orthosis for treadmill training," *Medical Engineering and Physics*, vol. 34, no. 10, pp. 1443–1458, 2012.
- [34] R. Lu, Z. Li, C.-Y. Su, and A. Xue, "Development and learning control of a human limb with a rehabilitation exoskeleton," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 7, pp. 3776–3785, 2014.
- [35] F. M. Botsali, M. Kalyoncu, M. Tinkir, N. Yilmaz, and Y. Sahin, "Design and actuator selection of a lower extremity exoskeleton," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 623–632, 2012.
- [36] K. Knaepen, P. Beyl, S. Duerinck, F. Hagman, D. Lefeber, and R. Meeusen, "Human-robot interaction: kinematics and muscle activity inside a powered compliant knee exoskeleton," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 6, pp. 1128–1137, 2014.
- [37] A. T. Asbeck, S. M. M. De Rossi, I. Galiana, Y. Ding, and C. J. Walsh, "Stronger smarter: softer next-generation wearable robots," *IEEE Robotics & Automation Magazine*, vol. 21, no. 4, pp. 22–33, 2014.
- [38] R. Frost, J. Skidmore, M. Santello, and P. Artemiadis, "Sensorimotor control of gait: a novel approach for the study of the interplay of visual and proprioceptive feedback," *Frontiers in Human Neuroscience*, vol. 9, pp. 1–8, 2015.
- [39] H. Yan and C. Yang, "Design and validation of a lower limb exoskeleton employing the recumbent cycling modality for post-stroke rehabilitations," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 228, no. 18, pp. 3517–3525, 2014.
- [40] B. M. Fleerkotte, B. Koopman, J. H. Buurke, E. H. F. van Asseldonk, H. van der Kooij, and J. S. Rietman, "The effect of impedance-controlled robotic gait training on walking ability and quality in individuals with chronic incomplete spinal cord injury: an explorative study," *Journal of Neuro Engineering and Rehabilitation*, vol. 11, no. 26, pp. 1–15, 2014.