Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

GC-IGTG: A Rehabilitation Gait Trajectory Generation Algorithm for Lower Extremity Exoskeleton*

Yong He[1,2,3], Xinyu Wu[1,2,4**], Yue Ma[1,2,3], Wujing Cao[1,2], Nan Li[1,2], and Jinke Li[1,2,3]
1. Guangdong Provincial Key Laboratory of Robotics and Intelligent System, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
2. CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
3. Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen 518055, China
4. Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong, China
***Correspondence: xy.wu@siat.ac.cn*

***Abstract*— It is important to offer a natural and personalized rehabilitation gait trajectory, especially in the early stages of walking rehabilitation, for the patients with lower limb disability. Lower extremity exoskeleton has been proven to be efficient to provide highly repeatable and accurate rehabilitation exercise, but most existing exoskeletons' gait trajectories won't vary with the users. This paper proposes an algorithm, named as gait cell based individualized gait trajectory generation (GC-IGTG), for the purpose of offering a natural and personalized gait trajectory reference for the lower extremity exoskeleton based on the body parameters of the patients. The GC-IGTG is based on extreme learning machine and AutoEncoder, which makes it achieve fast training speed and suitable for small sample training conditions. The gait cell concept is proposed to improve the efficiency and safety of the algorithm. The experimental results indicate that the generated trajectories with GC-IGTG are almost identical to the original ones.**

***Index Terms*— Lower Extremity Exoskeleton, Gait Generation, Extreme Learning Machine, AutoEncoder, Gait Cell.**

I. INTRODUCTION

The World Population Prospects 2019: Highlights shows that all countries are experiencing population ageing and estimates that one in six people in the world will be over age 65 (16%) by 2050 [1]. The aging individuals are at greater risk of suffering from stroke or spinal cord injury (SCI) which leads to loss of walking abilities. Surveys of persons with stroke or SCI indicate that mobility concerns are the most prevalent, and that chief among mobility desires is the ability to walk and stand [2]. Traditional rehabilitation training that helps the patient to walk and stand relies on physical labor. With the development of rehabilitation engineering, rehabilitation robotics have been developed to assist in delivering this repetitive and intensive therapy, and thus alleviate the load on physiotherapists [3].

*This work is partially supported by the NSFC-Shenzhen Robotics Research Center Project (U1613219), Peacock for Technology Innovation KQJSCX20170731164301774 and Innovation Fund Project (COMAC-SFGS-2019-262)

The lower limb rehabilitation robotics can be divided into the body weight support training devices and the lower limb exoskeletons [4]. This work focus on the lower limb exoskeleton. In 2014, the ReWalk exoskeleton by Argo Medical Technologies Ltd, became the first lower limb exoskeleton that received FDA approval [5]. The hips and knees of ReWalk were driven by motors while the ankles were passive. The walking of ReWalk users was controlled by subtle trunk motion and changes in center of gravity. A predefined reference trajectory determined by a finite-state-machine structure was enforced to the ReWalk. The researchers from Vanderbilt University described a powered lower-limb orthosis that is intended to provide gait assistance to SCI individuals by providing assistive torques at both hip and knee joints [6]. The joint angle trajectories were preprogrammed for each motion based on normal biomechanical walking trajectories, obtained from a recording of the joint angle trajectories generated by a healthy subject while wearing the orthosis. Another exoskeleton named walking supporting exoskeleton (WSE) has been developed at Necmettin Erbakan University [7]. WSE contains two legs powered by electrical actuators at hip and knee joints. Preprogrammed motion control technique is selected to be used in control of WSE. HAL of Tsukuba University was a lightweight power assist device [8]. The actuators are DC motors at the knees and hips. EMG electrodes on human leg muscles and ground reaction force sensors were used to estimate human inner force and motion information. It is necessary for above lower limb exoskeletons to use supplementary walking aids, such as crutches or a walking frame, to balance and change direction. REX is the only commercialized self-stabilizing lower limb exoskeleton that requires no supplemental upper body support to balance [9]. Similar to REX, the Auto-LEE from SIAT also owns the self-stability without external support [10].

The assistive strategy adopted in lower limb exoskeleton for movement assistance decides the performance of the human machine interaction. Generally, the human-exoskeleton interaction is bidirectional: the power and feedbacks infor-

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

mation are provided by robot to the human, and the intended movement information is received from the user [11]. The motion intention recognition, gait pattern generation, and stable gait control are main technical points during human-exoskeleton interaction [12]–[14]. Several motion trajectories generation methods have been studied. Gait trajectory generation methods based on zero moment point and center of gravity (COG) are usually adopted in humanoid robot control systems [15]. However, the accurate COG parameters of the lower limb exoskeleton system and the patient are difficult to acquire. Model based trajectory generation method was studied by Rui Huang in [16]. They proposed a method which obtains the angle trajectories of knee joints from the hip joints to generate the reference trajectory of the lower limb exoskeleton. The relationship between the angle trajectories of the knee and hip joints is acquired through kinematic models of the lower limb exoskeleton. Predefined gait trajectory control is the most acceptable assistive strategy in lower limb exoskeletons. The desired joint trajectory is pre-obtained from a healthy person, or predicted from a gait analysis data atlas, and then replicated on an exoskeleton in a predefined gait trajectory control mechanism. ReWalk, Vanderbilt Lower-Limb Exoskeleton, WSE and HAL fall into this category. Lower limb exoskeletons using this method usually apply to people with partly or completely lower limb dysfunction. However, the gait patterns of health people and patient are very different. The individual difference is also significant considering gender, age and other human body parameters. Thus, it is necessary for the lower limb exoskeleton to automatically generate suit gait trajectory for a given patient according to the specific parameters.

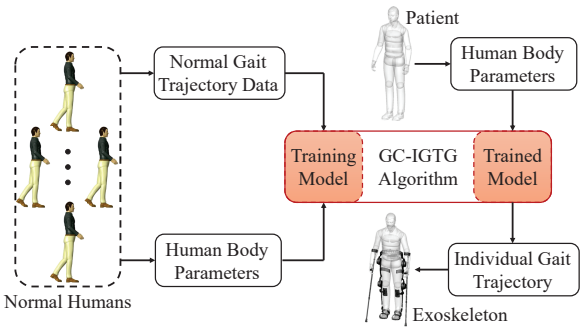

Fig. 1. The framework of individual gait trajectory generation based on GC-IGTG

In this work, the main purpose is to propose an algorithm named as gait cell based individualized gait trajectory generation (GC-IGTG), aiming at offering a natural gait trajectory reference for the lower extremity exoskeleton based on the body parameters of the exoskeleton user. Shown as Fig. 1, from a number (N_t) of normal humans, we can collect their normal gait trajectory data (NGTD) and human body parameters (HBP), then these NGTD-HBP data pairs can be used as the training set to train the GC-IGTG model. With the trained GC-IGTG model, an individual gait trajectory reference for exoskeleton can be acquired after entering the new HBP of a patient.

II. Data Acquisition and Preprocess

The training set for GC-IGTG algorithm consists of the normal gait trajectory data and human body parameters. The way to collect and preprocess these data will be described below.

A. Gait Trajectory Data Acquisition

The lower extremity exoskeleton we developed has six main degrees of freedom, including the hip flexion/extension, knee flexion/extension and ankle flexion/extension. Among them only the hip and knee joints are active, which are driven by servo motors [13]. Thus the gait trajectory data we need here are the joint angular trajectories of the right hip (RH), right knee (RK), left hip (LH) and left knee (LK) in the sagittal plane.

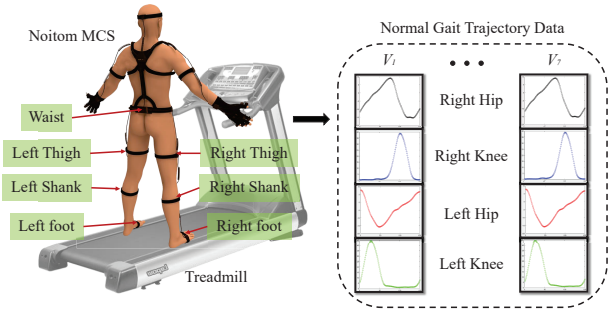

Fig. 2. The introduction of acquiring normal gait trajectory data

In order to acquire the four joint angle trajectories, the Noitom Motion Capture System (MCS) is employed. As Fig. 2 shows, seven inertial measurement units (IMU) are placed on the waist, left thigh, left shank, left foot, right thigh, right shank and right foot of the person being tested, which can record the joint angles in real time, and the sampling frequency of the system is set as 125 Hz. Besides that, the treadmill is used to collect the gait trajectory data at different walking speed, and seven walking speed ($V_1 - V_7$) are chosen, including 1.5 km/h, 2 km/h, 2.5 km/h, 3 km/h, 3.5 km/h, 4 km/h and 4.5 km/h.

B. Gait Trajectory Data Preprocess

The gait trajectory data acquired by Noitom MCS can not be used as the training set directly, for each of them contains many gait cycles, i.e. there is too much data. It is not practical to train an algorithm model to predict an individual gait trajectory like that. Human walking along a fixed direction on flat terrain is a repetitive periodic motion, and the gait trajectory can be approximately reproduced by copying an

one-cycle trajectory repeatedly [17]. Thus the gait trajectory data can be divided into many one-cycle trajectories, and the maximum angle of the left hip joint can be regarded as the separation mark of each cycle [12]. After the trajectory division process, we can select a fixed number (N_c) of stable one-cycle trajectories to form the training set. The one-cycle gait trajectory ($\mathbf{Tr}_j^k$) of each tester at every walking speed (V_i) can be described as Equation (1).

$$\mathbf{Tr}_j^k = \begin{bmatrix} \mathbf{t_r}_{-RH} \\ \mathbf{t_r}_{-RK} \\ \mathbf{t_r}_{-LH} \\ \mathbf{t_r}_{-LK} \end{bmatrix} = \begin{bmatrix} q_{RH1} & \cdots & q_{RHL_j} \\ q_{RK1} & \cdots & q_{RKL_j} \\ q_{LH1} & \cdots & q_{LHL_j} \\ q_{LK1} & \cdots & q_{LKL_j} \end{bmatrix}^{\mathrm{T}} \tag{1}$$

Here, L_j is the length of each one-cycle joint trajectory, and $j = 1, 2 \cdots N_c$. Then the one-cycle gait length (OCL) of a tester at walking speed V_i can be defined as:

$$OCL = \sum_{j=1}^{N_c} \frac{L_j}{N_c} \tag{2}$$

However, *OCL* varies from person to person at different walking speed. In order to make it convenient to extract the gait features, the one-cycle gait trajectory should be resampled to a fixed length L_0:

$$\mathbf{Tr}_j^k \Rightarrow \mathbf{Tr}_{L_0}^k = \begin{bmatrix} q_{RH1} & \cdots & q_{RHL_0} \\ q_{RK1} & \cdots & q_{RKL_0} \\ q_{LH1} & \cdots & q_{LHL_0} \\ q_{LK1} & \cdots & q_{LKL_0} \end{bmatrix}^{\mathrm{T}} \tag{3}$$

C. Gait-Related Human Body Parameters

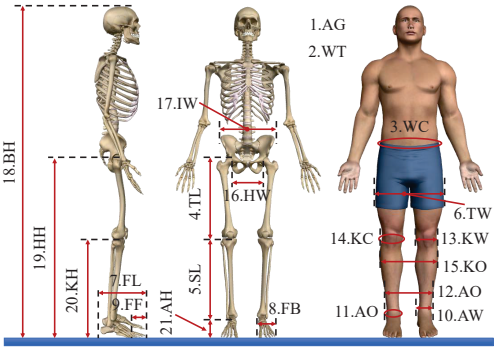

Fig. 3. The gait-related human body parameters

The gait trajectory of a person is related to many human body parameters, and there is little evidence to indicate which parameters affect the gait most. In this study, we select 21 body parameters according to [12]. As Fig. 3 shows, the gait-related human body parameters are: age (AG), weight (WT), waist circumference (WC), thigh length (TL), shank length (SL), maximum thigh width (TW), foot length (FL), foot breadth (FB), forefoot length (FF), ankle width (AW), ankle circumference (AC), bi-ankle outer width (AO), knee width (KW), knee circumference (KC), bi-knee outer width (HW), bi-hip width (IW), bi-iliac width (BH), body height (BH), hip height (HH), knee height (KH) and ankle height (AH).

With the gait trajectory and human body parameters ($\mathbf{BoP}$) described above, the training set data can be defined as below:

$$\mathbf{TS} = \{\mathbf{BoP},\ \mathbf{T}_{\mathbf{r}L_0},\ \mathbf{OCL}\}_{V_i}, V_i\ \varepsilon\ (V_1 \cdots V_7) \tag{4}$$

Here, $\mathbf{BoP} = [\mathbf{BoP}_1, \cdots, \mathbf{BoP}_k, \cdots, \mathbf{BoP}_{N_t}]$, $\mathbf{T}_{\mathbf{r}L_0} = \left[\mathbf{Tr}_{L_0}^1, \cdots, \mathbf{Tr}_{L_0}^k, \cdots, \mathbf{Tr}_{L_0}^{N_t}\right]$, and $\mathbf{OCL} = [OCL_1, \cdots, OCL_k, \cdots, OCL_{N_t}]$.

$\mathbf{BoP}_k$ is a 21×1 dimensional matrix, $\mathbf{Tr}_{L_0}^k$ is a $L_0\times4\times N_c$ dimensional matrix, and OCL_k is a scalar. $k = 1, 2 \cdots N_t$.

III. Method

Machine learning algorithms that require large sample data training can not get a well performed model due to the limitation of the sample size of the training set, and it takes a long time to train them. In order to break through these limitations, the GC-IGTG algorithm is proposed in this study to acquire the mapping relationship between human body parameters and gait trajectory. The GC-IGTG is based on extreme learning machine (ELM), for this algorithm does not need large sample size training set and long time to train the model. To reduce the complexity of the regression model, AutoEncoder is also employed to reduce the dimension of gait trajectory. Besides that, to further reduce the error range of the generated model and improve its safety, the gait cell (GC) concept is proposed.

A. Related Theories

1) Extreme Learning Machine: A typical extreme learning machine is a single-hidden layer feedforward neural network [18], shown as Fig. 4. It has three layers: the input layer, the hidden layer and the output layer. For a training set $[\mathbf{X}, \mathbf{T}]$ with N training samples, the jth input is $\mathbf{X}^j = \left[x_1^j, \cdots, x_m^j\right]^{\mathrm{T}}$, and the jth label is $\mathbf{T}^j = \left[t_1^j, \cdots, t_n^j\right]^{\mathrm{T}}$. $j = 1, 2 \cdots N$. If the number of the hidden layer nodes is P, then the output of the ELM network is:

$$\mathbf{Y}_{n\times N} = \mathbf{V}_{n\times P} \cdot g\left(\mathbf{W}_{P\times m} \cdot \mathbf{X}_{m\times N} + \mathbf{B}_{P\times N}\right) \tag{5}$$

Here, $\mathbf{W}$ is a $P \times m$ dimensional input weights matrix, $\mathbf{B}$ is a $P \times N$ dimensional input bias matrix, and $\mathbf{V}$ is a $n \times P$ dimensional output weights matrix. $g(x)$ is a transfer function. For the traditional algorithms based on gradient descent, $\mathbf{W}$, $\mathbf{B}$ and $\mathbf{V}$ have to be trained by minimizing the cost function, shown as Equation (6), with a long time.

$$E = \|\mathbf{Y} - \mathbf{T}\|^2 \tag{6}$$

For ELM, $\mathbf{W}$ and $\mathbf{B}$ are generated randomly at first, then $\mathbf{V}$ can be calculated as below:

$$\mathbf{V} = \left(g\left(\mathbf{W}_{P\times m}\cdot\mathbf{X}_{m\times N}+\mathbf{B}_{P\times N}\right)\right)^{\dagger}\cdot\mathbf{T} \tag{7}$$

Here, † is the Moore-Penrose of a matrix. Thus ELM has a faster learning speed than other algorithms.

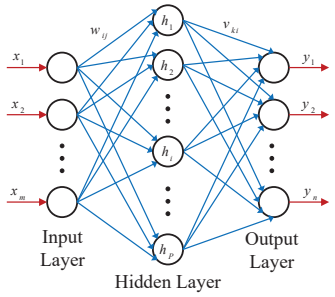

Fig. 4. The structure of ELM neural networ

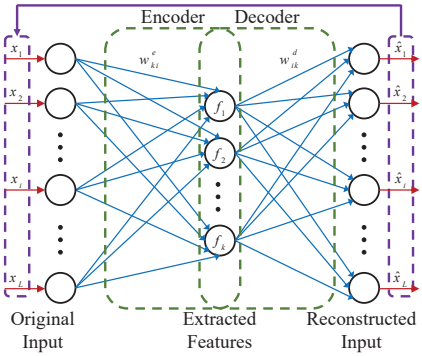

Fig. 5. The structure of AutoEncoder neural networ

2) AutoEncoder: The AutoEncoder we used is also a type of single-hidden layer neural network, shown as Fig. 5 [19]. The purpose of this network is to reduce the dimension of the original input to get the extracted features, and then the features can be reconstructed to the output, which should be almost the same as the input. AutoEncoder is a kind of unsupervised learning algorithm, in which we only need unlabeled training samples. For a training set $\{\mathbf{X}\}$, the jth input is $\mathbf{X}^j = \left[x_1^j, \cdots, x_L^j\right]^{\mathrm{T}}$. If the number of the hidden layer nodes is k, then the extracted features can be obtained as below:

$$\mathbf{F}^j = \mathbf{W}^e\cdot\mathbf{X}^j+\mathbf{B}^e \tag{8}$$

Here, $\mathbf{F}^j$ is a $k\times 1$ dimensional feature matrix, $\mathbf{W}^e$ is a $k\times L$ dimensional input weights matrix, and $\mathbf{B}^e$ is a $k\times 1$ dimensional input bias matrix. Then the reconstructed input can be calculated as below:

$$\hat{\mathbf{X}}^j = \mathbf{W}^d\cdot\mathbf{F}^j+\mathbf{B}^d \tag{9}$$

Here, $\mathbf{W}^d$ is a $L\times k$ dimensional output weights matrix, and $\mathbf{B}^d$ is a $L\times 1$ dimensional output bias matrix. The neural network training purpose is to get $\mathbf{W}^e$, $\mathbf{B}^e$, $\mathbf{W}^d$ and $\mathbf{B}^d$ by minimizing the error between $\mathbf{X}^j$ and $\hat{\mathbf{X}}^j$. Then the Encoder model is formed by $\mathbf{W}^e$, $\mathbf{B}^e$, and the Decoder model is formed by $\mathbf{W}^d$, $\mathbf{B}^d$.

B. Gait Cell

Inspired by incremental PID control algorithm, it may be more efficient and secure to find a method to generate trajectory-increment instead of the whole trajectory directly. After acquiring the training set from N_t normal testers, we find that the trajectories of the joints, which are from different testers at a fixed walking speed, are similar in shape. These trajectories vary within a small range. However, the difference between trajectories at different walking speeds increases dramatically. Thus, the gait cell (GC) concept is proposed, and a GC can be defined as: the average of the trajectories of a specific joint from N_t normal testers at a fixed walking speed. The GC contains two parts: the *OCL* part (GC_{OCL}) and the joint trajectory part (GC_{tr}). The GC of $joint_l$ at walking speed V_i can be calculated as below:

$$GC_l^{V_i} = \begin{cases} GC_{OCL} = \sum_{k=1}^{N_t}\frac{OCL_k}{N_t} \\ GC_{tr} = \sum_{k=1}^{N_t}\sum_{j=1}^{N_c} {}^{k}\mathbf{t}_{\mathbf{r}L_0}{}_{-l}^{\ j} \end{cases} \tag{10}$$

Here, ${}^{k}\mathbf{t}_{\mathbf{r}L_0}{}_{-l}^{\ j}$ is the jth resampled one-cycle trajectory of $joint_l$ from $tester_k$ at walking speed V_i, and $j \in (RH,\ RK,\ LH,\ LK)$.

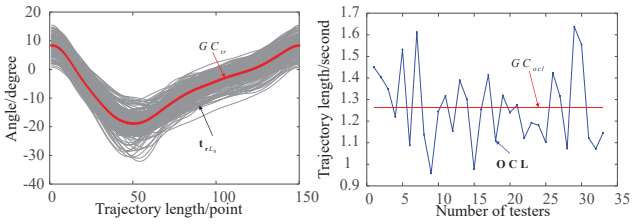

Fig. 6. The gait cell of joint RH at walking speed V_1

For the gait trajectory contains 4 joint trajectories, and each tester is tested at 7 different walking speeds. We need to build 28 GCs to train the algorithm model under different situations. The gait cell of joint RH at walking speed V_1 is shown as Fig. 6.

C. GC-IGTG Algorithm

Fig. 7 shows the framework of GC-IGTG algorithm. From N_t normal testers, we can get the training set for different walking speed, which ranges from V_1 to V_7, and every

training set includes a body parameters set and a normal gait trajectory set. Actually, the normal gait trajectory set contains 4 joint-trajectories, and each of them should be handled individually. The process of generating an individualized gait trajectory, referring to only one joint here, for an exoskeleton user can be broken down into 3 steps:

1) Training Set Selection: The training set $\mathbf{TS} = \{\mathbf{BoP},\ \mathbf{T}_{\mathbf{r}L_0},\ \mathbf{OCL}\}_{V_i}$ is decided by the patient's walking speed requirement V_i.

2) Model Training: The AutoEncoder model can be trained by the $\mathbf{T}_{\mathbf{r}L_0}$, then the normal gait features set $\mathbf{F}_{V_i}$ is acquired by entering the $\mathbf{T}_{\mathbf{r}L_0}$ into the trained Encoder. The GC_{OCL} is calculated from $\mathbf{OCL}$, and the GC_{tr} is from $\mathbf{T}_{\mathbf{r}L_0}$, which can be entered into the Encoder to get the gait cell features $\mathbf{F}_{gc}$. Then the ELM model can be trained by $\{\mathbf{BoP},\ [\mathbf{OCL} - GC_{OCL},\ \mathbf{F}_{V_i} - \mathbf{F}_{gc}]\}$.

3) Gait Trajectory Geneartion: We can get the increments of OCL and gait features $[\Delta OCL,\ \Delta\mathbf{F}]$ based on gait cell of the patient by entering the new body parameters $\mathbf{BoP}_{new}$ into the trained ELM model, then the patient's OCL and gait features $[OCL_{new},\ \mathbf{F}_{new}]$ can be acquired. A primary individual gait trajectory $\mathbf{t}_{\mathbf{r}Lo}^{new}$, whose length is Lo, can be obtained by entering $\mathbf{F}_{new}$ into the Decoder. This new trajectory should be filtered by a joint angle limitations filter. Besides that, $\mathbf{t}_{\mathbf{r}Lo}^{new}$ need to be resized to $\mathbf{t}_{\mathbf{r}L_{new}}^{new}$ according to OCL_{new}, and this one is the individualized gait trajectory for the patient.

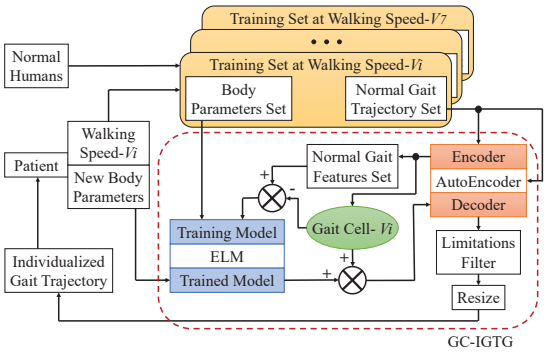

Fig. 7. The framework of GC-IGTG algorithm

IV. Experiment Research

A. Training Set Preparation

To get the training set for training the GC-IGTG model, 33 male normal humans are tested at 7 different walking speeds ($V_i \in [1.5,\ 2,\ 2.5,\ 3,\ 3.5,\ 4,\ 4.5]\ km/h$). More than 20 gait trajectory cycles are recorded of each tester at every walking speed, and 5 stable one-cycle gait trajectories are selected. The fixed length L_0 for resizing the trajectory is set as 150. Thus $N_t = 33$, $N_c = 5$ and $L_0 = 150$. The human body parameters of the 33 testers are shown as Fig. 8.

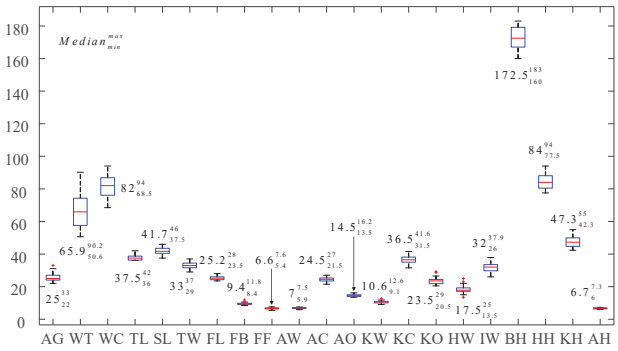

Fig. 8. The statistical analysis of human body parameters

B. Evaluation Criteria of Algorithm

The performance of a predictive generation algorithm is often evaluated by the difference between the generated gait trajectory and the original real one of the tester. In this study, the Root Mean Squared Error (RMSE) of the generated trajectory before being resized ($\mathbf{t}_{\mathbf{r}Lo}^{new}$) and the Deviation Error (DE) of the one-cycle gait length (OCL_{new}) are regarded as the evaluation criteria of GC-IGTG algorithm. OCL_{new} is a scalar, and $\mathbf{t}_{\mathbf{r}Lo}^{new}$ is a L_0 dimensional sequence. These two criteria can be calculated as below:

$$\begin{cases} C_{\mathbf{t_r}} = \sqrt{\frac{1}{L_0}\sum_{l=1}^{L_0}\left(\mathbf{t}_{\mathbf{r}l}^{new} - \mathbf{t}_{\mathbf{r}l}^{ori}\right)^2} \\ C_{ocl} = \left|\frac{OCL_{new} - OCL_{ori}}{OCL_{ori}}\right| \times 100\% \end{cases} \tag{11}$$

Here, $L_0 = 150$, $\mathbf{t}_{\mathbf{r}l}^{new}$ / $\mathbf{t}_{\mathbf{r}l}^{ori}$ is the generated/original gait trajectory, and OCL_{new} / OCL_{ori} is the generated/original OCL.

C. Experimental Results

Because of the limitation of our sample size, only one tester is regarded as the testing target at a time, and the remaining 32 testers' data are employed as training set. 5 testers, randomly selected from the total data set, form the testing set, and they are labeled as tester A, tester B, tester C, tester D, and tester E.

The mean $C_{\mathbf{t_r}}$ of the selected 5 testers, including the 4 joints trajectories (RH, RK, LH, LK) at 7 different walking speeds (V_1 to V_7), are shown in Table I. Besides that, the C_{ocl} of the testers are all less than 4%, which are not listed out here. The details of the experimental results of tester A at walking speed V_1 are shown in Fig. 9. The left figure shows the OCL of the 4 joints. The original OCL of tester A is 1.403 seconds, and the generated OCLs range from 1.396 to 1.414 seconds, which indicates that the C_{ocl} is less than 0.8%. The grey solid lines of the right figure are the original trajectories of the 4 joints of tester A in 5 cycles, and the

blue star lines are the averages of them. The red circle lines are the generated trajectories by GC-IGTG, which indicates the generated joints trajectories are almost identical to the original ones.

TABLE I

THE $C_{\mathbf{t_r}}$ OF 5 TESTERS AT 7 WALKING SPEEDS

Walking Speed	Mean $C_{\mathbf{t_r}}$ of 5 testers			
	RH	RK	LH	LK
V_1	1.27	3.20	1.45	2.48
V_2	1.33	2.48	1.63	2.71
V_3	1.58	2.57	2.09	2.61
V_4	1.66	2.61	1.62	2.86
V_5	1.59	2.68	1.63	3.44
V_6	1.51	1.75	1.93	2.09
V_7	1.28	2.04	1.35	2.15

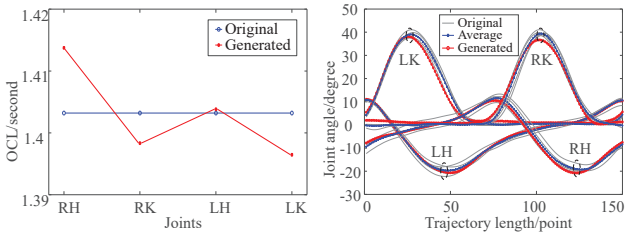

Fig. 9. The experimental results of tester A at walking speed V_1

V. CONCLUSIONS

This paper proposes the GC-IGTG algorithm, which aims at generating a natural and individual rehabilitation gait trajectory reference for the lower extremity exoskeleton based on the body parameters of the exoskeleton user. Extreme learning machine and AutoEncoder are the key components of the GC-IGTG algorithm, then this novel algorithm owns fast training speed and it is not limited to the sample size of the training set. Besides that, the gait cell concept, inspired by incremental PID control algorithm, makes it more efficient and secure. In this work, 21 gait-related human body parameters are regarded as the input of the GC-IGTG, and the output are the trajectories of the hip and knee joints in the sagittal plane. The experimental results validate the efficiency of the proposed algorithm.

ACKNOWLEDGMENT

The authors would like to thank Duxin Liu who offered the training set for this study.

REFERENCES

[1] D. o. E. United Nations and P. D. Social Affairs, "World population prospects 2019: Highlights," 2019.

[2] D. L. Brown-Triolo, M. J. Roach, K. Nelson, and R. J. Triolo, "Consumer perspectives on mobility: implications for neuroprosthesis design," *Journal of rehabilitation research and development*, vol. 39, no. 6, pp. 659–670, 2002.

[3] S. L. Nogueira, S. Lambrecht, R. S. Inoue, M. Bortole, A. N. Montagnoli, J. C. Moreno, E. Rocon, M. H. Terra, A. A. Siqueira, and J. L. Pons, "Global kalman filter approaches to estimate absolute angles of lower limb segments," *Biomedical engineering online*, vol. 16, no. 1, p. 58, 2017.

[4] 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.

[5] 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.

[6] R. J. Farris, H. A. Quintero, and M. Goldfarb, "Preliminary evaluation of a powered lower limb orthosis to aid walking in paraplegic individuals," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 6, pp. 652–659, 2011.

[7] Ü. Önen, F. M. Botsalı, M. Kalyoncu, M. Tınkır, N. Yılmaz, and Y. Şahin, "Design and actuator selection of a lower extremity exoskeleton," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 623–632, 2013.

[8] M. Sczesny-Kaiser, O. Höffken, M. Aach, O. Cruciger, D. Grasmücke, R. Meindl, T. A. Schildhauer, P. Schwenkreis, and M. Tegenthoff, "Hal® exoskeleton training improves walking parameters and normalizes cortical excitability in primary somatosensory cortex in spinal cord injury patients," *Journal of neuroengineering and rehabilitation*, vol. 12, no. 1, p. 68, 2015.

[9] N. Birch, J. Graham, T. Priestley, C. Heywood, M. Sakel, A. Gall, A. Nunn, and N. Signal, "Results of the first interim analysis of the rapper ii trial in patients with spinal cord injury: ambulation and functional exercise programs in the rex powered walking aid," *Journal of neuroengineering and rehabilitation*, vol. 14, no. 1, p. 60, 2017.

[10] Y. He, N. Li, C. Wang, L.-q. Xia, X. Yong, and X.-y. Wu, "Development of a novel autonomous lower extremity exoskeleton robot for walking assistance," *Frontiers of Information Technology & Electronic Engineering*, vol. 20, no. 3, pp. 318–329, 2019.

[11] 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.

[12] X. Wu, D.-X. Liu, M. Liu, C. Chen, and H. Guo, "Individualized gait pattern generation for sharing lower limb exoskeleton robot," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 4, pp. 1459–1470, 2018.

[13] D.-X. Liu, X. Wu, W. Du, C. Wang, and T. Xu, "Gait phase recognition for lower-limb exoskeleton with only joint angular sensors," *Sensors*, vol. 16, no. 10, p. 1579, 2016.

[14] C. Wang, X. Wu, Z. Wang, and Y. Ma, "Implementation of a brain-computer interface on a lower-limb exoskeleton," *IEEE Access*, vol. 6, pp. 38 524–38 534, 2018.

[15] J. Vermeulen, B. Verrelst, B. Vanderborght, D. Lefeber, and P. Guillaume, "Trajectory planning for the walking biped lucy," *The international journal of robotics research*, vol. 25, no. 9, pp. 867–887, 2006.

[16] R. Huang, H. Cheng, Y. Chen, Q. Chen, X. Lin, and J. Qiu, "Optimisation of reference gait trajectory of a lower limb exoskeleton," *International Journal of Social Robotics*, vol. 8, no. 2, pp. 223–235, 2016.

[17] Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. van de Panne, "Feedback control for cassie with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1241–1246.

[18] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[19] Q. Xiao and Y. Si, "Human action recognition using autoencoder," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2017, pp. 1672–1675.