Learning to Walk Using a Recurrent Neural Network with Time Delay

Boudour Ammar[1], Naima Chouikhi[1], Adel M. Alimi[1], Farouk Chérif[2], Nasser Rezzoug[3], and Philippe Gorce[3]

[1] REGIM: REsearch Groups on Intelligent Machines, University of Sfax, National Engineering School of Sfax (ENIS), BP 1173, Sfax, 3038, Tunisia
{boudour.ammar,adel.alimi}@ieee.org
[2] ISSATS, Laboratory of Math Physics, Specials Functions and Applications, LR11ES35, Ecole Supérieure des Sciences et de Technologie, 4002- Sousse- Tunisia
faroukcheriff@yahoo.fr
[3] HandiBio Laboratory, EA 3162, Avenue de l'université, University of South Toulon-Var, BP 20132, 83957 La Garde cedex, France

Abstract. Walking based on gaits is one of the most approved methodologies for walking robot. In this paper, we develop learning strategy for walking biped robot or human based on a self made database using biomechanical capture. This system is provided by a Recurrent Neural Network (RNN) with an internal discrete time delay. The role of the proposed network is the training of human walking data by giving an estimation of the biped's next position at each time and achieve a human-like natural walking. Different architectures of RNN are proposed and tested. In Particular, a comparative study is given and the results of the RNN mixed with extended Kalman filter are illustrated.

Keywords: Human walking, 3D human simulator, Recurrent Neural Network, Biomechanics, e-kalman filter.

1 Introduction

In a few years, robotics has become an important science that is constantly evolving. Within this issue, bipedal locomotion proposals [12] focus on different aspects of robot control. In fact, robots have demonstrated an ability to try to mimic humans in achieving activities in many fields such as industry, home automation, exploration, games and so forth. In all these fields, robots need to displace and move in order to realize their missions. The bipedal locomotion is supposed to be one of the best locomotion systems according to small support surface [2].

Artificial Neural Networks (ANNs) are computational tools, based mainly on the properties of biological neural systems as its network theory revolves around the idea that certain properties of biological neurons can be extracted and applied to simulations [14].

ANNs can be defined as an attempt to somehow imitate the neural network of a human being in a simple way. Although the ANNs can learn in the presence of noisy inputs, they cannot perform to react convincingly and dynamically to a real world situation. If we need to keep in memory the previous states of the network, it would be

V. Mladenov et al. (Eds.): ICANN 2013, LNCS 8131, pp. 511–518, 2013.

better to choose the RNN architecture to perform complex tasks ([1] and [9]). Several types of RNN have been used in the past years, for example: cellular neural networks, Hopfield neural networks, Cohen-Grossberg neural networks, bidirectional associative memory neural net-works. Besides, it is well known that there are time delays in the information processing of neurons. So, the RNN with time delays has received much more attention both in theory and in practice ([1], [18], [4]).

In addition, the use of the RNNs can be justified by their dynamism and powerful algorithms of training.

Throughout this paper, we are, mainly, interested in the learning of the human gait to a 3D biped simulator via a RNN. The main contributions of this paper are summarized below:

– Preparation of human walking database based on biomechanic study
– Novel model and architecture of RNN with discrete time delay
– Comparison between some RNN architectures and RNN mixed with e-kalman

By the end of the training, the simulator is able to estimate in each time step the next position during the gait trajectory. A simple comparison proves the effectiveness of a RNN with time delay .

The plan of this paper is organized as follows: in section II, we give an overview about the prepared database of gaits following biomechanic steps. We, also, introduce the RNN model which can be used in the training algorithm and the walking learning system. In Section III, the different architectures of RNN and results are given after training and a comparative study is presented. Finally, section IV draws the conclusions and provides suggestions for future work.

2 Training with RNN

The recurrent neural network is a class of neural network where connections between the units form a cycle. This can lead to an internal dynamic temporal behavior ([10] and [8]). As a Neural Network, the RNN has input, hidden and output layers where each layer has a number of neurons. Besides, in our system, the learning process is supervised and requires a training data set.

2.1 Training Data Set

The learning process consists of two main parts which are training and testing.

In fact, the data set is composed of five walking tests for six subjects. The test capture saves the changes of the joint angles of the different articulations of the two legs.

To do so, a set of distinguishable markers (59 markers) are placed on six human body landmarks. The scene is captured by an opto-electric system composed of a number of calibrated and synchronized cameras [16].

The system of measure consists of some steps according to biomechanical study. In order to establish a relation between the markers, a Human Body Model (HBM) is defined and our own database of gaits is built. Markers are detected on all the camera views and delivered as the input of a particle scheme where every particle encodes an instance of the estimated position of the HBM [17].

So, we save the human movements while walking and we extract the marker's trajectories. The joint trajectories are deducted from the markers position in each instance using some implemented functions that follow a biomechanical study. We mention here that not all the markers are used and only anatomical and technical markers in the lower trunk are implemented because we are concerned up to now with the learning of biped walking. To train the network, the database consists of 10 joint angles at instant *t* and *t+1* as inputs and outputs respectively. Once the database is finished, we can move to the training and test phases.

2.2 The Proposed RNN Model

Many phenomena exhibit a great regularity without being periodic. This is modeled using the notion of pseudo almost periodic functions which allows complex repetitive phenomena to be represented as an almost periodic process plus an ergodic component. Besides, it is well-known that there are time delays in the information processing of the neurons due to various reasons. For instance, the time delays can be caused by the finite switching speed of amplifier circuits in neural networks or deliberately introduced to achieve tasks of dealing with motion-related problems.

The proposed NN model is studied with the aim of achieving human-like performance, especially in the task of human walking. It is an application of the proposed RNNs with time-varying coefficients and discrete delay detailed in [1]. Using the technique of the Euler discretization of the simplified continuous formulation, the approximation of the equation is:

$$S_i(t+1) = S_i(t) + \sum_{j=0}^{n} (c_{i,j}(t) f_j(S_j(t)) + d_{i,j}(t) g_j(S_j(t-\tau))) + J_i. \tag{1}$$

Where: n is the number of neurons in each network, $S_i(t)$ denotes the state of the ith neuron at time t, f_j and g_j are the activation functions of the jth neuron at time t, $c_{(i,j)}(t)$ and $d_{(i,j)}(t)$ are the connection weights of the jth neuron on the ith neuron, τ is constant time delay and $J_i\,(t)$ is the external bias on the *i*th neuron.

The proposed architecture of the RNN is composed of ten inputs, ten outputs and a variable number of neurons in a hidden layer. We choose the joint angles at instant *t*, as inputs and the joint angles at instant *t+1*, as outputs.

This network aims to obtain a human-like robot behavior and estimates the next values of the different joints.

2.3 BPTT Training Algorithm

In this paper, we adopted the Back Propagation Through Time (BPTT) as a training algorithm. In fact, the RNNs are unfolded in time; which means that they can be transformed from the feedback structure to the feed forward structure [3].

Then, an error is calculated at each unfolding step beginning with the last layer and retro-propagated to the previous layer which propagates its error forth until reaching the first layer.

A database with ten inputs and ten outputs is introduced to the network. The angles of the articulations of the lower trunk at instant *(t)* and *(t+1)* are respectively the inputs and outputs. Then, the data set is divided into trained data set (2400 steps of 8 tests of human subjects) and test data set (300 steps of 1 test of human subject). The last database is used to calculate the performance of the RNN and the error is computed according to equation 2. For predictive analytics, the error function is the sum-of-squared errors [8]. This function is calculated by looking at the squared difference between what the network predicts for each training pattern and the target value, or observed value, for that pattern. The global error is, then, deduced from the equation 3.

$$E_k = \frac{1}{2}(S_d(t+k) - S(t+k))^2. \tag{2}$$

Where: k is the index of the step or sample, $S_d(t+k)$ and $S(t+k)$ are respectively the desired output vector and the output vector of the network in this step.

$$E = \frac{1}{N}\sum E_k. \tag{3}$$

Where: N is the number of samples.

Some parameters, which can be adjusted in the RNN, are increasing the iteration number or/and adding new hidden layers or/and altering the number of neurons in each hidden layer. The feedback between neurons from the input, hidden and output layers is tested and explained below. The update of weights can be performed with e-kalman filter. The last is an algorithm that uses a series of measurements observed over time, containing noise (random variations) and other inaccuracies, and produces estimates of weight that tend to be more precise than those based on BPTT algorithm alone [15].

Table 1. Error values of different architectures of NN with a variation of time delay

Characteristics of the NN	**Delay**	**Error**
Neural Network	0	1.7210^{-4}
RNN1	1	8.910^{-5}
RNN2	1	4.710^{-5}
RNN3	1	3.910^{-5}
RNN4	1	7.610^{-5}
RNN5	0	3.310^{-3}
RNN6	2	5.210^{-3}
RNN7	1	2.2410^{-4}
RNN8	1	1.510^{-4}
RNN9	1	2.2310^{-4}
RNN with e-kalman	**1**	1.8810^{-5}

Where: RNN1 : RNN with connections between **5** neurons in hidden layer.
RNN2 : RNN with connections between **8** neurons in hidden layer.
RNN3 : RNN with connections between **10** neurons in hidden layer.
RNN4 : RNN with connections between **12** neurons in hidden layer.
RNN5 : RNN with connections between **10** neurons in hidden layer where delay=0.

RNN6 : RNN with connections between **12** neurons in hidden layer where delay=2.
RNN7 : Elmann network with connections between **10** neurons in hidden layer and input layer.
RNN8 : Output recurrent network with connections between the output layer and input layer (10 neurons in hidden layer).
RNN9 : Recurrent neural network with connections between hidden (10 neurons) and input layers and between output and input layers.

3 Tests and Evaluation

Neural architecture considered in this investigation is a recurrent and multi-layer neural network. This architecture is composed of three layers arranged in a feed-forward model: The first layer is the input layer with a vector of 10 inputs, which are the joints of the lower trunk at instant t (two joints of each hip, one joint of each knee and two joints of each ankle). The second layer is the hidden layer where the number of neurons is fixed after many tests. The Third layer is the output vector that contains the 10 joint angles at instant $t+1$.

In order to evaluate the performance of each RNN, we calculate and compare the errors of many structures of the RNN. We start with the test of a simple NN. Then, we change the number of interconnected neurons in hidden layer (RNN1 to RNN6). After that, we add connections between neurons in different layers (RNN7 to RNN9). The RNN with e-kalman is, also, tested. It is a RNN for EKF-BPTT training. So, the filter of e-kalman are used in the BPTT algorithm and exactly in the step of update weights [5]. The goal of this evaluation is to compare several architectures of the RNN when a time delay is included in the dynamic equation.

So, table 1 gives a comparison between the error results of RNN with a variation of feedback connections between neurons and the different time delays (*delay=0, delay=1 and delay=2*). Results are summarized in table 1 and presented in figure 2 which clearly demonstrate the following:

1. The RNN with delay different to 1 have the worst performance. It reveals that wrong delay that occured at the beginning of each recursion will accumulate and propagate to the future when recursively, which will result in poor forecast accuracy.
2. RNN with ten neurons with connections in the hidden layer and a discrete delay equal to 1, gives a good performance.
3. the RNN mixed with e-kalman gives the best performance. It is clear that the last RNN is more important than the others. After computing, the error reached by the end of the training (600 iterations) is equal to 1.8810^{-5}. Figure 2 presents the desired right knee angle (in green), the estimated output of RNN with ten neurons with connections in the hidden layer (in red) and the output of the RNN mixed with e-kalman filter(in blue *). The two last curves are superposed because, as it is known, errors are in the order of 10^{-5}.

This architecture is described in figure 1 and characterized by one feedback connection which is a recursive connection between the neurons in the hidden layer. The numbers of the processing neurons and layers determined above are identified the best by a number of trials. In order to validate our RNN system, the output results are tested in a 3D simulator of the human-like robot. In each time step during the test phase, the Center

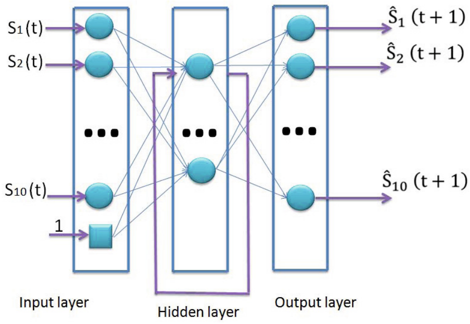

Fig. 1. The proposed RNN architecture

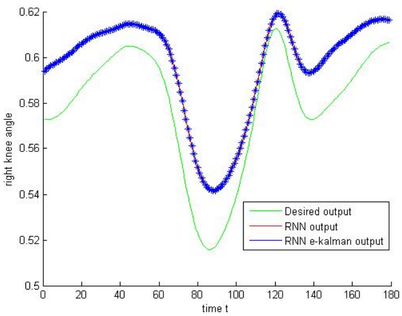

Fig. 2. Comparison between the desired output (green line), the RNN output (red line) and the output of RNN mixed with e-kalman (blue *)

Of Mass (COM) coordinates of the simulator are computed [13]. Then, the equations of stability of the simulator are calculated in order to avoid the risk of falling on the floor. When the simulator starts to walk, the stability can be controlled statically or dynamically. In both control manners, the COM of the simulator is computed.

Compared to [6] and [7], they predict positions of the center of mass of the biped. This prediction is controlled by the calculated ZMP stability. However, the COM coordinates information is not enough to control all the joints positions or angles.

In our case, the prediction of angles of articulations is controlled by the real data of the human subjects. The stability is controlled after the learning process.

Nakanishi et al report on their research for learning the biped locomotion from human demonstration. They choose to use database of walking joints of only one man from a book. Only the desired trajectories of the right leg are generated, while those of the left are concluded by shifting the phase of the oscillator of the right leg by π. The demonstrated trajectories are learned using locally weighted regression [11]. In the dynamical movement primitives, kinematic movement plans are described in a set of nonlinear differential equations with well-defined attractor dynamics. In our case, no differential equations of walking is used. The human motion is learned using RNN learning. The mentioned architecture of RNN using discrete delay is implemented with the technique of Euler discretization of the simplified continuous equation [1], the approximated equation is computed in the novel RNN architecture. The results of the proposed RNN are applied in 3D simulator.

The simulator is a geometric model similar to the human body. The 3D biped simulator has not only two legs and a pelvis but also a trunk, two hands and head. In our project, the higher part is immobile ; only the lower part is considered in the gait.

4 Conclusions

In this paper, the learning system of the human walking is proposed. This system is divided in two parts. The first is the building of the human walking model. A self made biomechanical data set of HBM walking is prepared. Only 10 joint angles at different instances are chosen from this data set. These joints are the inputs and the desired outputs of the RNN.

Different network architectures are tested. The training error is computed and the RNN with time delay and feedback connections in hidden layer and mixed with e-kalman has a good performance.

The second part is the prediction of the next joint angles in function of the actual joint angles. The role of the proposed RNN architecture is to obtain ”human-like robot behavior and predict the next values of the joint articulations”, but the RNN presented can only do prediction. To achieve control we, also, need other mechanisms (eg a PID or fuzzy controller on the joints). We mention that this system lacks the link between the learning step and the stability control.

As a future work, we attempt to expand the dynamic equation of the RNN by adding a continuous delay so as to manage the lateness of the network. We, also, aim at exploiting the described model to a Bidirectional Associative Memory (BAM) neural networks as a learning system in humanoid robotics. Another interesting topic for future work is by extending our approach with other databases of different motions (like: running, jumping, etc.). Furthermore, it is also planned to use this proposed system to learn elderly and handicapped persons.

Acknowledgment. The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program.

References

1. Ammar, B., Chérif, F., Alimi, A.M.: Existence and Uniqueness of Pseudo Almost-Periodic Solutions of Recurrent Neural Networks with Time-Varying Coefficients and Mixed Delays. IEEE Transactions on Neural Networks and Learning System 23(1) (2012)

2. Azevedo, C., Poignet, P., Espinau, B.: Artificial locomotion control: from human to robots. Robotics and Autonomous Systems 47, 203–204 (2004)
3. Baccour, N., Kaaniche, H., Chtourou, M., Ben Jemaa, M.: Recurrent Neural Network based time series prediction: Particular design problems. In: International Conference on Smart Systems and Devices, SSD (2007)
4. Bouaziz, S., Dhahri, H., Alimi, A.M.: Evolving Flexible Beta Operator Neural Trees (FBONT) for Time Series Forecasting. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part III. LNCS, vol. 7665, pp. 17–24. Springer, Heidelberg (2012)
5. Bousnina, S., Ammar, B., Baklouti, N., Alimi, M.A.: Learning system for mobile robot detection and tracking. In: International Conference on Communications and Information Technology, ICCIT, pp. 384–389 (June 2012)
6. Concalves, J.B., Zampieri, D.E.: Recurrent neural network approaches for biped walking robot based on zero-moment point criterion. J. Braz. Soc. Mech. Sci. and Eng. 25(1), 69–78 (2003)
7. Erbatur, K., Kurt, O.: Natural ZMP Trajectories for Biped Robot Reference Generation. IEEE Transactions on Industrial Electronics 56(3), 835–845 (2009)
8. Huynh, T.Q., Reggia, J.A.: Symbolic Representation of Recurrent Neural Network Dynamics. IEEE Transactions on Neural Networks and Learning Systems 23(10), 1649–1658 (2012)
9. Manabe, Y., Chakraborty, B., Fujita, H.: Structural learning of multilayer feed forward neural networks for continuous valued functions. In: The 47th Midwest Symposium on Circuits and Systems, vol. 3, pp. 77–80 (2004)
10. Mandic, D., Chambers, J.: Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability (2001)
11. Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., Kawato, M.: Learning from demonstration and adaptation of biped locomotion. Robotics and Autonomous Systems 47(23), 79–91 (2004)
12. Rokbani, N., Ammar, B., Alimi, A.M.: Toward intelligent biped-humanoids gaits generation. In: Humanoid Robots, pp. 259–271. I-Tech Education and Publishing, Vienna (2008)
13. Rokbani, N., Benbousaada, E., Ammar, B., Alimi, A.M.: Biped Robot Control Using Particle Swarm Optimization. In: IEEE International Conference on Systems Man and Cybernetics, SMC, pp. 506–512 (2010)
14. Stanton, C., Williams, M.-A.: Book Review: Robotics: State of the Art and Future Challenges. Artificial Intelligence (2008)
15. Welch, G., Bishop, G.: An introduction to the Kalman filter (1995)
16. Winter, D.A.: Biomechanics and motor control of human movement. John Wiley, USA (1990)
17. Wu, G.: Isb recommendation and definitions of joint coordinate system of various joints for the reporting of human joint motion. Part I: ankle, hip and Spine. Journal of Biomechanics 35 (2002)
18. Zhang, H., Liu, Z., Huang, G.: Novel Delay-Dependent Robust Stability Analysis for Switched Neutral-Type Neural Networks With Time-Varying Delays via SC Technique. IEEE Transactions on Systems, Man and Cybernetics -PART B: Cybernetics 40(6) (December 2010)