Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

ELSEVIER

Robotics and Autonomous Systems 42 (2003) 107–116

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Real time gait generation for autonomous humanoid robots: A case study for walking

Genci Capi [a,*], Yasuo Nasu [b], Leonard Barolli [c], Kazuhitsa Mitobe [b]

[a] *ATR Human Information Science Laboratories, Department 3, 2-2-2 Hikaridai, Seiko-cho, Soraku-gun, Kyoto 619-0288, Japan*
[b] *Faculty of Engineering, Yamagata University, Yamagata, Japan*
[c] *Faculty of Literature and Social Sciences, Yamagata University, Yamagata, Japan*

Received 19 February 2002; received in revised form 11 October 2002
Communicated by F.C.A. Groen

Abstract

As autonomous humanoid robots assume more important roles in everyday life, they are expected to perform many different tasks and quickly adapt to unknown environments. Therefore, humanoid robots must generate quickly the appropriate gait based on information received from visual system. In this work, we present a new method for real time gait generation during walking based on Neural Networks. The minimum consumed energy gaits similar with human motion, are used to teach the Neural Network. After supervised learning, the Neural Network can quickly generate the humanoid robot gait. Simulation and experimental results utilizing the "Bonten-Maru I" humanoid robot show good performance of the proposed method.
© 2002 Elsevier Science B.V. All rights reserved.

Keywords: Humanoid robots; RBF neural network; Genetic algorithm; Real time gait generation

1. Introduction

Humanoid robots, which have the same locomotive structure as humans, must operate in any environment in which a human can operate. Because the environments where humans operate are unpredictable, the robot must generate in real time the appropriate gait based on the information received from the visual system. The robot must walk with different step lengths, overcome obstacles, go up-stairs, etc. Until now, the humanoid robot gait is for the most part prescribed based on human motion [1]. However, measuring the angle trajectories during human walking for a wide range of step lengths and step times is difficult and time consuming [2]. In addition, the humanoid robot link dimensions and the number and distribution of degrees of freedom are not the same with those of humans. Therefore, the angle trajectories recorded from humans need to be manipulated to fit to humanoid robot specifications.

In order to generate a human like motion, we proposed a new method for optimal gait generation based on Genetic Algorithm (GA) [3]. The humanoid robot gait was generated using two different cost functions: minimum consumed energy (CE) and minimum torque change (TC). The results showed that minimum CE gait is similar to human gait. As well, the gait has a large effect on the energy needed during robot motion.

* Corresponding author. Tel.: +81-774-951031;
fax: +81-774-951259.
E-mail address: gcapi@atr.co.jp (G. Capi).

0921-8890/02/$ – see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0921-8890(02)00351-2

In real time situations when the humanoid robot has to change walking velocity or switch from walking to climbing stairs, the new optimal gait must be generated quickly. The GA converged to the optimal gait but time needed by GA was not suitable for real time situations. Prior works, which considered minimum energy gait of biped robots during walking by using conventional optimization methods [4,5], have also discussed the real time implementation problem. In [4], the authors suggest creating a database of pre-computed optimal gaits. However, the biped robot gait could only be generated for the step lengths and step times included in the database.

In order to generate a humanoid robot gait in a short enough time for real time applications, we utilize a Radial Basis Function Neural Network (RBFNN), which gives good results for the approximation problems. A Radial Basis Function (RBF) model provides high accuracy, fast training (identification), and is computationally and algorithmically simple. In many applications, the RBFNN approximation has superior accuracy and training time compared to Multilayered Perceptron Networks [6]. To teach the Neural Network (NN), the optimal gaits generated by GA are used as training data [3]. One of the advantages of RBFNN, compared with the database method suggested in [4] is that the RBFNN can be used to approximate any gait within the range of pre-computed optimal gaits. After training, the RBFNN can quickly generate the minimum CE gait.

In [3], we have shown that for every step length during walking there is an optimal step time for which the CE is minimal. Therefore, when the humanoid robot is constrained to walk with a particular velocity and step length, the RBFNN input variables will be step length and step time. When the only constrained is to walk with a particular step length, the optimal step time will be the best from the CE point of view. In this paper, we present the simulation and experimental results where the input variables of RBFNN are step length and step time.

The paper is organized as follows. Section 2 provides an introduction to RBFNN. Section 3 deals with the gait and body motion. In Section 4, the proposed method is discussed. Simulation and experimental results are given in Section 5. Finally, conclusions and future works are presented in Section 6.

2. RBFNN

The RBFNN belongs to a curve-fitting (approximation) problem in a high-dimensional space. According to this viewpoint, learning is equivalent to finding a surface in a multidimensional space that provides a best fit to the training data, with the criterion for "best fit" being measured in some statistical sense. Correspondingly, generalization is equivalent to the use of this multidimensional surface to interpolate the test data. Such a viewpoint is the motivation behind the method of radial-basis functions in the sense that it draws upon research work on traditional strict interpolation in a multidimensional space. In the context of an NN, the hidden units provide a set of "functions" that constitute an arbitrary "basis" for the input patterns (vectors) when they are expanded into the hidden space; these functions are called radial-basis functions. Radial-basis functions were first introduced in the solution of the real multivariate interpolation problem.

The RBFNN consists of one input layer, one output layer and one hidden layer. A schematic of RBFNN with L inputs and N outputs is depicted in Fig. 1, which looks like a common feedforward topology. Such a network implements a mapping $f:\mathbb{R}^L \rightarrow \mathbb{R}^N$. Each output of the network is computed according to the linear equation as follows:

$$\widehat{y} = f(x) = \sum_{k=1}^{M} w_k \varphi_k, \tag{1}$$

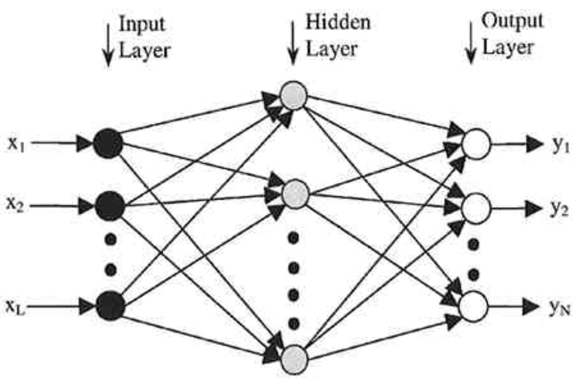

Fig. 1. Radial Basis Function Neural Network.

where

$$\varphi_k = \exp\left(-\frac{||x-c_k||^2}{\sigma_k^2}\right) \tag{2}$$

is the kth Gaussian function; $x \in \mathbb{R}^L$ is the input vector; w_k is the kth output of the hidden layer; $||\cdot||$ denotes the Euclidean norm; k is the index of the hidden nodes, $k = 1, 2, \ldots, M$; c_k is the center of the kth node; and σ_k stands for the width of the Gaussian function. Usually the nonlinear mapping of the hidden layer is carried out with the Gaussian function.

Let $x \in \mathbb{R}^L$, $y \in \mathbb{R}^N$ be the input and output of the system, respectively. Choosing the initial parameters of the RBFNN arbitrarily, the weight vector $w = [w_1, w_2, \ldots, w_M]^{\mathrm{T}}$ from the hidden layer to the output layer can be obtained from the linear equations as follows:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & a_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NM} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{bmatrix}, \tag{3}$$

where $a_{ij} = \varphi(||x_i - c_j||)$, $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M$; M is the number of the hidden nodes; and $e_i, i = 1, 2, \ldots, N$, are the errors of computation.

Based on the number of nodes in the hidden layer, the RBFNN are divided in generalized and regularization RBFNN. In the regularization network, the number of hidden neurons is the same with the training data. In the generalization RBFNN, the number of hidden neurons is smaller then the number of training samples. The hidden neurons are added one by one until the mean square error (mse) gets smaller than the desired one. The generalization RBFNN is used when the number of training data is large.

3. Biped model

During walking, the arms of the humanoid robot will be fixed on the chest. Therefore, it can be considered as a five-link biped robot in the saggital plane, as shown in Fig. 2. The motion of the biped robot is considered to be composed from a single support phase and an instantaneous double support phase. To satisfy repeatability conditions, the postures at the beginning and at the end of the step are considered to be the same. Also, the angular velocities of links 1 and 2 at the end of the step become equal to the angular velocities of links 5 and 4 at the beginning of the step, respectively. The friction force between the robot's feet and the ground is considered to be great enough to prevent sliding. During the single support phase, the ZMP must be within the sole length, so the contact between the foot and the ground will remain. In this paper, we calculate the ZMP by considering the link mass concentrated at one point [7]. To have a stable periodic walking motion, when the swing foot touches the ground, the ZMP must jump in its sole. We consider the body link acceleration to realize it. To have an easier relative motion of the body, the coordinate system from the ankle joint of the supporting leg is moved transitionally to the waist of the robot ($O_1X_1Z_1$). Referring to the new coordinate system, the ZMP position is written as

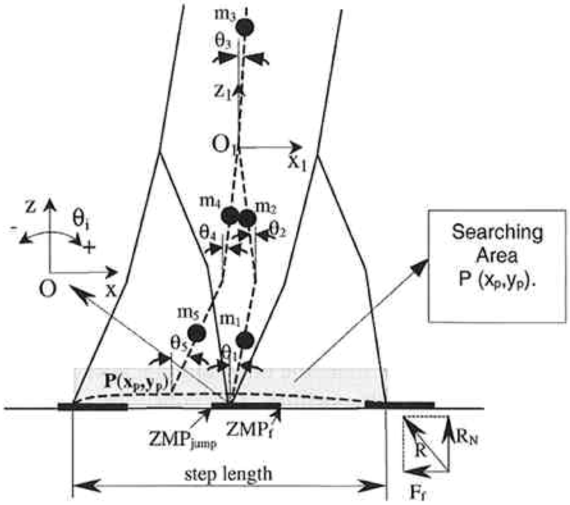

Fig. 2. Five-link biped robot.

$$\bar{X}_{\mathrm{ZMP}} = \frac{\sum_{i=1}^{5} m_i(\ddot{\bar{z}}_i + \ddot{z}_w + g_z)\bar{x}_i}{\sum_{i=1}^{5} m_i(\ddot{\bar{z}}_i + \ddot{z}_w + g_z)} - \frac{\sum_{i=1}^{5} m_i(\ddot{\bar{x}}_i + \ddot{x}_w)(\bar{z}_i + z_w)}{\sum_{i=1}^{5} m_i(\ddot{\bar{z}}_i + \ddot{z}_w + g_z)}, \tag{4}$$

where m_i is the mass of the particle i, x_w and z_w are the coordinates of the waist with respect to the coordinate system at the ankle joint of supporting leg, $\bar{x}_i$ and $\bar{z}_i$ are the coordinates of the mass particle i with respect to the $O_1X_1Z_1$ coordinate system, $\ddot{\bar{x}}_i$ and $\ddot{\bar{z}}_i$ the acceleration of the mass particle i with respect to the $O_1X_1Z_1$ coordinate system.

Based on formula (4), if the position, $\bar{x}_i$, $\bar{z}_i$, and acceleration, $\ddot{\bar{x}}_i$, $\ddot{\bar{z}}_i$, of the leg part ($i = 1, 2, 4, 5$), the body angle, θ_3, and body angular velocity, $\dot{\theta}_3$, are known, then because $\ddot{\bar{x}}_3$, $\ddot{\bar{z}}_3$ are functions of l_3, θ_3, $\dot{\theta}_3$, $\ddot{\theta}_3$, it is easy to calculate the body angular acceleration based on the ZMP position. Let subscripts (0) and (f) be the indexes at the beginning and at the end of the step, respectively. At the beginning of the step, $\ddot{\theta}_{30}$ causes the ZMP to be in the position $\mathrm{ZMP_{jump}}$. At the end of the step, the angular acceleration $\ddot{\theta}_{3\mathrm{f}}$ is calculated in order to have the ZMP at the position $\mathrm{ZMP_f}$, so that the difference between $\ddot{\theta}_{3\mathrm{f}}$ and $\ddot{\theta}_{30}$ is minimal. Therefore, the torque necessary to change the acceleration of the body link will also be minimal.

4. Real time gait generation method

We considered minimum CE as criterion for humanoid robot gait generation, because autonomous humanoid robots make difficult to use external power supply. In order to have a long operation time when a battery actuates the motors, the energy must be minimized. For minimum CE cost function, it can be assumed that the energy to control the position of the robot is proportional to the integration of the square of the torque with respect to the time. Because the robot joints are driven by torque, then the unit of torque, N m, is equal to the unit of energy, J. So, the cost function, J, can be defined as follows:

$$J = \frac{1}{2}\left(\int_0^{t_\mathrm{f}} \tau^\mathrm{T}\tau \,\mathrm{d}t + \Delta\tau_{\mathrm{jump}}^2 \,\Delta t + \int_0^{t_\mathrm{f}} C \,\mathrm{d}t\right), \quad (5)$$

where t_f is the step time, τ is the torque vector, $\Delta\tau_{\mathrm{jump}}$ and Δt is the addition torque applied to the body link to cause the ZMP to jump and its duration time, and C are the constraint function. The torque vector is calculated from the inverse dynamics of five-link biped robot [8] as follows:

$$J(\theta)\ddot{\theta} + X(\theta)\dot{\theta}^2 + Y\dot{\theta} + Z(\theta) = \tau, \quad (6)$$

where $J(\theta)$ is the (5×5) mass matrix, $X(\theta)$ is the (5×5) matrix of centrifugal coefficients, Y is the (5×5) matrix of Coriolis coefficients, $Z(\theta)$ is the (5×1) vector of gravity terms, τ is the (5×1) generalized torque vector, and θ, $\dot{\theta}$, $\ddot{\theta}$ are the (5×1) vectors of joint variables, joint angular velocities and joint angular accelerations, respectively.

To determine the humanoid robot optimal gait, a real-value GA was employed in conjunction with the selection, mutation and crossover operators. The GA generates randomly the first population, where every individual of the population presents a possible humanoid robot gait. According to formula (5), the minimum CE cost function is calculated and attached to every individual of the population. The GA moves from generation to generation, selecting parents and producing offspring until the termination criterion (maximum number of generations $\mathrm{GN_{max}}$) is met. In our method, we used normalized geometric ranking as a selection function. Based on the GA results, the minimum CE gait is obtained. In our simulations, GA needs about 10 minutes to determine the optimal gait.

In Fig. 3 is presented the method for creating different RBFNN modules. Every module presents a humanoid robot task. The creation of modules is carried out off time and separately for each task because the number of variables optimized by GA and searching spaces are different. GA generates the optimal gaits for a wide range of motion parameters, like step length and step time during walking, or step depth, step rise and step time during going up-stairs, which are used to teach the RBFNNs. Up to now, we have created the walking and going up-stairs modules.

In Fig. 4 is shown the schematic diagram of real time gait generation by walking module. The CCD cameras transmit the image information to image recognition module. After the image is analyzed, the RBFNN module and input variables are determined. As can be seen, there are two modules for walking, which differ from the number of input and output variables. The first module will be simulated to generate the humanoid robot gait when the robot is constrained to walk with a determined velocity for a given step length. In this case, the input variables of RBFNN are the step length and step time. The output variables are the same with variables optimized by GA. As presented in [3], every step length is optimal at a

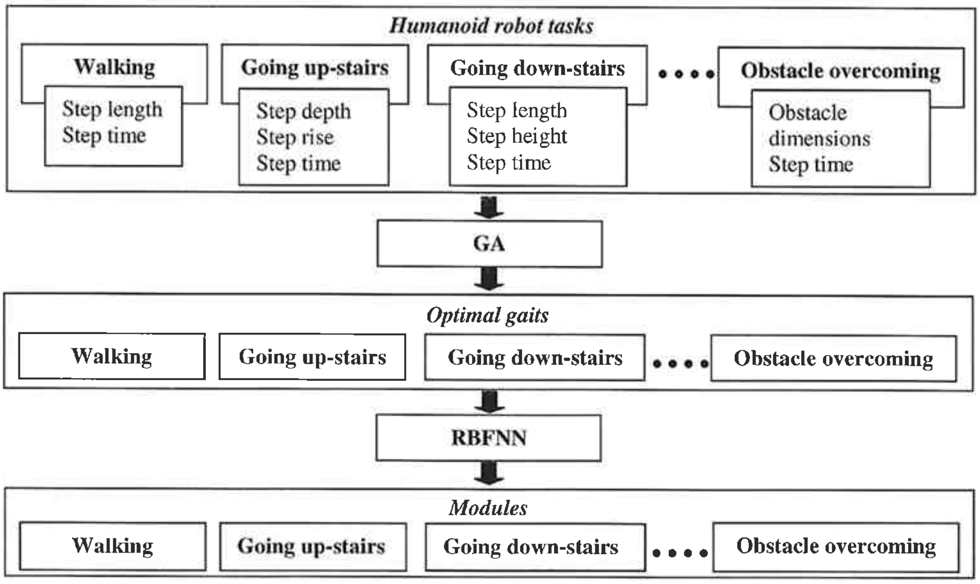

Fig. 3. RBFNN modules.

particular walking velocity. When the only constrained is to walk with a particular step length, the optimal step time will be the best from the CE point of view. Therefore, in the second module the input variable is only the step length. The output variables are the variables optimized by GA and the optimal step time. In this paper, the simulation and experimental results of the first module are presented.

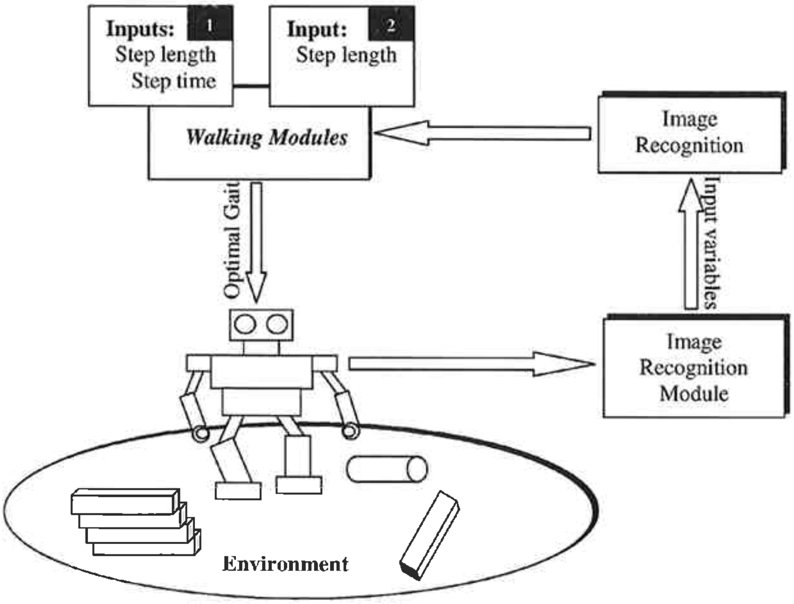

Fig. 4. Real time gait generation during walking.

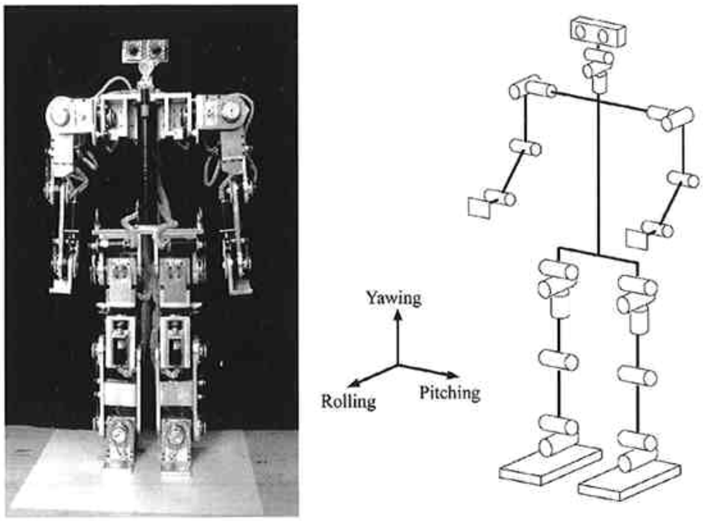

Fig. 5. "Bonten-Maru I" humanoid robot.

5. Simulation and experimental results

In the simulations and experiments, we use the "Bonten-Maru I" humanoid robot, which is developed in Robotics Laboratory of Yamagata University. The robot and the degrees of freedom are presented in Fig. 5. The "Bonten-Maru I" is 1.2 m high, each leg has 6 degrees of freedom and is composed by three segments: upper leg, lower leg and the foot. The foot length is 0.18 m. A DC servomotor actuates each joint. The control platform is based on Common Object Request Broker Architecture (CORBA), which allows an easy updating and addition of new modules [9].

The optimal gait generation belongs to an optimization problem and we use a real number GA as an optimization tool. The convergence of GA is shown in Fig. 6. The GA has a quick convergence during the first generations and it converges to the best solution after the 33rd generation. The minimum CE results generated by GA for the step length 0.42 m and step time 1.2 s are shown in Table 1. The joint angle trajectories (θ_i), the torque vector (τ_i) and the optimal motion are shown in Fig. 7. The torque is kept in low limits as shown in Fig. 7b. Based on the simulation results, we see that minimum CE gait similar to human walking (Fig. 7c).

For real time implementation, we use a regularization RBFNN, where the number of hidden nodes is the same with the training data. In order to teach the

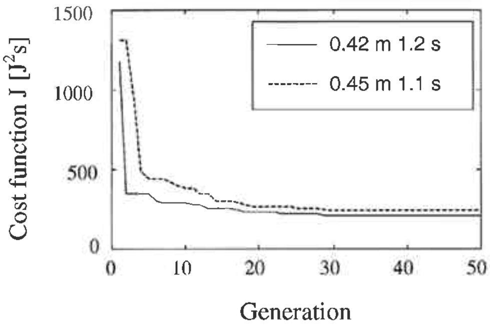

Fig. 6. Cost function J vs. generations.

Table 1
Variable space and GA results

GA var.	Step length 0.42 m, step time 1.2 s	
	Limits	CE
θ_{10}	−0.3 to 0.0	−0.122
θ_{20}	−0.7 to −0.3	−0.455
θ_{30}	0.0 to 0.3	0.1074
$\dot{\theta}_{1f}$	0 to 2	0.523
$\dot{\theta}_{30}$	−1 to 1	−0.031
θ_{3p}	−0.1 to 0.2	0.0840
t_3	0.2 to 0.8	0.5186
x_p	−0.2 to 0.2	−0.135
y_p	0.01 to 0.04	0.0163
t_p	0.0 to 1.0	0.441