



HAL
open science

Adaptive Neural Control with Backpropagation Algorithm

Henaff Patrick, Maurice Milgram

► **To cite this version:**

Henaff Patrick, Maurice Milgram. Adaptive Neural Control with Backpropagation Algorithm. IEEE Int. Symp. Signal Processing, Robotics and Neural Networks, Apr 1994, Lille, France. hal-01843723

HAL Id: hal-01843723

<https://hal.science/hal-01843723v1>

Submitted on 20 Nov 2024

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.

Adaptive Neural Control with Backpropagation Algorithm

P. HENAFF M. MILGRAM

Laboratoire de Robotique de Paris
10-12, Avenue de l'Europe
78 140 Vélizy Villacoublay
FRANCE
e-mail: henaff@robot.uvsq.fr

Abstract

This paper introduce an original approach to the Neural Network learning architecture for the control of robotics systems. The basic idea is to use multi-layer-network and the backpropagation algorithm without desired outputs, but with a quadratic criterion which spezify the control objective. Use a criterion avoid the computing of the inverse model of the robot which is required in the case of the desired outputs and difficult to obtain for complex systems. The method allows on-line learning that is impossible with classic backpropagation. We argued that on-line learning scheme made feasible adaptive control.

To illustrate the idea of a control criterion on backpropagation algorithm, we first considered a cartesian control problem of a simplified simulated planar manipulator. Next, we presents the results of a more difficult problem where the goal is to control the dynamic stability of a simulated planar biped. In this study, the results of the on-line learning scheme validate the adaptive control hypothesis.

Finally, we outline a more real problem, that is to control the dynamic equilibrium of a quadruped robot. We are specially studying reflex behavior to control unexpected situations.

1 INTRODUCTION

In dynamic reflex control of robots, one of problems is the required inversion of the dynamic model. This model can not, in general, be obtained analytically and it is not often uniquely determinable (kinematics re-

donduncy). So, the academics methods used criterion optimization technics to find a solution of this inversion.

Tough, these technics has limitations in reflex dynamic control of mechanical complex systems like walking machines. For example, when unexpected situations make lost the legged-ground contact and throw off balance the robot, the dynamic model and control laws to obtained the reflex behavior are often unknown.

On the other hand, the reduced computation time, the inherent robustness to real noisy data and the strong generalisation capability of multi-layer-networks (a neurocontroller can be designed even no explicit form of a control law is known) make them very suitable for control applications in robotics. In addition, neural learning algorithms are optimization algorithms.

For these reasons, it is interesting to use multi-layer-networks. A standart neural control scheme is build on the academic feedback control approach. The neural net replace the classical corrector (see Fig. 1).

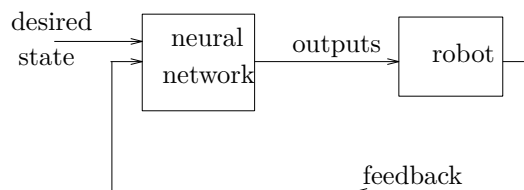


Figure 1: Neural feedback control

Multi-layer-networks are usually trained with the well known backpropagation learning algorithm schematically described on figure 2.

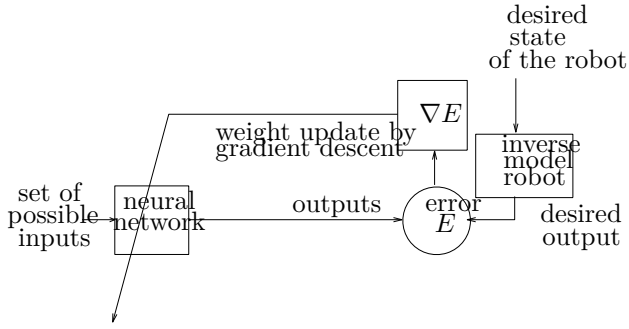


Figure 2: Standard backpropagation learning scheme

The goal of this algorithm is to minimize the quadratic error E_p (for every input pattern p) between the net and the desired outputs. Consequently, it 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).

So, the use of standard backpropagation in control applications raise some problems:

- On dynamic applications, calculate the desired output require to invert a model (geometric, kinematic or dynamic) of the robot. The choice of a solution of this inversion is based on criterion optimization (energy, manoeuvrability, dynamic equilibrium, ...) which increase the on-line learning complexity.
- It is impossible to learn on-line the desired command [2]. Then it is impossible to do adaptive control with the backpropagation algorithm.

As a result, we propose a learning method that doesn't need desired output, but only a criterion specifying directly the robotic task (distance minimization, equilibrium, ...).

2 LEARNING A CRITERION

The main basic idea is to minimize the arbitrary criterion J_p (for every input pattern p) as a function of the net output, instead of the error E_p between the net and the desired output (For more details see [1] or [3]).

In order to minimize this criterion, we use also a gradient descent method :

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

where

- w_{ji} is the weight from neuron i to neuron j ,
- o_p is the net output vector,

- $net_{pj} = \sum_i w_{ji} o_{pi}$ denotes the activation of neuron j and f_j its function (sigmoide).

We set $\delta_{pj} = -\frac{\partial J_p}{\partial net_{pj}}$ and obtain for a hidden neuron :

$$\delta_{pj}^{hid} = f'_j(net_{pj}) \sum_k \delta_{pk} w_{kj}$$

and for an output one :

$$\begin{aligned} \delta_{pj}^{out} &= -\frac{\partial J_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pj}} \\ &= -\frac{\partial J_p}{\partial o_{pj}} f'_j(net_{pj}) = \nabla J f'_j(net_{pj}) \end{aligned}$$

Instead of the actual and desired output error, we backpropagate the gradient of the criterion related to the corresponding output : $-\frac{\partial J_p}{\partial o_{pj}}$.

Therefore, the partial derivatives of the criterion must be analytically calculable. The computation of δ_{pj}^{hid} remains unchanged.

2.1 Off-line learning scheme

Figure 3 plots the new learning scheme (links to 2). The gradient criterion ∇J is computing on the new state of the simulated robot obtained with the neural outputs. Notes that we do not use a real robot in this training loop. Indeed, from the learning outset, neural outputs (command parameters) are physically incoherent.

Furthermore, we can use a robot simulation based on a degraded model. Then, we obtain a nearly optimal net which will be very easy to refine, during the control, with respect to the real robot. It is an on-line training method describe on the next subsection.

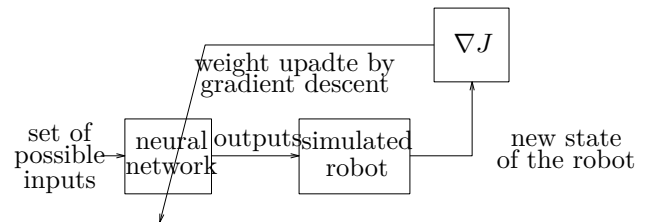


Figure 3: Learning off-line a criterion

2.2 On-line learning scheme

Figure 4 plots adaptive neural control structure where the learning loop permit to refine the weights to the real robot parameters at periodic times. Note that unlearning network should be prohibited on this loop, because of the incoherent outputs on the training outset.

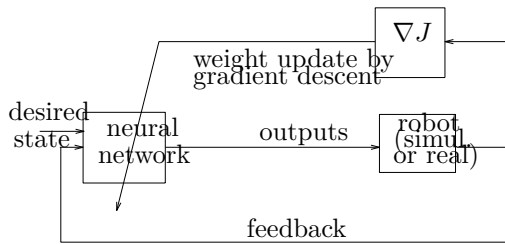


Figure 4: Learning on-line a criterion (adaptive neural control)

3 SIMPLE CARTESIAN CONTROL PROBLEM

In order to test our method, the new learning algorithm has been first applied to control the cartesian position of a three link simulated planar manipulator (without dynamic effects). The net output includes the three articular velocities. The net inputs includes the three articular positions, the three articular velocities and the cartesian errors. The criterion was the euclidean distance between the tool-center-point of the manipulator and the desired cartesian position. A two hidden layers neural net (respectively 9 and 5 neurons) was able to minimize the cartesian distance.

During tests, the tool-center-point was controlled at any cartesian position in an $1m \times 1m$ cartesian plan. Some examples of results are plotted on Figures 5 and 6. The initial articular configuration was $q_1 = 0^\circ, q_2 = 90^\circ, q_3 = 90^\circ$, and velocities are normalized with respect to $\pi \text{rad/s}$.

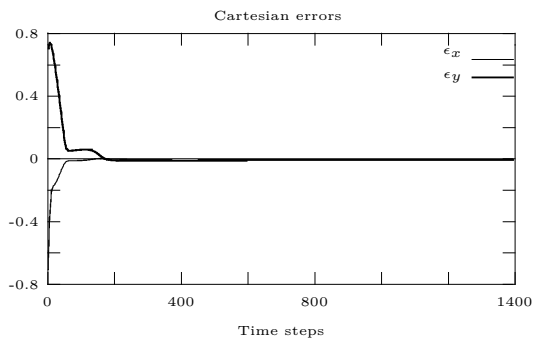


Figure 5: Cartesian control after Off-Line learning

We have tested the two learning schemes : off-line and one-line learning. The results proved that on-line learning can be used to adapt the network during the control. The interests of the on-line learning algorithm are presented in next section, for which the control problem is more difficult.

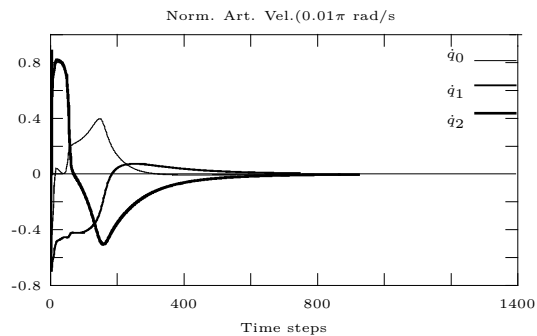


Figure 6: Cartesian control after Off-Line learning

4 DYNAMIC REFLEX CONTROL PROBLEM

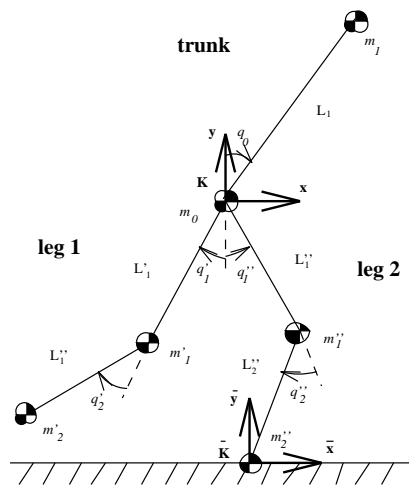


Figure 7: Model of the biped walking robot (refere to [5])

We applied our method to control the dynamic stability of a simulated planar biped (7). In this problem we proposed a neural network solution which calculates the trunk motion as a function of the actual state of the biped walking robot (q^t, \dot{q}^t) when an analytical pattern generator impose legs accelerations([8]). The goal of the network is to learned a reflex behavior to ensure the dynamic equilibrium during the single support phase.

The net output consists on the trunk acceleration \ddot{q}_0 . The net input includes:

- Articular positions, velocities and accelerations of the two legs.
- Articular position and velocitie of the trunk.

The dynamic stability of a walking machine can be define by the ZMP (*Zero-Moment-Point*) position ([7])

and[4]). It must fall always into the supporting surface. During the single-support-phase, it corresponds to a single point. Then, dynamic stability is guaranteed when the ZMP position is equal the contact point position.

Thus, the criterion J_p can be stated as follows:

$$J_p(\ddot{q}_0) = (x_{zmp} - x_{contact})^2$$

and gradient as

$$\frac{\partial J_p}{\partial \ddot{q}_0} = 2(x_{zmp} - x_{contact}) \frac{\partial x_{zmp}}{\partial \ddot{q}_0}$$

Is the only information used by the learning algorithm to find the optimal solution.

After off-line 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.

Additionally, learning algorithm can be used to adapt on-line the network 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.

Figure 8 plots the ZMP position during the on-line training for 150 biped strides. The network is updated at the end of each stride. From the outset of the weight adaptation, ZMP oscillation stems from the off-line learning result. The adaptive neural control decrease the amplitude of ZMP motion (i.e increase equilibrium). This establishes that on-line learning scheme adjust the neural controller in order to solve the control objective. So it is an adaptive neural control system.

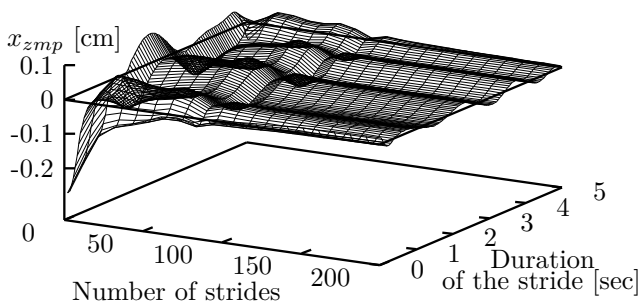


Figure 8: ZMP position during the On-line learning, establishes that adaptive neural control increase biped equilibrium.

5 CONCLUSION

Our goal was to propose the modification of the back-propagation algorithm in order to learn control objective without desired output. The principle of our approach is to express the control objective as a criterion. This criterion is backpropagated through the network instead of the error between desired and net output.

In conclusion, the significance of our control approach is :

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

We are now working on dynamic equilibrium control of a quadruped robot. We are studying reflex behavior when unexpected situations make lost the legged-ground contact and throw off balance the robot. The network compute legs torques to ensure the dynamic equilibrium. We used the 3-dimensional dynamic model of the robot which exists and which on classical control algorithm have been tested.

References

- [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 Engineering*, Vol. BME-17, No 1, 1970
- [8] P. Henaff, H. Schwenk, M. Milgram " A Neural Network Approach of the Control of Dynamic Biped Equilibrium", *International Symposium on Measurement and Control in Robotics*, Session As.II-19,1993