An adaptive learning mechanism for teaching a robotic hand to grasp

Gabriel Gómez*, Alejandro Hernandez A.†, Peter Eggenberger Hotz* and Rolf Pfeifer*
*Artificial Intelligence Laboratory
Department of Informatics, University of Zurich, Switzerland
Email:[gomez, eggen, pfeifer]@ifi.unizh.ch

†Developmental Cognitive Machines Lab
University of Tokyo, Japan
Email: alex@prince.pe.u-tokyo.ac.jp

Abstract— **In this paper we describe our ongoing work on the control of a tendon driven robotic hand by an adaptive learning mechanism evolved using a simulator developed over the last years. The proposed neural network allows the robotic hand to explore its own movement possibilities to interact with objects of different shape, size and material and learn how to grasp them. As the evolved neural controller is highly adaptive, it will allow us in the future to systematically investigate the interplay between morphology and behavior using the same, but adaptive neural controller.**

I. Introduction

Previously we have been working with an industrial robot arm, which exhibits precise, fast and controllable movements (see [1], [2], and [3]). In this paper we try to apply the same concepts in a more complex robotic setup: a robotic hand with 13 degrees of freedom, complex dynamics provided by a tendon driven mechanism of actuation and different types of tactile sensors.

We present the “ligand-receptor” concept that can be easily used by artificial evolution to explore the growing of a neural network, value systems and learning mechanisms systematically for a given task. In our implementation, receptors abstract proteins of specific shapes able to recognize specifically their partners molecules. Ligands, on the other hand, are molecules moving around, which also have specific shapes, and are basically used as information carrier for their receptors. The shape of a receptor determines which ligand can stimulate it, much in the same fashion, as notches of jigsaw pieces fit exactly into the molds of other pieces. When a receptor is stimulated by a matching ligand (signaling molecule), the following mechanisms are elicited on a neuron: connect to a neuron expressing a partner receptor, release a ligand molecule, express a receptor(see [4] and [5]). As a result of this evolutionary process, the specification of the neural network (i.e., number of neuronal fields, size of each neuronal field, a set of receptors expressed by each neuronal unit, a set of signals that can be released by the sensor neurons) was obtained and then embedded as a neural controller for a robotic hand.

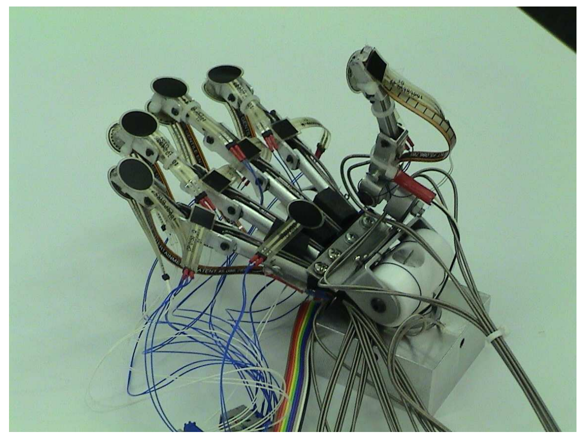


Fig. 1. Tendon driven robotic hand.

In the following section we describe our robotic setup, the tendon driven mechanism, and the position, type and number of sensors used. In section III we specify the robot’s task. In section IV we explain in more detail the “ligand-receptor” concept and its implications. Then we present some experimental results as well as a discussion and future work.

II. Robotic setup

Our robotic platform can be seen in Fig. 1. The tendon driven robot hand is partly built from elastic, flexible and deformable materials. For example, the tendons are elastic, the fingertips can be covered with deformable materials and between the fingers there is also deformable material (see [6] and [7]).

The hand apply an adjustable power mechanism develop by [8]. The wire’s guide is flexible, and moves proportionally to the load applied to the fingertip (see Fig. 3). When the load is small, the guide moves toward the fulcrum and the fingertip moves faster with low torque. In contrast, when a big load is applied to the finger the guide moves away from the fulcrum, resulting in a higher torque motion.

Due to the spring-like characteristics of the wire's guide, the finger's joints gain flexibility likes the human hand fingers. The relationship between the torque T, which is generated in the fingertip, and the force F, which pulls the wire, is:

$$T = LFsin(\theta_1) \tag{1}$$

Here, the angle θ_1 is defined as an angle θ_2 with the fulcrum-action line and a fulcrum-pulling force line, an angle β with the fulcrum-guide roll line and a pulling force line, and a distance x from the fulcrum to a point of lever (guide roll):

$$\theta_1 = Tan^{-1}\left(\frac{xsin(\theta_2 - \beta)}{L - cos(\theta_2 - \beta)}\right) \tag{2}$$

If the spring is connected near the fulcrum β is a constant value. The distance x is given by the following equation:

$$x = \frac{Fcos(\theta_1 + \theta_2 - \beta)}{k} \tag{3}$$

θ_1, θ_2 and L are defined in Fig. 2a, respectively. In a manipulator with multiple joints, the torque needed at the root joint is larger than those in the extreme joints. In the tendon driven mechanism, the torque from the other joints interferes with the torque at the root joint, resulting in a large torque at the base of the manipulator.

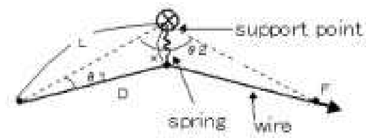
In order to acquire in a simple way the passive motion function in addition to the adjustable velocity and torque functions, the current robotic hand uses a spring type wire as an elastic guide for the inner wire. If the load does not require the application of a high torque on the finger, the wire remains straight. In the case of a heavy load that requires more torque, the outer wire bends drawing the inner wire taut (see Fig. 3). After performing several tests, we decided on stainless steel for the outer wire, and nylon for the inner wire, given that this combination provided with the smallest friction. Figures 2b and 2c, show the configuration of the mechanical parts.

The robotic hand has 13 degrees of freedom that are driven by 13 servomotors and has been equipped with two types of sensors: flex/bend and pressure sensors. For the flex/bend sensor, the bending angle is proportional to its resistance and responds to a physical range between straight and a 90 degree bend, they are placed on every finger as position sensors. The pressure sensors (force-sensing resistor (FSR)) are positioned on the fingers and on the back of the hand for tactile interaction with the environment) as detailed in table I.

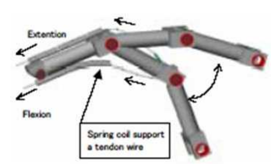
We control the robot hand using a TITech SH2 controller. The controller produces up to 16 PWM (pulse with modulation) signals for the sevomotors and acquire the values from the bending and pressure sensors. The motor controller receives the commands through a USB port.

We used two computers: one hosts the neural network and communicates with the sensorimotor control board to acquire the sensory data and produce the motor commands, the other computer is in charge of the visualization of the neural network. When learning takes place, the two computers exchange the synaptic changes via TCP/IP. The sensorimotor data was stored in a time series file for off-line analysis.

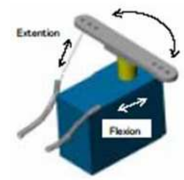
Using this setup we were able to see the actual behavior of the robot (observer's perspective), we had access to the sensorimotor data generated by the interaction of the hand and objects in the environment through the sensorimotor control board and additionally we could see inside the neural network and investigate how the learning mechanism, the sensory input and the interaction with the environment were shaping the neural structure (robot's perspective).



(a) schematic design

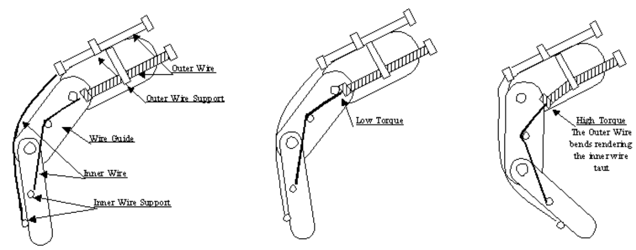


(b) motion of flexion and extension



(c) Power module by using RC servo motor

Fig. 2. Adaptive joint mechanism by using spring coil type of outer tube.



(a)Finger Structure (a) Low torque & HighSpeed (b) High torque & Low Speed

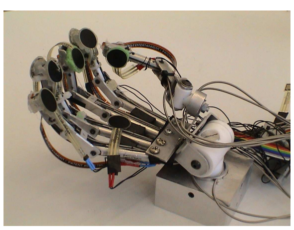
Fig. 3. Three functions of the adaptive joint mechanism.

TABLE I
ROBOTIC SETUP

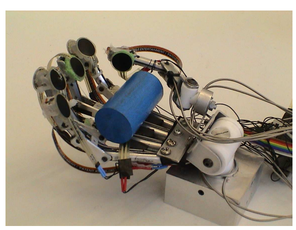
Number and type	Sensor position
5 bending	one per finger
1 pressure	on the palm
5 pressure	one per fingertip
5 pressure	on the back of the hand
4 pressure	on the middle of the fingers (except thumb)
Total number of sensors	20

III. Robot task

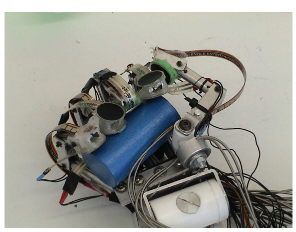
A typical experiment was performed as follows: at the beginning, the robot hand was initialized with all the fingers fully extended (see Fig. 4a). Then an object was placed on its palm whose weight was high enough to activate the pressure sensor located in the palm (see Fig. 4b), the hand started to explore its own movements by randomly moving the fingers, increasing and decreasing the position of the servo motors produced the pulling of the tendons, which made the fingers move back and forth (See figures 2b and 2c). Eventually the fingers encountered the object, and some pressure sensors were activated, by comparing the previous pressure readings with the current ones the learning mechanism taught the hand about the success of its exploratory efforts and finally the hand grasped the object successfully (see Fig. 4c). After that the hand remained grasping the object until it was taken away from it, in that case the sensory input was reduced to zero and the hand returned to the initial state with the fingers fully extended. A typical experiment can be see in Fig. 4.



(a) Initial state



(b) Object in the palm



(c) Final state

Fig. 4. Typical experiment.

IV. Ligand-receptor concept

To be able to explore both the connectivity, the learning, and the value systems, we use the ligand receptor concept (see [1], [2], and [3]). As can be seen in Fig. 5 a couple of ligand-receptor such as *signal0* and *receptor0* can be used to explore the connectivity, while a couple of ligand-receptor such as *signal2* and *receptor2* can be used for learning a specific task. There is not possible interaction between a couple of ligand-receptor such as *signal1* and *receptor3*.

We define the probability of an interaction between a ligand and a receptor as the affinity, which calculates an artificial binding between the two entities. This affinity parameter *aff* determines which molecule (signaling molecule) interacts with

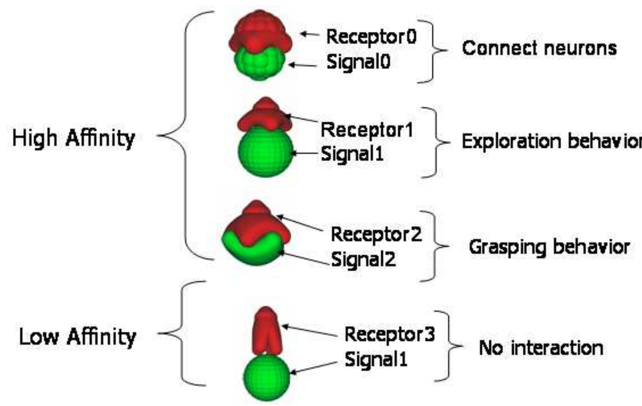


Fig. 5. Ligand receptor concept.

which partner (receptor). To each signal and receptor a real valued number and a function are assigned. This function is implemented as follows:

$$a_{12} = f_{aff}(aff_1, aff_2) = Exp^{-\alpha(aff_1 - aff_2)^2} \tag{4}$$

where:
aff_1, aff_2 are the real valued numbers representing the geometric properties of the substances. α is the affinity parameter with positive values. If α is high, the two substances have to be very similar (i.e., $aff_1 \sim aff_2$) to get a high functional f_{aff} value, if α is low, the substances can be more different to still get high f_{aff} values. Molecules compete for a docking site (receptor) and their success of binding depends on the affinity between the molecule and the docking site and on the concentrations of the competitors. Figure 5 shows examples of pairs of ligands and receptors with low and high affinity as well as their potential use.

A. Exploring the connectivity of the neural network

At the moment of the creation of the neural network, the system automatically assigns a set of receptors to each neuron and releases a signal to start connecting the neural network. Then the system attempts connections between neurons based on the receptors expressed by each neuron.

For instance, the system will create a synaptic weight to link a neuron from the *hiddenField* (see Fig. 6b) to a neuron in the *motorField* (see Fig. 6c) only if the two neurons had expressed the same matching receptors. The same applies for the synaptic weights connecting the neurons between the *sensorField* (see Fig. 6a) and the *hiddenField* (see Fig. 6b) as well as the synaptic weights connecting the neurons between the *motorField* (see Fig. 6c) and the *motorActivities* (see Fig. 6d).

The components of the neural structure and its connections to the robot hand are depicted in Figure 6. The number of

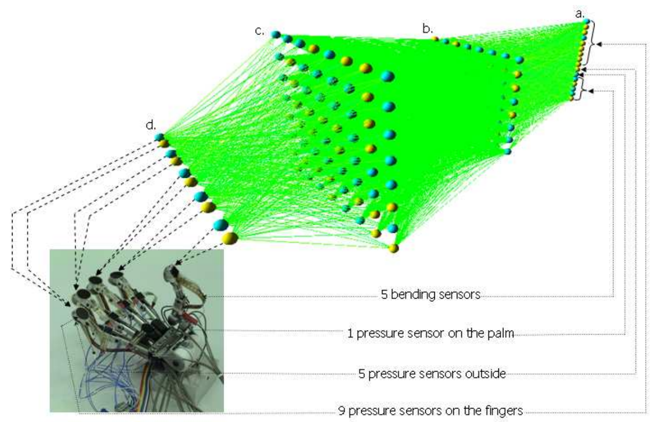


Fig. 6. Neural structure and its connections to the robot's sensors and motors. Neuronal areas: (a) sensorField. (b) hiddenField. (c) motorField. (d) motorActivities.

TABLE II

NEURAL STRUCTURE

Neuronal Area	number of neuronal units
sensorField	16
hiddenField	64
MotorField	64
MotorActivities	10
Total neuronal units	154

neuronal units in each neuronal area can be found in table II.

1) Sensory field: The size of the neuronal area *sensorField* (see Fig. 6a) was 1x16.

2) Neuronal field and motor field: The size of the neuronal area *hiddenField* (see Fig. 6b) was 8x8 and its neuronal units had a sigmoid activation function.
The size of the neuronal area *motorField* (see Fig. 6c) was 8x8.
The size of the neuronal area *MotorActivities* was 1x10 (see Fig. 6d), they directly control the fingers of the hand. Each finger was moved by two pairs of antagonistic neurons, one causes the finger to contract and the other to extend.

3) Synaptic connections: Neuronal units in the neural network were connected by using the ligand-receptor concept described above. All the weights were initialized randomly from a gaussian distribution with mean=0.0002 and sigma=0001. See Fig. 7.

B. Value system

Value systems are neural structures that are necessary for an organism to modify its behavior based on the salience or value of an environmental cue ([9], [10]). Such "value systems" have been implemented in robotic systems to study their role in adaptive behavior ([11] and [12]). In our case, the pressure sensors were the value system that taught

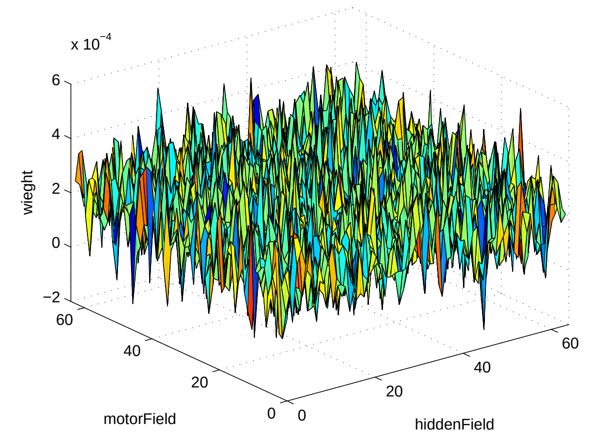


Fig. 7. Initial weights between the *hiddenField* and the *motorField*.

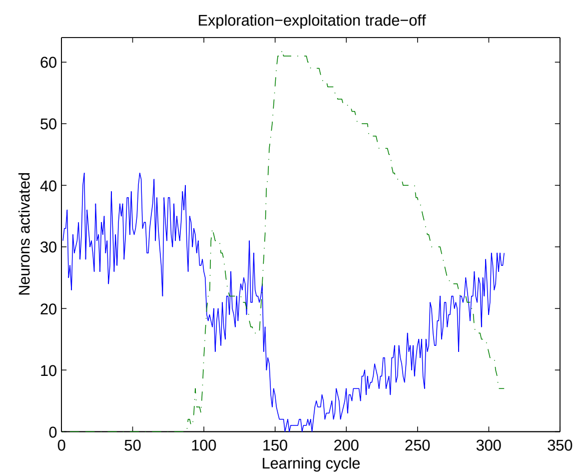


Fig. 8. Exploration-exploitation trade off (when $\mu = 0.5$, 50% of the motor neurons can fire randomly). The solid line represents the number of neurons that randomly fired at each learning cycle. The dashed line represents the number of motor neurons that learned and are activated by themselves.

the robot about the success of its exploratory movements. Whenever the pressure sensor on the palm was activated and this activation exceeded a given threshold (e.g., an object was put on the palm), the corresponding sensor neuron released a signaling molecule, the motor neurons that had expressed a receptor with high affinity to that signaling molecule will be allowed to fire randomly and therefore the hand will be in its "Exploration behavior". If a finger is in contact with an object, depending of the force and the direction of the movement, the reading of the pressure sensor on that finger will increase or decrease. In either case a corresponding sensor neuron will release a signal that will be used by the learning mechanism to *reward* or *punish* the active motor neurons at that particular moment.

C. Learning Mechanism

The proposed neural network allows the robotic hand to explore its own movement possibilities due to the random activation of the neurons in the *motorField* (see Fig. 6c). The outputs of the motor neurons are determined by the following formula:

$$O_i = \begin{cases} 1.0 & : \quad if \ \ \sigma(\sum_{j=1}^{n} \omega_{i,j} S_j) >= \theta_1 \\ 1.0 & : \quad if \ \ \sigma(\sum_{j=1}^{n} \omega_{i,j} S_j) < \theta_1 \ and \ \mu <= \theta_2 \\ 0.0 & : \quad otherwise \end{cases} \tag{5}$$

Where:
O_i output of the i-th neuron,
S_j input of the j-th neuron,
$\omega_{i,j}$ synaptic weight between the i-th and the j-th neurons,
σ is the sigmoidal function $\sigma(x) = \frac{1}{1+e^{-\alpha x}}$,
α is the slope parameter of the sigmoid function,
θ_1 is the threshold for the neuron to fire by itself,
μ is a uniform value between 0 and 1 randomly generated,
θ_2 is the threshold that determines how many motor neurons are allowed to fire randomly.

As can be seen, if the output exceeds a certain threshold (θ_1), the neuron will fire by itself. Otherwise the neuron is allowed to fire randomly, for that purpose an uniform value (μ) between 0 and 1 is randomly generated. By comparing μ with a second threshold θ_2, only a percentage of the population of motor neurons is allowed to fire.

The solid line in Fig. 8 represents the number of neurons that randomly fired at each learning cycle when $\mu = 0.5$ (50% of the neurons were allowed to fire randomly). The dashed line represents the number of motor neurons that learned and were activated by themselves. Figure 8 elicits the exploration-exploitation trade-off. At the beginning of the experiment about half of the motor neurons are randomly activated, so the robot hand will explore its environment, over time, the learning mechanism modified the synaptic weights causing more neurons to learn, that means more motor neurons will have an output that exceeds the threshold θ_1 and will fire by themselves and less and less motor neurons will fire randomly, so the more it learns the less than it has to explore. Around the learning cycle number 160, the object was taken away from the robot hand, so there is no sensory input, therefore the robot hand restarted its exploratory activity.

Synaptic modification was determined by both pre and post synaptic activity and resulted in either strengthening or weakening of the synaptic efficacy between two neuronal units. The active neurons controlling the robot hand were "rewarded" if the movement of the fingers exerted a higher activation of the pressure sensors and "punished" otherwise. In this way the synaptic connections between the neuronal areas *hiddenField* (see Fig. 6b) and *motorField* (see Fig. 6c) were updated.

Figure 9 gives a graphical explanation of the sequence of events resulting in the change of a synaptic weight.

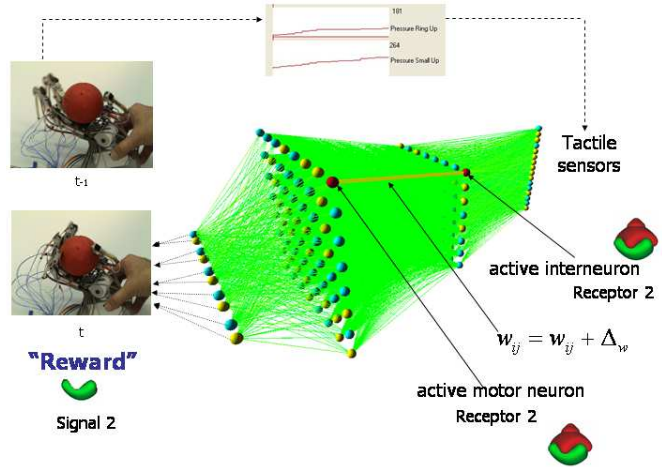


Fig. 9. Learning mechanism. As the robotic hand enclosed an object with its fingers, some pressure sensors were activated, a signaling molecule was then released, and eventually a synaptic weight between a pair of active neurons was updated (only if the respective neurons had expressed a receptor that matched with the signaling molecule).

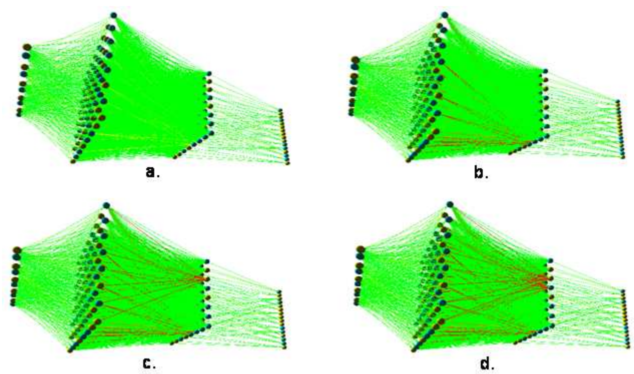


Fig. 10. Synaptic changes over time during a typical grasping experiment.

V. Experimental results

Figure 10 illustrates the learning process. The synaptic changes can be seen in an OpenGL animation resulting from the real time data exchange between the computer hosting and running the neural network and the computer visualizing the learning process. All the synaptic weights were kept between -1 and 1. A learning cycle (i.e., the period during which the current sensory input is processed, the activities of all neuronal units are computed, the connection strength of all synaptic connections are computed, and the motor outputs are generated) had a duration of approximately 0.08 seconds. Figure 11 shows the final states of experiments performed with several objects of different color, shape and material.

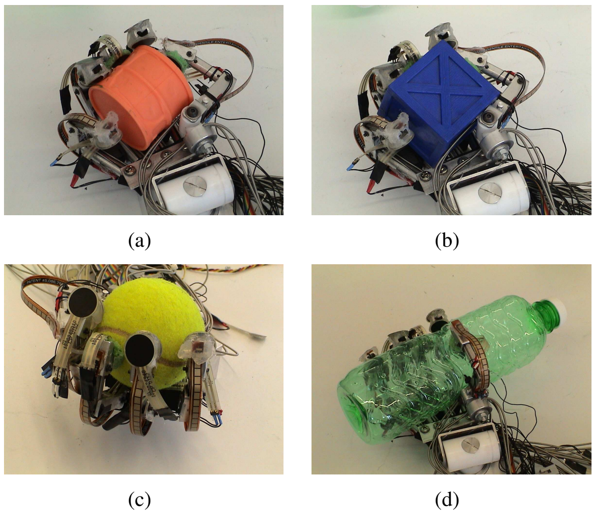


Fig. 11. Grasping different objects.

VI. Discussion and future work

Predefining all possible sensors and movement capabilities of a robot, will certainly reduce its adaptivity. In order to be adaptive, a neural controller must be able to reconfigure itself to cope with environmental and morphological changes (e.g., additional sensors, not only in number, but different types of sensors could be added, damaged sensors over time, additional or fewer degrees of freedom, etc.). As we cannot predict the above possible changes, the system should be allowed to explore its own movements and coherently adapt its own behavior to the new situation; something that cannot be achieved by a purely reflex-based system.
Until now no engineering methods exist how to tackle with unforseen changes, we expect to contribute to the solution of these problems with our approach.

The robustness of the evolved neural controller will be tested by making systematic changes in the hand's morphology (e.g., position and number of the pressure sensors, different types of sensors, stronger motors, covering materials in order to increase the friction with objects) to investigate how the neural controller reacts to unforeseen perturbations.

Now that we have a system able to learn to foveate by using visual information (see [1], [2], and [3]), and a system able to learn to grasp objects based on tactile information, it will be interesting to see how the two systems can be integrated and how far the system can go to solve more demanding tasks (e.g., object manipulation, reaching, catching moving objects).

Acknowledgment

Gabriel Gómez was supported by the EU-Projects ADAPT: Artificial Development Approach to Presence Technologies (IST-2001-37173) and ROBOT-CUB: ROBotic Open-architecture Technology for Cognition, Understanding and Behavior (IST-004370). Alejandro Hernandez was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (B), 2004, No.16360118.

References

[1] P. Eggenberger Hotz, G. Gómez, and R. Pfeifer, "Evolving the morphology of a neural network for controlling a foveating retina - and its test on a real robot." in *Proc. of Artificial Life VIII*, 2002, pp. 243–251.

[2] G. Gómez and P. Eggenberger Hotz, "An evolved learning mechanism for teaching a robot to foveate." in *Proc. of the 9th Int. Symp. on Artificial Life and Robotics (AROB-9)*, 2004, pp. 655–658.

[3] ——, "Investigations on the robustness of an evolved learning mechanism for a robot arm," in *Proc. of the 8th Int. Conf. on Intelligent Autonomous Systems (IAS-8)*, 2004, pp. 818–827.

[4] P. Eggenberger Hotz, "Evolving morphologies and neural controllers based on the same underlying principle: Specific ligand-receptor interactions," in *Hara, F. and R. Pfeifer, (eds.), Morphofunctional Machines: The New Species (Designing Embodied Intelligence)*, 2003, pp. 217–236.

[5] ——, "Combining developmental processes and their physics in an artificial evolutionary system to evolve shapes," in *On Growth, Form and Computers. Edited by Sanjeev Kumar and Peter Bentley.*, 2003.

[6] H. Yokoi, A. Hernandez Arieta, R. Katoh, W. Yu, I. Watanabe, and M. Maruishi, *Mutual Adaptation in a Prosthetics ApplicationIn Embodied artificial intelligence. Lecture Notes in Computer Science.*, F. Iida, R. Pfeifer, L. Steels, and Y. Kuniyoshi, Eds. Springer, ISBN: 3-540-22484-X, 2004.

[7] R. Pfeifer and F. Iida, "Morphological computation: Connecting body, brain and environment." *Japanese Scientific Monthly*, vol. 58, No. 2, pp. 48–54, 2005.

[8] Y. Ishikawa, W. Yu, H. Yokoi, and Y. Kakazu, "Research on the double power mechanism of the tendon driven robot hand," *The Robotics Society of Japan*, pp. 933–934, 1999.

[9] G. M. Edelman, *Neural Darwinism: The theory of neural group selection*. New York:Basic Books, 1987.

[10] K. J. Friston, G. Tononi, G. N. Reeke, O. Sporns, and G. M. Edelman, "Value-dependent selection in the brain: Simulation in a synthetic neural model," *Neuroscience*, vol. 59, no. 2, pp. 229–243, 1994.

[11] O. Sporns, N. Almassy, and G. Edelman, "Plasticity in value systems and its role in adaptive behavior." *Adaptive Behavior*, vol. 8, pp. 129–148, 2000.

[12] O. Sporns and W. Alexander, "Neuromodulation and plasticity in an autonomous robot." *Neural Networks (Special Issue)*, vol. 15, pp. 761–774, 2002.