Iterative Learning Control for Biped Walking Robot with Varying Iteration Lengths*

Zhang Wei, Wang Lanjing, Shen Dong[1]

College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, P. R. China
E-mail: 123-ymxc@163.com, 42912366@qq.com, shendong@buct.edu.cn

Abstract: The iterative learning control (ILC) is addressed in this paper for biped walking robot with varying iteration lengths. Two update laws are provided for the simulation studies. One is the conventional P-type update law and the other one introduces an iterative averaging operator to the previous tracking information. It is illustrated that both algorithms could achieve well tracking performance.

Key Words: Biped Walking Robot, Iterative Learning Control, Iteration Varying Lengths

1 Introduction

Iterative learning control (ILC) is a branch of intelligent control, which is much suitable for those systems that could complete some given task in finite time repeatedly. The aim of ILC is to design the input updating law such that the corresponding output can follow some desired trajectory asymptotically along the iteration axis [1]. This control method was first proposed in 1984 by Arimoto et al for robot systems [2]. As has been developed for three decades, ILC is very fruitful both in theoretical analysis and practical applications nowadays. It is a D-type update law that was first used in [3] and then more alternatives such as P-type, PD-type, PID-type, higher-order type, and feedback feedforward iterative type, are proposed in the ILC field. Meanwhile, ILC also has been applied to various systems, such as delayed, sampled, and distributed parameter systems [4, 5].

It is noticed that the inherent mechanism of ILC is to track a given reference well by successive learning along iteration axis. Thus it is usually required that the reference is iteration invariant so that ILC could learn from previous experiences. In this framework, if the condition on iteration invariant reference is not satisfied, the learning algorithm has to learn again from the beginning for new references. This further motivates academicians to consider ILC for systems with varying tracking references. Cheah studied a model reference based iterative learning law to track the dynamic output of the reference model in [6]. Chen and Wen discussed the problem of varying trajectory in the monograph [7], and it was pointed out that the final convergence boundary of system tracking error is affected by the supremum of the changes about the desired trajectory. Saab et al studied the continuous-time nonlinear system with slowly varying references in [8], where D-type, PD-type and PID-type update law were used to generate control signal for tracking problems, where the reference of next iteration was assumed to have a small deviation from the current iteration. In [9, 10], the direct learning control (DLC) approach was proposed, which mainly handled two cases of varying references. One case is that the references have the same pattern but different time scales, and the other one is that the references are with different magnitudes scales. In [11], the author proposed an adaptive ILC approach to cope with a class of high-order discrete-time system with multiple unknown time-varying parameters and unknown control gain, where the references were also iteration-varying, and the parameter estimation and controller design were provided.

As ILC is first proposed for industrial robots, there are many experiments study on the robot control. It is worth mentioning that Japan's humanoid robot has come to a very high level, and the biped walking robots are also studied. Biped walking model is a simple basic imitation of human movements, and is widely regarded as a passive mechanical motion process [12]. As given in [13], the passive dynamic walking (PDW) can make a biped walking robot walk down a gentle slope only using gravity effect and generate a stable periodic gait. In [14], it was proposed that two forces were involved in the simplest PDW model, one of which is the impulsive push along the stance leg and the other is a hip torque on the stance leg using the torso as a base. In this paper, the authors focus on the biped walking robot with varying iteration lengths. In [15] and [16], two update laws for the discrete-time affine nonlinear system and linear system with randomly varying iteration lengths were proposed and analyzed respectively. One is the conventional P-type update law and the other one uses an iterative average operator to make more robustness. Both algorithms can ensure the system output to track the desired reference accurately. These two update laws are applied to the biped walking robot in this note and related performances are discussed based on simulations.

The rest of the note is arranged as follows. Section 2 gives the description of biped walking robot and the problem form formulation; Section 3 presents the two ILC algorithms, while simulations results are shown in Section 4. Section 5 concludes this note.

*This work is supported by National Natural Science Foundation of China (61304085), Beijing Natural Science Foundation (4152040).

[1]D. Shen is the corresponding author of this paper.

2 Biped Walking Robot and Problem Formulation

2.1 The biped walking robot model

The biped walking robot model consists of a hip joint and two rigid legs, and the main mass at the hip and feet are M and m respectively, as shown in Fig. 1. The two rigid legs are distinguished as one stance leg and one swing leg in each step. According to the process of biped walking, the movement is separated into two phases, i.e., the swing phase and the heel-strike collision phase.

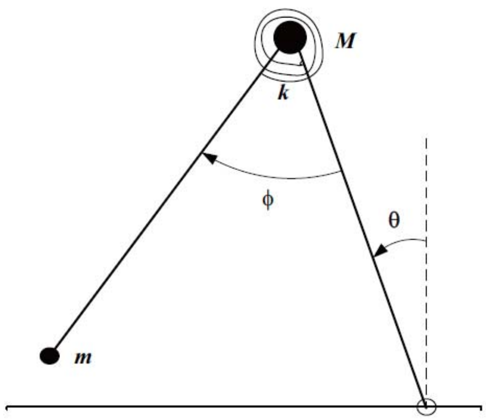

Fig.1. Illustration of biped walking robot.

a. The swing phase

The motion model of the swing phase is described by the following second-order differential equation [13].

$$\begin{bmatrix} 1+2\rho(1-\cos\varphi) & -\rho(1-\cos\varphi) \\ \rho(1-\cos\varphi) & -\rho \end{bmatrix}\begin{bmatrix} \ddot{\theta} \\ \ddot{\varphi} \end{bmatrix}+\begin{bmatrix} -\rho\sin\varphi(\dot{\varphi}^2-2\dot{\theta}\dot{\varphi}) \\ \rho\dot{\theta}^2\sin\varphi \end{bmatrix}$$
$$+\begin{bmatrix} (\rho g/l)[\sin(\theta-\varphi)-\sin\theta]-g/l\sin\theta \\ (\rho g/l)\sin(\theta-\varphi) \end{bmatrix}=\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{1}$$

where $\rho = m/M$, and θ is the angle between the stance leg and the ground normal, φ is the angle between the swing leg and the stance leg. Both angles are a function of time t. Besides, g is the acceleration of gravity, k is the stiffness of spring at the hip joint, and l is the length of the leg.

b. The heel-strike collision phase

In [11], the stance leg and the swing leg switches when

$$\varphi(\tau) = 2\theta(\tau) \quad (\tau \text{ is the step period}). \tag{2}$$

It is assumed that the impulse between the heel and the ground is instantaneous. Moreover, this impulse is an inelastic collision, such that the momentum is conserved. Then we have

$$\begin{bmatrix} \theta \\ \dot{\theta} \\ \varphi \\ \dot{\varphi} \end{bmatrix}^+ = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & \cos 2\theta & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & \cos 2\theta(1-\cos 2\theta) & 0 & 0 \end{bmatrix}\begin{bmatrix} \theta \\ \dot{\theta} \\ \varphi \\ \dot{\varphi} \end{bmatrix}^- \tag{3}$$

where the superscript " + " means ‘ just after the heel-strike ’ (i.e., after the touchdown), and the superscript " " means ‘ just before the heel-strike ’ (i.e., before the touchdown). Now, (3) can be written as

$$\begin{aligned} \theta^+ &= -\theta^- \\ \varphi^+ &= -2\theta^- \\ \dot{\theta}^+ &= \cos 2\theta(0)\dot{\theta}^- \\ \varphi^+ &= \cos 2\theta(0)(1-\cos 2\theta(0))\dot{\theta}^- \end{aligned} \tag{4}$$

From (4), the equation of the motion after switching the stance leg and the swing leg can be attained. Successive repetition of these two processes forms the actual walking. In order to facilitate the simulation experiments later, the system state and output are defined as follows.

$$x = [x_1, x_2, x_3, x_4]^T = [\theta, \varphi, \dot{\theta}, \dot{\varphi}]^T$$
$$u = [u_1, u_2]^T = [\tau_1, \tau_2]^T$$

The model dynamic equation can be rewritten in the form of an affine nonlinear system.

$$\dot{x} = f(x) + g(x)u$$

where

$$f(x) = \begin{bmatrix} \dot{\theta} \\ \dot{\varphi} \\ \dfrac{\begin{pmatrix} -(2\dot{\theta}\dot{\varphi}-\dot{\varphi}^2+(\cos\varphi-1)\dot{\theta}^2)\rho\sin\varphi \\ +\rho\cos\varphi\sin(\varphi-\theta)+(\rho+1)\sin\theta \end{pmatrix}}{1+\rho\sin^2\varphi} \\ \dfrac{\begin{pmatrix} (\dot{\theta}^2-\rho(\cos\varphi-1)(2\dot{\theta}^2-2\dot{\theta}\dot{\varphi}+\dot{\varphi}^2))\sin\varphi \\ -(\rho+1)(\cos\varphi-1)\sin\theta-\sin(\varphi-\theta)(1- \\ \rho(\cos\varphi-1)) \end{pmatrix}}{1+\rho\sin^2\varphi} \end{bmatrix}$$
$$= \begin{bmatrix} x_3 \\ x_4 \\ \dfrac{\begin{pmatrix} -(2x_3x_4-x_4{}^2+(\cos x_2-1)x_3{}^2)\rho\sin x_2 \\ +\rho\cos x_2\sin(x_2-x_1)+(\rho+1)\sin x_1 \end{pmatrix}}{1+\rho\sin^2 x_2} \\ \dfrac{\begin{pmatrix} (x_3{}^2-\rho(\cos x_2-1)(2x_3{}^2-2x_3x_4+x_4{}^2))\sin x_2 \\ -(\rho+1)(\cos x_2-1)\sin x_1-\sin(x_2-x_1)(1- \\ \rho(\cos x_2-1)) \end{pmatrix}}{1+\rho\sin^2 x_2} \end{bmatrix}$$
$$g(x) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \dfrac{1}{1+\rho\sin^2 x_2} & \dfrac{1-\cos x_2}{1+\rho\sin^2 x_2} \\ \dfrac{1-\cos x_2}{1+\rho\sin^2 x_2} & \dfrac{1+2\rho(1-\cos x_2)}{\rho+\rho^2\sin^2 x_2} \end{bmatrix} \tag{5}$$

Take a small sampling interval $T_s = 0.1$, and discretize the continuous system. We can have the following discrete-time model.

$$x_k(t+1) = f(x_k(t)) + g(x_k(t))u_k(t)$$

$$f(x_k(t))=\begin{bmatrix} T_s x_3(t)+x_1(t) \\ T_s x_4(t)+x_2(t) \\ \dfrac{\begin{pmatrix}-(2x_3(t)x_4(t)-x_4^2(t)+(\cos x_2-1)\cdot \\ x_3^2(t))\rho\sin x_2(t)+\rho\cos x_2(t)\sin(x_2(t) \\ -x_1(t))+(\rho+1)\sin x_1(t)\end{pmatrix}}{(1+\rho\sin^2 x_2(t))/T_s}+x_3(t) \\ \dfrac{\begin{pmatrix}(x_3^2(t)-\rho(\cos x_2(t)-1)(2x_3^2(t)-2x_3(t)x_4(t) \\ +x_4^2(t)))\sin x_2(t)-(\rho+1)(\cos x_2(t)-1)\sin x_1(t) \\ -\sin(x_2(t)-x_1(t))(1-\rho(\cos x_2(t)-1))\end{pmatrix}}{(1+\rho\sin^2 x_2(t))/T_s}+x_4(t)\end{bmatrix}$$

$$g(x_k(t))=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \dfrac{1}{(1+\rho\sin^2 x_2(t))/T_s} & \dfrac{1-\cos x_2(t)}{(1+\rho\sin^2 x_2(t))/T_s} \\ \dfrac{1-\cos x_2(t)}{(1+\rho\sin^2 x_2(t))/T_s} & \dfrac{1+2\rho(1-\cos x_2(t))}{(\rho+\rho^2\sin^2 x_2(t))/T_s}\end{bmatrix}$$

(6)

2.2 Problem Formulation

In the traditional ILC problem, the reference trajectory is generally assumed to be iteration-unchanged. Therefore, each iteration will take the same operation length. If this condition is required for the biped walking robot, then it is required that all step movements of a biped walking robot should be completed in the same time interval. However, it is more practical that the time length of a step movement may varies in a range. This case is closer to the actual walking. Thus in this paper, ILC is designed for a biped walking robot with iteration varying lengths. For this problem, we first formulate the system as follows.

$$\begin{cases} x_k(t+1)=f(x_k(t))+g(x_k(t))u_k(t) \\ y_k(t)=Cx_k(t) \end{cases} \tag{7}$$

where $f(\cdot)$, $g(\cdot)$ are shown in (4), and $C=\begin{bmatrix}0&0&1&0\\0&0&0&1\end{bmatrix}$

The symbols $x_k(t)\in R^n$, $u_k(t)\in R^p$, and $y_k(t)\in R^r$ denote state, input and output, respectively; k denotes the iteration index, $k=0,1,2,\ldots$; t denotes the discrete time, $t=0,1,2,\ldots,N_d$, where N_d is a positive integer labeling the desired iteration length. Here it is assumed $x_d(0)=x_k(0)$.

3 ILC laws with Iteration-Varying Lengths

Two ILC update laws are presented for the problem of varying iteration lengths. One is the conventional P-type update law and the other one uses an iterative average operator to make more robustness.

3.1 Formulation on Iteration-Varying Lengths

The biped walking robot may end a cycle of walking gait before the desired length N_d. In order to model this issue, let N_k be a random variable valued in $\{N_m,\ N_m+1,\ldots N_d\}$, where N_{m} denotes minimal iteration length. Define A_{N_k} be the event that the k-th iteration length is N_k, and $P\{A_{N_k}\}$ be the probability of the k-th iteration length being N_k, where $P\{A_{N_k}\}>0$.

Thus it is easy to get $P\{A_{N_k}\}=p(N_k)-p(N_k+1)$, where $p(t)$ is the corresponding probability value for each time, $N_{\mathrm{m}}\le t\le N_d$ and $\sum_{t=N_m}^{N_d}P\{A_t\}=1$.

Note that the iteration length varies among different iterations, thus part tracking information could not be obtained if the case $N_k=N_d$ is not satisfied. In this situation, i.e., $N_k<N_d$, the output information from N_k to N_d is not available, and thus could not be used to calculate tracking error for the corresponding input updating. To deal with this case, denote the tracking errors at these time instances as zero. That is, we could define a modified tracking error as follow.

$$e_k^*(t)=\begin{cases} e_k(t) & 1\le t\le N_k \\ 0 & N_k<t<N_d \end{cases} \tag{8}$$

where $e_k(t)=y_d(t)-y_k(t)$ is the original tracking error.

According to the above definition of $e_k^*(t)$, (7) could be reformulated as follows

$$e_k^*(t)=1(t\le N_k)e_k(t) \tag{9}$$

where $1(t\le N_k)$ is an indicator function whose value is 1 if the indicated event is true and 0 otherwise. When $t\le N_k$, it is regarded as the process of control is continuous occurrence. Then

$$P\{t\le N_k\}=P\{1(t\le N_k)\}=p(t) \tag{10}$$

Thus, we can get

$$E\{1(t\le N_k)\}=1\cdot p(t)+0\cdot(1-p(t))=p(t) \tag{11}$$

3.2 ILC update laws

Now we could give the following update laws for input signal.

The first one is the original P-type updating law with modified tracking errors.

$$u_{k+1}(t)=u_k(t)+Le_k^*(t+1) \tag{12}$$

where L is the learning gain, $L\in R^{p\times r}$

The second one is also P-type but with an average operator of historical tracking errors.

$$u_{k+1}(t)=A\{u_k(t)\}+\frac{k+2}{k+1}\cdot L\cdot\sum_{i=0}^{k}e_i^*(t+1) \tag{13}$$

where $A\{f_k(\cdot)\}=\frac{1}{k+1}\sum_{i=0}^{k}f_i(\cdot)$ denoting the average operator given in [16], and L is the learning gain, $L\in R^{p\times r}$.

From [15], the convergence of (12) is established, while the convergence of (13) is analyzed in [16] for linear systems.

Remark 1. *In [16], the probability on randomly varying iteration length is first given and then calculates the probability of the occurrence of output at each time instance. However, in [15], the order is exchanged, i.e., the probability of the occurrence of output at each time instance is first given and then calculates the probability of random iteration length. However, the internally logical relationship is identical.*

4 Experiment Results

Without loss of generality, let the desired trajectory be

$$y_{1d} = 0.5\pi T_s \sin(\frac{\pi}{2} T_s t)$$

$$y_{2d} = -\pi T_s \sin(\frac{\pi}{3} T_s t) \quad (14)$$

where $T_s = 0.1s$ and t is the time instance number. We assume a step gait is completed in 2s, thus the expected iteration length is $N_d = 20$. To model the iteration varying lengths, we allow the iteration length varies from 15 to 20 in the simulation. As just a simple case for illustration, assume N_k satisfy discrete uniform distribution during the discrete set $\{15, 16, 17, 18, 19, 20\}$.

The input at the initial iteration is simple given as zero for all time instances. The initial state is to be set as $x_k(0) = [0.1, -0.2, -0.8, 2.1]$, the mass ratio is $\rho = 0.5$. The simulations are all run 300 iterations.

4.1 The Conventional P-type iterative update law

When the update law (12) is applied, the parameter is set as $L = [0.5, 0.5]$.

The angular velocities of θ, φ at the last iteration are shown in Fig.2. As we can see, the actual output achieves perfect tracking performance.

Here it should be noted that $y_1(t)$, $y_2(t)$ represents the derivative of θ, φ. That is, $y_i(t)$ is angular velocity. Because the application is a simple model of bipedal walking, the model can be seen as a pendulum when switching has not occurred.

The tracking errors of the $y_1(t)$, $y_2(t)$ of the whole time interval at the 60th, 120th, 240th, and 300th iterations are shown in Fig.3. It is observed that the magnitude of the errors shows a decreasing trend as iteration number increase. The maximal error at the 120th iteration has been very small. Since the operation length randomly varies among different iterations, the curves in Fig. 3 are not with the same length.

The maximal tracking errors along iterations are shown in Fig.4. This reflects the worst tracking performance of each iteration. As one can see from Fig. 4, the maximal tracking error of $y_1(t)$ is almost zero after the 120th iteration, while the maximal tracking error of $y_2(t)$ is almost zero after the 100th iteration.

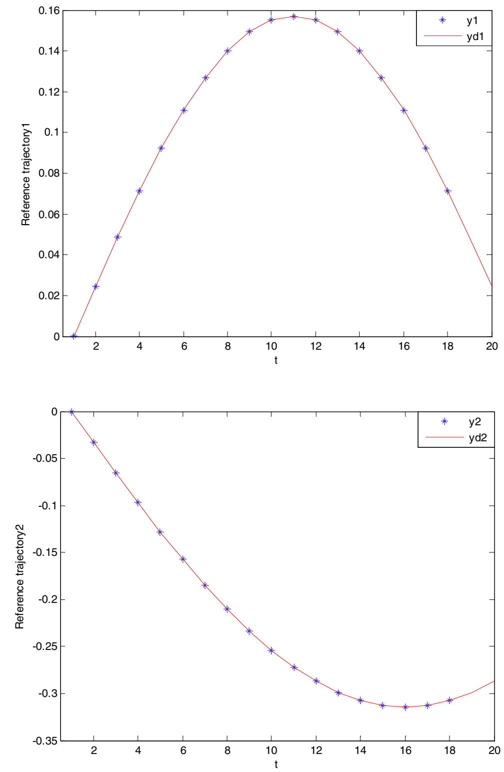

Fig. 2. Desired trajectory and output of the last iteration.

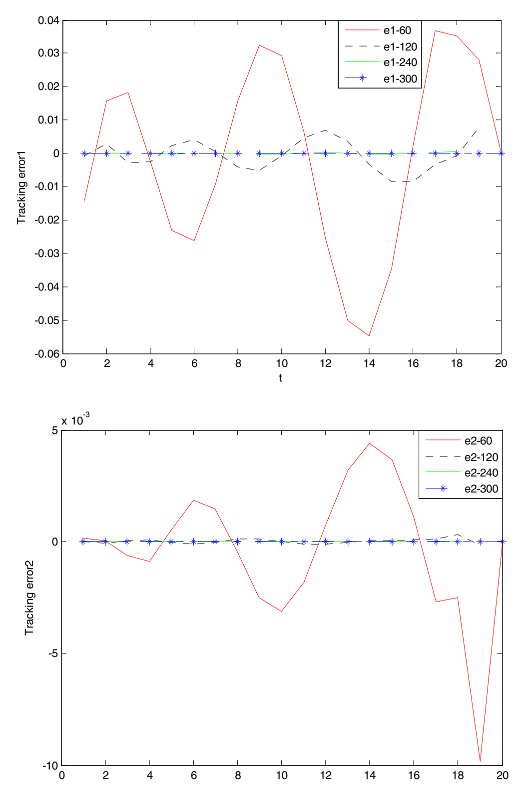

Fig. 3. Tracking errors at the 60th, 120th, 240th, and 300th iterations.

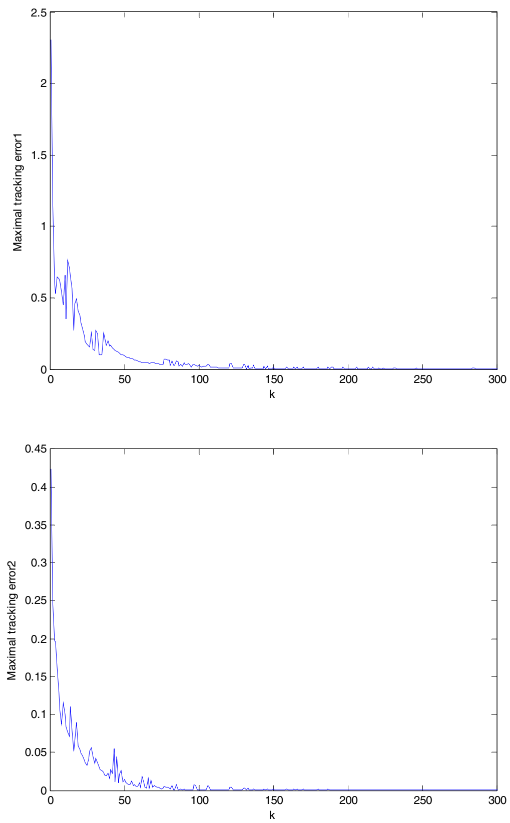

Fig. 4. Maximal tracking along iterations

4.2 The average operator iterative update law

When the update law (13) is applied, the parameter is also given as $L=[0.5, 0.5]$

The angular velocities of θ, φ at the last iteration are shown in Fig.5. It can be seen that the actual output achieves perfect tracking performance

The tracking errors of the $y_1(t)$, $y_2(t)$ at the 60th, 120th, 240th, and 300th iterations are shown in Fig.6. Similar to Fig. 3, it can be seen that the errors also show a decreasing trend as the iteration number increases. However, the decreasing speed of the first output for the update law (13) is a little slower than for the update law (12).

To make detailed comparison, the maximal tracking errors along iteration axis are shown in Fig. 7. As one can see, the maximal error for the average operator update law also decreases to zero. However, compared with the conventional P-type update law, the speed of (13) is a little slower. We believe the reason is that the average operator retains part information of previous iterations and this mechanism slows the learning speed. On the other hand, this mechanism generally owns much robustness if large disturbances happen to the system.

Remark 2. *As one can see, in [16], the condition on L is some loose; however, in [15], the requirements on L is a little restrictive, but no information on probability is requested any longer. Thus it is more suitable for practical applications.*

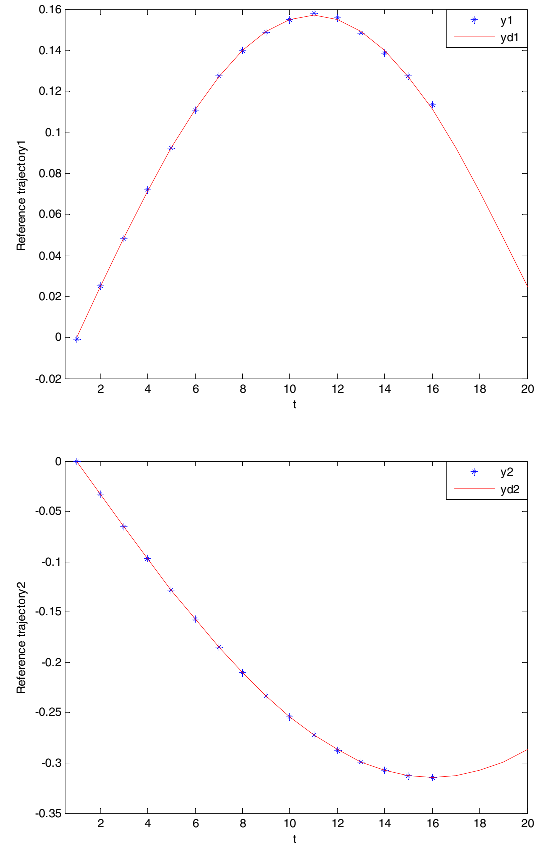

Fig. 5. Desired trajectory and output of the last iteration.

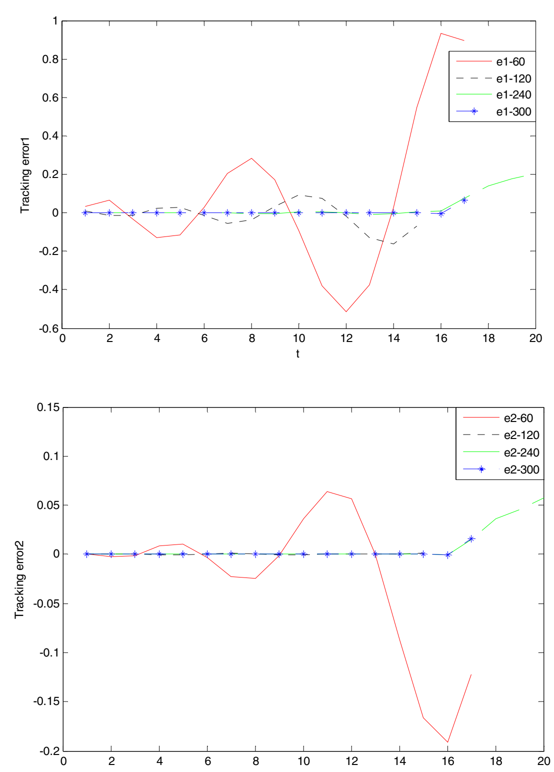

Fig. 6. Tracking errors at the 60th, 120th, 240th, and 300th iterations.

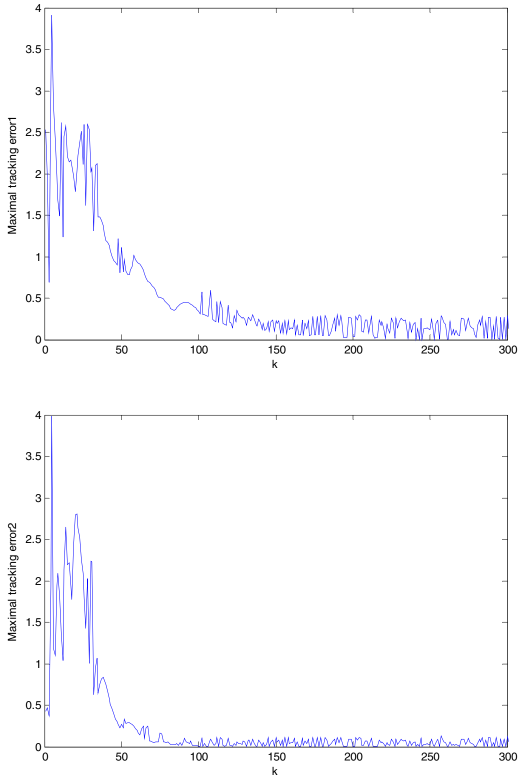

Fig. 7. Maximal tracking along iterations

Through the simulation experiments of the biped walking robot model with two update laws in the same parameters condition, respectively, we observe that the convergence speed of the P-type update law may be faster due to its sensitivity to innovation information. By the results of experiments, we can also easily verify the practical value of engineering application of the proposed algorithm (12).

5 Conclusions

Two ILC update laws for the biped walking robot with varying iteration lengths are discussed in this note. One is the conventional P-type update law and the other one introduces an iterative averaging operator to previous tracking information. From the simulation results, it is observed that the former update law ensures a faster learning speed than the latter one under the same simulation conditions. More analysis and comparison for various cases such as varying initial states and disturbances will be studied in the next step.

References

[1] Douglas A. Bristow. A Survey of iterative learning control: a learning based method for high performance tracking control. *IEEE Control System Magazine*, 97-114, 2006.

[2] Uchiyama M. Formation of high speed motion pattern of mechanical arm by trial[J]. *Trans of the Society of Instrumentation and Control Engineer*s, 1(5): 706-712, 1978.

[3] Arimoto S, Kawamura S, Miyazaki F. Bettering operation of robots by learning [J]. *J of Robotic Systems*, 1(2): 123-140, 1984.

[4] M. Sun, B. Huang, *Iterative Learning Control*, Beijing: National Defense Industry Press, 1999.

[5] Dong Shen, Yun Xu. Iterative Learning Control for Discrete -time Stochastic Systems with Quantized Information. *IEEE/CAA Journal of Automatica Sinica*, 3(1): 59-67, 2016

[6] Cheah C, Wang D. A model reference learning control scheme for a class of nonlinear systems[J]. *International ，Journal of Contro*l, 66(2): 271-288, 1997.

[7] Y Q． Chen，c． Y． Wen． Iterative Learning Control: Convergence，Robustness and Application[M]. *Springer-Ver la*g， 1999.

[8] Samer S. Saab, William G. Vogt, Marlin H. Mickle. Learning control algorithms for tracking “slowly” varying trajectories. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 27(4): 657-679, 1997.

[9] J.-X. Xu. Direct learning of control efforts for trajectories with different magnitudes scales. *Automatica*, 33(12): 21912195, 1997.

[10] J.-X. Xu. Direct learning of control efforts for trajectories with different time scales. *IEEE Transcation on Automatic Control*, 43(7): 1027-1030, 1998.

[11] Ronghu Chi, Zhonsheng Hou, Jianxin Xu. Adaptive ILC for a class of discrete-time systems with iteration-varying trajactory and random initial condition. Automatica, 44(8): 2207-2213, 2008.

[12] Garcia M, Chatterjee A, Ruina A, Michael Coleman. The Simplest Walking Model: Stability, Complexity, and Scaling[J]. *Asme Journal of Biomechanical Engineering*, 120(2): 281-28 8, 1998.

[13] Mcgeer T. Passive Dynamic Walking[J]. *International Journal of Robotics Research*, 9(6): 62-82, 1990.

[14] A. D. Kuo. Energetics of Actively Powered Locomotion Using the Simplest Walking Model[J]. *Journal of Biomechanical Engineering*, 124(1): 113-120, 2002.

[15] Wei Zhang, Dong Shen. Iterative Learning Control for Discrete-Time Affine Nonlinear System with Iteration Varying Lengths. *the 34st CCC*, 3062-3067, 2015.

[16] X. Li, J.-X. Xu, D. Huang, An iterative learning control approach for linear systems with randomly varying trial lengths. *IEEE Transactions on Automatic Control*, 59(7): 1954-1960, 2014.