International Journal of Networked and Distributed Computing, Vol. 4, No. 3 (July 2016), 145-158

Adaptive Omni-directional Walking Method with Fuzzy Interpolation for Biped Robots

Haobin Shi[1,*], Xuesi Li [1], Huahui Chen [2], Shixiong Wang [2]

[1] School of Computer Science, Northwestern Polytechnical University,
127 West Youyi Road
Xi'an, 710072, China
E-mail: shihaobin@nwpu.edu.cn, lixuesi@mail.nwpu.edu.cn

[2] School of Electronics and Information, Northwestern Polytechnical University,
127 West Youyi Road
Xi'an, 710072, China
E-mail: chh_chh_@mail.nwpu.edu.cn, wangshixiong@mail.nwpu.edu.cn

Abstract

Omni-directional walking is an important part for biped robots. Whereas, the problems of complex model, low stability, low flexibility, etc., existing in walking planning of biped robots, attract interests of many researchers. This paper proposes an adaptive omni-directional walking method with fuzzy interpolation (OW-FI) applied to a fast-stable omni-directional walking. A new separated omni-directional walking method consisting of straight walking, based on the improve Hermite interpolation, and rotation is designed to simplify the walking model and achieve a good key frame data selection. The mapping relationship between poses and actuators' angular speeds is then constructed by walking transforming model based on kinematics. In order to implement a stable and adaptive walking, a fuzzy approach to walking parameters adjustment is presented, where the step length and rotational speed of omni-directional walking are controlled fuzzily. The results of simulations and experiments demonstrate that the proposed method has better properties in adaptability and stability than the competing methods.

Keywords: Omni-directional walking, Biped robot, Fuzzy control, Hermite Interpolation, Robot kinematics.

1. Introduction

Nowadays, biped robots have broad application in many fields, such as servicing, rescuing, detecting and so on. Researches on biped robots have been developed for many years, and attract more and more interests of scholars. Among the robotic research fields, the walking planning is one of the core technologies [1-2].

Different from wheeled robots, the walking planning of biped robots is a complex model, and many approaches have been implemented to achieve a good walking. In general, walking planning methods of biped robots can be categorized into two areas roughly [3]: non-model based walking and model based walking. Non-model based walking, proposed in many researches, such as [4], [5], [6], and so on, achieves the biped robots walking through the historical walking data and does not calculate by kinematics or dynamics. In non-model based methods, approaches with neural network are usually used for the biped robots' walking [5-6], where the mapping relationship between 3D pose of walking and angular degrees of the actuators is fitted through a lot of offline training of walking data. These methods can achieve a continuous and stable walking effectively. However, the large amount of walking training data are usually hard to be obtained. What's worse, the coverage rate of domain of training data can affect the accuracy of fitted mapping relationship deeply.

Published by Atlantis Press

Contrast to non-model based walking, model based walking, proposed in many researches, such as [3], [7] and so on, is based on the robot kinematics or dynamics. Robot inverse kinematics [8-9] is often applied to calculate the real-time angular degrees of actuators according to the 3D pose data during the model based walking planning instead of fitting in non-model based walking planning. However, the selection of 3D pose data is not introduced in robot inverse kinematics, which causes that if the pose data are not calculated correctly, the walking planning may be broken down. The 3D pose data of walking can be calculated by 3D inverted pendulum model [10-11] with taking the gravity and accelerated speed into consideration. These methods have a better performance in efficiency and stability compared with the neural network. Whereas, these approaches conduct the walking planning with open-loop control, and it may result in a poor adaptability faced with complex terrain. Feedback regulation control of walking planning are introduced in some researches, such as [3], [12], [13] and so on. In these methods, during the walking, a biped robot can perform a real-time correction with the feedback information of environment obtained by sensors. However, the complex model and poor generality exist in these method, which results in that these method may often be only suitable for the specific robot model. Walking planning based on machine learning is popular recently [14-15], where the learning algorithm, such as the reinforcement learning [16-17]. Through machine learning, the walking of biped robots are more flexible and can conduct the adaptive adjustment as the environment changes. Nevertheless, an outstanding walking planning result with machine learning may need a great amount of online training under various scenes, which will be a huge cost on time and energy. Approaches with interpolation also are a popular method in walking planning, and there are many different interpolation functions are used, such as the high-order polynomial method [18], cubic spline interpolation [19], B-spline interpolation [20], and so on. The parameters of interpolation are calculated by key frame data and the continuous pose data are calculated by the interpolation. While, like the inverted pendulum model, simple interpolation is an open-loop control method, the poor adaptability exists in interpolation too.

In this paper, an OW-FI, with a high stability and adaptability, is proposed. The method is based on the Hermite interpolation [21] and adjusts the parameters of Hermite interpolation according to the real-time environmental information. In order to simplify the walking model and achieve a better selection of key frame data, a new separated omni-directional walking method consisting of straight walking, based on the improve Hermite interpolation, and rotation is presented. The mapping relationship from pose data to angular degrees of actuators is then constructed by walking transforming model based on robot kinematics. Finally, a fuzzy approach to walking parameters adjustment is proposed, where the relative position and the acute angle between a robot and a target are regarded as the input of the fuzzy control, for an adaptive and stable closed-loop walking planning. The overall method architecture is improved, resulting in the proposition of this new method architecture.

This paper is organized as follows: Section 2 presents the new separated omni-directional walking method, where the improved Hermite interpolation is introduced. The walking transforming model based on kinematics is presented in Section 3, where the key frame data sent from the interpolation is mapped to the angular degree of actuators. Section 4 presents the adaptive fuzzy parameters adjustment method, where the fuzzy control is used to conduct a feedback control. Simulations and real experiments are given in Section 5 to illustrate the performance of the proposed method. Section 6 presents the conclusion.

2. New Separated Omni-directional Walking of Biped Robot

Considering that the traditional walking planning methods conduct the calculation in the Cartesian space directly, which causes the walking planning model is a little complex, the separated walking model is designed, which consists of straight walking and rotation. The straight walking and rotation are conducted in parallel and the final walking data can be sent to the transforming model to be transformed into the actuators' control data.

2.1. Straight walking based on improved Hermite interpolation

As shown in Fig. 1, x-y-z is called the local coordinate system of a biped robot, whose origin is the center of two biped robot's feet. The calculation of walking planning is based on the local coordinate system.

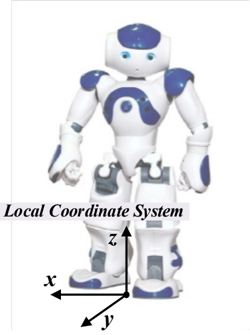


Fig. 1. Local coordinate system of biped robot.

In general, the walking of biped robot can be divided into two phases: the double-leg support phase and the single-leg support phase [22]. In double-leg support phase, the supporting leg changes and a simple pose alignment is conducted. In single-leg support phase, in order to reduce the complexity of planning model, the robot leg are set to keep horizontal and a walking trajectory of single-leg support phase is shown in Fig. 2. In Fig. 2, $2d$ is the robot step length; h is the step height; the solid lines are the key frames of swinging leg; and the dotted lines are the key frames of supporting leg.

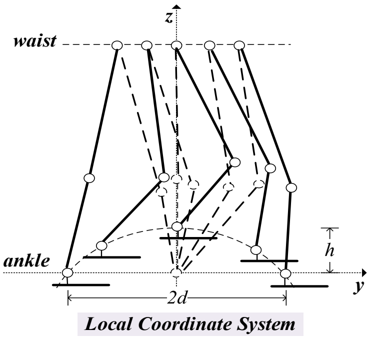


Fig. 2. Diagram of key frames of straight walking.

There are many key frames calculation methods, such as the high-order polynomial method [18], cubic spline interpolation [19], B-spline interpolation [20], and so on. This paper introduces a improve Hermite interpolation, where considering the practical physical model of walking, the condition that the second derivative is zero in traditional Hermite interpolation is changed into that the first derivative is zero, for walking planning. The specific steps of the proposed straight walking planning based on improve Hermite interpolation are introduced as bellow:

Step1. Simplify the walking model.

Since it is a linear motion, the coordinates of the actuators of robot leg on the ***x*** axis are regarded as the same, i.e. the robot leg keeps horizontal. The walking can be simplified as a motion in ***y***-***z*** plane, as shown in Fig. 2.

Step2. Calculate the key frame coordinates.

From Fig. 2, in ***y***-***z*** plane, five waist points $\{(y_i^w, z_i^w) \mid i = 0,1,...,4\}$ and five ankle points $\{(y_i^a, z_i^a) \mid i = 0,1,...,4\}$ of swinging leg are selected. Under the premise of meeting the requirement of zero moment point (ZMP) [4], they can be calculated by,

$$\begin{cases} z_i^w = C_w \\ y_i^w = d(i-1)/4 \\ z_i^a = (2-|i-2|)h/2 \\ y_i^a = d(i-2)/2 \end{cases}, i = 0,1,...,4 \qquad (1)$$

where, C_w is a constant, which means the height of the waist of swinging leg in the single-leg support phase remain the same.

Step3. Improved Hermite interpolation of ankle.

The key frame coordinates are regarded as the input parameters of Hermite interpolation to obtain the smooth moving trajectory functions, as shown in Fig. 2. The interpolations of waist and ankle are the same, here take ankle as an example.

Assuming that the moving trajectory function of ankle in each interval $[y_i^a, y_i^a], i = 0,1,2,3$ is $z^a = I_i(y^a)$. According to the basic constraints of Hermite interpolation:

i. In any interval, $z^a = I_i(y^a)$ and $I(y^a)$ is a polynomial of no more than three orders.

ii. $I(y^a)$ is a continuous function.

iii. The first derivative of $I(y^a)$ is continuous function.

iv. The second derivative of $I(y^a)$ is continuous function.

Meanwhile, the speeds of start and end, in single-leg support phase, are zero. Assuming that $I'(y_i^a) = \xi_i^a, i = 0,1,...,4$, hence, $\xi_0^a = 0$, $\xi_4^a = 0$. The improved Hermite interpolation formula in this paper is shown in (2).

According to the constraint iv, the constraint vector $(\xi_0^a, \xi_1^a, \xi_2^a, \xi_3^a, \xi_4^a)^T$ can be calculated by (3).

where $\mu_i^a = \Delta y_i^a / (\Delta y_i^a + \Delta y_{i+1}^a)$; $\lambda_i^a = 1 - \mu_i^a$;

$\varphi_i^a = 3(\mu_i^a (z_{i+1}^a - z_i^a)/\Delta y_{i+1}^a + \lambda_i^a (z_i^a - z_{i-1}^a)/\Delta y_i^a)$; and $\Delta y_{i+1}^a = y_{i+1}^a - y_i^a$.

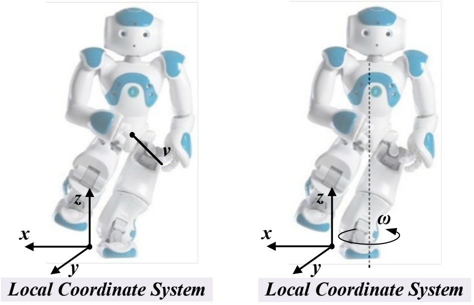
$$\begin{aligned} I_i(y^a) = {} & (1-2\frac{y^a - y_{i-1}^a}{y_{i-1}^a - y_i^a})(\frac{y^a - y_i^a}{y_{i-1}^a - y_i^a})^2 z_{i-1}^a + \\ & (1-2\frac{y^a - y_i^a}{y_i^a - y_{i-1}^a})(\frac{y^a - y_{i-1}^a}{y_i^a - y_{i-1}^a})^2 z_i^a + \\ & (y^a - y_{i-1}^a)(\frac{y^a - y_i^a}{y_{i-1}^a - y_i^a})^2 \xi_{i-1}^a + \\ & (y^a - y_i^a)(\frac{y^a - y_{i-1}^a}{y_i^a - y_{i-1}^a})^2 \xi_i^a \end{aligned} \tag{2}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \lambda_1^a & 2 & u_1^a & 0 & 0 \\ 0 & \lambda_2^a & 2 & \mu_3^a & 0 \\ 0 & 0 & \lambda_3^a & 2 & \mu_3^a \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_0^a \\ \xi_1^a \\ \xi_2^a \\ \xi_3^a \\ \xi_4^a \end{bmatrix} = \begin{bmatrix} 0 \\ \varphi_1^a \\ \varphi_2^a \\ \varphi_3^a \\ 0 \end{bmatrix} \tag{3}$$

The expression of $I(y^a)$ can be calculated by (2) and (3). Hence, the moving trajectory function of ankle of swinging leg can be calculated.

Step4. Repeat **Step3**, the waist trajectory function can also be calculated.

2.2. Simplified omni-directional walking



(a) (b)

Fig. 3. Simplified omni-directional walking model. (a) Straight walking model. (b) Rotation model.

Through analyzing omni-directional walking of biped robots carefully, this paper discovers that the essence of the biped robots' motion is a resultant motion of straight walking and rotation around the *z* axis of local coordinate system, as shown in Fig. 3. Thus, this paper proposed a separated walking model consisting of straight walking and rotation and the two motion are independent. The straight walking can be achieved by improved Hermite interpolation, introduced in Chapter 2.1, as shown in Fig. 3(a). The rotation can be achieved by an angular velocity ω_i^z w.r.t. *z* axis, as shown in Fig. 3(b). At last, the calculated data of two motion are sent to the transforming model.

3. Walking Transforming model based on Kinematics

In order to transform the motion data, sent from the separated omni-traditional walking planning method, into actuators' control data, a walking transforming model based on kinematics is designed. Different from the traditional inverse kinematics, the proposed model considers the straight walking and rotation by kinematics separately. A robot inverse kinematics is applied to transform the motion data, calculated by straight walking, into actuators' control data. Meanwhile, the rotation data only control the actuator, which controls the ankle's rotation of supporting leg around the *z* axis. Finally, the control data obtained from the two motion are combined together and sent to the robot.

3.1. Straight walking transforming model based on robot inverse kinematics

The processing of calculating the angular degrees of actuators of a robot's legs according to poses of waist and ankle is called inverse kinematics. Since the inverse kinematics is relative to the concrete robot model, this paper conduct the inverse kinematics on a NAO robot platform [23-24] for an example.

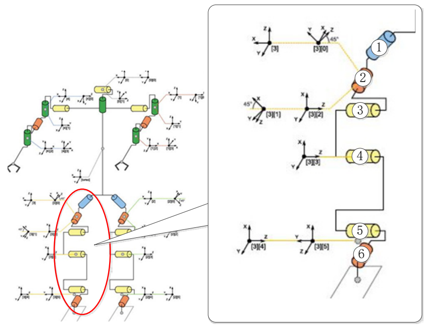


Fig. 4. Diagram of a NAO robot's joint model.

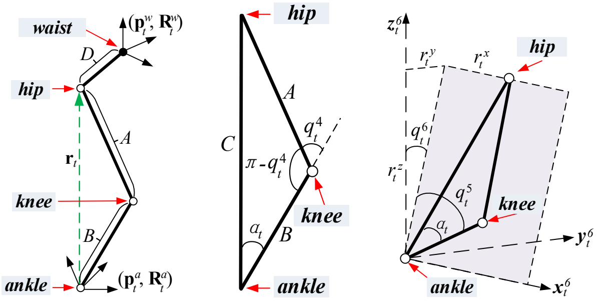
The diagram of a NAO robot's joint model is shown in Fig. 4. In Fig. 4, every leg has six joints, i.e. six actuators; hip has three actuators; ankle has two actuators; and knee has only one actuator. The specific calculation steps of right leg are:

Step1. Calculate real-time pose data.

Since the interpolation function of straight walking is obtained, according to the cycle times of single-leg support phase, the discrete real-time pose data of waist and ankle can be obtained from the function curve. Assuming that now time is *t*, and the corresponding pose data of waist and ankle are $(\mathbf{p}_t^w, \mathbf{R}_t^w)$ and $(\mathbf{p}_t^a, \mathbf{R}_t^a)$, where $\mathbf{p}_t^w$ and $\mathbf{p}_t^a$ are the coordinates in the local coordinate system; $\mathbf{R}_t^w$ and $\mathbf{R}_t^a$ are the orientation vector.

Step2. Calculate the knee's angular degrees.

The space pose diagram is shown in Fig. 5. In Fig. 5, six angular degrees of actuators are $\{q_t^i \mid i = 1,...,6\}$; $\mathbf{r}_t$ is the distance vector from ankle to hip; *D* is the distance between waist and hip; *A* is the distance between hip and knee; *B* is the distance between knee and ankle; *C* is the distance between hip and ankle; α_t is the space angular degree of ankle; x_t^6 - y_t^6 - z_t^6 is the local coordinate system of 6th actuator; and the pose data of six actuators are $\{(\mathbf{p}_t^i, \mathbf{R}_t^i) \mid i = 1,...,6\}$.



(a) (b) (c)

Fig. 5. Diagram of space pose. (a) Space pose of one leg. (b) Space pose of knee. (c) Space pose of ankle.

Consequently,

$$\mathbf{p}_t^3 = \mathbf{p}_t^w + \mathbf{R}_t^w \cdot [0 \quad D \quad 0]^T \tag{4}$$

$$\mathbf{p}_t^6 = \mathbf{p}_t^a \tag{5}$$

$$\mathbf{r}_t = (\mathbf{R}_t^6)^T (\mathbf{p}_t^3 - \mathbf{p}_t^6) \tag{6}$$

r_t^x, r_t^y, and r_t^z are the value of $\mathbf{r}_t$ in three axes. Thus,

$$C = \sqrt{r_t^{x2} + r_t^{y2} + r_t^{z2}} \tag{7}$$

At the same time, according to Fig. 5(b), it can be obtained that,

$$C^2 = A^2 + B^2 - 2AB\cos(\pi - q_t^4) \tag{8}$$

According to (8), q_t^4 can be calculated,

$$q_t^4 = \pi - \arccos((A^2 + B^2 - C^2)/(2AB)) \tag{9}$$

Step3. Calculate the ankle's angular degrees.

From Fig. 5(b), it can be obtained that,

$$C/\sin(\pi - q_t^4) = A/\sin\alpha_t \tag{10}$$

Thus,

$$\alpha_t = -\arcsin(A\sin(\pi - q_t^4)/C) \tag{11}$$

From Fig. 5(c), it can be calculated that,

$$q_t^5 = -\arctan 2(r_t^y, sign(r_t^z)\sqrt{r_t^{x2} + r_t^{z2}}) - \alpha_t \tag{12}$$

$$q_t^6 = \arctan 2(-r_t^x, r_t^z) \tag{13}$$

Step4. Calculate the hip's angular degrees.

The relationship between $\mathbf{R}_t^a$ and $\mathbf{R}_t^w$ are,

$$\mathbf{R}_t^a = \mathbf{R}_t^w \cdot \mathbf{R}_t^1 \cdot \mathbf{R}_t^2 \cdot \mathbf{R}_t^3 \cdot \mathbf{R}_t^4 \cdot \mathbf{R}_t^5 \cdot \mathbf{R}_t^6 \tag{14}$$

Assuming $\mathbf{R}_t = \mathbf{R}_t^1 \cdot \mathbf{R}_t^2 \cdot \mathbf{R}_t^3 \in \Re^{3\times3}$, it can be obtained that,

$$\mathbf{R}_t = (\mathbf{R}_t^w)^T \cdot \mathbf{R}_t^a \cdot (\mathbf{R}_t^6)^T \cdot (\mathbf{R}_t^5)^T \cdot (\mathbf{R}_t^4)^T \tag{15}$$

Then,

$$\mathbf{R}_{11} - \mathbf{R}_{31} = \cos q_t^2 + \sin q_t^2 \tag{16}$$

$$\mathbf{R}_{12} - \mathbf{R}_{32} = -\sin q_t^3 (\cos q_t^2 - \sin q_t^2) \tag{17}$$

$$\sqrt{2}\mathbf{R}_{21} = \sin q_t^1 (\cos q_t^2 - \sin q_t^2) \tag{18}$$

where $\mathbf{R}_{ij}$ is the element of **R** in *i*th row and *j*th clolumn.

Therefore, q_t^1, q_t^2, and q_t^3 can be calculated,

$$q_t^2 = \arcsin(\sqrt{2}(\mathbf{R}_{11} - \mathbf{R}_{31})/2) - \pi/4 \tag{19}$$

$$q_t^1 = \arcsin(\sqrt{2}\mathbf{R}_{21}/(\cos q_t^2 - \sin q_t^2)) \tag{20}$$

$$q_t^3 = \arcsin(-(\mathbf{R}_{12} - \mathbf{R}_{32})/(\cos q_t^2 - \sin q_t^2)) \tag{21}$$

In the same way, the real-time angular degrees of actuators of left leg, assuming they are $\{q_t^i \mid i = 7,...,12\}$, can be calculated.

3.2. Combination of control data

In order to add the rotation into the walking, a combination method of control data is proposed. The real-time desired angular degrees of actuators of a robot's legs, $\{q_t^i \mid i=1,2,...,12\}$, in straight walking, can be calculated according to Chapter 3.1. The real-time current angular degrees of actuators, assuming they are $\{\hat{q}_t^i \mid i=1,2,...,12\}$ can be obtained by sensors as a feedback information. Thus, according to the feedback information, the angular speeds of actuators can be calculated in a closed-loop control mode,

$$\omega_t^i = (q_t^i - \hat{q}_t^i) \big/ \Delta t \, , i = 1,...,12 \qquad (22)$$

where Δt is the cycle time unit.

Then, the rotational speed can be obtained according to Chapter 2.2, with value being ω_t^z and the rotation only controls the actuator, which controls ankle' motion of supporting leg around the z axis, i.e. 6th or 12th actuator. Hence,

$$\begin{cases} {}^*\omega_t^6 = \omega_t^6 + \omega_t^z & \text{, if supporting leg is right leg} \\ {}^*\omega_t^{12} = \omega_t^{12} + \omega_t^{12} & \text{, else} \end{cases} \qquad (23)$$

Finally, the combined angular speeds, $\{\omega_t^1, \omega_t^2, ..., {}^*\omega_t^6, ..., \omega_t^{12}\}$ or $\{\omega_t^1, \omega_t^2, ..., \omega_t^6, ..., {}^*\omega_t^{12}\}$, are sent to a biped robot for achieving the omni-directional walking.

For a more clear understanding to the separated omni-directional walking, the frame of proposed method is shown in Fig. 6. According to Fig. 6, a completed separated omni-directional walking can be constructed, and it has three parameters: step length d, step height h, and rotational speed ω_t^z.

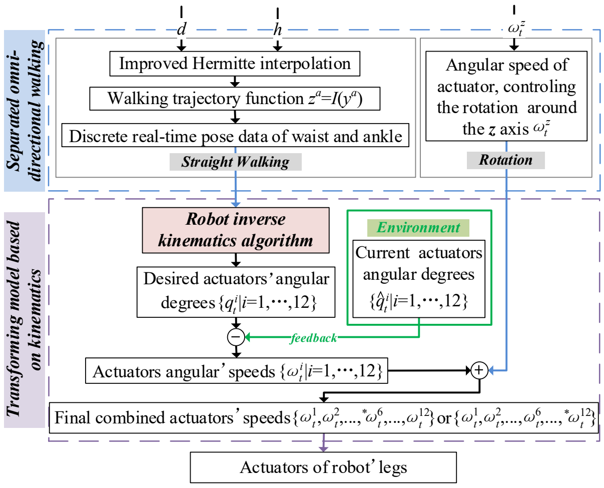

d
h
ω_t^z
Separated omni-directional walking
Improved Hermitte interpolation
Walking trajectory function $z^a=I(y^a)$
Discrete real-time pose data of waist and ankle
Straight Walking
Angular speed of actuator, controling the rotation around the z axis ω_t^z
Rotation
Transforming model based on kinematics
Robot inverse kinematics algorithm
Desired actuators' angular degrees $\{q_t^i|i=1,\cdots,12\}$
Environment
Current actuators angular degrees $\{\hat{q}_t^i|i=1,\cdots,12\}$
feedback
Actuators angular' speeds $\{\omega_t^i|i=1,\cdots,12\}$
Final combined actuators' speeds $\{\omega_t^1,\omega_t^2,...,{}^*\omega_t^6,...,\omega_t^{12}\}$ or $\{\omega_t^1,\omega_t^2,...,\omega_t^6,...,{}^*\omega_t^{12}\}$
Actuators of robot' legs

Fig. 6. Frame of separated omni-directional walking method.

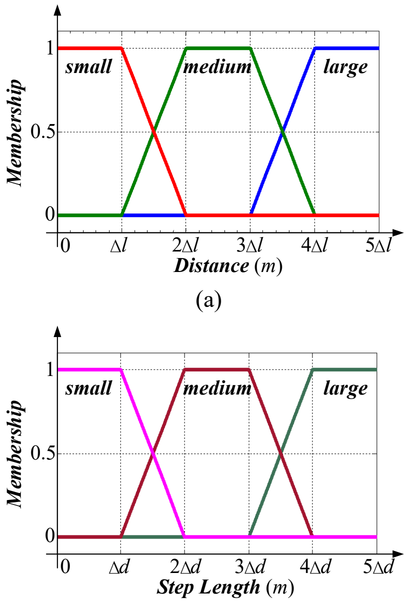
4. Adaptive Fuzzy Parameters Adjustment Method

During the robot walking, in order to keep the stability of walking, an adaptive parameters of proposed omni-directional walking should be made. Simply speaking, since the prosed method has three parameters, when a robot is closed to a target, it should walk in a low speed, i.e. a small step length and small step height. On the contrary, when the distance from a robot to a target, the robot should walk in a high speed. The same situation can be found in rotation. In order to conduce the dynamic control, some approaches are introduced: A second speed regulation optimization method is proposed in [25] for a walking speed control where two speed levels are set in the robot system and the walking speed changes between two levels according to the environment information. Whereas, the control result may be not a continuous speed function, which may cause a poor stability of robot system. Proportional-Integral-Derivative (PID) control is introduced in [26-27], and the parameters are controlled by a PID controller. While, it cannot be a genetic approach to all, even similar, tasks, because the parameters of PID controller are quite circumstantial.

This paper, combining the thought of fuzzy control [28-29], designs an adaptive fuzzy parameters adjustment method. At the same time, considering the proposed omni-directional walking planning method, the fuzzy control is divided into two independent parts: the adjustment of straight walking and the adjustment of rotation. In straight walking adjustment part, the relative distance between a robot and a target is regarded as the input of fuzzy control and the step length and height are the output. In rotation walking adjustment part, the relative acute angle between a robot and target is regarded as the input and the rotational speed is the output. Especially, the relative distance and acute angle can be obtained by some methods, such as image processing methods [30-33], localization techniques [34-36], mapping techniques [37-38], noise reducing method [39-40] and so on.

4.1. Adaptive Fuzzy adjustment of straight walking

The fuzzy description of relative distance between a robot and a target is set as "small, medium, large". In the same way, the step length is described as "small, medium, large" fuzzily. The corresponding membership function are shown in Fig. 7.



(a)

(b)

Fig. 7. Diagram of Membership function of straight walking. (a) Distance/input membership function. (b) Step length/output membership function.

In Fig. 7, Δl and Δd are constants, and their values are set according to the real environment. The corresponding membership functions are as bellow.

$$\mu_{small}^{l}(l)=\begin{cases}0 & ,l>2\Delta l\\ 2-l/\Delta l & ,\Delta l<l\le 2\Delta l\\ 1 & ,l\le\Delta l\end{cases} \tag{24}$$

$$\mu_{medium}^{l}(l)=\begin{cases}0 & ,l>4\Delta l\\ 4-l/\Delta l & ,3\Delta l<l\le 4\Delta l\\ 1 & ,2\Delta l<l\le 3\Delta l\\ l/\Delta l-1 & ,\Delta l<l\le 2\Delta l\\ 0 & ,l\le\Delta l\end{cases} \tag{25}$$

$$\mu_{large}^{l}(l)=\begin{cases}1 & ,l>4\Delta l\\ l/\Delta l-3 & ,3\Delta l<l\le 4\Delta l\\ 0 & ,l\le 3\Delta l\end{cases} \tag{26}$$

where $\mu_{small}^{l}(l)$, $\mu_{medium}^{l}(l)$, and $\mu_{large}^{l}(l)$ are the member functions relative to the fuzzy description "small, medium, large" of relative distance, i.e. the input.

$$\mu_{small}^{d}(d)=\begin{cases}0 & ,d>2\Delta d\\ 2-d/\Delta d & ,\Delta d<d\le 2\Delta d\\ 1 & ,d\le\Delta d\end{cases} \tag{27}$$

$$\mu_{medium}^{d}(d)=\begin{cases}0 & ,d>4\Delta d\\ 4-d/\Delta d & ,3\Delta d<d\le 4\Delta d\\ 1 & ,2\Delta d<d\le 3\Delta d\\ d/\Delta d-1 & ,\Delta d<d\le 2\Delta d\\ 0 & ,d\le\Delta d\end{cases} \tag{28}$$

$$\mu_{large}^{d}(d)=\begin{cases}1 & ,d>4\Delta d\\ d/\Delta d-3 & ,3\Delta d<d\le 4\Delta d\\ 0 & ,d\le 3\Delta d\end{cases} \tag{29}$$

where $\mu_{small}^{d}(d)$, $\mu_{medium}^{d}(d)$, and $\mu_{large}^{d}(d)$ are the member functions relative to the fuzzy description "small, medium, large" of step length, i.e. the output.

N^s discrete points $\mathbf{I}^s=\{I_i^s\,|\,i=1,2,...,N^s\}$ are selected from the input domain of straight walking uniformly. Every point has three membership value to the three fuzzy description of input. Hence, the input membership discrete matrix $\mathbf{IM}^s=\{IM_{ji}^s\}\in\Re^{3\times N^s}$ of straight walking can be calculated by,

$$\begin{cases}IM_{1i}^s=\mu_{small}^{l}(I_i^s)\\ IM_{2i}^s=\mu_{medium}^{l}(I_i^s)\\ IM_{3i}^s=\mu_{large}^{l}(I_i^s)\end{cases} \tag{30}$$

Similarly, M^s discrete points $\mathbf{O}^s=\{O_i^s\,|\,i=1,2,...,M^s\}$ are selected from the output domain of straight walking uniformly. The corresponding output membership discrete matrix $\mathbf{OM}^s=\{OM_{ji}^s\}\in\Re^{3\times M^s}$ can be expressed as,

$$\begin{cases}OM_{1i}^s=\mu_{small}^{d}(O_i^s)\\ OM_{2i}^s=\mu_{medium}^{d}(O_i^s)\\ OM_{3i}^s=\mu_{large}^{d}(O_i^s)\end{cases} \tag{31}$$

The fuzzy rules is appointed as "If distance is large, step length is large; if distance is medium, step length is medium; if distance is small, step length is small". According to the max-min compose operation, the fuzzy inference engine $\mathbf{R}^s=\{r_{ij}^s\}\in\Re^{N^s\times M^s}$ can be obtained by,

$$r_{ij}^s=\mathop{\vee}_{k=1}^{3}(IM_{ki}^s\wedge OM_{ki}^s),i=1,...,N^s,j=1,...,M^s \tag{32}$$

where $\vee$ is the operation to find the maximal value; $\wedge$ is the operation to find the minimal value.

When a distance value l_0 is measured, it should be blurred into the fuzzy input vector $\mathbf{B}^s = \{b_i^s \mid i = 1,2,...,N^s\}$, according to the N^s discrete points $\mathbf{I}^s$ of input. In this paper, a PID-inspired blurring method is designed and the blurring function is,

$$b_i^s = \begin{cases} 0 & ,|l_0 - I_i^s| \ge \varepsilon^s \\ (\varepsilon^s - |l_0 - I_i^s|)/\varepsilon^s & ,|l_0 - I_i^s| < \varepsilon^s \end{cases} \qquad (33)$$

where ε^s, which is a constant, is the range of maximal allowable fuzzy distance.

According to the max-min compose operation of fuzzy input vector $\mathbf{B}^s$ and fuzzy inference engine $\mathbf{R}^s$, the calculation formula of corresponding fuzzy output vector $\mathbf{G}^s = \{g_j^s \mid j = 1,2,...,M^s\}$ is,

$$g_i^s = \mathop{\vee}_{k=1}^{N^s} (b_k^s \wedge r_{kj}^s), j = 1,2,...,M^s \qquad (34)$$

The output value, relative to l_0, can be calculated by the weighted average method for de-fuzzy.

$$sl_0 = \sum_{j=1}^{M^s} (OM_j^s \times g_j^s \Big/ \sum_{i=1}^{M^s} g_i^s) \qquad (35)$$

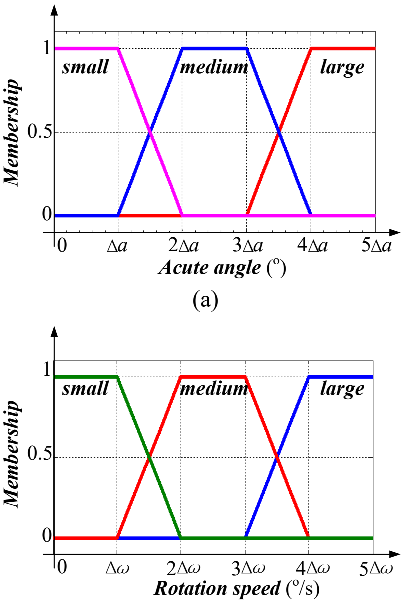
Finally, $d_0 = sl_0/2$. Meanwhile, to reduce the computational complexity, this paper regards the step height *h* has a proportional relationship with step length *d*. The proportional function is,

$$h = d \times h_{\max} / d_{\max} \qquad (36)$$

where $d_{\max}$ is the maximal step length and $h_{\max}$ is the maximal step height. Hence, $h_0 = d_0 \times h_{\max} / d_{\max}$. The values of d_0 and h_0 will be sent to the separated omni-directional walking for the parameter values of *d* and *h* in Fig. 6.

4.2. Adaptive Fuzzy adjustment of rotation

Like straight walking, in adaptive fuzzy adjustment method of rotation, the relative acute angle between a robot and a target is selected as the input, the rotational speed is selected as the output. The fuzzy description of input is "small, medium, large", and the rotational speed, i.e. the output is also described as "small, medium, large" fuzzily. The fuzzy rules are the similar that "If acute angle is large, rotational speed is large; if acute angle is medium, rotational speed is medium; if acute angle is small, rotational speed is small". The diagram of membership function of rotation is shown in Fig. 8.



(a)

(b)

Fig. 8. Diagram of Membership function of rotation. (a) Acute angle/input membership function. (b) Rotational speed/output membership function.

In Fig. 8, Δa and $\Delta\omega$ are constants, and their values are set according to the real environment. The corresponding membership functions are as bellow.

$$\mu_{small}^a(a) = \begin{cases} 0 & ,a > 2\Delta a \\ 2 - a/\Delta a & ,\Delta a < a \le 2\Delta a \\ 1 & ,a \le \Delta a \end{cases} \qquad (37)$$

$$\mu_{medium}^a(a) = \begin{cases} 0 & ,a > 4\Delta a \\ 4 - a/\Delta a & ,3\Delta a < a \le 4\Delta a \\ 1 & ,2\Delta a < a \le 3\Delta a \\ a/\Delta a - 1 & ,\Delta a < a \le 2\Delta a \\ 0 & ,a \le \Delta a \end{cases} \qquad (38)$$

$$\mu_{large}^a(a) = \begin{cases} 1 & ,a > 4\Delta a \\ a/\Delta a - 3 & ,3\Delta a < a \le 4\Delta a \\ 0 & ,l \le 3\Delta a \end{cases} \qquad (39)$$

$$\mu_{small}^\omega(\omega) = \begin{cases} 0 & ,\omega > 2\Delta\omega \\ 2 - \omega/\Delta\omega & ,\Delta\omega < \omega \le 2\Delta\omega \\ 1 & ,\omega \le \Delta\omega \end{cases} \qquad (40)$$

$$\mu^{\omega}_{medium}(\omega)=\begin{cases}0 & ,\omega>4\Delta\omega\\ 4-\omega/\Delta\omega & ,3\Delta\omega<\omega\leq 4\Delta\omega\\ 1 & ,2\Delta\omega<\omega\leq 3\Delta\omega\\ \omega/\Delta\omega-1 & ,\Delta\omega<\omega\leq 2\Delta\omega\\ 0 & ,\omega\leq\Delta\omega\end{cases}\quad(41)$$

$$\mu^{\omega}_{large}(\omega)=\begin{cases}1 & ,\omega>4\Delta\omega\\ \omega/\Delta\omega-3 & ,3\Delta\omega<d\leq 4\Delta\omega\\ 0 & ,\omega\leq 3\Delta\omega\end{cases}\quad(42)$$

where $\mu^{a}_{small}(a)$, $\mu^{a}_{medium}(a)$, $\mu^{a}_{large}(a)$, $\mu^{\omega}_{small}(\omega)$, $\mu^{\omega}_{medium}(\omega)$, and $\mu^{\omega}_{large}(\omega)$ are the member functions of input and output.

Meanwhile, N^r discrete points $\mathbf{I}^r=\{I^r_i\mid i=1,2,...,N^r\}$ in the domain of input and M^r discrete points $\mathbf{O}^r=\{O^r_i\mid i=1,2,...,M^r\}$ in the domain of output are selected. Then, the input membership discrete matrix $\mathbf{IM}^r=\{IM^r_{ji}\}\in\Re^{3\times N^r}$, the output membership discrete matrix $\mathbf{OM}^r=\{OM^r_{ji}\}\in\Re^{3\times M^r}$, and the fuzzy inference engine $\mathbf{R}^r=\{r^r_{ij}\}\in\Re^{N^r\times M^r}$ can be obtained according to (30)-(32).

In the blurring phase, the measured value of angle a_0 are blurred into $\mathbf{B}^r=\{b^r_i\mid i=1,2,...,N^r\}$ according to (33) with the ε^r being the range of maximal allowable fuzzy distance. After the fuzzy output vector $\mathbf{G}^r=\{g^r_j\mid j=1,2,...,M^r\}$ is obtained according to (34), the weighted average value of corresponding rotational speed is rs_o according to (35).

In particular, if the position of target is in the first or third quadrants of ***x-y*** plane, which is belong to the local coordinate system of biped robot, the final rotational speed parameter value should be $\omega_0=-rs_0$. Otherwise, $\omega_0=rs_0$. The values of ω_0 will be sent to the separated omni-directional walking for the parameter values of ω^z_t in Fig. 6.

4.3. Frame of OW-FI

According to the above description, the proposed OW-FI is completed, and its frame is shown in Fig. 9.

In fuzzy control phase, called FCP, the environment feedback information is used for fuzzy control and the final fuzzy parameters, including the step length, step height, and the angular speed, are obtained.

In omni-directional walking planning phase, called WPP, the fuzzy parameters are used in the straight walking and the rotation.

In transforming model based on kinematics, the calculated improved Hermite interpolation curve of straight walking and rotational speeds are combined together to obtain the final angular speeds of actuators by kinematics.

Finally, the angular speeds are sent to the robot to execute.

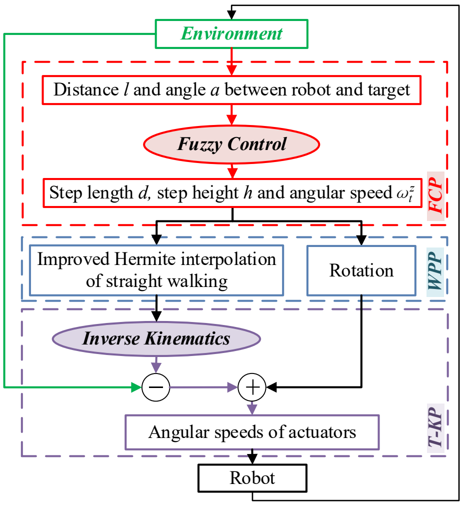
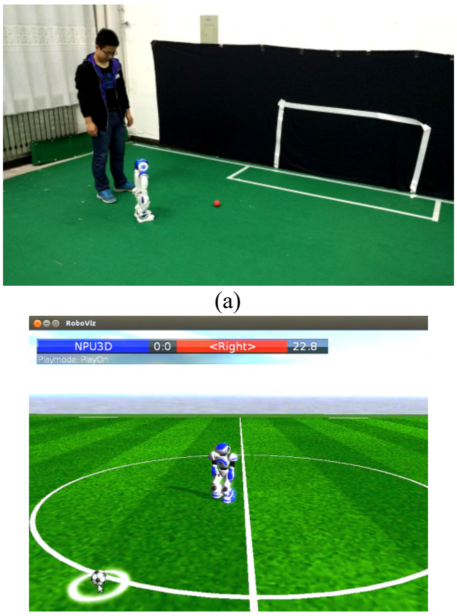

Environment
Distance l and angle a between robot and target
Fuzzy Control
Step length d, step height h and angular speed ω^z_t
FCP
Improved Hermite interpolation of straight walking
Rotation
WPP
Inverse Kinematics
−
+
Angular speeds of actuators
T-KP
Robot

Fig. 9. Frame of proposed OW-FI.

5. Simulations and Experiments

In the experiments, walking test is conducted to demonstrate the practicality of proposed separated omni-directional walking method. Then, the proposed OW-FI, PID control [26-27], where the proportional control is used, and the sectional control [25] are compared in the Matlab, where the control output results are compared, real environment, where a red ball is selected as the target for tracking comparison, and the simulation platform, where a large amount of dribbling ball tests are implemented, to prove the adaptability, efficiency, and stability of proposed OW-FI.

The experiments are implemented on the Matlab, a real NAO robot [23-24] and the simulation platform, RoboCup3D simulation platform [41], as shown in Fig. 10. Fig. 10(a) is the real experimental environment, which is a $10m\times 10m$ space with a green carpet, and the NAO robot is tested on the green carpet for tracking the red ball. Fig. 10(b) is the RoboCup3D simulation platform, where a NAO robot, which is the same as real NAO is built in it, and NAO robot can obtain the ball position through image processing. During the experiments and simulations, the real-time ball positions, robot positions, and other information are recorded to conduct the comparison. At the same time, all the comparisons are under the same conditions.



(a)

(b)

Fig. 10. Experimental system. (a) Real environment. (b) RoboCup3D simulation platform.

5.1. Walking test

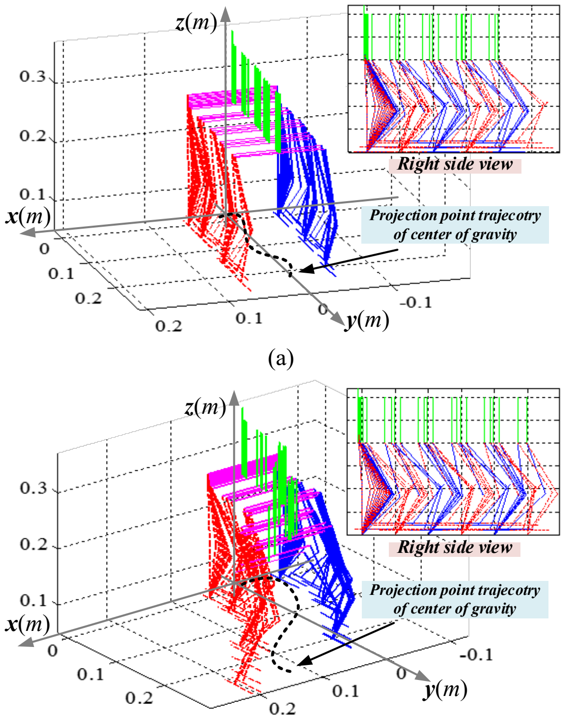
To illustrate the practicality of proposed separated omni-directional walking, the experiments are implement in the Matlab and a real NAO robot. The walking parameters are set as constants and their values are shown in Table 1.

Table 1. Omni-directional walking parameters.

Identified Parameters	Value
d	0.075m
h	0.08m
C_w	0.27m
ω_t^z	6 °/s
Δt	0.02s

In the Matlab, the planned leg trajectories are calculated by kinematics, as shown in Fig. 11. In Fig. 11, the red lines are the right leg trajectories; blue lines are the left leg trajectories; green lines are the waist trajectories; black dotted lines are the projection point trajectories of center of gravity.

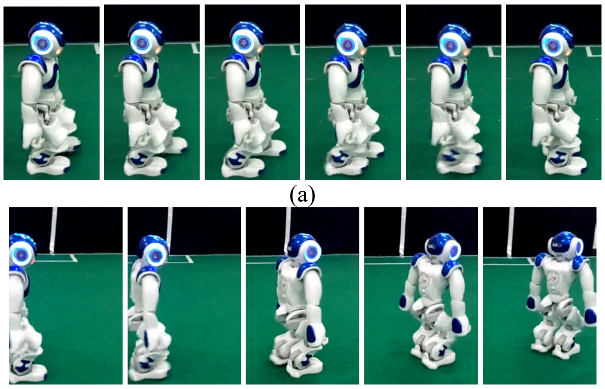
From Fig. 11, the projection point trajectories of center of a robot's gravity are meeting the requirement of ZMP after analysis, and it can be obtained from the simulation results that the separated omni-directional walking can be achieved. Then, the real NAO robot is used for walking test, as show in Fig. 12. In Fig. 12(a), the robot is tested for the straight walking and in Fig. 12(b), the robot is tested for the rotation.



(a)

(b)

Fig. 11. Walking simulation results in Matlab. (a) Simulation result of straight walking. (b) Simulation of rotation.



(a)

(b)

Fig. 12. Real walking test. (a) Straight walking test. (b) Rotation test.

Published by Atlantis Press

Through testing, the average speed of straight walking can be obtained and its value is 0.413*m*/*s*, which means the moving speed of NAO robot is fast. The rotational speed is 5.617°/*s*, which is much closed to the setting value, 6 °/*s*. According to the experimental results, it can be obtained that the separated omni-directional walking has a high practicality for the robots' walking and simplifies the walking model.

5.2. Fuzzy control test

The high walking speed may reduce the stability of a robot, so, the fuzzy control thought is used for the walking speed control. The parameters of fuzzy control are shown in Table 2.

Table 2. Parameters of fuzzy control.

Identified Parameters	Value
Δl	0.5m
Δd	0.03m
Δa	16°
$\Delta \omega$	1.2°/s
N^s	10
M^s	10
N^r	10
M^r	10
$d_{\max}$	0.075m
$h_{\max}$	0.08m
ε^s	2.5m
ε^r	90°

From the Table 2, the input membership discrete matrix of straight walking $\mathbf{IM}^s$ and the output membership discrete matrix of straight walking $\mathbf{OM}^s$ can be obtained by (30) and (31), as shown in Table 3 and Table 4.

Table 3. Input membership discrete matrix of straight walking.

$10^{-2}m$	0	25	50	75	100	125	150	175	200	225	250
Small	1	1	1	0.5	0	0	0	0	0	0	0
Medium	0	0	0	0.5	1	1	1	0.5	0	0	0
Large	0	0	0	0	0	0	0	0.5	1	1	1

Table 4. Output membership discrete matrix of straight walking.

$10^{-3}m$	0	15	30	45	60	75	90	105	120	135	150
Small	1	1	1	0.5	0	0	0	0	0	0	0
Medium	0	0	0	0.5	1	1	1	0.5	0	0	0
Large	0	0	0	0	0	0	0	0.5	1	1	1

From Table 3 and Table 4, the fuzzy engine $\mathbf{R}^s$ of straight walking can be obtained by (32). Finally, the fuzzy results can be calculated by (33)-(36). A linear growth distance values are set as the input and the output results under fuzzy control are shown in Fig. 13.

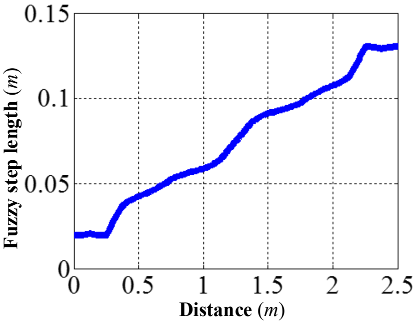


Fig. 13. Step length results under fuzzy control.

In the same way, a linear growth angle values are set as the input and the output results under fuzzy control are shown in Fig. 14.

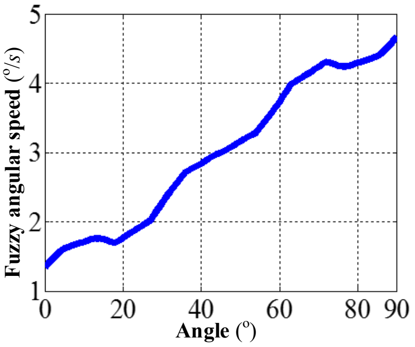


Fig. 14. Angular speed results under fuzzy control.

It can be obtained from Fig. 13 and Fig. 14 that linear inputs become smooth curve output through fuzzy control, which has certain adaptability for the objects with complex motion models, such as robots, and could improve the robustness and stability of the system.

5.3. Target tracking comparison

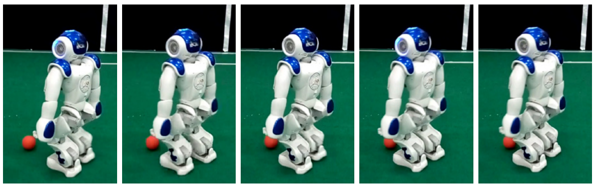


Fig. 15. Ball tracking photo sequence in real environment.

In the target tracking test, as show in Fig. 10(a), red rolls, controlled by a person, and a NAO robot can get the distance information through image processing and then the real-time fuzzy control is conducted. A ball tracking photo sequence is shown in Fig. 15. To a further proving for the practicality and efficiency of proposed OW-FI, the propose method is compared with PID control [26-27] and the sectional control [25] under the same conditions and the same ball motion trajectories. Especially, every methods are test for 30 times and the final average tracking trajectories of three methods are show in Fig. 16 and Fig. 17. Fig. 16 is the average tracking trajectories of three methods and in Fig. 16, the red lines are the ball motion trajectories and blue lines are the robot motion trajectories. Fig. 17 is the change of average distance between the robot and the red ball, i.e. the distance error, as the time goes. The positions of ball and robot are recorded every 1*s*. In Fig. 17, the red line is the distance error change of OW-FI; the blue line is the PID control's; and the green line is the sectional control's.

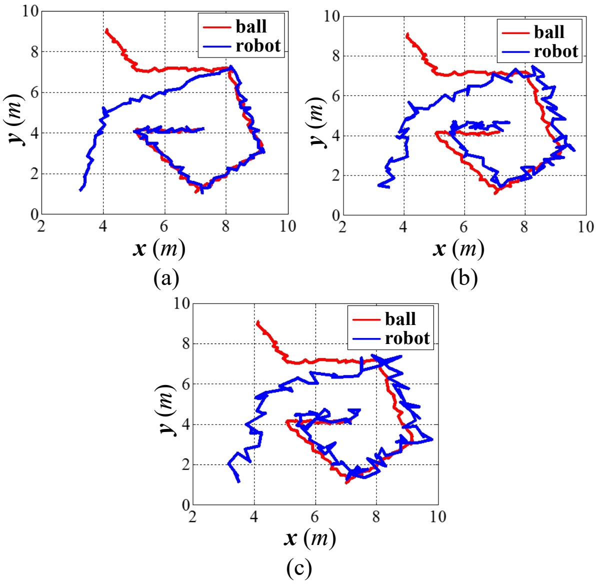


Fig. 16. Target tracking comparison of three methods. (a) Average tracking trajectory of OW-FI. (b) Average tracking trajectory of PID control. (c) Average tracking trajectory of sectional control.

From Fig. 16 and Fig. 17, with the OW-FI, the NAO robot catches up with the ball after about 27 time slices; with the PID control, it costs about 42 time slices; with the sectional control, it costs about 57 time slices. And when the robot dribbles the ball, after catches up with the ball, the stable distance error is 0-0.3*m* with OW-FI. With PID control and sectional control, the stable distance error ranges are 0-0.7*m* and 0-0.8*m*. The above data means that the OW-FI can track the target more closely than other methods. Hence, it can be obtained that the OW-FI is more suitable for robots than the competing methods and better practicality and efficiency it has.

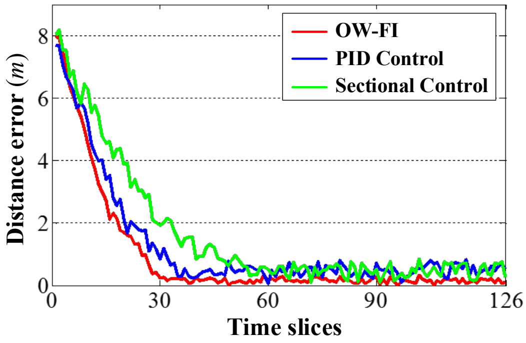


Fig. 17 Convergence rate comparison of three methods in target tracking.

5.4. Stability comparison

In order for a further proof in the stability of OW-FI. The three walking control methods, proposed OW-FI, PID control, and sectional control, are tested in the RoboCup3D simulation platform for the large amount of dribbling ball comparisons. The testing environment is shown in Fig. 10(b) and robot can find the real-time ball position through image processing. A dribbling ball photo sequence in Robocup3D simulation platform is shown in Fig. 18.

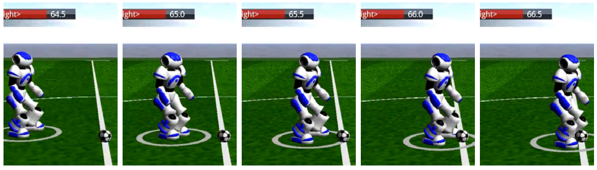


Fig. 18. Ball tracking photo sequence in simulation.

Table 5. Comparison of dribbling ball test.

	Average falling	Average shooting
OW-FI	3.5	6.77
PID control	13.33	3.12
Sectional control	15.81	2.75

As shown in Fig. 18, every method is tested for 100 times and every time takes 10 minutes. At the same time, the average falling times during dribbling ball, the average shooting times of three method in every time are recorded as shown in Table 5. Especially, the average falling times reflects the stability of the control method, and the average shooting times reflects the average dribbling ball speed.

From the Table 5, the average falling times with OW-FI is 3.5 times, which is less than the PID control's result, 13.33 times, and the Sectional control's result, 15.81 times. Hence, the high stability of proposed OW-FI can be proved. According to the average shooting times comparison, it can be obtained that the dribbling ball speed of OW-FI is the most fast, the reason is that the OW-FI can reduce the action switching times with the fuzzy control.

6. Conclusion

Facing the problems of omni-directional walking for biped robots, such as the complex model, low stability, low flexibility and so on, an adaptive OW-FI method, where a closed-loop control is conducted, is proposed to tackle these problems. This paper introduces a new separated omni-directional walking method, consisting of straight walking and rotation, to simplify the walking model. Real-time pose data can be obtained from the omni-directional walking function, and then transformed into the angular speed of legs' actuators by kinematics. By the adaptive fuzzy parameters adjustment, the walking parameters change according to the environment feedback information to improve the stability and flexibility of the robot system. The experiment results demonstrate the proposed OW-FI has better properties in stability and adaptability than the competing methods. This method can play a certain role in the development of biped robot gait planning.

Acknowledgement

The research work was supported by National Natural Science Foundation of China(No.61373120), National Natural Science Foundation of Shaanxi(No.2015JM6308), The fund of national ministries(No.9140A15090115HK03216), The Fundamental Research Funds for the Central Universities (No.3102015JSJ0005). The National Students' Innovation Project (No. 201510699071).

References

1. Sato T, Sakaino S, Ohashi E, et al, Walking trajectory planning on stairs using virtual slope for biped robots, *IEEE Transactions on Industrial Electronics*. 58(4) (2011) 1385-1396.
2. Dehghani R, Fattah A and Abedi E, Cyclic gait planning and control of a five-link biped robot with four actuators during single support and double support phases, *Multibody System Dynamics*. 33(4) (2013) 389-411.
3. Wang S, Hu M, Shi H, et al, Humanoid robot's omnidirectional walking, In *2015 IEEE International Conference on Information and Automation (ICIA)* (Lijiang, China, 2015), pp. 381-385.
4. Kim D W, Kim N H and Park G T, ZMP based neural network inspired humanoid robot control, *Nonlinear Dynamics*. 67(1) (2012) 793-806.
5. Juang C F, Chang Y C and Hsiao C M, Evolving gaits of a hexapod robot by recurrent neural networks with symbiotic species-based particle swarm optimization, *IEEE Transactions on Industrial Electronics*. 58(7) (2011) 3110-3119.
6. Kumar M R, Lathan L S and Vundavilli P R, Dynamically balanced obstacle crossing gait generation of a biped robot using neural networks, *International Journal of Mechanisms and Robotic Systems*. 2(3) (2015) 232-253.
7. Shin H K and Kim B K, Energy-efficient gait planning and control for biped robots utilizing the allowable ZMP region, *IEEE Transactions on Robotics*. 30(4) (2014) 986-993.
8. Ames A D, Human-inspired control of bipedal walking robots, *IEEE Transactions on Automatic Control*. 59(5) (2014) 1115-1130.
9. Liu C, Wang D and Chen Q, Central pattern generator inspired control for adaptive walking of biped robots, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 43(5) (2013) 1206-1215.
10. Zijlstra A and Zijlstra W, Trunk-acceleration based assessment of gait parameters in older persons: A comparison of reliability and validity of four inverted pendulum based estimations, *Gait & posture*. 38(4) (2013) 940-944.
11. Ali F, Shukor A Z H, Miskon M F, et al, 3-D biped robot walking along slope with dual length linear inverted pendulum method (DLLIPM), *International Journal of Advanced Robotic Systems*. 10 (2013).
12. Ramezani A, Hurst J W, Hamed K A, et al, Performance analysis and feedback control of ATRIAS, a three-dimensional bipedal robot, *Journal of Dynamic Systems, Measurement, and Control*. 136(2) (2014).
13. Grizzle J W, Chevallereau C, Sinnet R W, et al, Models, feedback control, and open problems of 3D bipedal robotic walking, *Automatica*. 50(8) (2014) 1955-1988.
14. Hwang K S, Lin J L and Yeh K H, Learning to Adjust and Refine Gait Patterns for a Biped Robot, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 45(12) (2015) 1481-1490.
15. Wang S, Chaovalitwongse W and Babuška R, Machine learning algorithms in bipedal robot control, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*. 42(5) (2012) 728-743.
16. Kuremoto T, Otani T, Mabu S, et al, One-D-R-A-G-SOM

and its Application to a Hand Shape Instruction Learning System, *International Journal of Networked and Distributed Computing*. 2(3) (2014) 166-174.

17. Uğurlu Y. Utilizing Human-Computer Interaction Data to Extract User Interests from Web-based Learning Systems, *International Journal of Networked and Distributed Computing*. 1(4) (2013) 187-195.
18. Gong C, Travers M J, Astley H C, et al, Kinematic gait synthesis for snake robots, *The International Journal of Robotics Research*. 35(1) (2016) 100-113.
19. Hong Y D and Kim J H, 3-D command state-based modifiable bipedal walking on uneven terrain, *IEEE/ASME Transactions on Mechatronics*. 18(2) (2013) 657-663.
20. Kwon H J, Xiang Y, Bhatt R, et al, Backward walking simulation of humans using optimization, *Structural and Multidisciplinary Optimization*. 50(1) (2014) 169-179.
21. Bao J, Huang Q, Wang X H, et al, A New Wideband Mutual Coupling Compensation Method for Adaptive Arrays Based on Cubic Hermite Interpolation, *Progress In Electromagnetics Research M*. 44 (2015) 161-170.
22. Li X, Shi H and Li R, An improved analytical method of inverse kinematics of a humanoid robot, In *Future Computer and Information Technology* (Tianjin, China, 2013), pp. 563-571.
23. Jovanović M Š D, Vujović V V, Rodić A, et al, KINEMATIC MODEL OF NAO HUMANOID ROBOT, *Robotica & Management*. 19(1) (2014).
24. Kofinas N, Orfanoudakis E and Lagoudakis M G, Complete analytical forward and inverse kinematics for the NAO humanoid robot, *Journal of Intelligent & Robotic Systems*. 77(2) (2015) 251-264.
25. Powell M J, Hereid A and Ames A D, Speed regulation in 3D robotic walking through motion transitions between human-inspired partial hybrid zero dynamics, In *2013 IEEE International Conference on Robotics and Automation (ICRA)* (2013), pp. 4803-4810.
26. Chevalier A, Ionescu C M and De Keyser R. Model-based vs auto-tuning design of PID controller for knee flexion during gait, In *2014 IEEE International Conference on Systems, Man and Cybernetics (SMC)* (2014), pp. 3878-3883.
27. Wang F C, Yu C H and Chen C H, Design and implementation of multivariable robust PID control for an active gait trainer, In *Proceedings of 2012 SICE Annual Conference (SICE)* (2012), pp. 2226-2231.
28. Li H, Liu H, Gao H, et al, Reliable fuzzy control for active suspension systems with actuator delay and fault, *IEEE Transactions on Fuzzy Systems*. 20(2) (2012) 342-357.
29. Precup R E and Hellendoorn H, A survey on industrial applications of fuzzy control, *Computers in Industry*. 62(3) (2011) 213-226.
30. Hayashi T, Ooi T and Sasaki M, Contour Completion of Partly Occluded Objects Based on Figural Goodness, *International Journal of Networked and Distributed Computing*. 3(3) (2015) 185-192.
31. Wu C F, Lin C J and Lee C Y, Applying a functional neurofuzzy network to real-time lane detection and front-vehicle distance measurement, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*. 42(4) (2012) 577-589.
32. Yamanishi R, Fujimoto R, Iwahori Y, et al, Hybrid Approach of Ontology and Image Clustering for Automatic Generation of Hierarchic Image Database, *International Journal of Networked and Distributed Computing*. 3(4) (2015) 234-242.
33. Jameel T and Lin M, Test Image Generation using Segmental Symbolic Evaluation, *International Journal of Networked and Distributed Computing*. 2(3) (2014) 135-147.
34. Gu F, He Y and Han J, Active Persistent Localization of a Three-Dimensional Moving Target Under Set-Membership Uncertainty Description Through Cooperation of Multiple Mobile Robots, *IEEE Transactions on Industrial Electronics*. 62(8) (2015) 4958-4971.
35. Anbao W and Bin Z. Realize Node Localization Based on OLSR Protocol in Ad Hoc Networks, *International Journal of Networked and Distributed Computing*. 1(1) (2013) 61-71.
36. Hoang-Van H, Mizutani K, Miyoshi T, et al, P2P Traffic Localization by Forcing Packet Loss, *International Journal of Networked and Distributed Computing*. 1(4) (2013) 251-259.
37. Endres F, Hess J, Sturm J, et al, 3-D mapping with an RGB-D camera, *IEEE Transactions on Robotics*. 30(1) (2014) 177-187.
38. Ishii N, Torii I, Nakashima T, et al, Generation and Mapping of Multi-Reducts Based on Nearest Neighbor Relation, *International Journal of Networked and Distributed Computing*. 2(1) (2014) 1-10.
39. Otsuka T and Ito T, Flexible WSNs Aims Easy Installation With Noise Reduce Method For Elderly People Care, *International Journal of Networked and Distributed Computing*. 1(3) (2013) 174-186.
40. Karami A, Heylen R and Scheunders P. Band-Specific Shearlet-Based Hyperspectral Image Noise Reduction, *IEEE Transactions on Geoscience and Remote Sensing*. 53(9) (2015) 5054-5066.
41. Fan B and Liang Z, Omnidirectional kick in RoboCup3D simulation, In *2014 IEEE International Conference on Mechatronics and Automation (ICMA)* (2014), pp. 1058-1062.