Neural Network Model for Balancing a Biped Robot

Daniel Zaldívar [1,2], Erik Cuevas [1,2], Marco A. Pérez-Cisneros[2], Raúl Rojas [1]
[1] Freie Universität Berlin, Institüt für Informatik, Takustr. 9, D-14195
Berlin, Germany
{zaldivar, cuevas, rojas}@inf.fu-berlin.de
http://www.inf.fu-berlin.de/~zaldivar
[2] Instituto de Robótica y Sistemas Inteligentes, CUCEI, Universidad de Guadalajara
marcopc@cucei.udg.mx
Blvd. Marcelino García Barragan 1421, C.P. 44430, Guadalajara, Jal., México
http://irsi.cucei.udg.mx

(Paper received on August 12, 2006, accepted on September 27, 2006)

Abstract. In bipedal walking, stable balance and walking sequence are essential. In this work, one neural network is proposed to model the balance dynamic of a biped robot. The generalizing ability of back propagation neural network is used to agilely characterize the performance of a fuzzy PD incremental algorithm based on the Zero Moment Point (ZMP) criteria to balance a real biped robot structure. The effectiveness of the implemented neural model is demonstrated by comparison between its output -the predicted robot's ZMP, and the real robot's ZMP value. Some training algorithms are used to model the biped balance and its results are reported.

1. Introduction

In recent years, there is enthusiasm to study the bipedal walking as private companies such as Sony, Honda, etc., alongside other research institutes and universities have invested huge amounts of human and economic resources to develop sophisticated biped robots prototypes [1], [2], [3]. However, some researchers have followed a rather low-cost biped robot design philosophy. Such kind of biped robots is similar to its costly counterpart affording similar capacities to study and improve new biped walking algorithms that in turn have resulted rather convenient. Therefore the trend of building low-cost biped robots has been increasing worldwide [4], [5].

In traditional legged robots, stability is maintained by keeping at least three contact points on the ground surface at all times. Within biped machines, only two points are actually contacting the ground surface that endorses the importance of implementing novel algorithms to achieve balance.

There are some techniques to implement a balance control of a biped robot. Many of them are implemented using classic control techniques while some others use either soft-computing or artificial intelligence techniques. In this work, an incremental fuzzy PD controller is employed to achieve balance on a biped robot [6]. One hybrid dynamic approach model for biped robots is proposed. It combines the inverted pendulum model approach to characterize the biped's walking and one back-propagation

Special Issue in Neural Networks and Associative Memories
Research in Computing Science 21, 2006, pp. 141-150

neural network system identification approach to model the biped's balance. The neural network, predicts the behavior of the ZMP during walking.

In order to test the balance control, the "Dany Walker" biped robot structure was designed. It has 10 degrees of freedom (DOF) with each joint being driven by a DC-servo motor (Fig. 1 left). One modular design was chosen to allow an easy assembly and to ease reconfiguration to support several DOF setups. In real-time biped robot structures, a feedback-force system at each foot has to be implemented to obtain the ZMP that is then fed into the incremental fuzzy PD controller, calculating the ZMP error. Then the controller adjusts the lateral robot's positions to keep the ZMP point within the support region [6].

The dynamic of a biped robot is closely related to its structure and its mass distribution [5]. Therefore the movement of the Center of Mass (COM) will have a significant impact on the overall robot stability.

In order to achieve static stability, we place the COM as lower as possible. To such a purpose, a short leg's position was used (Fig. 1, left and right).

To compensate the disturbances during walking, lateral movements of the robot were enabled by mechanical design. Thus, it was possible to control the lateral balance of the robot by swinging the waist using 4 motors, two at the waist and two over the ankles (Fig. 1 right).

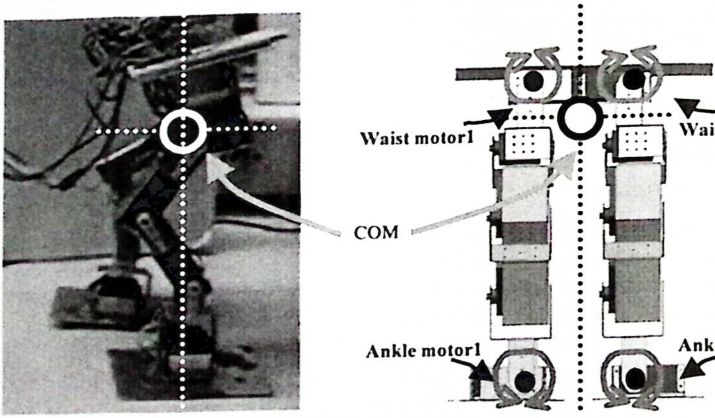


Figure 1. Designed biped robot structure "Dany Walker": left) Real structure, right) CAD design.

This paper is organized as follows: In section 2, the biped balance theory is briefly described. Section 3 presents the dynamic robot's model while Section 4 discusses some conclusions.

2. Biped balance theory

In dynamic walking, the important control criterion is to keep the *Zero Moment Point* (*ZMP*) within the support region (from now on, this criteria is mentioned as the "*ZMP criteria*"). The use of *ZMP criteria* has been broadly used to generate biped control algorithms [2], [3].

2.1. ZMP

The *ZMP* represents a point on the ground where the sum of all momentums is zero. Using this principle, the *ZMP* can be computed as follows:

$$x_{ZMP} = \frac{\sum_i m_i(\ddot{z}_i+g)x_i - \sum_i m_i \ddot{x}_i z_i - \sum_i I_{iy}\ddot{\theta}_{iy}}{\sum_i m_i(\ddot{z}_i+g)} \tag{1}$$

$$y_{ZMP} = \frac{\sum_i m_i(\ddot{z}_i+g)y_i - \sum_i m_i \ddot{y}_i z_i - \sum_i I_{ix}\ddot{\theta}_{ix}}{\sum_i m_i(\ddot{z}_i+g)} \tag{2}$$

where *(x_{ZMP}, y_{ZMP})* are the *ZMP* coordinates, *(x_i,y_i,z_i)* is the mass centre of the link *i* in the coordinate system, *m* is the mass of the link *i*, and *g* is the gravitational acceleration. I_{ix} and I_{iy} are the inertia moment components, θ_{iy} and θ_{ix} are the angular velocity around the axes *x* and *y* (taken as one point from the mass centre of the link i). The biped balance is achieved when the ZMP is controlled and continuously corrected to fall inside of the boundaries of the support region [6].

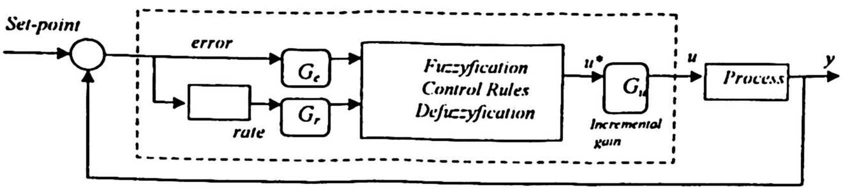


Fig. 2. Fuzzy PD incremental algorithm structure.

2.2. Balance control algorithm

In this work, the fuzzy PD algorithm for incremental control is implemented to balance the robot [6]. The fuzzy PD incremental control algorithm has the structure illustrated in Fig. 2. Gains *Gu, Ge* and *Gr* represent the output gains as determined by tuning. They correspond respectively to the error (*ZMP error*) and error rate (*ZMP rate*) gains. The value *u** is the defuzzyficated output, also known as "crisp output". The value *u* is defined by:

Fig. 6 shows the architecture used to train a back-propagation neural network and identify the biped robot's ZMP dynamic model. First, from the real biped robot structure, (real robot's dynamics) the ZMP is obtained (ZMP (k)) and feed it to the incremental fuzzy PD controller. The controller produces an output *M(k)* (lateral motors output) to correct the ZMP inside of the support polygon. *M(k-1)*, *M(k-2)* and *ZMP(k-1)* are respectively the controller output delayed one time unit, two time unit, and the ZMP delayed one time unit.

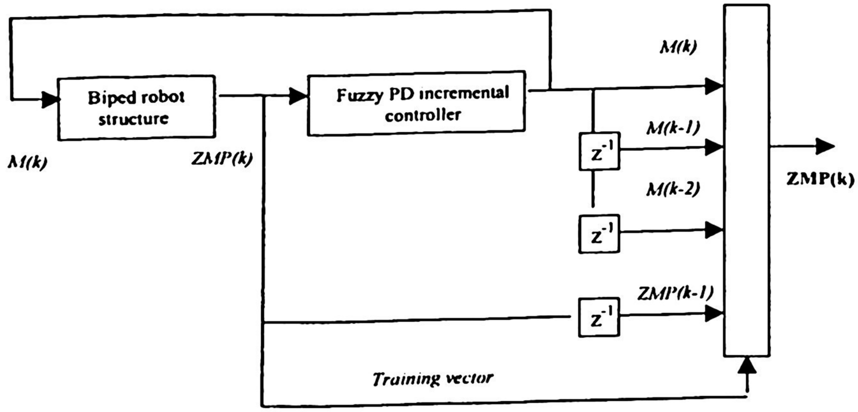


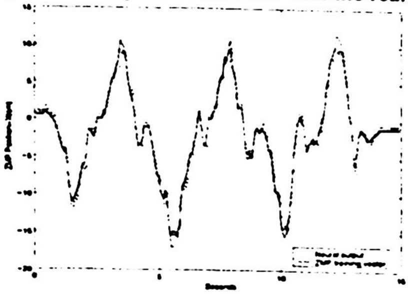
Fig. 6. Architecture to identify the biped robot's ZMP dynamic model.

Thus, to model the biped robot's balance dynamics, a back propagation neural network with four input neurons and an output neuron and with linear output activation function, was choose. The network was trained offline in batch mode, using data collected from the real walking operation of the biped robot. Some different training algorithms were tested for the network training, each; obtain a different biped robot's ZMP dynamic model performance.

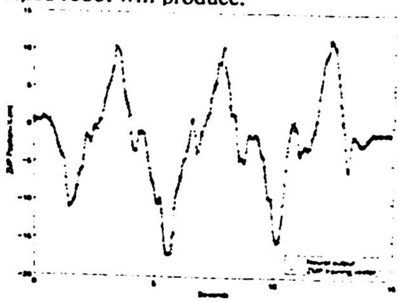
3.1.1. Neural Network model's performance

The in general a neural network performance could depend on many factors, including the complexity of the problem, the number of data points in the training set, the number of weights and biases in the network, the error goal, and the application it self (discriminant analysis, regression, etc). The last, is our case, since the goal is to find, means a neural network, a function approximation which model the biped robot's ZMP dynamic. The criteria to know which training algorithm better describes the ZMP robot's dynamic at walking will be a compromise between the velocity and economy of the algorithm.

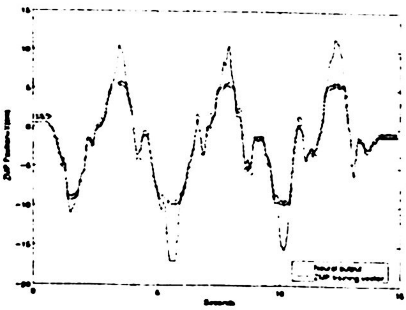
The neural network was training using different training methods. To test the performance of each of them, the controller's output at walking was feed to the neural network. Expecting that the neural network, now trained with the biped's ZMP dynamics, be able to predict the ZMP that the real biped robot will produce.



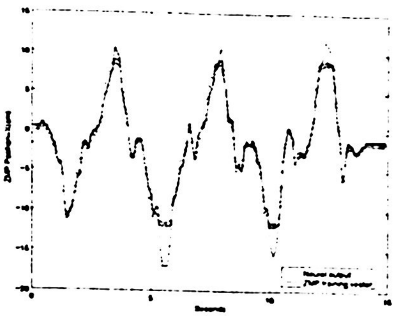
a) Levenberg-Marquardt



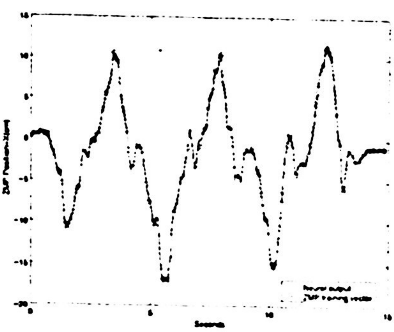
b) Resilient Back-propagation



c) Scaled Conjugate Gradient



d) One-Step Secant



e) BFGS Quasi-Newton

Fig. 7. Performance of some neural network training algorithms to proximate the real ZMP.

In Figure 7 a data set of ZMP values obtained at real walking, are compared with the ZMP produced by the neural network using different training algorithms. Fig. 7, shows the performance with the next training algorithms: a) Levenberg-Marquardt, b) Resilient Back-propagation, c) Scaled Conjugate Gradient, d) One-Step Secant and e) BFGS Quasi-Newton (Broyden, Fletcher, Goldfarb, and Shanno (BFGS)).

4. Conclusions

A neural network used to model the nonlinear biped robot's lateral movements dynamic was implemented. The strategy was to use a neural network as a system identifier; in this case the system to be identified is the biped robot's lateral movement's dynamics. A part of the lateral movements are generated by the fuzzy controller to correcting the ZMP. The ZMP dynamic was the parameter learned by the neural network. Some different training methods were used to compare the performance of the neural network to approximate the real robot's ZMP dynamic at walking. In all the different training algorithms, a back-propagation neural network architecture was choose. From each tested algorithms can be concluded:

a) Levenberg-Marquardt training algorithm

In general, this algorithm has the fastest convergence on function approximation problems. This advantage is especially noticeable if very accurate training is required. In many cases, Levenberg-Marquardt training algorithm is able to obtain lower mean square errors than any of the other algorithms tested. However, as the number of weights in the network increases, the advantage of the Levenberg-Marquardt training algorithm decreases. The performance of the algorithm to approximate the ZMP dynamics was quite gut (Figure 7(a)). A disadvantage, was that the storage requirements of Levenberg-Marquardt training algorithm were larger than the other tested algorithms.

b) Resilient Back-propagation training algorithm

The Resilient Back-propagation training algorithm is the fastest algorithm on discriminant analysis problems. However, in general it does not perform well on function approximation problems. Its performance also degrades as the error goal is reduced. An advantage is that its memory requirements are relatively small in comparison to the other tested algorithms. Figure 7(b), shows the performance of the resilient back propagation training algorithm to model the biped robot's ZMP dynamics.

c) Scaled Conjugate Gradient (SCG) training algorithm

The SCG algorithm demonstrated to be almost as fast as the Levenberg-Marquardt training algorithm on the approximation of the biped balance dynamics. How ever,

Figure 7 c) shows that its performance to model the biped robot's ZMP dynamics was inferior to the two first training algorithms. An important advantage is that the conjugate gradient algorithm has relatively modest memory requirements.

d) One-Step Secant training algorithm (OSS)

The one step secant (OSS) method is an attempt to bridge the gap between the conjugate gradient algorithms and the quasi-Newton (secant) algorithms. This algorithm does not store the complete Hessian matrix; it assumes that at each iteration, the previous Hessian was the identity matrix. This has the additional advantage that the new search direction can be calculated without computing a matrix inverse. However, Figure 7(d) shows that the performance of the OSS training algorithm to model the biped robot's ZMP dynamics was even inferior to the first tree training algorithms. An advantage is that it required less storage and computation per epoch than the BFGS algorithm, but required slightly more storage and computation per epoch than the conjugate gradient algorithms. It can be considered a compromise between full quasi-Newton algorithms and conjugate gradient algorithms.

e) BFGS Quasi-Newton training algorithm

In Figure 7(e) and 7(a) a similar performance between the Quasi-Newton and Levenberg-Marquardt training algorithm can be observed. The Quasi-Newton does not require as much storage as Levenberg-Marquardt training algorithm, but the computation required does increase geometrically with the size of the network, since the equivalent of a matrix inverse must be computed at each iteration.

As a result from these test, the BFGS Quasi-Newton training algorithm to model the robot's ZMP dynamics is prefer for its convenient relationship between computational economy and fast convergence.

References

[1] Hirai, K.H., Haikawa, M., Takenaka, Y. (1998). The development of Honda humanoid robot. Proceedings. IEEE International Conference in Robotics and Automation. Leuven, Belgium. (1998)

[2] Takanishi, A., Ishida, M., Yamazaki, A., Kato, I. (1985). The realization of dynamic walking robot WL-10RD, in Proc. 1985 Int. Conf. Advanced Robotics, pp. 459-466.

[3] Takanishi, A., Egusa, Y., Tochizawa, M., Takeya, T., Kato, I. (1988). Realization of dynamic biped walking stabilized with waist motion. Proceeding of the Seventh CISMIFTOMM Symposium on Theory and Practice of Robots and Manipulators, pp 68-79.

[4] Nicholls, E.: Bipedal dynamic walking in robotics (1998). Dissertation thesis, University of Western Australia, Department of electrical and Electronics Engineering.

[5] Cuevas, E. V., Zaldivar, D., Rojas, R. (2004). Bipedal robot description. Technical Report B-03-19, Freie Universität Berlin, Fachbereich Mathematik und Informatik, Berlin, Germany.

[6] Cuevas, E. V., Zaldivar, D., Rojas, R. (2005). Incremental fuzzy control for a biped robot balance IASTED International Conference on ROBOTICS AND APPLICATIONS ~RA2005~ Cambridge, USA.
[7] Norgaard, M., Ravn, O., Poulsen, N. K., Hansen, L. K. (1999). Neural Networks for Modelling and Control of Dynamic Systems. Springer Verlag, Berlin.