J Intell Robot Syst (2007) 49:311–323
DOI 10.1007/s10846-006-9040-x

A Neural Tactile Architecture Applied to Real-time Stiffness Estimation for a Large Scale of Robotic Grasping Systems

J. L. Pedreño-Molina · A. Guerrero-González · J. Calabozo-Moran · J. López-Coronado · P. Gorce

Received: 10 February 2005 / Accepted: 2 May 2006 /
Published online: 21 March 2007

Abstract This paper presents a model for solving the problem of real-time neural estimation of stiffness characteristics for unknown objects. For that, an original neural architecture is proposed for a large scale robotic grasping systems applied for unknown object with unspecified stiffness characteristics. The force acquisition is based on tactile information from force sensors in robotic manipulator. The proposed model has been implemented on a robotic gripper with two parallel fingers and on a one d.o.f. robotic finger with opponent artificial muscles and angular displacements. This self-organized model is inspired of human biological system, and is carried out by means of Topographic Maps and Vector Associative Maps. Experimental results demonstrate the efficiency of this new approach.

Keywords Adaptive learning · Grasping · Neural estimator · Robotic force control · Stiffness

J. L. Pedreño-Molina
Department of Information Technologies and Communications,
Polytechnical University of Cartagena, Cartagena, 30202 Murcia, Spain

A. Guerrero-González · J. López-Coronado
Department of System Engineering and Automatic, Polytechnical University of Cartagena,
Cartagena, 30202 Murcia, Spain

J. Calabozo-Moran
Department of Electrical and Electronics Engineering, University of Leon,
Avenida de Portugal 41, 24071 Leon, Spain

P. Gorce (✉)
Handibio-LESP EA 3162, University of the South Toulon-Var,
BP 20132 83957 La Garde Cedex, France
e-mail: Philippe.gorce@iut-cachan.u-psud.fr

1 Introduction

Since few years, the design and development of biologically inspired robotic grasping system are extensively studied. One of the main objective of such researches is to develop flexible and adaptive systems able to interact with an unknown environment. In such a context, non linear stiffness behaviour between applied displacement and resulting force is a great deal. In literature, this problem is tackled considering the deformable material on the object and on the fingertip. The majority of them are based on analysis of contact between robotic fingers covered by deformable material and a rigid object. Using this approach Hyun-Yong in [1] estimates the deformation of a soft human and artificial fingertip by means of area variation. A function of the displacement in the contact zone between the finger and the object is used. Experimental results with human fingers demonstrate that nonlinear stiffness function can be approached by exponential or potential functions. Ambrosi [2] has extended those works for discriminating softness, based on kinesthetic and cutaneous tactile information in humans. Here the CASR (Contact Area Spread Rate) paradigm is introduced and applied in psychophysical experiments.

The second important point to consider in reproducing a grasp is to take into account the natural compliance of muscle and tendons. Many works have proposed impedance laws [3] for robotic systems applied to jumping [4], walking [5], robot manipulator [6]... In grasping field, Nguyen [7] simulates a 3 d.o.f. robotic finger with a deformable material with an impedance controller. This work consider the dynamic equation and a fingertip force sensor. Reznik [8] uses a dynamic model for control systems for deformable fingertips, based on the discretization of soft volume into a interconnected virtual masses. This is an adaptative approach as the parameters are modified along the contact process. Howard [9] extend these works to the grasp of 3D non-rigid objects with two cooperative manipulators.

A lot of proposed works on deformation estimation of objects are based on vision sensors approach [10] and few of them suggest to use tactile sensors in robotic grasping systems. Because force sensors decrease the control complexity, we propose a solution for real-time stiffness control in non-rigid objects based on tactile exploration. This model has been implemented in robotic platform to demonstrate its performances.

The remainder of this paper is organized as follows. Section 2 describes the requirements for the control system in order to develop a general model that can be applied to several robotic manipulators with different joint controllers and dynamic environments. The structure of the proposed neural model is presented in Section 3. Experimental results are shown in Section 4. Finally, a conclusion is given in Section 5.

2 Requirements in the Control System for Stiffness Estimation

Starting from the robotic grasping characteristics, we impose several requirements in our control system. Firstly, the estimation of the stiffness characteristic of each object has to be based on learned neuron maps, which give the nonlinear behaviours of contact between robotic joint and the non-rigid objects. The second is the on-line capability of adding to the neural structure new samples of stiffness curves (adaptive

learning). This is based on a decision parameter, which is a function of the distance from the contact behaviour to the learned stiffness maps. Third requirement is to carry out the maximum convergence velocity of force applied to unknown objects without any oscillations (better compliance). It allows robotic fingers to quickly reach the estimated displacement based on desired final force to be applied on unknown non-rigid object. The fourth is the positioning control of several robotic manipulator devices with the ability of moving the strains of two joints agonist–antagonist artificial muscles by means of biologically inspired neural models. Finally, this model has to be able to control all the joint of a complex robotic grasping manipulator.

3 Structure of the Neural Model

The proposed architecture links three connected neural models in order to solve all above described requirements. The input of the controller is the desired final force to grasp the object with unknown stiffness characteristic. Indeed, a K co-contraction parameter can be used to supply the controller joint in robotic hands with torque control. The K parameter represents the force in joint when the equilibrium position is reached. In Fig. 1, the general structure of the neural model is shown.

3.1 Displacement Neural Estimator

This neural architecture is dedicated to obtain the accurate final position of the manipulator joint as a function of the desired final force in contact point (between the object to be grasped and the tactile sensors of the joint) and the values corresponding to behaviours of learned objects. This joint position is estimated starting from initial samples that determine the nearest stiffness curve.

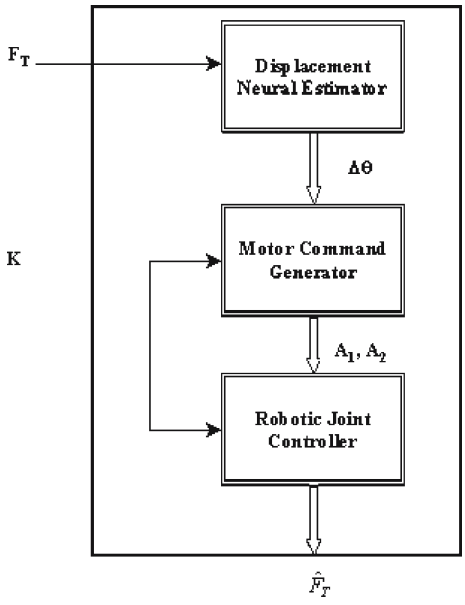

Fig. 1 General structure of neural model

In the learning phase, few objects with different characteristics of stiffness train the system. In each case, the joint produces small increments of displacement ($\Delta\theta$) starting from an initial small contact. It produces a force increment (ΔF). The vector formed by (ΔF, $\Delta\theta$) is stored in a two dimensional map. Figure 2 shows the obtained curves, in this phase, for different objects.

For the actuation phase, the displacement neural estimator model is divided into three modules: samples acquisition module, stiffness estimator and joint displacement generator. In Fig. 3, scheme of these connected modules is represented.

Samples acquisition module allows to obtain the initial samples as a function of the joint position control or force control. The selection depend on how the robotic joint controller and tactile acquisition are designed. In force control selection, the system measures joint position increments corresponding to fixed force values belong to the operation range of the sensor.

By the other hand, in joint position control the system has to produce displacement increments belong to force variation range of each object when pressed by the sensor. It allows the system to obtain series of significant points of the object that has been pressed.

In this case, each new sample of position increment depends on the previous position. This dependence is function of the slope of the curve that fits all previous points (ΔF, $\Delta\theta$). So, it is necessary to estimate the slope of the curve in the first sample. Then, next joint increment will be obtained from a mathematical expression as a function of slopes of learned curves. Figure 4 shows the polynomial that fits to this expression.

This function has been obtained from learned curves of force-position increments and the desired displacement in order to get a minimal number of representative samples for the stiffness estimation.

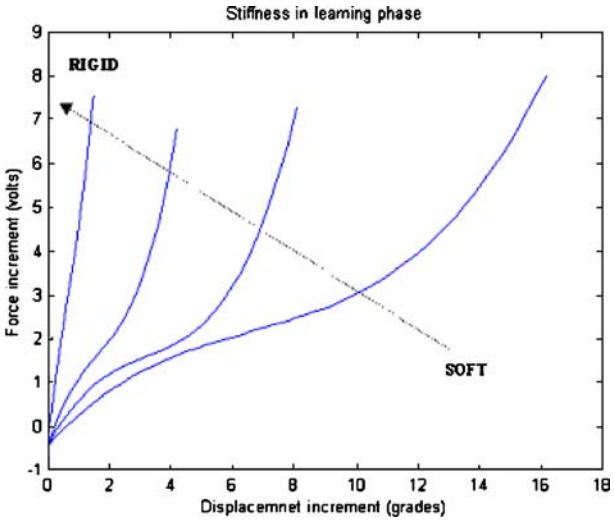

Fig. 2 Force and displacement in learning phase

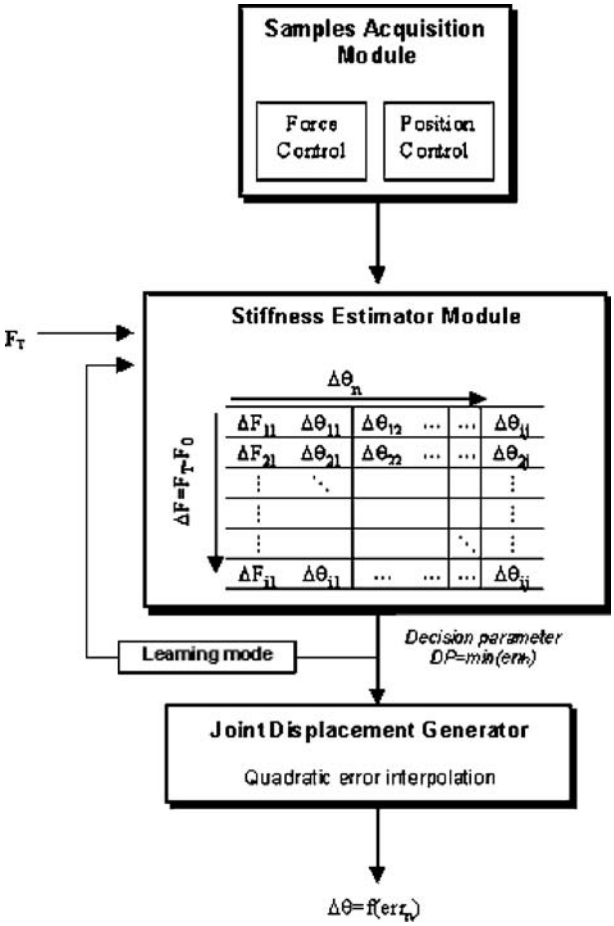

Fig. 3 Displacement neural estimator scheme

The next joint increment is given by:

$$\Delta\theta_{i+1} = \begin{cases} \sum_{i=0}^{i=samples-1} a_i \cdot R^i, & R \le R_{max}/2 \\ R_0, & R > R_{max}/2 \end{cases} \tag{1}$$

the slope is

$$R_= \frac{\Delta F_i}{\Delta\theta_i} \tag{2}$$

and a is the vector of coefficients of the polynomial

The "stiffness estimator" module is based on Topographic Map [11] architecture. It consists on a two-dimensional map, which is a neural representation of the incremental joint position displacement (linear or angular) and of the obtained force. Each cell is a neuron of the network whose weight is the desired value ($\Delta\theta$). The internal representation of the neural topographic map is shown in Fig. 3.

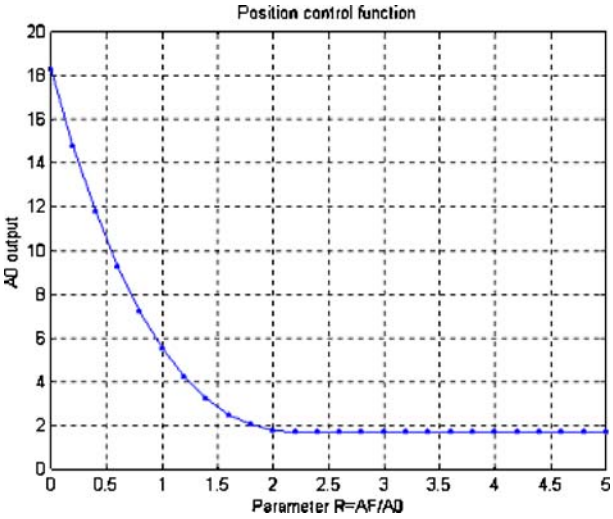

Fig. 4 Function of position control

The inputs of this module are the force increments from each sample vector from the unknown object. The force increments (ΔF) active the row, directly. For the column selection a quadratic error parameter between the learned stiffness curves (neuron weights) and each initial force-angle samples is evaluated. For each sample n,a neuron weight Zij is selected. Then a decision parameter DP is compared with updating umbral, where the error parameter for sample n is defined by:

$$err_n = \sqrt{\Delta\theta_n^2 - \left(Z_{ijn}\right)^2} \tag{3}$$

and the decision parameter is:

$$DP = \max_{n=1}^{samples} (err_n) \tag{4}$$

If DP is greater than a prefixed umbral (U), the system will decide to add the force-position characteristics of this object in the topographic map. In that case, the system generates a sequence of position increments, in order to add in the cell of the neural net the new obtained values.

By the opposite, if DP parameter is less than the umbral, the system makes an interpolation of error signals err_n from initial samples. So, the incremental values of forces from the obtained samples (ΔF_n) and their related error parameter (err_n) the interpolation is made by means of a polynomial function. Let b be the obtained k-dimensional coefficient vector, the desired incremental force (ΔF_T) will generate on error by means of the expression:

$$err_T = \sum_{k=0}^{k=orden-1} b_k \cdot \Delta F_T^k \tag{5}$$

So, from Eq. 3, the displacement increment is obtained by:

$$\Delta\theta_T = \sqrt{err_T^2 + \left(Z_{ijT}\right)^2} \tag{6}$$

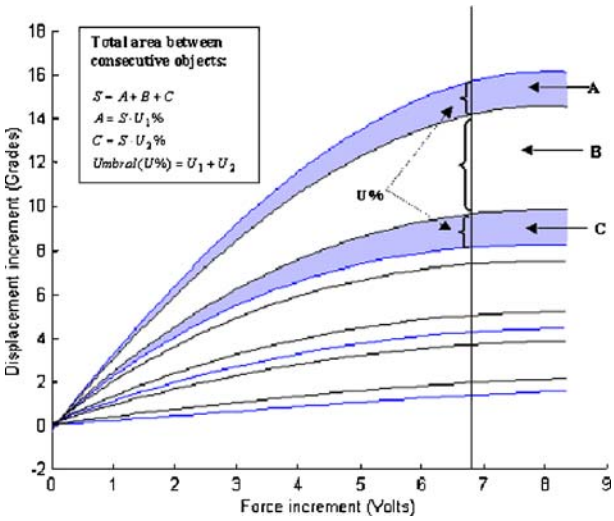

Fig. 5 Umbral for decision parameter DP

The weight value Zij_T corresponds to the activated cell of the topographic map as a function of $\Delta F_T = F_T - F_0$, where:

F_T: Desired final force.
F_0: Initial contact force of reference.
i: This index selects the activated file by ΔF_T.
j: This index selects the column corresponding to the learned curve, which is nearest to the behaviour of the current object.

The chosen value of umbral parameter (U)is critical for the control system performance and has to control two requirements. On one hand, U has to be small in order

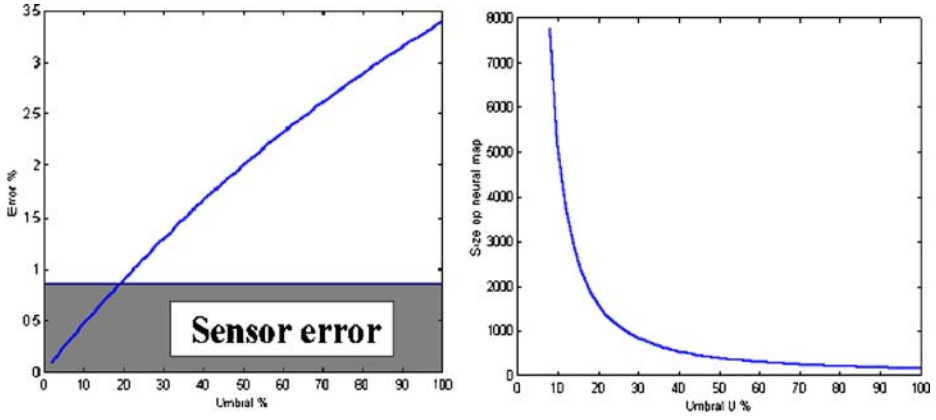

Fig. 6 Error and map dimension

to the produced error for $\Delta\theta$ estimation when $DP<U$, was small. But on the other hand, small U implies a high processing time for the control due to the probability of $DP>U$ to be greater. In this case, the neural topographic map increases as a function of the number of new learned objects. The umbral is computed as a percentage of area between two learned consecutive curves. This is shown in Fig. 5.

If U=100%, no new objects will be added to the learning map, because DP<U, and the produced error will be maximum. By the opposite, if U=0%, the error produced is minimum but the dimension of topographic map groves without any limits. The error percentage and the size of the map, as a function of U parameter are represented in Figs. 6a and b, respectively. Due to the precision characteristics of tactile sensor, a minimal error e_{sensor}, always remains.

Finally, the "joint displacement generator" carries out the final position $\Delta\theta$, as a function of input force that it is desired for grasping the body, the interpolated error parameter, and DP. If this signal is less than predefined umbral, the force control or position control will determine the final position $\Delta\theta$, directly.

3.2 Motor Command Generator

The second model is a tension generator. This neurobiological self-organized neural network is a VAM (*Vector Associative Map*) algorithm (developed in Boston University) [12].

This algorithm has been applied to tactile-motor integration in robotic grasping tasks [13] and visually guided reaching in robotic stereoheads [14]. In Fig. 7, the scheme of this model is shown. This neural structure learns the mapping between the input vector, formed by final displacement increment ($\Delta\theta$) and co-contraction K parameter (for manipulator that requires torque control) and the tensions (ΔA1, ΔA2) to control the opponent movement of a robotic finger. Figure 8 shows the projections of spatial coordinates into motor coordinates.

In learning phase, neuron weights (*Wij*) are adapted by Difference Vector (DV) block that computes the difference between the sensorial and motor target position and the actual position. The samples are obtained by means of an Endogenous Random Generator (ERG) block that generates tensions to the joint controller. The weight matrix is obtained Applying the Least Square Method (LMS).

The adaptation ability of this VAM model allows the system to update the neural weights in actuation phase. It is based on iterative method that modifies this matrix

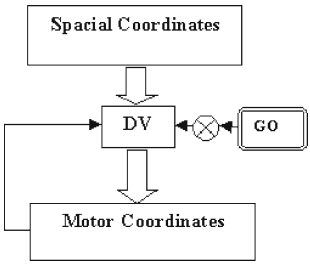

Fig. 7 Schematic diagram of the mode

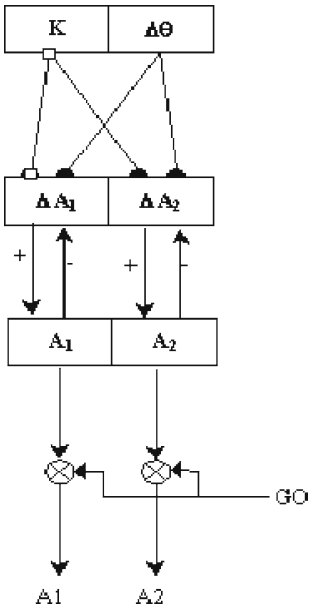

Fig. 8 Spatial and motor coordinates mapping

in the opposite direction of the quadratic error gradient. The iterative equation that calculates the new *(Wij)* values is given by:

$$\frac{dw_{ij}}{dt} = \eta \cdot s_j \cdot \left(r_i - \sum_{j=1}^{M} w_{ij} \cdot s_j \right) \tag{7}$$

where:

- i index selects the element of (ΔA1, ΔA2)
- r vector – motor positions-
- j is the element of (K, $\Delta\theta$),
- s vector – spatial positions- that fire the *Wij* neuron weight.
- M is the dimension of s vector
- η parameter represents the learning rate that determines the weight convergence speed of the neural net.

In order to smooth the movement of the robotic joint towards the estimated final position, corresponding to a predefined force, a feedback sigmoid function *GO* is applied to (A1, A2) outputs. The shape of this signal defines the convergence speed of the finger joint in the tactile process. Figure 9 shows the modulator function GO.

3.3 Robotic Joint Controller

This model directly acts on the controller of the robotic manipulator. It receives the (A1, A2) tensions form motor command generator and the K co-contraction parameter from the initial conditions of grasping process. The output are the commands for positioning the robotic finger in the estimated position that produces the predefined grasping force as a function of the stiffness of unknown objects.

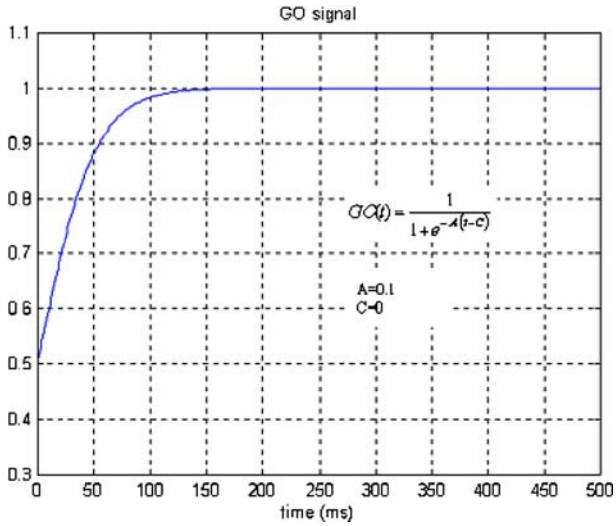

Fig. 9 GO modulator function

The structure of this neural module depends on the design of the joint controller. An example is presented in [15], where a FLETE (Factorization of Length an Tension Muscle) algorithm is employed to control the force and longitude of an agonist-antagonist system of opponent muscles of a robotic arm. In [16], Mo-Yuen applied a neural CMAC (Cerebellar Model Articulation Controller) algorithm, developed by Albus [17], to approximate the opponent movement of the human arm with two-link robot arm.

4 Experimental Results

In order to evaluate the performance of this model, two robotic manipulators Fig. 10, with different operation ways, have been programmed for real-time stiffness

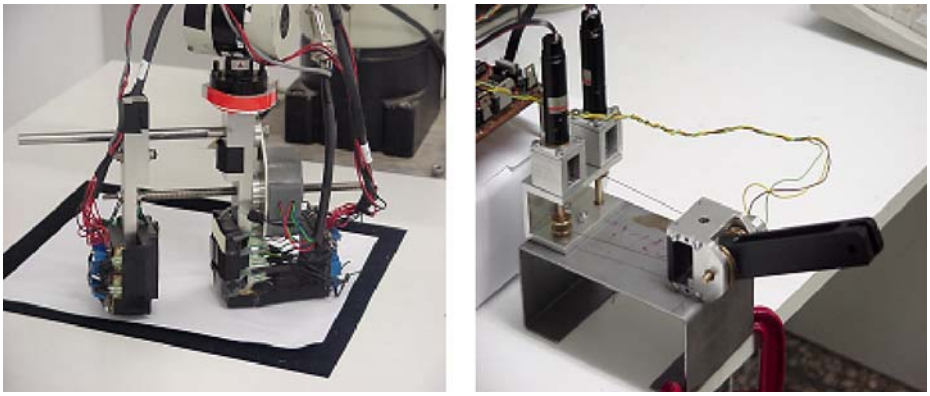

Fig. 10 Experimental robotic platforms

estimation in a grasping process. Figure 10a shows the parallel robotic gripper with artificial tactile skins and in Fig. 10b a robotic finger with artificial agonist–antagonist muscles an fingertip sensor is shown.

In the learning phase, several objects with different stiffness characteristics have trained the displacement neural estimator module. In Fig. 2, these samples are shown. Indeed, in motor command generator module, an initial neuron weight matrix has been obtained by means of the ERG block. It has generated the spatial-motor coordinates projective map.

In the operation phase, unknown objects with different stiffness behaviour have been used. In all cases, only three initial samples have been chosen employing force control and position control. The input information to the system is the final force (F_T) with which the object has to be grasped and the K co-contraction parameter. The vector (F_T,K) is the input to the displacement neural estimator module. Depending of the controller of each platform, force-control or displacement-control has been used for sampling. In all cases, only three initial samples have been necessary to estimate the displacement increment in Topographic Map architecture. It is carried out by means of DP decision parameter, err_n vector, the learned neuron map and the joint displacement generator module. In Fig. 11, the three samples and the estimated final displacement is shown. The four curves correspond to the stiffness behaviours from learned objects.

Once $\Delta\theta_T$ has been obtained, the (K, $\Delta\theta_T$) vector supplies the motor command generator, in order to carry out the desired tensions when the equilibrium position is reached. Applied the GO signal of Fig. 9, (A_1,A_2) output vector is carried out by means of neural model based on VAM algorithm. The response is shown in Fig. 12.

In this case a K=0.2 parameter has been predefined. In absence of load, K can be small, but if any external force is applied to the joint, it is necessary to increase this parameter in order to overcome the variation of dynamic behaviour conditions. In order to increase the convergence velocity of tension signals, it is possible to modify A coefficient of the sigmoid GO expression. The new input and the generated output

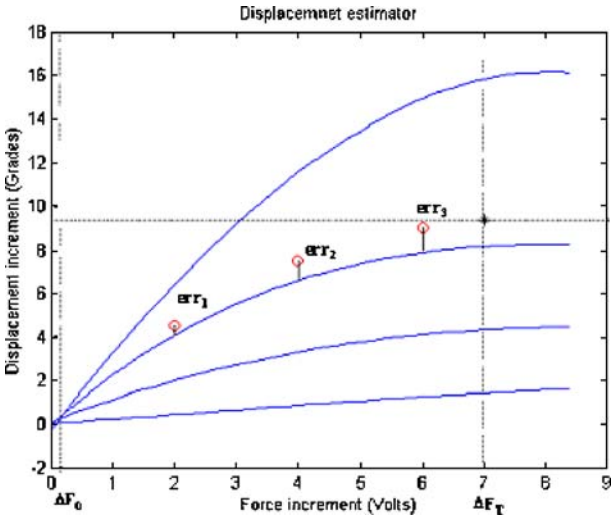

Fig. 11 Displacement estimation

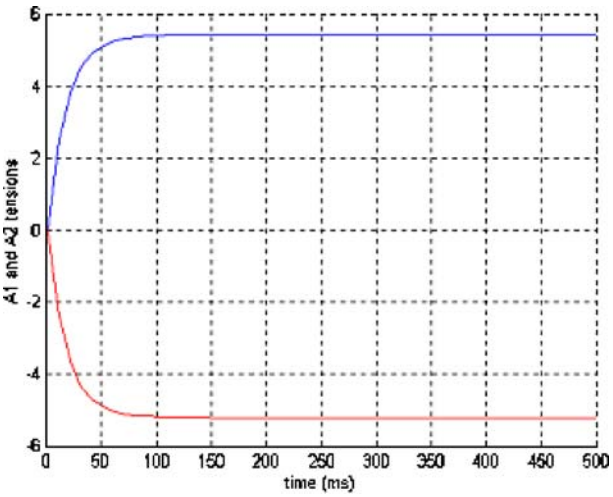

Fig. 12 Tensions for joint opponent controller

are used to update the neuron weights of the neural net, by means of the expressed iterative equation. So the proposed model is able to modify its performances when variations on environment exit.

Finally this tension signals are the input to the robotic joint controller. It acts over the pair the muscles of joint in order to reach, with the better compliance, the estimated final position. Because the robotic platform in our group has been mechanically designed with the capacity of opponent movement, the referenced FLETE algorithm has been used to place the joint in the desired final position.

5 Conclusion

As conclusion, the integration of these three learning neural model results in a real-time feedback system that is able to smoothly control the movement of an anthropomorphic finger, for any force and co-contraction (K) value, with no dependence of nonlinear stiffness characteristic of objects. Indeed, this neural controller avoids the system to know the interactive dynamic relationships of the contact points between the robotic system and each touched object.

Acknowledgements The authors would like to thank the Spanish Research Foundation (CICYT) by the support given to the project TIC99-0446-C02-01, and to the European Community for the support to the Brite Research Project SYNERAGH, ref. BRE-CT98-0797.

References

1. Hyun-Yong, H., Kawamura, S.: Analysis of stiffness of human fingertip and comparison with artificial fingers. In: Proc. IEEE Int. Conf. Syst. Man, and Cybern. vol. 2, pp. 800–805 (1999)
2. Ambrosi, G., Bicchi, A., De Rossi, D., Scilingo, E.P.: The role of contact area spread rate in haptic discrimination of softness. In: Proc. IEEE Int. Conf. on Robotics and Automation, vol. 1, pp. 305–310 (1999)

3. Hogan, N.: Impedance control: An approach to manipulation. Part I- Theory, Part II- Implementation, part III- Applications. Trans ASME J. Dyn. Syst. Meas. Control **107**, 1, 24 (1985)
4. Guihard, M., Gorce, P.: Simulation of dynamic vertical jump. Robotica, Cambridge **19**(1), 87–91 (2000)
5. Gorce, P., Guihard, M.: Joint impedance pneumatic control for multichains systems. Trans. ASME J. Dyn. Syst. Meas. Control **121**(2), 293–297 (1999)
6. Slotine, J.J.E., Li, W.: On the adaptative control of robot manipulator. Int. J. Rob. Res. **6**, 49–59 (1987)
7. Nguyen, T., Arimoto, S., Hyun-Yong, H., Kawamura, S.: Control of physical interaction between a deformable finger-tip and a rigid object. In: Proc. IEEE Int. Conf. on Syst. Man, and Cybern, vol. 2, pp. 812–817 (1999)
8. Reznik, D., Laugier, C.: Dynamic simulation and virtual control of a deformable fingertip. In: Proc. IEEE Int. Conf. Robotics and Automation, vol. 2, pp. 1669–1674 (1996)
9. Howard, A.M., Bekey, G.A.: Recursive learning for deformable object manipulation. In: Proceedings of the 8th Int. Conf. on Advanced Robotics, vol. 1, pp. 939–944 (1997)
10. Chen, C., Zheng, Y.F.: Deformation identification and estimation of one-dimensional objects by using vision sensors. In: Proc. IEEE Int. Conf. on Robotics and Automation, vol. 3, pp. 2306–2311 (1991)
11. Guenther, F.H., Bullock, D., Greve, D., Grossberg, S.: Neural representations for sensorimotor control. Part III: Learning a body – centered representation of a three-dimensional target position. J. Cogn. Neurosci. **6**, 341–358 (1994)
12. Gaudiano, P., Grossberg, S.: Vector associative maps: unsupervised real-time error-based learning and control of movement trajectories. Neural Netw. **4**, 147–183 (1991)
13. López-Coronado, J., Pedreño-Molina, J.L., Guerrero González, A., Gorce, P.: A Neural model for a Visal-tactile-motor integration in robotic reaching and grasping tasks. Robotica, Cambridge **20**(1), 23–31 (2001)
14. Guerrero, A., Coronado, J.L., Garcia-Cordova, F.: A neural network for target reaching with a robot arm using a stereohead. In: Proc. IEEE Int. Conf. on Syst. Man, and Cybern, vol. 1, pp. 842–847 (1999)
15. Bullock, D., Contreras-Vidal, J.L., Grossberg, S.: A neural network model for spino-muscular generation of launching and braking forces by opponent muscles. In: Proc. Int. Joint Conf. on Neural Networks, vol. 3, pp. 450–455 (1992)
16. Mo-Yuen, C., Menozzi, A.: A self-organized CMAC controller. In: Proc. IEEE Int. Conf. on Industrial Technology, vol. 1, pp. 68–72 (1994)
17. Albus, J.S.: A new approach to manipulator control: the cere bellar model articulation controller (CMAC). Trans. ASME J. Dyn. Syst. Meas. Control **97**, 220–227 (1975)