HAL
open science

A Neural Network Approach of the Control of Dynamic Biped Equilibrium

Patrick Henaff, Holger Schwenk, M. Milgram

To cite this version:

Patrick Henaff, Holger Schwenk, M. Milgram. A Neural Network Approach of the Control of Dynamic Biped Equilibrium. 3rd International Symposium on Measurement and Control in Robotics (ISMCR '93), Sep 1993, Turin, Italy. pp.AS.II-19. hal-01348277

HAL Id: hal-01348277

https://hal.science/hal-01348277v1

Submitted on 8 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ISMCR '93

Technical Committee on Robotics (TC 17)

Proceedings of the Third International Symposium on Measurement and Control in Robotics

September 21-24, 1993

AMMA - Via Vela 17 - Torino, Italy

Edited by
Carlo Ferrero
Giancarlo Genta

Consiglio Nazionale delle Ricerche
Progetto Finalizzato ROBOTICA

IMEKO

AMMA
ASSOCIAZIONE INDUSTRIALI METALLURGICI MECCANICI AFFINI

ORGANIZED BY
IMEKO Technical Committee on Robotics (TC 17)
AMMA - (Associazione Industriali Metallurgici Meccanici Affini)
CNR - Progetto Finalizzato Robotica

A Neural Network Approach of the Control of Dynamic Biped Equilibrium.

Patrick HENAFF, Holger SCHWENK and Maurice MILGRAM

Laboratoire de Robotique de Paris - UPMC-UVSQ-CNRS
10-12, av de l'Europe 78140 Vélizy- FRANCE.

Abstract :
In this paper, we propose a Neural Network learning architecture for the reflex-control of complex systems. The multi-layer-network is trained by a new learning algorithm which does not need a desired output, but directly minimizes a criterion which spezify the control objective. We propose two learning concepts: off-line learning and on-line learning (similar to adaptive control). We have tested our method for a simplified dynamic problem: the dynamic stability of a simulated planar biped. To obtain dynamic equilibrium during the single-support-phase, the robot trunk is controlled such that a zero-moment-point (ZMP) criterion is satisfied. We show that although no gait-specific knowledge is used during off-line learning, dynamic walking for different step lengths and heights could be realized and controled. The capability of the proposed learning algorithm to adapt the trunk control on-line to the actually performed leg trajectories allows further improvements and adaptation to real environment conditions or perturbations.

Keywords : Neural network, Learning control, Robotic, Adaptive control, Biped locomotion

1. INTRODUCTION

Many scientists are currently working on the design and control of autonomous, mobile robots. In particular, there is a growing interest in walking robots. They have good mobility, and therefore, they could replace human beings in dangerous environments. On the other hand, the large mechanical complexity of walking robots, and the necessary of dynamic control imply some very difficult control algorithms. The main problem is the require inversion of the dynamic model. This model can not, in general, be obtained analytically and it is even often not uniquely determinable. The choice of a solution of this inversion is usualy based on criterion optimization (energy, manoeuvrability, equilibrium, ...) which makes on-line control very difficult. For this reason, most of recent walking robots use off-line calculated walking-patterns (e.g. [5]). Several improvements of this basic sheme have been proposed in recent papers (adaptation to the actual measured ZMP (*zero-moment-point*) [4], on-line consideration of unknown external force [6]).

To solve these difficulties, some new methods have been developed, such as Neural Networks and Fuzzy Logic. In the case of Neural networks, multi-layer-networks trained with the well known backpropagation learning algorithm are frequently used. However, this algorithm requires the a priori knowledge of the desired output (e.g. the desired command) corresponding to the net input (e.g. the actual state of the system). To calculate this desired output, it is necessary to invert a model (geometric, kinematic or dynamic).

Lee and El Maraghy for instance, use a multi-layer-network to control a simplified planar biped walking robot [2]. Since the exact solution is difficult to derive, they generate examples using three different linear controllers with different parameters. In this case, the learning data does not represent the optimal solution, so that the neural net can not perform the task perfectly. Therefore we propose a new learning method that doesn't need the desired output, but only a criterion specifying directly the goal.

Here, this learning algorithm is applied to control the dynamic equilibrium of a biped walking robot. The neural network provides, (on-line), the optimal trunk acceleration to compensate for an imposed leg-movement. As this is the first time we test our method for a dynamic problem, we have used a simulated planar biped, but we believe it can be extended to the general case, i.e. a 3-dimensional real biped robot.

2. LEARNING ALGORITHM

We use standard multi-layer-networks in our approach, so that we omit the description of the network architecture. These networks are normally trained with well known backpropagation learning

algorithm ([1] and [3]). It uses a cost function E_p which compares the net ouput vector o_p with the desired output vector o_p^d:

$$E_p = \frac{1}{2}\sum_j (o_{pj}^d - o_{pj})^2$$

The goal of the learning is to minimize E_p for every input pattern p by a gradient descent method. Since the output vector depends only on the weights w_{ji} (weight from neuron i to neuron j), the following gradient must be calculated:

$$\nabla E = \frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial w_{ji}} = \frac{\partial E_p}{\partial net_{pj}} o_{pi}$$

We set $\delta_{pj} := -\frac{\partial E_p}{\partial net_{pj}}$ and then we have:

$$\delta_{pj}^{out} = f_j'(net_{pj})(o_{pj}^d - o_{pj}) \quad (1)$$

$$\delta_{pj}^{hid} = f_j'(net_{pj}) \sum_k \delta_{pk} w_{kj} \quad (2)$$

where equation (1) is applied for an output neuron and (2) for hidden ones. In the above equations $net_{pj} = \sum_i w_{ji} o_{pi}$ denotes the activation of neuron j and f_j its function (sigmoide). Using this notations, we get the following update rule :

$$w_{ij}^{k+1} = w_{ij}^k + \Delta w_{ij}^{k+1}$$

where

$$\Delta w_{ij}^{k+1} = -\lambda \frac{\partial E_p}{\partial_{ij}^k} + \eta \Delta w_{ij}^k$$

$\lambda \in [0, 1]$ denotes the learning rate. The last term anticipes the old weight change. This technique, known as momentum, speeds up learning and reduces oscillations in valleys of the energy landscape.
Although the inherent robustness to real noisy data and the strong generalization capability of multilayer-networks make them very suitable for control applications in robotics, the need for the desired outputs presents some problems. This is particularly true for dynamic applications. In this case it is often not possible to determine the exact solution and frequently linearized models are used.

On the other hand, it is often easy to formulate a *quality function* to minimize, e.g. the movement of the tool-center-point of a manipulator to a point corresponds to the minimization of the euclidean distance.

The basic idea of the proposed learning algorithm is to mimimize directly an arbitrary criterion J_p specifying the goal and the constraints of the problem, instead of the error E_p between the net- and the desired output. In order to minimize this criterion we use also a gradient descent method:

$$\nabla J = \frac{\partial J_p}{\partial w_{ji}} = \frac{\partial J_p}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial w_{ji}} = \frac{\partial J_p}{\partial net_{pj}} o_{pi}$$

In similar manner to what was done above, we define $\delta_{pj} := -\frac{\partial J_p}{\partial net_{pj}}$ and obtain for a ouptut neuron:

$$\delta_{pj}^{out} = -\frac{\partial J_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pj}} = -\frac{\partial J_p}{\partial o_{pj}} f_j'(net_{pj})$$

In place of the difference between actual and desired output we backpropagate the gradient of the criterion related to the corresponding output. Therefore, the partial derivates of the criterion must be analytically calculable. The computation of δ_{pj}^{hid} remains unchanged. Figure 1 compares the the two learning algorithms. The advantage of learning with criterion

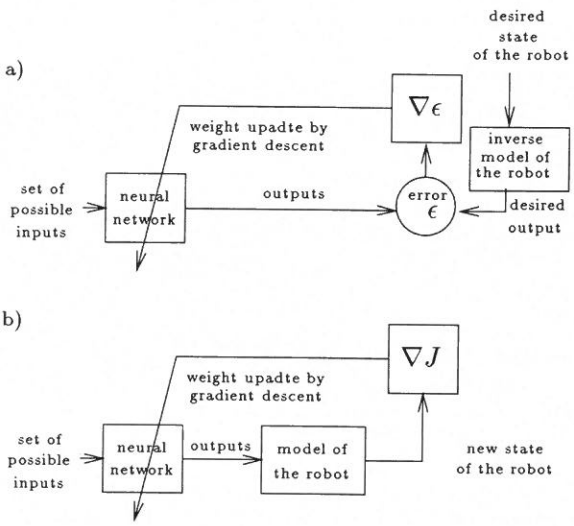

Figure 1: *standard backpropagation (a) and learning with a criterion (b)*

is that we need not any desired output, and only a cost function is required. The neural network determines itself the way to achieve this goal. This also allows for training the network on-line.

3. MODELING AND STABILITY OF THE BIPED

The dynamics of human-like walking is very complicated and up to now, it is not possible to formulate its complete dynamic equations. Accordingly, the study of the feasibility of a neural network approach to dynamic stability control of biped walking must be based on a simplified model. In order to obtain a model for a simulated biped walking robot which can be handled mathematically, we made the following assumptions (see also fig. 2):

- The model is constrained to the sagittal plane.
- To allow full manoeuvrability two active degrees of freedom (DOF) are assumed, i.e. a rotatory articulation at the hip and at the knee.
- The trunk, the head and the arms are modeled as one particle.
- The mass is considered to be concentrated in the articulations, at the end of the trunk and at the feet respectively.
- The feets are considered as point contact.

The parameters of the robot model are:

$$m_0 = 2.0\,kg$$
$$L_1 = 1.0\,m \qquad m_1 = 3.0\,kg$$
$$L_1' = L_1'' = 0.7\,m \qquad m_1' = m_1'' = 1.0\,kg$$
$$L_2' = L_2'' = 0.6\,m \qquad m_2' = m_2'' = 0.5\,kg$$

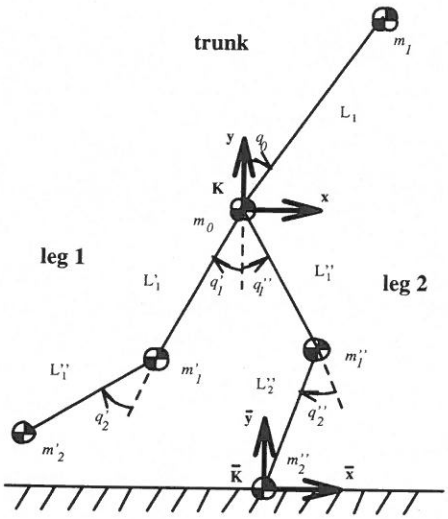

Figure 2: *Model of the biped walking robot*

In fact, we used a two-dimensional version of the model of the WL-12 walking robot built at Waseda-University [5]. On the other hand, the punctate feet of our model require a very exact stability control method.

3.1 Stability criterion

In this study we use a *zero-moment-point* (ZMP) criterion [7] to control dynamic stability during the single-support-phase, when only one foot is in contact with the ground. Stability control during the double-support-phase is not considered since the two dimensional structure makes it easy to control.

Using the above model, equations for the ZMP expressed in the fixed coordinate system $\bar{K}$ can be derived as follows:

$$\bar{x}_{zmp} = \frac{\sum_i m_i(\ddot{\bar{y}}_i + g_y)\bar{x}_i - \sum_i m_i(\ddot{\bar{x}}_i + g_x)\bar{y}_i}{\sum_i m_i(\ddot{\bar{y}}_i + g_y) - \sum_k F_{yk}} + \frac{\sum_j M_j + \sum_k (F_{xk}\bar{y}_{sk} - F_{yk}\bar{x}_{sk})}{\sum_i m_i(\ddot{\bar{y}}_i + g_y) - \sum_k F_{yk}} \quad (3)$$

where

$(\bar{x}_i, \bar{y}_i)^T$	:	coordinates of mass m_i in $\bar{K}$
$(\bar{x}_{sk}, \bar{y}_{sk})^T$	:	position vector where external force F_k acts
$(F_{xk}, F_{yk})^T$	:	external forces F_k
M_j	:	external moment
$(g_x, g_y)^T$	:	gravitational acceleration
$(\bar{x}_{zmp}, 0)^T$	:	coordinates of the ZMP

An additional moving coordinate system K is defined at the hip in order to obtain a symmetric study of the legs. Coordinates of the origin of K are $(\bar{x}_{pl}, \bar{y}_{pl})^T$ measured relative to the fixed coordinate system $\bar{K}$.

In this study we don't consider any external forces and moments. So, the particle coordinates, equation (3),come:

$$\bar{x}_{zmp} = \frac{Fy_x - Fx_y}{Fy} \quad (4)$$

$$\text{with } Fx_y = \sum_i m_i(\ddot{x}_i + \ddot{\bar{x}}_{pl} + g_x)(y_i + \bar{y}_{pl})$$

$$Fy_x = \sum_i m_i(\ddot{y}_i + \ddot{\bar{y}}_{pl} + g_y)(x_i + \bar{x}_{pl})$$

$$Fy = \sum_i m_i(\ddot{y}_i + \ddot{\bar{y}}_{pl} + g_y)$$

$\bar{x}_{zmp}$ is a function of the articular variables q_i, $\dot{q}_i$ and $\ddot{q}_i$. $(x_{pl}, y_{pl})^T$ and the second derivates are provided by the simulation, or by sensors for a real biped.

A system is dynamically stable, if the ZMP falls always into the supporting surface. During the single-support-phase this corresponds to a single point, so that dynamic stability is garanteed when the ZMP is equal to zero.

3.2 Control structure

High adaptibility of a biped walking robot to different environment conditions can only be obtained when the stability and the movement functions are separated: legs perform the displacement while the trunk is controlled such that dynamic stability is guaranteed. In this paper we propose a neural network approach to calculate the compensating trunk movement. The walking patterns of the legs are generated analytically.

In order to obtain a system which is able to maintain equilibrium independently of the movement of the legs, and which tolerates consequent perturbations of the leg-trajectories, the compensating trunk movement must be calculated on-line.

In the general case however, i.e. a 3-dimensional robot, the ZMP equations can not be solved analytically and the numerical resolution is time-consuming. To solve this problem, Takanishi et al. have proposed to plan a ZMP-trajectory for a given fixed gait, and to solve the resulting equations by using FFT and modifying the fourier coefficients [5]. All these calculations are performed *before* walking takes place. Afterwards, the biped walking robot uses only the saved walking patterns to control its joints on-line. We

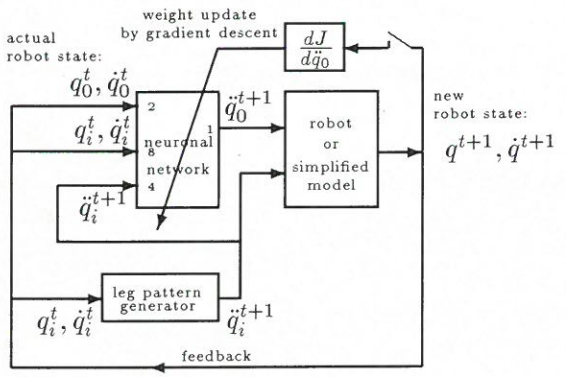

Figure 3: *Neural control structure. The lower part shows the combination of analytical leg and neural trunk control. Additionally, on-line learning can be activated. During off-line learning only the upper part is present and the net inputs come from an example set.*

have decided to control the simulated planar biped by accelerations in order to have full dynamic control. In this case, an analytical solution of the second order differential equation 4 ($\bar{x}_{zmp} = 0$) can be obtained by solving in respect to $\ddot{q}_0$. Since this solution is not a function of the new trunk velocity $\dot{q}_0^{t+1}$ and position q_0^{t+1} which result from this $\ddot{q}_0^{t+1}$, the ZMP is in general not exactly zero after a simulation step.

Several simulation have shown that, because of the punctate feet, these effects have siginficant influence on the quality of the control, such that a dynamically stable gait couldn't be obtained (see section **4.2** for more details).

Therefore, we propose a neural network architecture which calculates, on-line, the trunk motion as a function of the actual state of the biped walking robot ($\mathbf{q^t}$, $\dot{\mathbf{q}}^\mathbf{t}$) and the next walking pattern of the legs ($\ddot{\mathbf{q}}^\mathbf{t}$). The last ones are provided by an analytical walking pattern generator. Figure 3 shows schematically this control structure.

It is worthwhile noting that the above mentioned problems of the analytical solution are solved by using a simplified model of the biped robot during off-line learning (see fig. 1b).

4. OFF-LINE LEARNING

For dynamic stability during the single-support-phase, the ZMP must fall exactly in the point supporting area. Thus, the criterion J_p and the partial derivate with respect to the net output $\ddot{q}_0$ can be stated as follows:

$$\begin{aligned} J_p &= \bar{x}_{zmp}^2 \\ \frac{\partial J_p}{\partial \ddot{q}_0} &= 2\,\bar{x}_{zmp} m_1 L_1 \left(\frac{Fy_x - Fx_y}{Fy^2} \sin q_0 \right. \\ &\quad \left. + \frac{(y_1 + \bar{y}_{pl}) \cos q_0 - (x_1 + \bar{x}_{pl}) \sin q_0}{Fy} \right) \end{aligned}$$

with Fx_y, Fy_x and Fy given by equation 4. This gradient is the only information used by the learning algorithm to find the optimal solution.

4.1 Example generation

The goal of off-line learning is to obtain a neural network able to perform (almost) stable dynamic biped walking. In this phase it is more important to obtain good trunk control for a great varity of leg movements than a perfect one for few special trajectories since the network can be improved afterwards by learning on-line.

In order to obtain this behavior, the neural network must have 'seen' a representative subset of the possible configurations of the biped. This can normally be achieved by generating random examples that span the whole input data space. However, the 14 inputs, a space $\Re^{14}$, would lead to a very large example set.

On the other hand, some general constraints can be formulated which are true for any reasonable biped walking gait:

1. The ranges of the joints are limited (trunk $\pm\frac{\pi}{2}$, hip $\pm\frac{\pi}{2}$ and knee $-\pi \ldots 0$).
2. Geometric constraints on the swinging leg are:
   (a) The foot is always located between the ground and the height of the hip.
   (b) Foot-trajectories with large step heights can be discarded.
   (c) The foot is neither on the opposite side of the contact point from the hip, nor between these two points.

Unfortunatly it is not easy to formulate similar qualitative conditions for the greater number of the kinematic inputs, i.e. the velocity and acceleration information, although there exists many dependencies between these measures for each joint.

For this reason, we have solved this problem on a higher level: for every complete input vector we test if a solution of $\bar{x}_{zmp} = 0$ exists in the output range of the neural network. Since the ZMP is a continous function of $\ddot{q}_0$, this is the case if the ZMP lies on different sides of the contact point for the minimal and maximal net output respectively. The condition 'ZMP is reachable' rejects dynamically unfavorable situations, and thus, for a natural gait, uninteresting examples.

To respect these geometric and kinematic conditions we have developed two example generation methods. They are described in the following.

Example generation in the robot configuration space

The basic idea of this method is to generate every component of the input vector randomly in their specific range of values. Afterwards, we verify if this input vector represents a valid robot configuration. We obtain the following procedure (see also fig. 4a):

1. Generate randomly q_1'' and q_2'' until the hip is located in the rectangle with the corners $(-\frac{l_{max}}{4}, y_{pl} - dy_{pl})$ and $(\frac{l_{max}}{4}, y_{pl} + dy_{pl})$.

2. Generate randomly q_1' and q_2' until the foot 2 is located in the rectangle with the corners $(x_{pl} - \frac{l_{max}}{4}, 0)$ and $(x_{pl} + \frac{l_{max}}{4}, h_{max})$.

3. If condition 2c is not satisfied, go to 2)

4. Generate random values for the kinematic inputs until the ZMP is reachable.

Experiments have shown that generation of random values for angles often implied an inharmonious distribution of the foot end points in the rectangle $h_{max} \times l_{max}$. To fix it, we have developed a second method.

Example generation in the problem space

Now, we generate directly potential geometric robot configurations and calculate afterwards the corresponding angles by using the inverse geometric model. Note that this doesn't correspond to an inversion of the whole problem, which would be a dynamic one. The kinematic components of the input vectors are still generated randomly using the condition 'ZMP is reachable'.

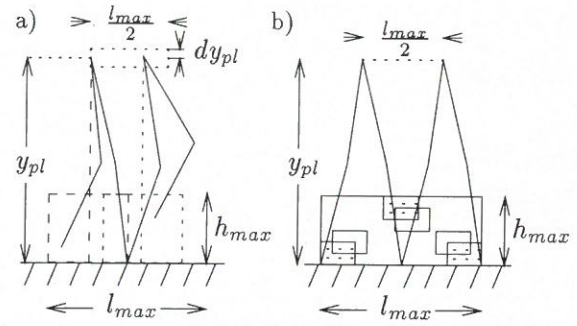

Figure 4: *Example generation in the robot configuration space (a) and in the problem space (b).* l_{max} *is the maximal step length,* h_{max} *the maximal step height.*

The hip positions lie on the line with the end points $(-\frac{l_{max}}{4}, y_{pl})$ and $(\frac{l_{max}}{4}, y_{pl})$. For each of these positions a small rectangle was introduced, in which foot positions of the leg 2 were distributed linearly. The position of these rectangles varies with the hip position, whereby overlapping has been foreseen (see fig. 4b).

Remember that the first method uses no a-priori knowledge of the later performed leg-trajectories, and that the second one only assumes different foot heights as a function of the hip position.

4.2 Simulation results

The two example generation methods were used to build two learning sets of 800 examples used to train a neural network off-line. The maximum step length in both cases was 0.3 m and the maximum step height was 0.1 m. The neural networks had 14 inputs, one hidden layer with 12 neurons and one output neuron. Both networks converged very quickly and after 10,000 learning cycles a very low final value of the total-criterion $J = \sum_p J_p$ was obtained.

Figure 5 shows the ZMP-trajectory for a step of length 25 cm and height 7.5 cm. For the analytical leg movement we have used a parametrized cycloid function in order to obtain zero impact forces. The trunk was controlled by a neural network trained off-line with an example set generated in the robot configuration space (curve 1) and in the problem space (curve 2) respectively. For comparison the ZMP-trajectory for the analytic trunk control is also given. Although no leg-trajectories information was used during off-line learning, both neural networks were able to control the dynamic equilibrium. The ZMP was maximal 0.5 cm beside the contact point.

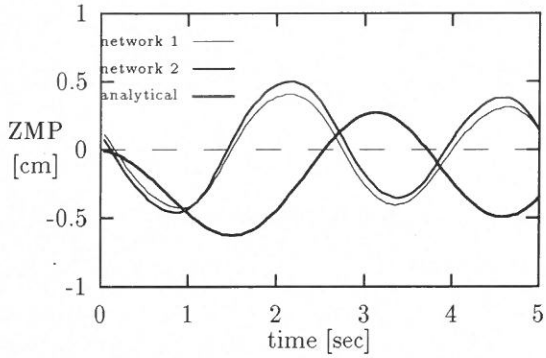

Figure 5: *ZMP-trajectories with neural trunk control (curves 1, 2) and with analytical solution (curve 3)*

The ZMP-trajectory for the analytical trunk control has almost the same amplitude as neural control, but has a bad position. This stem from the not respected new trunk velocities and positions, and leads to a destabilization, which is in the same direction. The whole biped walking robot begins to turn around the contact point. Table 1 shows the influences of this rotation on the final distances of the swinging leg from the ground, after a complete step. As expected, the errors are very small for the neural trunk control, while a dynamic stable gait with an analytically controlled trunk is not possible. The final distance in this case is greater than 0.6 cm and hence the foot can not be placed correctly on the ground.

trunk controled by	net 1	net 2	analyt.
error in [cm]	0.18	0.11	0.61

Table 1: *Final distances of the swinging leg to the ground (step length 25 cm, step height 7.5 cm)*

The rotation of the whole biped around the waist during one step is shown in figure 6.

We have made simulation experiments with five representative cycloid leg-trajectories inside the rectangle $l_{max} \times h_{max}$ in figure 6. It can be easily seen that the network is able to control the equilibrium of the biped robot for every leg-trajectory. In the beginning of the movement, the biped robot begins slowly to rotate, but afterwards the network provides the optimal control, so that the waist doesn't continue to rotate. At the end, rotations lay in the range 0.005 - 0.01 rad. This corresponds to maximal ground distances at the end of the step between 0.06 and 0.16 cm.

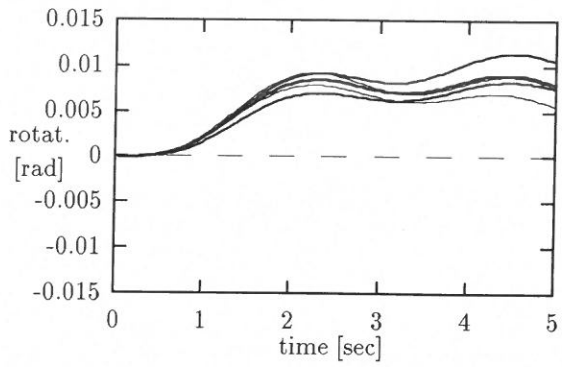

Figure 6: *Rotation of the biped around the waist for different leg-trajectories (trunk controlled by net 2).*

5. ON-LINE LEARNING

The goal of on-line learning is to adapt the general neural network to the actual leg movement. This is only possible if the network is able to control the equilibrium such that the movement can be achieved without large errors (can cause the robot to fall over). Unlearning should also be prohibited. On the other hand, we expect a good generalization by the network, i.e. the control of (similar) leg-trajectories that have not been learned on-line should also be improved.

To demonstrate this, the neural network 2 was trained with only one representative leg trajectory in the middle of the rectangle $l_{max} \times h_{max}$. Weights were changed after each complete step. The network converged very fast and after only 17 iterations the final distance between the swinging leg and the ground was smaller than 0.006 cm. Figure 7 shows the rotation of the whole robot around the waist during the control of the five different test-trajectories (compares with figure 6). The neural network is now able to control perfectly the on-line learned leg-trajectory.

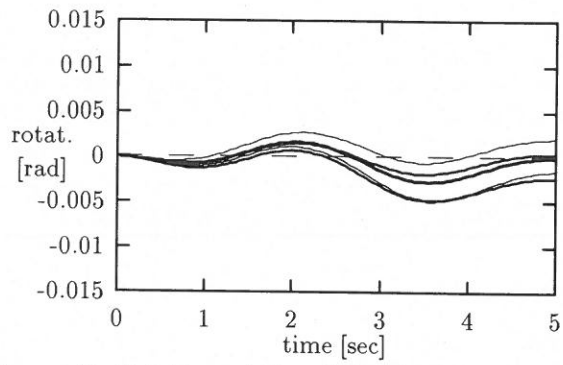

Figure 7: *Rotation of the whole biped for the control of the test-trajectories after learning on-line.*

The corresponding curve of the biped rotation is well centered and has a zero value at the end. More over, the control of the not learned on-line trajectories could also be considerably improved. Distances

trajectory	1	2	3	4	5
distance	-0.014	-0.030	0.006	0.030	0.009

Table 2: *Final distances in cm of the swinging leg to the ground after learning on-line of trajectory 3.*

are now smaller than 0.5 mm for every leg-trajectory (see table 2). Further tests with random step length and heights have shown that this is also true for any cycloid leg-trajectory falling inside the rectangle.

6. CONCLUSION

The proposed learning algorithm was accurated for the control of dynamic problem we studied. After offline learning, the neural net was able to provide the necessary trunk acceleration to maintain dynamic equilibrium. Since no gait specific a-priori knowledge was used for the generation of learning examples, the neural net should also be able to control noncycloidleg-movements. This is currently investigated.

Additionally, the learning algorithm can be used to adapt the network, on-line, to a specific leg-trajectory. The network used has only 182 weights, so it can be easily applied on-line, in contrast to a numerical approach. The significance of control approach is:

- It is a learning control method without a need for a desired output.
- Its neural controler outputs are simply the command parameters.
- It can be adaptive.
- Its control objective depends on the command parameters.

Perspective

We are now working on dynamic equilibrium control of a quadruped robot. We are studying reflex behavior when unexped situations make lost the legged-ground contact and throw off balance the robot. The 3-dimensional dynamic model of the robot exists and classical control algorithm have been tested.

REFERENCE

[1] Y. Le Cun, "Modèles connexionnistes de l'Apprentissage", *thèse de l'Université Paris 6 (Pierre et Marie Curie)*, France, june 1987

[2] D.M.A. Lee, W.H. ElMaraghy, "A Neural Network Solution for Biped Gait Synthesis", *Int. Joint Conference on Neural Networks*, Boston, Vol II, pp. 763-767,1992

[3] D.E Rummelhart, J.L. McClelland, "Parallel Distributed Processing", *The MIT Press*, Vol 1, pp. 318-362, 1986

[4] Q. Li, A. Takanishi, I. Kato, "Learning Control of Compensative Trunk Motion for Biped Walking Robot based on ZMP Stability Criterion", *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 597-603,1992

[5] A. Takanishi, et al., "Realization of Dynamic Biped Walking Stabilized with Trunk Motion", *Proceedings of ROMANSY*, 1988

[6] A. Takanishi, et al., "A Control Method for Dynamic Biped Walking under Unknown External Force", *IEEE International Workshop on Intelligent Robots and Systems, IROS* , pp. 795-801,1990

[7] A. Vukobratovic, A. Frank, D. Juricic, "On the Stability of Biped Locomotion", *IEEE Transactions on Biomedical Engeneering*, Vol. BME-17, No 1, 1970