Particle Filter Based Lower Limb Prediction and Motion Control for JAIST Active Robotic Walker

Takanori Ohnuma, Geunho Lee, and Nak Young Chong

Abstract— **This paper presents an interactive control for our assistive robotic walker, the JAIST Active Robotic Walker (JARoW), developed for elderly people in need of walking assistance. The focus of our paper is placed on how to estimate the user's walking parameters by sensing the locations of lower limbs and to predict his or her walking patterns. For this purpose, a particle-filter-based prediction technique and a motion controller are developed to help JARoW smoothly generate the direction and velocity of its movements in a way that reflects the prediction. The proposed scheme and its implementation are described in detail, and outdoor experiments are performed to demonstrate its effectiveness and feasibility in everyday environments.**

I. Introduction

A walker is a device for elderly people who need additional support to maintain balance and stability while walking. Basic traditional walkers mainly consist of a frame surrounded by four legs on the front and sides; body support is provided by the user holding onto the top of the sides. As common structures, four- (or three-) wheel type traditional walkers can be rolled around daily environments with little effort. The merits of these walkers include low cost, simple design, and compact size. However, users must take overly cautious steps so as not to push the walker too far forward, since it lacks a feedback control system. Other disadvantages of traditional walkers include their difficulties navigating carpets, uneven floors with dips and depressions, thresholds, and ascending or descending ramps often encountered in daily routines. Several papers reported that these walkers may increase the risk of falling [1-2].

Robotic walkers with similar mechanical structures to traditional walkers have emerged, but the installation of mechanical and/or electronic components is often required to promote safer ambulatory assistance. Recent technological advances have allowed the incorporation of a range of features into robotic walkers, which can be specialized with physical [3], sensory [4], and maneuverability [5] assistances. Moreover, users utilize their own remaining ambulatory capability when walking with such devices, which plays an important role in helping users exercise. The walker therefore needs to be comfortable and easy to use.

The robotic walker prototype discussed in this paper, the JAIST Active Robotic Walker (JARoW) [6-8], was developed with these considerations in mind (see Fig. 1). Specifically, JARoW does not require specific manual controls or additional equipment. Furthermore, the JARoW's circular shape, light weight, and compact size make the walker easy to use in daily life. Ultimately, JARoW encourages elderly people to lead more active lives, with reduced need for assistance. Toward the practical use of JARoW, a walking intent based movement control [8] was proposed, allowing JARoW to accurately generate the direction and location of its movement in a way that corresponds to the user's walking behaviors. The success of the control for JARoW was confirmed through extensive experiments where elderly subjects currently using traditional walkers participated. To provide the more widespread use of JARoW, there still remains an issue of how to realize the JARoW's smooth motion generation rather than discrete or intermittent movements. The smooth and accurate motions can be directly connected with the improvement of its usability and reliability.

Fig. 1. Prototype of JAIST Active Robotic Walker (JARoW) and its interface

In general, the human gait is nonlinear, and the gait parameters of individuals are not always steady during walking. Furthermore, JARoW should be able to accommodate various individual levels of physical capability. For the purpose, our challenge aims at analyzing the different gait parameters of users and applying the analysis into the JARoW's control. As the main contribution, a particle-filter-based estimation and prediction technique is presented to estimate and predict the locations of the user's lower limbs. Based on the proposed technique, the filtering function is implemented as one function in the main controller. Moreover, as a tracking function, a motion controller reflecting the estimation and prediction is proposed, enabling JARoW to control its discrete or intermittent motions. After the realization of the filtering and the tracking functions, JARoW can autonomously control its smooth motions adapting to the user's walking patterns. The objective of this paper is to introduce this particle-filter-based prediction and motion control technique.

T. Ohnuma and N. Y. Chong are with the School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan. e-mail: {t-ohnuma, nakyoung}@jaist.ac.jp

G. Lee is with the Department of Environmental Robotics, University of Miyazaki, 1-1 Gakuen Kibanadai-nishi, Miyazaki 889-2192, Japan. e-mail: geunho@cc.miyazaki-u.ac.jp

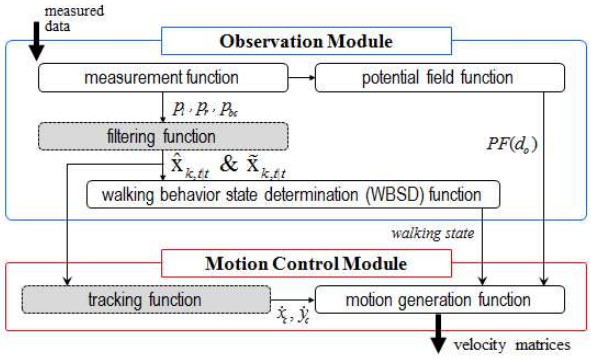


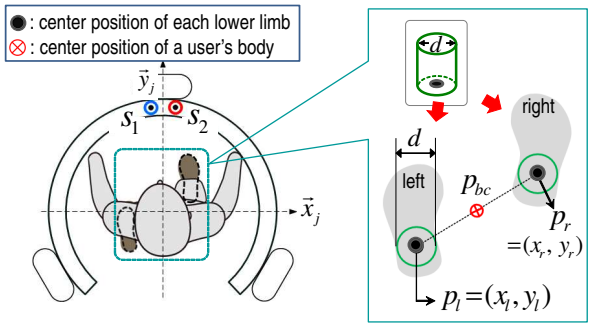
Fig. 2. Control flow in the JARoW's main controller

II. JARoW: System Description

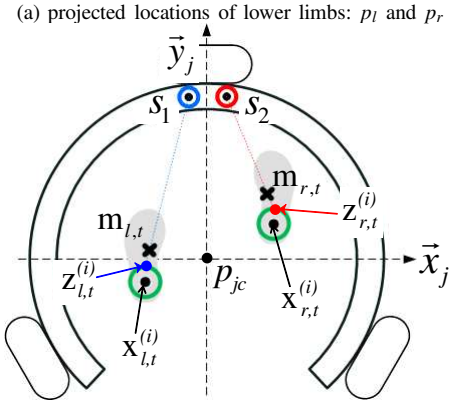
JARoW and its interface system are shown in Fig. 1. JARoW's design is compact, and its footprint circular, which reduces the potential for collisions with obstacles or walls. JARoW has three main structural parts: a base frame, an upper frame, and connecting rods. The base frame supports the superstructure, and is directly connected to the drive-train and equipped with two Hokuyo URG-04LX laser range finders (LRFs) as the interface system. The length of the connecting rod can be adjusted according to the height of users. Users are able to lean their upper body forward and place their forearms onto the upper frame. Details on the JARoW's mechanical specification and its first-order kinematics can be found in [7].

Next, the control components of JARoW consist of the drive-train with three omni-directional wheels, the interface system, and the main controller mounted on top of the upper frame. A laptop PC running Microsoft's Windows 7 is used as the main controller. To begin, the interface system is able to detect the locations of the user's lower limbs, as well as obstacles or area borders. In the interface, two LRFs are represented as S_1 and S_2, indicating the left and the right LRF, respectively. Each LRF outputs a 240 degree scan and measures up to 4000mm with 100ms sweep interval; pair of LRFs can cover a full 360 degree spectrum. The sampling time, counted according to individual laser beams emitted in the synchronized S_1 and S_2, is defined as t. At each t, the interface system outputs the measured data, which are fed to the main controller.

The solution proposed in this paper is embedded in the filtering and tracking functions of the main controller, shown in Fig. 2. The input to the controller includes measurement data obtained from the interface, and the output of the controller is velocity matrices to the drive-train. More specifically, the observation and the motion control modules play roles in recognizing users' walking patterns and generating JARoW's motions. A detailed explanations of how the filtering and tracking functions were designed will be described in Section 3 and 4, respectively. Descriptions of the other functions can be found in [8].



(a) projected locations of lower limbs: p_l and p_r



(b) locations estimated by measurement data

Fig. 3. Definitions and notations used in the particle-filter-based estimation and prediction technique

III. Filtering Function: Particle-Filter-Based Lower Limb Estimation and Prediction

The particle-filter-based technique is composed of two phases: estimation and prediction. The technique is designed and implemented as follows.

A. Definition and Notations

As depicted in Fig. 3, JARoW's local coordinates are x_j (horizontal axis) and y_j (vertical axis) where y_j is defined as the direction of its forward motion. Its center position is denoted by $p_{jc} = (x_{jc}, y_{jc})$. Measurement data are calculated with respect to the JARoW's local coordinates initialized at the beginning of each sweep interval. Accordingly, p_{jc} is $(0,0)$ with respect to its x_j and y_j as soon as it updates at each sweep interval. The lower limbs are modeled as cylinders with a diameter d, representing each shin that is vertically projected onto a two-dimensional plane (for simplicity, $x_j y_j$) with respect to x_j and y_j. In Fig. 3-(a), the projected centers of the individual cylinders correspond to the center positions of a user's shins; the right and left projected centers are defined as $p_r = (x_r, y_r)$ and $p_l = (x_l, y_l)$, respectively. At each t, the center position vector is represented by $\mu_{k,t} = [p_{r,t}\ p_{l,t}]^T$. Next, a valid region for the location measurement of lower limbs is set to a rectangle with 900mm×800mm (length and width) on $x_j y_j$ plane. We assume that $p_{r,t}$ and $p_{l,t}$ remain within the valid region at each t. In other words, these assumptions indicate that $p_{r,t}$ and $p_{l,t}$ are always observable within the region.

The variables of interest are the locations of the shins with respect to x_j and y_j. In Fig. 3-(b), the variables at each t are defined as a set of n samples (i.e., particles $s_{k,t|t}^{(i)} = \{\mathbf{x}_{k,t|t}^{(i)}, w_{k,t}^{(i)}\} : i = 1,2,\cdots,n)$, where k and i denote the right or the left side and the $i-th$ order, respectively. Moreover, each particle consists of its 2-D location vector $\mathbf{x}_{k,t|t}^{(i)}$ and its associated weight $w_{k,t}^{(i)}$. In particular, $\mathbf{x}_{k,t|t}^{(i)}$ with respect to x_j and y_j is defined as:

$$\mathbf{x}_{k,t|t}^{(i)} = [x_{r,t|t}^{(i)} \; y_{r,t|t}^{(i)} \; x_{l,t|t}^{(i)} \; y_{l,t|t}^{(i)}]^T. \tag{1}$$

The state vector for a given particle $\mathbf{x}_{k,t|t}^{(i)}$ is defined as $X_{k,t}$. $M_{k,t}$ denotes the sensor observation vector for measurement data $\mathbf{m}_{k,t}$.

B. Location Estimation

Given $M_{k,t}$, to obtain the posterior probability $p_k(X_{k,t}|M_{k,t})$ of $X_{k,t}$, the location estimation phase is realized as follows.

- step-1: initialization

As a set of particles estimating the locations of the shins at $t = 0$, the number of n initial particles are generated and denoted by $\{s_{k,0|0}^{(i)} | 1 \le i \le n\}$, where $s_{k,0|0}^{(i)}$ indicates $\{\mathbf{x}_{k,0|0}^{(i)}, \; w_{k,0|0}^{(i)}\}$. Specifically, by the use of the Metropolis-Hastings (for simplicity, M-H) algorithm [9], $\mathbf{x}_{k,0|0}^{(i)}$ is obtained from the variance vector $\alpha_{k,0}^2$ with a predefined distribution around $\mu_{k,0}$. It is also assumed that $w_{k,0}$ is a constant.

- step-2: system model at time t

The system model representing the forward and backward movements of both lower limbs is formalized as:

$$\mathbf{x}_{k,t|t-1}^{(i)} = \begin{bmatrix} x_{r,t|t-1}^{(i)} \\ y_{r,t|t-1}^{(i)} \\ x_{l,t|t-1}^{(i)} \\ y_{l,t|t-1}^{(i)} \end{bmatrix} = \begin{bmatrix} x_{r,t-1|t-1}^{(i)} \\ y_{r,t-1|t-1}^{(i)} \\ x_{l,t-1|t-1}^{(i)} \\ y_{l,t-1|t-1}^{(i)} \end{bmatrix} + \begin{bmatrix} n_{r,t-1}^{(i)} \\ \dot{y}_{t-1}^{(i)} \delta t \\ n_{l,t-1}^{(i)} \\ -\dot{y}_{t-1}^{(i)} \delta t \end{bmatrix}, \tag{2}$$

where δt denotes the sampling period, and $n_{k,t-1}^{(i)}$ indicates the system noise for $\mu_{k,t}$ sampled by using the M-H algorithm. In addition, $\dot{y}_{r,t-1}^{(i)}$ is autonomously determined according to the following three modes:

$$\dot{y}_{t-1}^{(i)} = \left\{ \begin{array}{ll} -\frac{\pi R_{t-1}^{(i)}}{2T_{t-1}^{(i)}} \sin(\frac{\pi \tau_{t-1}^{(i)}}{T_{t-1}^{(i)}}) & \text{(right)} \\ \frac{\pi R_{t-1}^{(i)}}{2T_{t-1}^{(i)}} \sin(\frac{\pi \tau_{t-1}^{(i)}}{T_{t-1}^{(i)}}) & \text{(left)} \\ 0 & \text{(otherwise)} \end{array} \right\} \tag{3}$$

where $R_{t-1}^{(i)}$ and $T_{t-1}^{(i)}$ denote step length and gait cycle parameters, respectively. Moreover, the time required to reset each mode is defined as $\tau_{t-1}^{(i)}$. Using (2) and (3), $\mathbf{x}_{k,t|t-1}^{(i)}$ in $s_{k,t|t-1}^{(i)}$ are computed from $\mathbf{x}_{k,t-1|t-1}^{(i)}$ in $s_{k,t-1|t-1}^{(i)}$.

- step-3: computation of the i-th particle's likelihood

Given $X_{k,t} = [\mathbf{x}_{k,t|t-1}^{(1)} \; \mathbf{x}_{k,t|t-1}^{(2)} \; \cdots \mathbf{x}_{k,t|t-1}^{(n)}]^T$, the i-th particle's likelihood $p_k^{(i)}(X_{k,t}|M_{k,t})$ of the observation vector $M_{k,t}$ is computed:

$$p_k^{(i)}(X_{k,t}|M_{k,t}) = \frac{1}{\sqrt{2\pi}\sigma_s} \exp(-\frac{D_k^{(i)^2}}{2\sigma_s^2}), \tag{4}$$

where σ_s denotes the standard deviation for the permissible location error. Next, the Euclidean distance $D_k^{(i)}$ between $\mathbf{m}_{k,t}$ and the point vector $\mathbf{z}_{k,t}^{(i)}$ intersecting between the circle with the radius d at the center of $\mathbf{x}_{k,t|t-1}^{(i)}$ and the beams emitted from the LRFs (see Fig. 3-(b)) are defined:

$$D_k^{(i)} = \|\mathbf{z}_{k,t}^{(i)} - \mathbf{m}_{k,t}\|. \tag{5}$$

To reduce computational loads practically, $\mathbf{x}_{k,t|t-1}^{(i)}$ is not considered as the representation for $\mu_{k,t}$ in the following four cases: 1) $\mathbf{x}_{k,t|t-1}^{(i)}$ located outside the valid region, 2) $\|\mathbf{x}_{k,t|t-1}^{(i)} - \mathbf{m}_{k,t}\| < d$, 3) having no intersection points $\mathbf{z}_{k,t}^{(i)}$, and 4) $\mathbf{m}_{k,t}$ located d or more away from $\mathbf{x}_{k,t|t-1}^{(i)}$ along the $-y_j$ direction. In our implementation, $p_k^{(i)}(X_{k,t}|M_{k,t})$ corresponding to the four cases is set to 0.

- step-4: computation of the i-th weight

The associated weight $w_{k,t}^{(i)}$ is given:

$$w_{k,t}^{(i)} = \frac{p_k^{(i)}(X_{k,t}|M_{k,t})}{\sum_{i=1}^{n} p_k^{(i)}(X_{k,t}|M_{k,t})}. \tag{6}$$

- step-5: re-sampling [10]

For $\mathbf{x}_{k,t|t-1}^{(i)}$, our re-sampling is to eliminate $s_{k,t|t-1}^{(i)}$ with small weights, and then is to concentrate and replicate on $s_{k,t|t-1}^{(i)}$ with large wights in order to best explain $\mathbf{m}_{k,t}$ according to their likelihoods. Through such a re-sampling, $s_{k,t|t}^{(i)} = \{\mathbf{x}_{k,t|t}^{(i)}, \; w_{k,t}^{(i)}\}$ is obtained.

- step-6: location estimate of both shins

Using $\mathbf{x}_{k,t|t}^{(i)}$ obtained in step-5, as the expectations $\hat{\mathbf{x}}_{k,t|t}^{(i)}$, the location estimates of both shins $\mathbf{x}_{k,t|t}$ are calculated:

$$\hat{\mathbf{x}}_{k,t} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_{k,t|t}^{(i)}. \tag{7}$$

Then, the processes from step-2 to step-6 are reiterated.

C. Location Prediction

After re-sampling $R_{t-1}^{(i)}$, $T_{t-1}^{(i)}$, and $\tau_{t-1}^{(i)}$ used in (2) and (3) according to the manner of step-5, the expectations of $\hat{R}_{t-1}^{(i)}$, $\hat{T}_{t-1}^{(i)}$, and $\hat{\tau}_{t-1}^{(i)}$ are obtained. One merit of re-sampling is that the location state represented by re-sampling particles with large $w_{k,t}^{(i)}$ at each t can correspond with the most appropriate $\hat{R}_{t-1}^{(i)}$, $\hat{T}_{t-1}^{(i)}$. Therefore, employing $\hat{R}_{t-1}^{(i)}$, $\hat{T}_{t-1}^{(i)}$, the predicted locations of both shins $\tilde{\mathbf{x}}_{k,t|t}$ are given:

$$\tilde{\mathbf{x}} = \begin{bmatrix} \hat{x}_{r,\,t} \\ \hat{y}_{r,\,t} + y_r \\ \hat{x}_{l,\,t} \\ \hat{y}_{l,\,t} - y_r, \end{bmatrix} \tag{8}$$

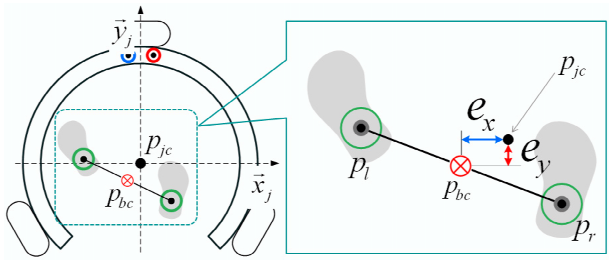


Fig. 4. Position difference between p_{jc} and p_{bc}

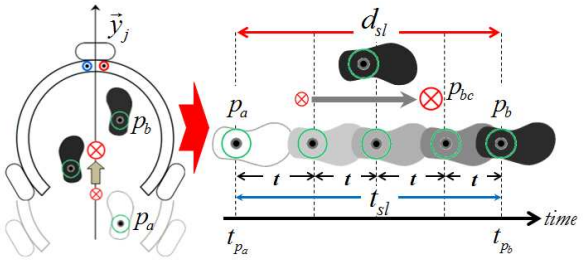


Fig. 5. Definitions and notations for stride length d_{sl}

where y_r is autonomously determined according to the following three modes:

$$y_r = \left\{ \begin{array}{ll} -\frac{1}{2}\hat{R}_{t-1}\{1+\cos(\frac{\pi\hat{\tau}_{t-1}}{\hat{T}_{t-1}})\} & \text{(right)} \\ \frac{1}{2}\hat{R}_{t-1}\{1+\cos(\frac{\pi\hat{\tau}_{t-1}}{\hat{T}_{t-1}})\} & \text{(left)} \\ 0 & \text{(otherwise)} \end{array} \right\}. \quad (9)$$

IV. Tracking Function: Motion Controller

We occasionally observed a problem occurred due to intermittent or discrete motions of JARoW caused by relative locations based on $\vec{x}_j$ and $\vec{y}_j$ redefined according to each sweep interval. For example, JARoW was swaying back and forth like a pendulum while walking. To further enhance its accurate and smooth motion generation, a more sophisticated controller is required. For this, a PID_VEP controller is newly designed, integrating the PID controller and a velocity-estimation-and-prediction (VEP) technique for p_{bc}.

To begin, the PID controller enables JARoW to figure out differences between p_{jc} and p_{bc}, as errors, in $\vec{x}_j$ and $\vec{y}_j$ directions, respectively (see Fig. 4). And then, JARoW reflects the errors to its motions in order to remain coincident with each other. The reasons why p_{jc} needs to coincide with p_{bc} are twofold: (1) JARoW assists the well-proportioned walking of a user so that the user body is not leaning to one side, and (2) avoids bumping into the user while rotating.

As illustrated in Fig. 4, errors between p_{bc} and p_{jc} in $\vec{x}_j$ and $\vec{y}_j$ directions are defined as $e_x = x_{bc} - x_{jc}$ and $e_y = y_{bc} - y_{jc}$, respectively. To minimize errors, the PID controller is realized:

$$\begin{array}{rcl} \dot{x}_c & = & K_{p,x}\ e_x + K_{i,x}\ \int e_x dt + K_{d,x}\ \dot{e}_x \\ \dot{y}_c & = & K_{p,y}\ e_y + K_{i,y}\ \int e_y dt + K_{d,y}\ \dot{e}_y \end{array}, \quad (10)$$

where $\dot{x}_c$ and $\dot{y}_c$ are the output velocities of JARoW, and K_p, K_i, and K_d denote the proportional, integral, and derivative gains, respectively.

Next, despite the use of the PID controller, since a user within JARoW moves continuously, errors of varying degrees may persist. Moreover, since the geometrically relative locations are directly connected to the intermittent motions of JARoW, its motions need to harmonize with the gait cycle of the user. In other words, a control input considering the gait cycle is indispensable for generating the motions more smoothly. Accordingly, the VEP technique integrating an absolute and predicted velocity for p_{bc} is designed.

Before the explanation of the VEP technique, we consider a scene for forward movement as depicted in Fig. 5. The user shifted the right shin from the one location p_a to the other p_b while the left shin is fixed. Here, p_r is represented as p_a and p_b according to t. The stride length d_{sl} is defined from p_a to p_b, and t_{sl} denotes the sum of some sampling times as long as it took to shift from p_a to p_b.

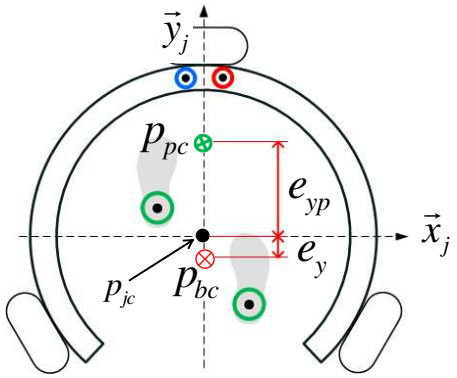


Fig. 6. Definitions and notations of the distance e_{yp} between p_{jc} and p_{pc} in $\vec{y}_j$ direction

Given the dominant walking state in human gait, the VEP technique puts restrictions on the forward movements, namely d_{sl} to both shins only in $+\vec{y}_j$ direction. Furthermore, it is assumed that the shins move two times faster than p_{bc} during forward movement. Under this assumption, the absolute velocity v_{av} of p_{bc} is obtained:

$$v_{av} = \frac{d_{sl}}{2t_{sl}}. \quad (11)$$

From (8) and (9), the predicted body center $p_{pc} = (x_{pc}, y_{pc})$ can be obtained (see Fig. 6). Similar to e_x and e_y, an error distance e_{yp} for p_{pc} according to $\vec{y}_j$ direction is defined as $e_{yp} = y_{jc} - y_{pc}$. Moreover, an error distance to y_{bc} from y_{pc} in $\vec{y}_j$ direction is given as $e_{yp} + e_y$. Here, we recall the case of shifting the right shin from p_a to p_b in Fig. 5. From the viewpoint of t_{sl}, the predicted velocity v_{pv} of p_{pc} is computed:

$$v_{pv} = \frac{e_{yp} + e_y}{t_{sl}}. \quad (12)$$

When v_{av} combines with v_{pv} to form an absolute and predicted velocity for p_{bc}, the control input considering the gate cycle of a user is formalized:

$$v_{vep} = \frac{v_{pv} + v_{av}}{2} = \frac{d_{sl} + 2e_{yp} + 2e_y}{4t_{sl}}. \quad (13)$$

Fig. 7. Experimental scenes for performances and validity

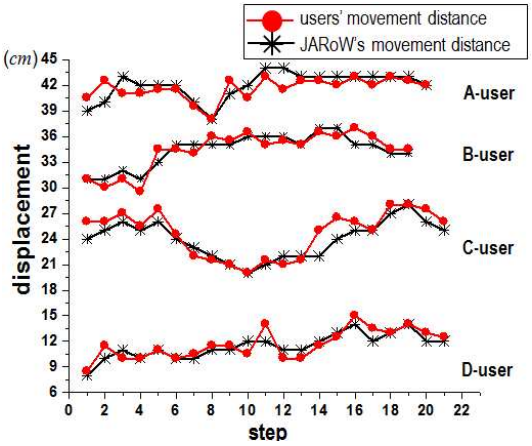
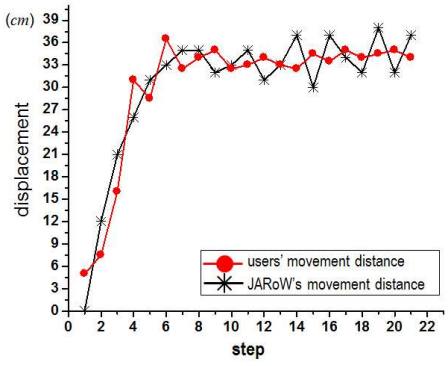
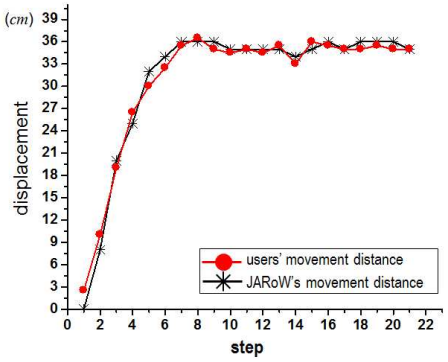


Fig. 8. Experimental results for JARoW displacements according to the forward movements of individual users



(a) results performed under the previous control [8]



(b) results performed under the proposed control

Fig. 9. Comparison results for JARoW's displacements based on the previous control and the PID_VEP controller

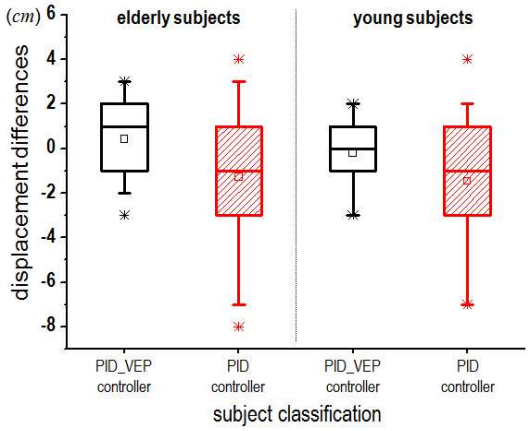


Fig. 10. Comparison results for differences between individual displacements and JARoW movements based on the PID controller proposed in [7] and the PID_VEP controller, respectively, according to elderly and young subjects

Note that, to promote the smooth variations of JARoW's movements, v_{vep} is standardized as an average velocity. Finally, for the forward movements, a PID_VEP controller adding v_{av} into (10) is summarized:

$$\dot{y}_c = v_{vep} + K_{p,y}\ e_y + K_{i,y} \int e_y dt + K_{d,y}\ \dot{e}_y. \qquad (14)$$

The tracking function computes $\dot{x}_c$ and $\dot{y}_c$ for the PID_VEP controller by using e_x and e_y. From the walking state obtained by the WBSD function [8], either $\dot{x}_c$, $\dot{y}_c$, or both $\dot{x}_c$ and $\dot{y}_c$ is selected. First, for forward/backward movement states, JARoW is controlled according to (14). Second, only $\dot{x}_c$ in (10) is applied in the step left/right states. Third, during the turning left/right states, JARoW is controlled according to the combination of $\dot{x}_c$ and $\dot{y}_c$ in (10).

V. Evaluation Results and Discussion

This section presents results of experiments conducted to evaluate features and performances for two proposed functions embedded in the JARoW's main controller. As shown in Fig. 7, outdoor experiments were performed by 6 elderly subjects (male: 2, female: 4, age: 75-82 years, height: 152-175cm). Before participating in our experiments, written informed consent for the publication of this study and any accompanying images was obtained from the subjects involved in all experiments. During the experiments, JARoW moves under the maximum linear velocity of $1.333m/s$ throughout all experiments. When it generates a rotational motion, the magnitude of the angular velocity is $0.5rad/s$. The gains for the PID controller are set $K_{p,x} = K_{p,y}$ =2.2, $K_{i,x} = K_{i,y}$ =0.5, and $K_{d,x} = K_{d,y}$ =0.05.

Before our experiments, we briefly explained the use of JARoW and demonstrated its basic performance. The elderly subjects were given five or ten minutes in order to become familiar with its use. Then, they were requested to walk along an assigned path of a distance of 15 meters. When the assigned task was performed by individual subjects, we recorded their displacements and the JARoW's movements

according to their displacement under the proposed control. Fig. 8 shows the experimental results for JARoW displacements controlled by the proposed PID_VEP controller. Despite different forward movements of individual users, JARoW could follow their walking behaviors. To clear the difference between the previous and the proposed controls, the same task was conducted under the previous control [8] and recorded. Compared with Fig. 9-(a), JARoW's displacement variations in Fig. 9-(b) became almost similar. It was confirmed that the proposed PID_VEP controller allowed JARoW to generate motions which closely corresponded to the walking behaviors of the users. Moreover, the controller helped JARoW facilitate smooth forward movements.

After the same numbered young subjects in their twenties were selected, the same experiments were performed and experimental data depending on the individuals were recorded. Furthermore, the same task was conducted under the previous control [7] and recorded. Fig. 10 shows comparison results where the error bars represent 95% confidence intervals and the boxes indicate distributions of measured data in the range of 25-75%. Despite their different stride lengths, it was verified that the proposed control allowed JARoW to track the displacements of the subjects accurately. As we expected, the proposed control appeared to yield better performance due to less variations for the relative differences. More notably, the proposed control allowed potential users to easily control JARoW without requiring any mental or physical effort.

VI. Conclusions

This paper presented an enhanced interactive scheme for the JARoW's smooth motion generation that does not requiring manual user controls or additional control equipment. To accomplish this, we first proposed the particle-filter-based estimation and prediction scheme to estimate and predict the locations of the user's lower limbs. Secondly, based on the proposed scheme, the filtering function was implemented as one function in the main controller. Thirdly, the use of a PID_VEP controller is proposed. This controller has an effect on controlling its discrete or intermittent motions caused by relative relations. To demonstrate the effectiveness and the feasibility of the proposed control, a series of experiments were performed. These results were analyzed and compared to our previous findings. From these results, we are able to confirm that the proposed control and its realization enhanced the JARoW's usability and reliability, since it effectively controlled discrete or intermittent motions. Although for the purpose of this study we considered potential users with only a narrowly defined range of ambulatory capability, in future, a more refined version of JARoW will need to incorporate an enhanced interactive control scheme that enables JARoW to accommodate potential users with unusual gait.

References

[1] J.A Stevens, K. Thomas, L. Teh, and A.I. Greenspan, "Unintentional fall injuries associated with walkers and canes in older adults treated in U.S. emergency departments," *Journal of the American Geriatric Society*, vol.58, no.8, pp.1464-1469, 2009

[2] H. Bateni and B.E. Maki, "Assistive devices for balance and mobility: benefits, demands, and adverse consequences," *Archives of Physical Medicine and Rehabilitation*, vol.86, no.1, pp.134-145, 2005

[3] K. Kong, H. Moon, B. Hwang, D. Jeon, and M. Tomizuka, "Impedance compensation of SUBAR for back-drivable force-mode actuation," *IEEE Trans. Robotics*, vol.25, no.3, pp.512-521, 2009

[4] A.F. Neto, J.A. Gallego, E. Rocon, J.L. Pons, and R. Ceres, "Control and path planning of a walk-assist robot using differential flatness," *BioMedical Engineering*, vol.9, no.37, pp.1-16, 2010

[5] M. Nokata and W. Hirai, "Unrestraint support robot for elderly gait rehabilitation," *Proc. 11th IEEE Int. Conf. Rehabilitation Robotics*, pp.614-620, 2009

[6] G. Lee, T. Ohnuma, and N.Y. Chong, "Design and control of JAIST active robotic walker," *Journal of Intelligent Service Robotics*, vol.3, no.3, pp.125-135, 2010

[7] G. Lee, E.-J. Jung, T. Ohnuma, N.Y. Chong, and B.-J. Yi, "JAIST robotic walker control based on a two-layered Kalman filter," *Proc. IEEE Int. Conf. Robotics and Automation*, pp.3682-3687, 2011

[8] G. Lee, T. Ohnuma, N.Y. Chong, and S.-G. Lee, "Walking Intent Based Movement Control for JAIST Active Robotic Walker," *IEEE Trans. Systems, Man and Cybernetics: Systems*, vol.44, no.5, pp.665-672, 2014

[9] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings algorithm," *American Statistician*, vol.49, no.4, pp.327-335, 1995

[10] G. Kitagawa, "Monte Carlo filtering and smoothing method for nonlinear non-Gaussian state space models," *Journal of Computational Graphical Statistics*, vol.5, no.1, pp.1-25, 1996