2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

Target Walking Speed Generation and Parameters Identification by Feedback Control of 1-DOF Limit Cycle Walker

Qingqing Wei[1],Xuan Xiao[2], Qingliang Meng[2] and Fumihiko Asano[3]

Abstract— **This paper studies a model-based feedback controller which can generate limit cycle walking at target walking speed, and identify the physical parameters through neural network. First, a combined rimless wheel is developed, and the feedback control is proposed by dynamic planning its equation of motion. Second, the numerical simulations are conducted to analyse walking speed and other properties when the physical parameters are assumed unknown and the prediction parameters are used instead. The controller has a certain adaptability to the prediction error, and the target walking speed can be generated with little error (0.001%). Finally, based on the model-based properties of the control, the physical parameters can be predicted through a proposed neural network model with an average error of 2%. In general, the model-based feedback controller provides us a new approach for simultaneously controlling walking speed and identifying physical parameters.**

I. INTRODUCTION

As the extension of "Passive Dynamic Walking" [1], the paradigm of "Limit cycle walking" was proposed to make the walker obtain the stable periodic walking without locally stabilizing the walking motion at every instant during gait cycle. Utilizing their own physical dynamics and passivity, limit cycle walkers have advantages in generating natural and energy-efficient dynamic gaits, but thus far they have not been as versatile [2][3]. As an aspect of versatility, accurately generating a target walking speed is a challenge for limit cycle walkers because of the irreversibility of time and space. Therefore, an overall planning for a discrete time period and a space segment in one step is the essential of the target speed controlling for walkers.

Several approaches to accurately control the walking speed have been proposed. Hobbelen studied how walking speed could be varied, which way was energetically beneficial and how walking speed affected a walker's ability to handle disturbances in limit cycle walking [4][5]. Kajita et al. proposed a method by changing the foothold of a biped walker to modify the initial condition of the support phase to control the walking speed based on a PD feedback controller [6]. In addition, Juang et. al. proposed a learning scheme which trained the neuro-fuzzy controller to follow the designed trajectory as closely as possible for generating walking gaits at a certain speed [7]. Focusing on the convergence gaits of limit cycle walking, Xiao et. al. proposed a model-based control to keep target walking speeds when handling the disturbance.

As the property of model-based controller, however, its performance generally depends on all the physical model parameters. Thus a vary of parameter identification methods [8][9] to predict the parameters in advance becomes a solution, and proposing the robust controller which can handle the prediction error is another option. Since the prediction parameter error directly reflects on the performance of walkers, conversely, the performance of model-based controller can help us to predict the physical parameters. In other words, the controller performance and the prediction physical parameters are promising to optimize each other during walking.

The discrete dynamics progress of walking makes mathematical analysis a great challenging work, and thus some data-driven methods are considered as solutions. Neural Networks are computing systems inspired by the biological neural networks that constitute animal brains. They are common tools of controlling and parameter identification which can catch hidden and strongly non-linear dependencies, even when there is a significant noise in the training set. In this paper, a model-based feedback controller is proposed to generate limit cycle walking at target speeds, and the physical parameters are predicted based on the walking performance through neural networks during walking. As the contribution, this study provides a new approach to reduce the dependence of model-based controllers on the physical parameters when the walker is generating target states.

II. ACTIVE COMBINED RIMLESS WHEEL

A. Modeling

Fig.1 shows the model of a planar active combined rimless wheel (CRW). It consists of two eight-legged rimless wheels

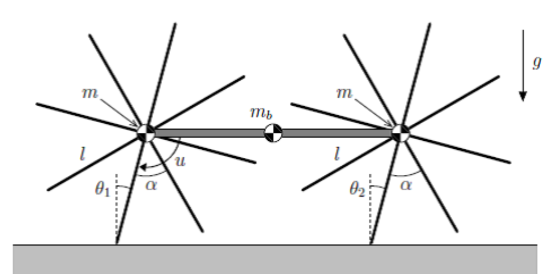

Fig. 1: A planar active combined rimless wheel

[1]Qingqing Wei is with Beijing Institute of Spacecraft System Engineer, Beijing, 100094, China weiqingqing51@sina.com
[2]Xuan Xiao and Qingliang Meng are with the School of Aerospace Engineering, Tsinghua University, 1 Tsinghua Yuan, Beijing, 100084, China {xxiao,mengql16}@mail.tsinghua.edu.cn
[3]Fumihiko Asano is with the School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan fasano@jaist.ac.jp

(RWs) combined by a body frame. A motor on the frame exerts a joint torque, u, between the rear stance-leg and the body frame. We assume the following statements.

- The fore and rear stance legs always contact with the ground without sliding.
- The inertia moments about the CoMs of all the frames can be neglected.
- The fore and rear RWs perfectly synchronize or rotate maintaining the relation $\theta_1 \equiv \theta_2$.

In the CRW, the steady walking speed can be calculated by ratio between the step period and step length. Therefore, the target steady walking speed state can be generated by controlling the step period due to the constant step length.

B. Equation of Motion and Its Linearization

A four-bar linkage is configured by the body frame, the two stance legs and the ground surface. Exerting the joint torque is thus equivalent to exerting the ankle-joint torque. In addition, the torque of the joint viscosity, $f_{\mathrm{v}} = -k_{\mathrm{v}}\dot{\theta}$ is also taken into consideration. Thus, the dynamics of the rear RW then becomes identical to that of an active RW with an ankle-joint torque and viscosity friction, that is,

$$\ddot{\theta} = \omega^2 \sin\theta + \frac{u}{Ml^2} - \frac{k_{\mathrm{v}}\dot{\theta}}{Ml^2} \tag{1}$$

where $M := m_b + 2m$ [kg] is the total mass of the CRW, $\theta(= \theta_1 = \theta_2)$ is the stance-leg angle, and $\omega := \sqrt{g/l}$ [rad/s]. By linearizing $\theta \approx \sin\theta$ around 0 [10], the state-space realization of the RW dynamics becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega^2 & -\widehat{k_{\mathrm{v}}} \end{bmatrix}\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 1/Ml^2 \end{bmatrix} u \tag{2}$$

Where $\widehat{k_{\mathrm{v}}} = k_{\mathrm{v}}/Ml^2$. Thus we denote Eq. (2) as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u. \tag{3}$$

It should be clear that Eq. (2) is only used for the control system designing. All the simulations are conducted through the kinetic equation in Eq. (1). Obviously, when the control system is designed based on Eq. (1), the generated walking state in the simulation can not completely meet the expected results due to the linearization in Eq. (2). According to the results of follow sections, the errors are acceptable and can be eliminated.

C. Collision Equation

We define the state vector immediately before the (i)th impact as $\boldsymbol{x}_i^-$ and the state vector immediately after the (i)th impact as $\boldsymbol{x}_i^+$. In the collision phase we assume that the rear leg frame at impact (the previous stance leg) begins to leave the ground immediately after the landing of the fore leg frame (the next stance leg) according to the law of inelastic collision. Thus based on the research of Coleman [3], the transition equation for the angular velocity becomes

$$\dot{\theta}_i^+ = \mu\dot{\theta}_i^-, \tag{4}$$

where

$$\mu = \frac{I_{\mathrm{c}} + Ml^2\cos\alpha}{I_{\mathrm{c}} + Ml^2},$$

and I_{c} is the inertia moments of the CRW. Considering inertia moments are neglected in this paper ($I_{\mathrm{c}} = 0$), μ is simplified as $\mu = \cos\alpha$.

In addition, when the CRW walks on the level ground, the initial and terminal angular positions are $\mp\frac{\alpha}{2}$. The transition equation for angular position is also determined as

$$\theta_i^+ = -\theta_i^- = -\frac{\alpha}{2}.$$

D. Control System

In the control system, the torque is dynamically updated based on current time and state. The moment immediately after impact is defined as 0 [s] and the time parameter will be reset immediately after every impact. The torque must supply enough kinetic energy to make the CRW overcome the potential barrier. In addition, if the walking speed is so fast that the control cannot be completed before the next impact, the target walking speed cannot be guaranteed.

III. CONTROL LAW

Control law is proposed based on the constant torque controller. Instead of designing a specified position trajectory, the motor is made to follow the desired constant-torque strategy. In detail, when the disturbance happens, the walker can propose a new planning according to the current state and the target one. By updating the control planning for the rest time, the disturbance can be handled and the target walking states can be guaranteed. The principle of model-based feedback control is common in our daily life, especially in mapping. The detail of the feedback control is introduced as follows.

A. Terminal Boundary Condition Prediction

Based on Eq. (2), the dynamic equation of RW has been proposed. Thus as the solution of differential equation $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u$, the state vector immediately before the $(i+1)$th impact, $\boldsymbol{x}_{i+1}^-$, is written by the state vector immediately after the (i)th impact, $\boldsymbol{x}_i^+$, as [11]

$$\boldsymbol{x}_{i+1}^- = \mathrm{e}^{\boldsymbol{A}T_i}\boldsymbol{x}_i^+ + \int_{0^+}^{T_i^-} \mathrm{e}^{\boldsymbol{A}(T_i - s)}\boldsymbol{B}u_i(s)\ \mathrm{d}s. \tag{5}$$

Similarly, the state vector at t [s], $\boldsymbol{x}_i(t)$ can be written by the time t and the initial state vector $\boldsymbol{x}_i(0)$ as

$$\boldsymbol{x}_i(t) = \begin{bmatrix} \theta_i(t) \\ \dot{\theta}_i(t) \end{bmatrix} = \mathrm{e}^{\boldsymbol{A}t}\boldsymbol{x}_i(0) + \int_{0^+}^{t} \mathrm{e}^{\boldsymbol{A}(t-s)}\boldsymbol{B}u_i(s)\ \mathrm{d}s. \tag{6}$$

Therefore, based on the Eq. (6), at t [s] we assume to keep a constant control input $u_i(t)$ for the rest $T_{\mathrm{s}}^* - t$ [s] to generate the terminal condition $\boldsymbol{x}_i(T_i)$ as follows [12].

$$\boldsymbol{x}_i(T_i) = \mathrm{e}^{\boldsymbol{A}(T_i - t)}\left(\boldsymbol{x}_i(t) + \int_{0^+}^{T_i - t} \mathrm{e}^{-\boldsymbol{A}s}\boldsymbol{B}u_i(t)\ \mathrm{d}s\right) \tag{7}$$

By expanding Eq. (7), the terminal angular position $\theta_i(T_i)=\frac{\alpha}{2}$ can be derived by t and $u_i(t)$ in detail, and thus the equation of angular position is simplified for designing the controller as follows [13].

$$\frac{u_i(t)(F_1-\mathrm{e}^{k_\mathrm{r}})}{\mathrm{e}^{k_\mathrm{r}}\lambda}+\frac{\theta_i(t)F_1+2\dot{\theta}_i(t)\sinh\left(\frac{T_\mathrm{r}k_\mathrm{e}}{2}\right)}{\mathrm{e}^{k_\mathrm{r}}}=\frac{\alpha}{2} \quad (8)$$

Where $k_\mathrm{e}=\sqrt{\widehat{k_\mathrm{v}}^2+4\omega^2}$, $k_\mathrm{r}=\frac{1}{2}T_\mathrm{r}\widehat{k_\mathrm{v}}$, $T_\mathrm{r}=T_\mathrm{s}^*-t$, $\lambda=l^2M\omega^2$ and

$$F_1=k_\mathrm{e}\cosh\left(\frac{T_\mathrm{r}k_\mathrm{e}}{2}\right)+\widehat{k_\mathrm{v}}\sinh\left(\frac{T_\mathrm{r}k_\mathrm{e}}{2}\right). \quad (9)$$

As a result, after the analysis of the equation of angular position above, $u_i(t)$ is derived for generating the target step period state, $T_i=T_\mathrm{s}^*$, as

$$u_i(t)=\lambda\frac{\alpha k_\mathrm{e}\mathrm{e}^{k_\mathrm{r}}-4\dot{\theta}_i(t)\sinh\left(\frac{T_\mathrm{r}k_\mathrm{e}}{2}\right)-2\theta_i(t)F_1}{-2k_\mathrm{e}\mathrm{e}^{k_\mathrm{r}}+2F_1}, \quad (10)$$

Even though the control law has been proposed, there are still some details to be solved. However,

$$\lim_{T_\mathrm{r}\to 0}2F_1-2k_\mathrm{e}\mathrm{e}^{k_\mathrm{r}}\approx 0,$$

and a system error will be caused in Eq. (10). Here, a simple solution is proposed:

- When $T_\mathrm{r}<0.001$ [s], we set $u_i(t)=0$.

In addition, the physical parameter of viscosity friction k_v, however, usually cannot be obtained accurately, which obviously affects the performance of controller. The other parameters, M and l are also assumed as unknown parameters. Therefore, when the input torque is calculated in Eq. (10), M and l and k_v are all replaced by the prediction parameters of M_p and l_p and k_vp in some reasonable range instead. As a result, the flow chart of the feedback control is illustrated as Fig. 2. If the CRW can feedback the walking state constantly, and thus the control system calculates a new control input $u_i(t)$ based on the current time t and the walking state $\boldsymbol{x}_i(t)$ by Eq. (10) constantly, a target walking speed can be generated and the ability of disturbance handling can be improved. In addition, the model-based control updates the control torque dynamically, and thus it is difficult to analyse the trajectory in each step in advance. The proof of stability of our feedback control is left as a future work.

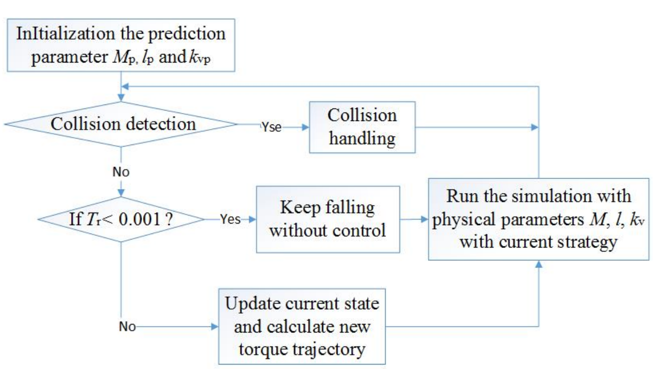

Fig. 2: Flow chart of the control law

TABLE I: Physical parameters

m/[kg]	l/[m]	k_v
1.0	1.0	0.5

TABLE II: Prediction parameters of simulation

Prediction parameter	Set 1	Set 2
m_p/[kg]	1.0	1.2
l_p/[m]	1.0	1.2
k_vp	0.5	1.0

To test the performance of our controller, the numerical simulations were conducted under prediction parameters. The RW was made to walk at the target speed of 0.4 [s] per step. 2 particular sets of prediction parameters were selected: one set was exactly equal to the real physical parameters, and the other one was larger. In detail, the real physical parameters and the two sets of prediction parameters were listed in Table I and Table II.

As a result, the simulations were conducted and the gait properties were recorded. The target walking speed were guaranteed under two sets of prediction parameters, and the target step period gait was generated with little error (0.001%) during all the steps. In addition, Figs. 3 and 4 illustrated the evolutions of control torque. When the prediction parameters were exactly equal to the physical parameters, the control torque in each step had some small fluctuations caused by the linearization in Eq. (2). On the contrary, the control torque significantly changed to handle the disturbance caused by the inaccurate model. When the rest time were close to 0, the torque increase or decrease drastically. Another interesting property was illustrated in Figs. 5 and 6, the Poincare Map of both cases. The simulation with larger prediction error had a faster convergence speed which could be explained by the deceleration effect [11].

Furthermore, 40 simulations under different sets of prediction parameters were conducted for testing the boundary conditions. In each simulation, the walker was driven by our feedback control to walk 40 steps at the target walking

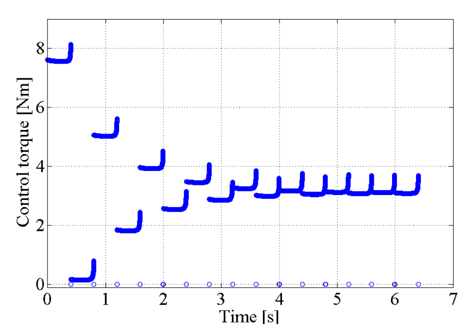

Fig. 3: Time-evolution of control input under the prediction parameters of Set 1

speed (0.4[s] per step) by using a set of random prediction parameters. The real parameters were same as Table I and the prediction parameters were picked randomly in the sufficient ranges from Table III. When the walking time error was larger than 0.1% (totally $\pm$ 0.016[s]), the simulation would be marked as a failure. Through analysing the prediction parameters in the failure cases, a rough range of prediction parameters would be discovered.

As a result, the distribution of the simulations was illustrated in Fig. 7. The pink star point meant the real physical parameters, and three failure cases were found as the red points cases. It turned out that both prediction parameters of m and l are relatively lower than the real ones could cause failure cases. If the initial angular velocity of one step was very fast, however, the controller would incorrectly design the walking trajectory and make the planning angular velocity immediately before impact a negative value, and caused a failure case. Therefore, qualitatively speaking, when the physical parameters were unclear, setting a positive prediction error in the reasonable range would be a good choice.

As a conclusion, three properties of our feedback control were concluded as follows.

- The feedback control had a certain adaptability to keep the target speed when the a vary of prediction parameters were close to the real physical ones. Relative large prediction error would be preferred.
- The fluctuations of torque were caused by both linearization and prediction error. Even if the prediction parameters were exactly equal to the physical ones, small fluctuations were also caused by the linearization. A large prediction error caused rapid change of the torque.
- The prediction parameters could affect some gait properties, such as convergence speed and energy-efficiency.

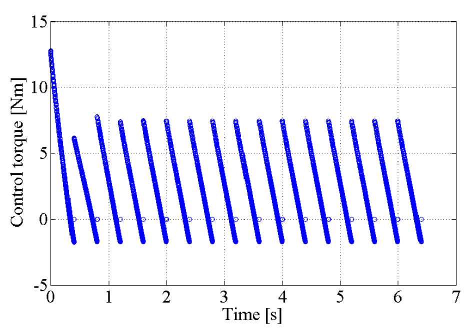

Fig. 4: Time-evolution of control input under the prediction parameters of Set 2

TABLE III: Pattern information

Prediction parameter	Random range
m_{p}/[kg]	[0.8,1.3]
l_{p}/[m]	[0.8,1.3]
k_{vp}	[0.0,2.0]

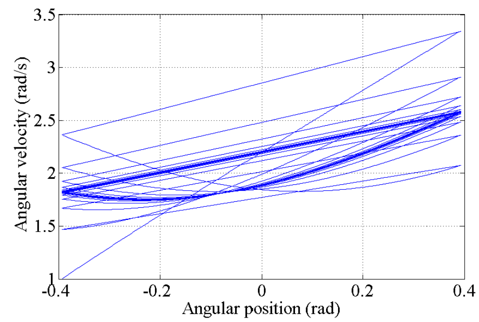

Fig. 5: Poincare Map under prediction parameters of Set 1

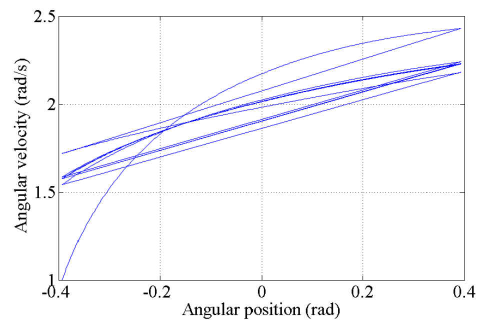

Fig. 6: Poincare Map under prediction parameters of Set 2

IV. PARAMETERS IDENTIFICATION BASED ON NEURAL NETWORK

Based on the analysis of the numerical simulation under 37 sets of random prediction parameters in Fig. 7, the torque trajectory of each case depends on the prediction parameters change. There is a relationship in the non-linear dynamic system, but it's a great challenge to extract the rules through mathematical methods, and thus the data-driven methods becomes an option.

Artifitial Neural Networks (ANNs) are computing systems inspired by the biological neural networks. An ANN consists of basic processing units, the neurons, and weighted connections between these neurons. By building the training sets from the plenty of simulation results, the prediction

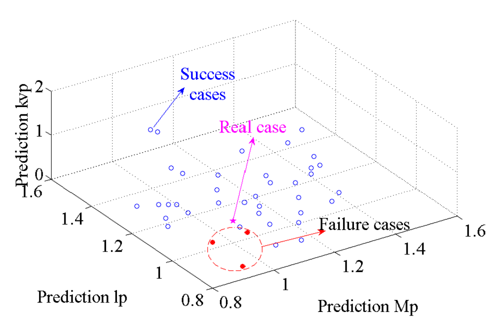

Fig. 7: The distribution of the 40 sets of random prediction parameters

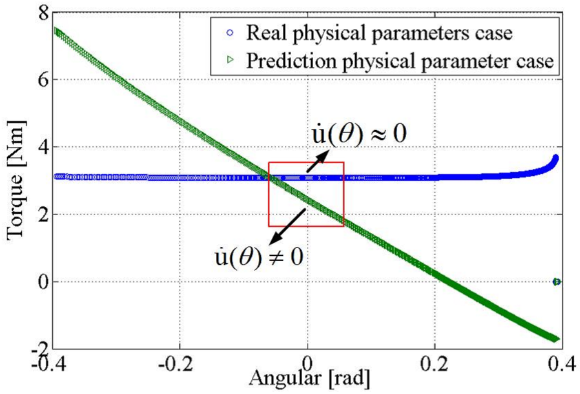

Fig. 8: The method of predicting physical parameters through the trajectory of $u(\theta)$ in one step

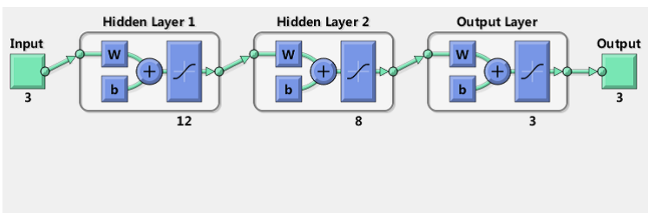

Fig. 9: The architecture of the neural network and the training parameters

model of the physical parameters can be proposed. Thus the physical parameters can be predicted precisely, and then basis of prediction parameters will help to optimize the control input.

A. Neural Networks Training

The training sets must be built to reflect the relationship between performance of controller and the prediction error. According to the control strategy, the flat torque trajectory in each step demonstrates small prediction error. However, the performance error was caused by both the linearization and the prediction error. Therefore, $\lim_{\theta \to 0} \theta \approx \sin\theta$ is taken into consideration to exclude the influences of the error of linearization. As shown in Fig. 8, when $-0.05 \leq \theta \leq 0.05$, the trajectory of θ evolution of control input should be approximately an unknown constant value (which means $\dot{u}(\theta) = 0$) if there is no prediction error, otherwise the performance error totally comes from the prediction error. Thus the physical parameters can be predicted according to these extracted trajectories of $\dot{u}(\theta)$ in simulations.

The Levenberg-Marquardt algorithm was used for training ANN. All functions were available in the neural networks toolbox of Matlab. The architecture of the neural network and the training parameters were shown in Fig. 9

As the result, the training sets were built as the flow chart in Fig. 10. For every successful simulations, the trajectory of θ evolution of $\dot{u}(\theta)$ when $-0.05 \leq \theta \leq 0.05$ was extracted. Then the trajectory was fitted by quadratic curve fitting, and the parameters of quadratic curve were recorded as the

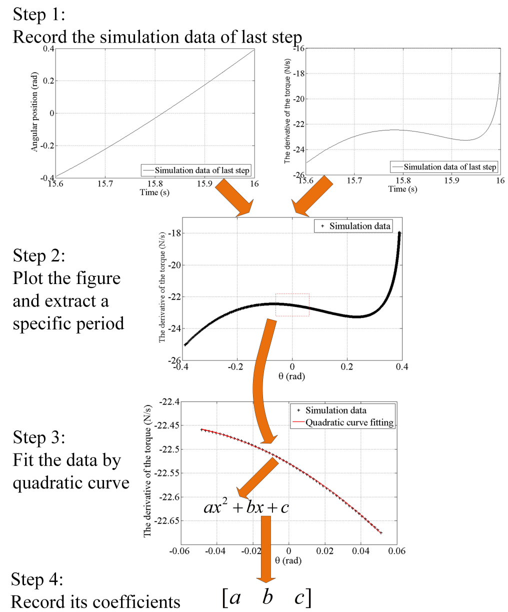

Fig. 10: Flow chart of building training sets

training set with the random prediction parameters. Thus the physical parameters could be predicted through setting $a = b = c = 0$ (the trajectory $\dot{u}(\theta) = 0$).

As a result, the training dataset consists of 37 training patterns $(\boldsymbol{x}_j, \boldsymbol{t}_j)$, where j is the index number. The input vector and output vector are shown below

$$\boldsymbol{x}_j = [a_j, b_j, c_j] \in R^3,$$
$$\boldsymbol{t}_j = [m_j, l_j, k_j] \in R^3.$$

The procedure used for designing the neural network for identification can be summarized as follows:

1) Conduct the simulations under 37 sets of random physical parameters $[m, l, k]$ around the certain range.
2) In each simulation, the trajectory of θ evolution of $\dot{u}(\theta)$ when $-0.05 \leq \theta \leq 0.05$ was extracted and fitted by quadratic curve fitting to obtain the coefficient parameters $[a, b, c]$.
3) Build the entire training set. The curve-fitting parameters $[a, b, c]$ were assigned as input, while the physical parameters $[m, l, k]$ were assigned as output of the neural network.
4) Train and build the neural network, calculate the solution of $a = b = c = 0$ as the prediction parameters.

Through this method, the prediction error was greatly reduced, however, the error could not reach 0 because the influences of linearization could not be completely eliminated.

The real value, prediction and relative error of the physical parameters were shown in Table IV. The performances of ANN were illustrated in Figs. 11 and 12. As a result, the torque curve under the predicted physical parameters are illustrated in Fig. 13. Several ANNs were built based on the same training sets, and the average error of prediction was about 2%.

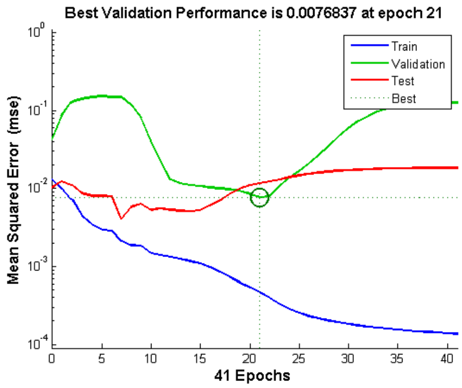

Fig. 11: The performance of proposed ANN

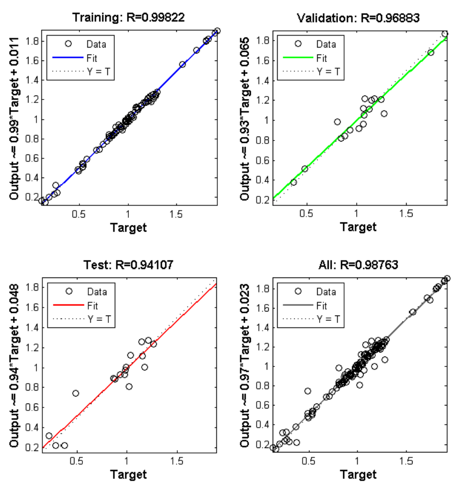

Fig. 12: The linear regression plot on training validation and test sets

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed the model-based feedback control to generate target walking speed and identify the physical parameters on a limit cycle walker, thus provided us a novel strategy of model-based controller optimization. The CRW was proposed and its dynamic equation of motion was derived. Thus the feedback control of walking speed was developed through the dynamic torque planning. The ability of tolerating prediction error was tested on numerical simulations by using a vary of prediction parameters. As a result, the controller was proved to have a certain adaptability to the prediction error, and the target walking speed could be guaranteed with little error (0.001%). In addition, the neural network parameter identification model was also proposed with the error of 2% based on the walker performance.

The ability of error toleration will be further tested as the future work. In addition, the controller will be extended to some other walkers with multiple DoFs.

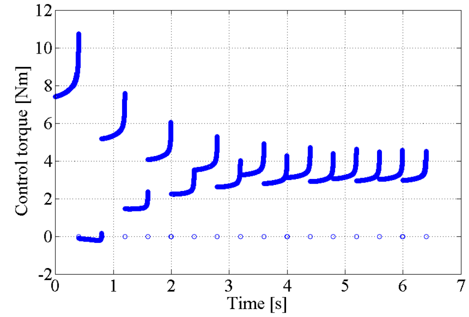

Fig. 13: Torque input of simulation under the predicted parameters

TABLE IV: Predicted parameters through ANN

Parameter	Physical	Predicted	Prediction error
m/[kg]	1.0	0.9990	0.1%
l/[m]	1.0	0.9711	2.89%
k_{v}	0.5	0.4984	0.32%

REFERENCES

[1] T. McGeer, "Passive dynamic walking," *Int. J. of Robotics Research*, Vol. 9, No. 2, pp. 62–82, 1990.

[2] M. J. Coleman, A. Chatterjee and A. Ruina, "Motions of a rimless spoked wheel: a simple three-dimensional system with impacts," *Dynamics and Stability of Systems*, Vol. 12, No. 3, pp. 139–159, 1997.

[3] M. J. Coleman, "Dynamics and stability of a rimless spoked wheel: a simple 2D system with impacts," *Dynamical Systems*, Vol. 25, No. 2, pp. 215–238, 2010.

[4] D. G. E. Hobbelen and M. Wisse, "A disturbance rejection measure for limit cycle walkers: The gait sensitivity norm," *IEEE Trans. on Robotics*, Vol. 23, No. 6, pp. 1213–1224, 2007.

[5] D. G. E. Hobbelen and M. Wisse, "Controlling the Walking Speed in Limit Cycle Walking," *Int. J. of Robotics Research*, Vol. 27, No. 9, pp. 989–1005, 2008.

[6] S. Kajita, F. Kanehiro, K. Kaneko, et al. "The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation," *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on. IEEE*, pp. 239-246, 2001.

[7] J G. Juang, "Fuzzy neural network approaches for robotic gait synthesis," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, Vol. 30, No. 4, pp. 594-601, 2000.

[8] K. S. Narendra and K. Parthasarathy, *Identification and control of dynamical systems using neural networks. IEEE Transactions on neural networks*, Vol. 1, No. 1, pp. 4-27, 1990.

[9] X. M. Ren, A. B. Rad, P. T. Chan, W. L. Lo, *Identification and control of continuous-time nonlinear systems via dynamic neural networks. IEEE Transactions on Industrial Electronics*, Vol. 50, No. 3, pp. 478-486, 2003.

[10] F. Asano, "Stability analysis of passive compass gait using linearized model," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 557–562, 2011.

[11] F. Asano and X. Xiao, "Role of deceleration effect in efficient and fast convergent gait generation," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 5649-5654, 2013.

[12] X. Xiao, and F. Asano, "Generating 1-DOF limit cycle walking at target walking speed by feed-forward and feedback limit cycle control," *Multibody System Dynamics* Vol.40, No. 2, pp. 155-175, 2017.

[13] X. Xiao, O. Ma, and F. Asano, "Control walking speed by approximate-kinetic-model-based self-adaptive control on underactuated compass-like bipedal walker, " *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 4729-4734, 2017.