Inverse Kinematics for Humanoid Robots Using Artificial Neural Networks

Javier de Lope, Rafaela González-Careaga, Telmo Zarraonandia, and Darío Maravall

Department of Artificial Intelligence
Faculty of Computer Science
Universidad Politécnica de Madrid
Campus de Montegancedo, 28660 Madrid, Spain
{jdlope,rafaela,telmoz,dmaravall}@dia.fi.upm.es

Abstract. The area of inverse kinematics of robots, mainly manipulators, has been widely researched, and several solutions exist. The solutions provided by analytical methods are specific to a particular robot configuration and are not applicable to other robots. Apart from this drawback, legged robots are inherently redundant because they need to have real humanoid configurations. This degree of redundancy makes the development of an analytical solution for the inverse kinematics practically unapproachable. For this reason, our proposed method considers the use of artificial neural networks to solve the inverse kinematics of the articulated chain that represents the robot's legs. Since the robot should always remain stable and never fall, the learning set presented to the artificial neural network can be conveniently filtered to eliminate the undesired robot configurations and reduce the training process complexity.

1 Introduction

The design and development of legged robots has been one of the main topics in advanced robotic research. Basically, projects addressing legged robots study the stability and mobility of these mechanisms in a range of environmental conditions for applications where wheeled robots are unsuitable. The research in this field is been oriented to the development of humanoid robots. There are successful projects, such as the Honda, Waseda and MIT humanoid robots.

The main interest has focused on getting the robot to maintain stability as it walks in a straight line. Two methods are usually used for this purpose. The first and most widely used method aims to maintain the projection of the center of masses (COM) of the robot inside the area inscribed by the feet that are in contact with the ground. It is known as *static balance*. The second method, also referred to as *dynamic balance*, uses the Zero Moment Point (ZMP), which is defined as the point on the ground around which the sum of all the moments of the active forces equal zero [1]. If the ZMP is within the convex hull of all contact points between the feet and the ground, the biped robot is stable and will not fall over [2,3].

R. Moreno-Díaz and F. Pichler (Eds.): EUROCAST 2003, LNCS 2809, pp. 448–459, 2003.

This paper proposes a method for obtaining relative robot foot positions and orientations to achieve several sets of postures, apart from walking in a straight line, like turning around, moving sideways, stepping backwards, etc. The standing and walking postural schemes are discussed.

To achieve these postures, we must give the new foot coordinates to the control system, and it will be able to determine the way the leg should move in order to place the foot at the new position and orientation. For this purpose it will be necessary to compute the inverse kinematics of the legged robot.

A direct kinematic model of the system will have to be built to provide the artificial neural network with a set of training cases. Because of the redundancy of the degrees of freedom (DOF), one feet position can map several angle configurations and a stability criterion will have to be applied to determine the best suited configuration.

Several works have been reported on training neural networks to resolve the inverse kinematics of a robot arm. Self-organizing maps have been widely used for encoding the sample data on a network of nodes to preserve neighborhoods. Martinetz *et al.* [4] propose a method to be applied to manipulators with no redundant DOF or with a redundancy resolved at training time, where only a single solution along a single branch is available at run time. Jordan and Rumelhart [5] suggest first training a network to model the forward kinematics and then prepending another network to learn the identity, without modification of the weights of the forward model. The prepended network is able to learn the inverse. Other alternatives, like recurrent neural networks [6], are also used to learn the inverse kinematics of redundant manipulators.

Kurematsu *et al.* [7] use a Hopfield-type neural network to solve the inverse kinematics of a simplified biped robot. The joint positions are obtained from the position of the center of gravity and the position of the toes is calculated from the equation of an inverted pendulum.

2 Mechanical Description of the Humanoid Robot

Currently a real robot is been designed and constructed. Light weight materials, such as aluminium, are being used. The joints are driven by servo modules for radio controlled models with a limited and reduced torque, which are not applied directly over the joint to increase servo motor performance.

At this stage of research, we are considering a simplified humanoid robot which is made up two legs and the hip. So, we are considering a biped robot. Each leg is composed of six degrees of freedom, which are distributed as follows: two for the ankle —one rotational on the pitch axis and the other one rotational on the roll axis—, one for the knee —rotational on the pitch axis—, and three for the hip —each of them rotational on each of the axes—.

The ankle joint is one of most complex joints in a biped robot. It integrates two rotations of the foot in the roll and pitch axes. The former rotation moves the leg forward during a step and the latter rotation is used, in conjunction with a lateral rotation of the hip around its roll axis, to balance the body and adjust

the position of the COM projection. Therefore, two servos are required to rotate the foot around the roll and pitch axes.

The servos are usually situated in the shank, rotated 90 degrees with respect to the other to allow both movements as, for example, in the PINO [8] and Robo-Erectus [9] humanoids. This configuration is not very well suited because the movement produced by the leg as a whole is not like human movement. With this arrangement of the axes, the pitch rotation is made immediately above the foot, but the roll rotation takes place in the middle of the shank. The human ankle can produce both rotations at the same point, so another kind of mechanism must be used/designed for making more natural, human movements.

A possible solution is provided by humanoid robots like P3 [10], BIP [11] or HRP [12]. These robots use a double axis joint by means of a cardan and two parallel motors on the shank. The axis of every motor is fixed to the foot and prismatic movements of the motors mean the foot can rotate around the pitch and roll axes.

This solution can also be adopted using servos. Two servos can be mounted on the shank with the rotation axes at right angles to the shank and parallel to the foot plane. The two rotations of the foot can be achieved by actuating both servos simultaneously. This joint, which is referred to as *cardan-type joint*, is briefly studied in the next section.

The knee joint only includes a rotation around the pitch axis. In this case, the servo can directly drive the joint, but the required servo torque can be reduced if we consider the use of a simple lever and the mechanical advantage that it provides. Adjusting the position of the fulcrum of the lever on the leg, we will reduce both the required torque and the range of movement of the joint. The *lever-type joint* will be discussed in the following.

Three diferent rotations must be contemplated for the hip joint, around each one of the rotation axes. These rotations can be divided into two groups: a single rotation around the pitch axis, similar to the knee joint, and a double rotation around the roll and yaw axes. The single rotation needs more torque, because it has to lift up the entire leg by a greater range and can be implemented as a lever-type joint. We can use a cardan-type joint for the double rotation, as required ranges and torques for movements are lower.

2.1 Cardan-Type Joint

Fig. 1 shows a cardan-type joint driven by two servos. Let α and α' be the angular position of the right and left servos, respectively. It is known that the range of positions for these angles in servos for radio controlled models is 180 degrees, approximately. Moving the servos in the same direction, clockwise or counterclockwise, the contiguous link rotates around the roll axis, because L grows at the same rate as L' decreases and vice versa. When the servos are moved in opposite direction, the contiguous link rotates around the pitch axis, because L and L' grow or decrease at the same rate.

Let us consider just the right-hand servo. L_1 is the servo arm. One end of the segment L_2 is fixed to the servo arm and the other end to the contiguous

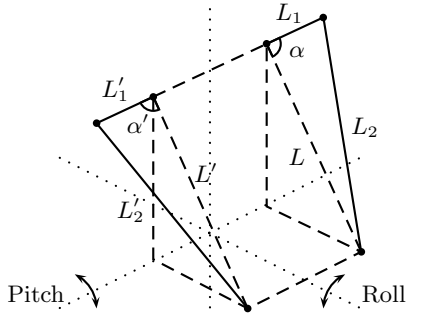

Fig. 1. Rotation axes of a cardan-type joint

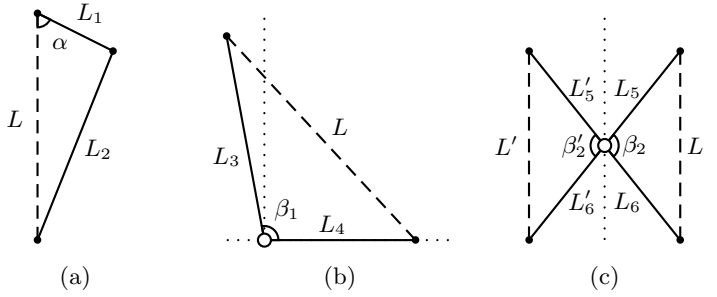

Fig. 2. The ankle joint and its parameters

link. In this way, as L_1 and L_2 are rigid, the length of L is determined by the servo angle α (see Fig. 2a) according to the following expression:

$$L_2^2 = L_1^2 + L^2 - 2L_1 L \cos\alpha \tag{1}$$

and L can be expressed as function of the other parameters, which are known or must be estimated during the design phase

$$L = L_1 \cos\alpha \pm \sqrt{L_1^2 \cos^2\alpha - L_1^2 + L_2^2} \tag{2}$$

If the servos are moved in opposite directions, the value of L and L' will be the same and will grow or decrease according to the value of the angles α and α'. Therefore, by increasing or decreasing the length of L (and L'), the contiguous link will experiment a pitch rotation around the axis. The magnitude of this rotation is denoted as β_1 in Fig. 2b. L_3 is the distance between the servo and pitch axes. L_4 determines the distance between the pitch axis and the fixing point. By modifying the length of all these parameters, we can determine the movement range around the pitch axis.

The parameter β_1 is determined by the expression:

$$L^2 = L_3^2 + L_4^2 - 2L_3 L_4 \cos\beta_1 \tag{3}$$

By operating, we get the value of β_1:

$$\beta_1 = \arccos \frac{L_3^2 + L_4^2 - L^2}{2L_3L_4} \tag{4}$$

where L is determined by (2).

Now, let us consider the rotation around the roll axis. This movement is produced when the servos move in the same direction, and L and L' increase or decrease at the same rate (if L increases, L' decreases and vice versa). L can also be expressed as follows:

$$L^2 = L_5^2 + L_6^2 - 2L_5L_6 \cos \beta_2 \tag{5}$$

where β_2 is the rotation angle around the roll axis, L_5 is the distance between the servo and roll axes, and L_6 is the distance between the roll axis and the fixing point, as shown in the Fig. 2c.

By operating, we get the value of β_2 as a function of the other parameters:

$$\beta_2 = \arccos \frac{L_5^2 + L_6^2 - L^2}{2L_5L_6} \tag{6}$$

where L is determined by (2).

Note that β_2 and β_2' are two *complementary* angles in this kind of movement. We have assumed that when a servo rotates clockwise, the other one also rotates clockwise, so when β_2 increases, β_2' decreases and the rotation around the roll axis is produced. The range of movement of β_2 and β_2' to the left and right is different, if the same interval is taken for α and α'. Obviously, due to mechanical and physical restrictions, the minimum of both β_2 angles must be selected in order to determine the range around the roll axis.

As we are using a cardan joint, the roll axis cut the pitch axis at a point at which both rotations are produced. Accordingly, we can assume that $L_3 = L_5$ and $L_4 = L_6$. If the axes are crossed, but not cut, for example, when two different axes are used rather than a cardan, this assumption is not valid.

To get the range of movement around each axis, all the L_i parameters must be fixed.

2.2 Lever-Type Joint

Fig. 3 shows a simplified diagram of the lever-type joint and the used parameters for the computation of the joint range. L_1 is the length of the servo arm, L_2 is the length of a rigid segment that connects the servo arm and the contiguous link, L_3 is the distance between the servo and joint axes, and L_4 is the distance between the rotation axis and the point at which the rigid segment is fixed to the contiguous link. α is the rotation angle of the servo and, as mentioned above, it is constrained approximately to a range of 180 degrees. The angle $\gamma = \gamma_1 + \gamma_2$ defines the joint position and its range of values can be calculated from the L_2, L_3 and L_4 parameters, and it will be selected considering the most appropriate range for the required movements and the generated force/torque.

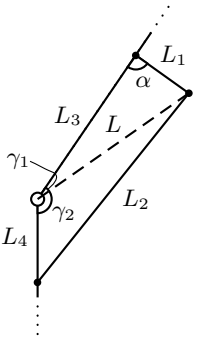

Fig. 3. The lever-type joint and its parameters

The length of the segment L is determined by the following expression:

$$L^2 = L_1^2 + L_3^2 - 2L_1L_3\cos\alpha \tag{7}$$

L_1 and L_2 parameters can be expressed as:

$$L_1^2 = L^2 + L_3^2 - 2LL_3\cos\gamma_1 \tag{8}$$

$$L_2^2 = L^2 + L_4^2 - 2LL_4\cos\gamma_2 \tag{9}$$

where L is determined by (7).

By operating, we get the value of γ by means of:

$$\gamma = \gamma_1 + \gamma_2 = \arccos\frac{L^2 + L_3^2 - L_1^2}{2LL_3} + \arccos\frac{L^2 + L_4^2 - L_2^2}{2LL_4} \tag{10}$$

For the construction point of view, the lengths of L_3 and L_4 are limited by the length of the links. Moreover, the lengther L_4 is, the larger the distance between L_2 and the leg is. If this distance is too large, the workspace needed for the leg will also be larger and troubles when the joint is completely bent could exist.

On the other hand, we can also consider some dynamical aspects related to the moment arm and the servo torque. Assuming that the link is almost perpendicular to the ground most of the time during operation, we can choose a value for L_4 which establish an advantageous moment arm to compensate a lower torque rating of the servo. The mechanical advantage is defined by the L_1 and L_4 lengths considering the whole system as a lever (see the Fig. 3). For example, considering a value for $L_4 = 2L_1$, we can approximate that the requiring torque ratio for the servo to a half.

2.3 Physical Dimensions

Table 1 shows the maximum ranges for each robot joint. θ_1 and θ_2 correspond to the pitch and roll axes of the right ankle (θ_{12} and θ_{11}, for the left ankle). θ_3

Table 1. Range of angle positions for each joint

Right Leg			*Left Leg*		
θ_i	*Min*	*Max*	θ_i	*Min*	*Max*
θ_1	-42.55^o	27.82^o	θ_7	-35.31^o	32.21^o
θ_2	-27.82^o	27.82^o	θ_8	-32.21^o	32.21^o
θ_3	-130^o	0^o	θ_9	-14.73^o	104.43^o
θ_4	-104.43^o	14.73^o	θ_{10}	0^o	130^o
θ_5	-32.21^o	32.21^o	θ_{11}	-27.82^o	27.82^o
θ_6	-35.31^o	32.21^o	θ_{12}	-27.82^o	42.55^o

and θ_{10} are the angles associated to the right and left knees, respectively. The rest of angles are belonged to the hips.

The total height of the robot is 274 mm. The links dimensions are 10, 106 and 158 mm for the links between the ground and the ankle, the ankle and the knee, and the knee and the hip respectively. The hip length is 100 mm.

3 Direct Kinematics

The direct kinematics of the robot as regards the relative position and orientation of one foot from the other can be solved easily considering our model as a robotic chain of links interconnected to one another by joints. The first link, the base coordinate frame, is the right foot of the robot. We assume it to be fixed to the ground for a given final position or movement, where the robot's global coordinate frame will be placed. The last link is the left foot, which will be free to move. The assignment of the coordinate frames to the robot joints is illustrated in Fig. 4.

Obviously, to consider the positions or movements in the inverse case, that is, when the left foot is fixed to the ground and the right foot is able to move, the coordinate frames must be assigned similarly. This does not lead to any important conclusions concerning the proposed method and will not be discussed in this paper.

The position and orientation of the left with respect to the right foot will be denoted by the homogeneous coordinate transformation matrix:

$$
{}^{R}T_L = \begin{bmatrix} {}^{R}R_L & {}^{R}\boldsymbol{p}_L \\ 0 & 1 \end{bmatrix} = {}^{R}T_0 \times {}^{0}T_1(\theta_1) \times \ldots \times {}^{5}T_6(\theta_6) \times {}^{6}T_7(\theta_7) \times \ldots \times {}^{11}T_{12}(\theta_{12}) \times {}^{12}T_L \tag{11}
$$

where ${}^{R}R_L$ denotes the rotation matrix of the left foot with respect to the right foot coordinate frame, ${}^{R}\boldsymbol{p}_L$ denotes the position matrix of the left foot with respect to the right foot coordinate frame and each ${}^{i}T_j$ denotes the homogeneous coordinate transformation matrix from link i to link j. Coordinate frames from 1 to 6 correspond to the right leg, and from 7 to 12 are associated to the left leg.

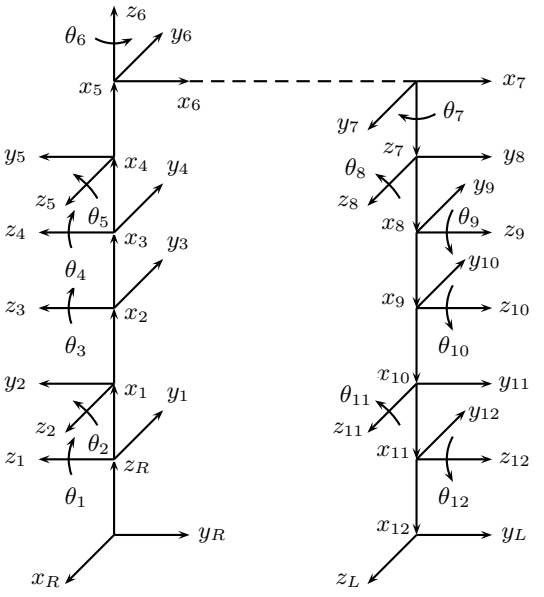

Fig. 4. Coordinate frames associated to the robot joints

4 Postural Schemes

Based on the model of the robot and the direct kinematics described above, a set of joint and Cartesian coordinates pairs has been generated to be used as training data for the neural network. Two sets of postural schemes have been considered. The first one will be composed of positions with both feet on the ground, straightened legs, which are uniformly distributed accross a given coordinate range. The second scheme includes the positions involved in the execution of a single robot step.

For each postural scheme, a different artificial neural network will be used, and the respective network will be selected depending on the posture to be achieved. Each artificial neural network will be trained separately with the respective generated data under the restrictions explained below.

For each case, the redundancy of the degrees of freedom will mean that different joint configurations can produce the same feet position. Therefore, the criteria we will use to choose which configurations will be used as training data need to be established. Ideally, we want the network to learn the most stable positions from the range of all the possible solutions. Using a static balance criterion, a position is considered to be more stable the closer the vertical projection of the COM of the humanoid is to the center of the support polygon formed by the feet in contact with the ground. The height of the COM will also affect system stability, where the system will be more stable the lower it is placed. However, we are not interested in postures that place the hip too low, as we want the postures of the robot to be as natural as possible.

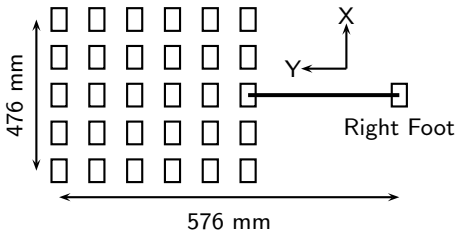

Fig. 5. Positions of the free foot in the training set of the standing postural scheme

Table 2. Range of angle positions for the standing postural scheme

Right Leg			*Left Leg*		
θ_i	*Min*	*Max*	θ_i	*Min*	*Max*
θ_1	-30^o	25^o	θ_7	-30^o	30^o
θ_2	-25^o	0^o	θ_8	0^o	30^o
θ_3	-90^o	0^o	θ_9	-12^o	30^o
θ_4	-30^o	12^o	θ_{10}	0^o	90^o
θ_5	0^o	30^o	θ_{11}	0^o	25^o
θ_6	-30^o	30^o	θ_{12}	-25^o	30^o

4.1 Standing Postural Scheme

The aim of this first scheme is to learn robot postures along the roll and pitch axes. These postures are restricted to both feet being in full contact with the ground, the knees not being bent and there being no rotation around the hip roll axes. Fig. 5 shows the final positions of the free foot.

A set of joint and Cartesian coordinates pairs whose values fulfill these constraints will be generated to feed the neural network. The training set is composed of 1000 pairs. The range of angle positions is shown in Table 2.

A two-layer backpropagation network is used to learn a representative set of input-target pairs, which has been introduced in elsewhere [13]. The input layer has 3 neurons, one for each element of the Cartesian coordinates of the free foot. The output layer has 12 neurons, associated to the joint coordinates of the humanoid robot. After trying out several configurations, the hidden layer contains 35 neurons.

The transfer function for the output layer has to be a *purelin* function, because the outputs can take any value. The *logsig* transfer function was chosen for the hidden layer because it achieves a lower error rate than the other transfer functions compared. The Levenberg-Marquardt training algorithm was the fastest training algorithm and 32 epochs were needed to reach the error goal 0.001. By comparison, the epochs needed to reach an error goal of 0.01, ten times greater than the above, using, for example, the variable learning rate algorithm, is six orders of magnitude.

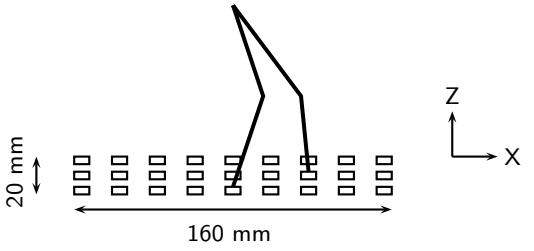

Fig. 6. Positions of the free foot in the training set of the walking postural scheme

Table 3. Range of angle positions for the walking postural scheme

Right Leg			*Left Leg*		
θ_i	*Min*	*Max*	θ_i	*Min*	*Max*
θ_1	0°	25°	θ_7	—	—
θ_2	−15°	15°	θ_8	−15°	15°
θ_3	−30°	0°	θ_9	0°	30°
θ_4	−30°	0°	θ_{10}	0°	30°
θ_5	−15°	15°	θ_{11}	−15°	15°
θ_6	—	—	θ_{12}	−25°	0°

4.2 Walking Postural Scheme

The aim of the second scheme is to learn all the robot postures required to take a step on the roll axis. For this to be achieved, we have to generate a set of joint and Cartesian coordinates pairs which describes possible postures of the robot to be adopted along the step gait (Fig. 6).

Any of these postures must accomplish the COM stability criteria, and for this second scheme we can make a distinction between two different cases:

(a) The robot has only one foot in contact with ground. In this case the polygon of support is formed only by the right foot, and the projection of the COM must remain as close as possible to the center of this rectangle. For this to be achieved, some amount of swinging movement of the hip will be necessary.
(b) The robot has both feet in full contact with the ground. Here the polygon of support will be the one formed by the two feet. In order to determine its shape, we calculate the convex hull of the polygon [14]. To calculate the center position of the polygon, we get the arithmetic average of the cut points of the lines perpendicular to the polygon sides.

The process for outputting the data is to generate a set of values that uniformly covers all the angle ranges of the roll and pitch axes. The values of the angles for the roll joints of the right hip and the left ankle are calculated to assure that the hip is parallel to the ground and that the left foot always remains parallel to the floor. Table 3 shows the used ranges.

Another set of angles must be generated to obtain the desired balancing movement of the hip and maintain the COM projection inside the support polygon. These values will be applied appropriately so that the pitch axes of the ankle and hip joints of both legs maintain the hip parallel to the ground.

Once we have the set of values for each angle, we generate all the possible combinations, calculate the final position of the left foot, the position of the COM and apply the restrictions described above.

The next filter to be applied, removes the less stable repeated positions (the ones with the projection of the COM further from the center of the polygon), leaving unique and optimally stable positions to train the neural network.

Finally, we restrict the length and height of the step. The movement ranges for every axis are ± 80 mm in the roll axis, ± 8 mm in the pitch axis and 20 mm for the maximum high of the feet over the ground [2].

The result is the final set of joint and Cartesian coordinates pairs that will be used to train the neural network. This set is composed of 300 input-target pairs, approximately.

The neural network for this postural scheme has the same architecture as the first artificial neural network described. Nevertheless, as the complexity of the input data have increased considerably, the number of neurons in the hidden layer has been increased to 100.

5 Conclusions and Further Work

A method for solving the inverse kinematics of a humanoid robot has been presented. The method is based on artificial neural networks and makes it unnecessary to develop an analytical solution for the inverse problem, which is practically unapprochable because of the redundancy in the robot joints.

An artificial neural network learns relative foot positions and orientations to achieve several postures. Each artificial neural network is trained separately with the generated data under a set of constraints and stability criteria.

The proposed neural architecture is able to approximate, at a low error rate, the learning samples and to generalise this knowledge to new robot postures. The error in the roll, pitch and yaw axes are close to 1 millimeter, less than the 1% considering the leg height.

The next stage of our work will be to define a mechanism for the selection of the most appropriate postural schemes for each robot movement. Also, the generation of other postural schemes apart from the standing and walking postural schemes will be implemented.

Acknowledgements. This has been partially supported by the Spanish Ministry of Science and Technology, project DPI2002-04064-CO5-05.

References

1. Vukobratovic M., Brovac, B., Surla, D., Sotkic, D. (1990) Biped Locomotion. Springer-Verlag, Berlin
2. Huang, Q. *et al.* (2001) Planning walking patterns for a biped robot. IEEE Trans. on Robotics and Automation, **17**(3), 280–289
3. Furuta, T., *et al.* (2001) Design and construction of a series of compact humanoid robots and development of biped walk control strategies. Robotics and Autonomous Systems, **37**(2–3), 81–100
4. Martinetz, T.M., Ritter, H.J., Schulten, K.J. (1990) Three-dimensional neural net for learning visuomotor coordination of a robot arm. IEEE Trans. on Neural Networks, **1**(1), 131–136
5. Jordan, M.I., Rumelhart, D.E. (1992) Forward models: Supervised learning with a distal teacher. Cognitive Science, **16**, 307–354
6. Xia, Y., Wang, J. (2001) A dual network for kinematic control of redundant robot manipulator. IEEE Trans. ond Systems, Man, and Cybernetics – Part B: Cybernetics, **31**(1), 147–154
7. Kurematsu, Y., Maeda, T., Kitamura, S. (1993) Autonomous trajectory generation of a biped locomotive robot. IEEE Int. Conf. on Neural Networks, 1961–1966
8. Yamasaki, F., Miyashita, T., Matsui, T., Kitano, H. (2000) PINO, The humanoid that walk, Proc. of The First IEEE-RAS Int. Conf. on Humanoid Robots
9. Zhou, C., Meng, Q. (2003) Dynamic balance of a biped robot using fuzzy reinforcement learning agents. Fuzzy Sets and Systems, **134**(1), 169–187
10. Hirai, K. *et al.* (1998) The development of Honda humanoid robot. Proc. of the IEEE Int. Conf. on Robotics and Automation, 1321–1326
11. Espiau, B., Sardain, P. (2000) The antropomorphic biped robot BIP2000. Proc. of the IEEE Int. Conf. on Robotics and Automation, 3996–4001
12. Kaneko, K. *et al.* (2002) Design of prototype humanoid robotics platform for HRP. Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, 2431–2436
13. De Lope, J., Zarraonandia, T., González-Careaga, R., Maravall, D. (2003) Solving the inverse kinematics in humanoid robots: A neural approach. Proc. of Int. Work Conf. on Artificial and Natural Neural Networks, IWANN'03 (to appear)
14. Lee, D.T. (1983) On finding the convex hull of a simple polygon. J. of Algorithms, **4**, 324–331