Proceedings of the 2006 IEEE
International Conference on Robotics and Biomimetics
December 17 - 20, 2006, Kunming, China

A BioInspired Neural Controller For a Mobile Robot

Michele Folgheraiter, Giuseppina Gini, Alessandro Nava, Nicola Mottola
Department of Electronic and Information
Politecnico di Milano
Piazza L. da Vinci, MILANO , I-20133, Italy32
{folghera, gini}@elet.polimi.it

Abstract— **This paper focuses on the study of a bio-inspired neural controller used to govern a mobile robot. The network's architecture is based on the understanding that neurophysiologists have obtained on the nervous system of some simple animals, like arthropods or invertebrates. The neuronal model mimics the behavior of the natural cells present in the animal, and elaborates the continuous signals coming from the robot's sensors. The output generated by the controller, after scaling, commands the wheel rotation and therefore the robot's linear and angular velocity. The mobile robot, thanks to the controller, presents different behaviors, like reaching a sonorous source, avoiding obstacles and finding the recharge stations. In the network architecture different modules, charged of different functionality, are regulated and coordinated using an inhibition mechanism. In order to test the control strategy and the neural architecture, we implemented the system in Matlab and finally in hardware using a dedicated dual processor board equipped with an ARM7TDMI micro-controller. Results show that the neural controller can govern the robot efficiently with performances comparable with those described about the animal.**

Keywords: Biorobotics, Neural controller, Robot navigation.

I. Introduction

Service robotics today requires synthesizing robust automatic systems able to cope with a complex and dynamic environment. Even for simple behaviors, like autonomous navigation and obstacle avoidance, the most advanced systems sometime fail, especially in presence of noisy information. However, if we look at nature, we can see that in very "simple" animal, insects or invertebrates, the deambulation behavior is always accomplished [1],[2].

Biorobotics, in this context, tries to give an answer to these issues mimicking [3], in the machine, the behaviors and the structure of living creatures. Studying the anatomy and the physiology of the animal it is possible to understand how nature has attempted to solve crucial functional issues. Many scientists are focusing their attention on the part of the animal's nervous system that is involved in the sensorimotor coordination. This part, considering the phylogenetic evolution of the living organism, is the simplest and oldest one [4]. From the functional point of view, it covers a primary role because it permits the animal to perceive, explore and change the environment where it lives. Because it is relatively simple and accessible, we have a deeper understanding on how it works in comparison with the higher nervous centers.

II. The neural Controller Architecture

Many researchers have considered a bio-inspired control system in order to control a robot [5], [6], [7], [8]. Sometimes the animal not only inspires the control strategy for the robot, but also its kinematics and functionalities. In our point of view there are two possible goals for bio-robotics: the first is to use the robotic system to test and validate the models we have for the animals, the second is to use the proposed models to design new kinds of robots. Reaching both these goals at the same time is very difficult and at times dangerous because a compromise is required. In this work we are more focused on the second goal, with the main idea to use the knowledge we have from the biological studies of the animal to synthesize a "better" robotic system. Better, from the functionalities point of view, than a similar system not based on biological knowledge.

The neural controller we implemented is based on the early studies conducted by Braitenberg [9] twenty years ago on very simple automata vehicles, and on the more recently studies that Barbara Webb et al. [10],[8] carried out on a robot cricket, whose principal behavior is to follow sonorous sources.

Inspired by these studies we tried to implement new paradigms that do not have any evidence in the biological studies of the animal. Sometimes it is near impossible to perform a complete comparison between our model and the biological model, since we are more interested in the robotic functionalities than in mimicking the animal. Nevertheless we are convinced that studying the living organism gives us a big opportunity to synthesize new kinds of "intelligent" machines. In the neural architecture we propose (Figure1) it is possible to individuate two neuron layers: a sensory layer and a motor layer. The sensory layer is composed by 7 neurons connected with different sensors: contact sensors, sound sensors, energy stations sensors, and an energy level sensor. The motor layer is composed by two neurons whose outputs, opportunely scaled, control the velocity of the two robot's wheels. The synapses of each neuron can be excitatory or inhibitory, so to regulate the activation level and therefore the neuron output. In the network we can also distinguish four principal parts that are assigned to four different behaviors: collision avoidance, reaching the sound emitter, reaching the recharge platforms,

1-4244-0571-8/06/$20.00

energy level monitoring. In the next four paragraphs we will enter in detail in each of these single parts.

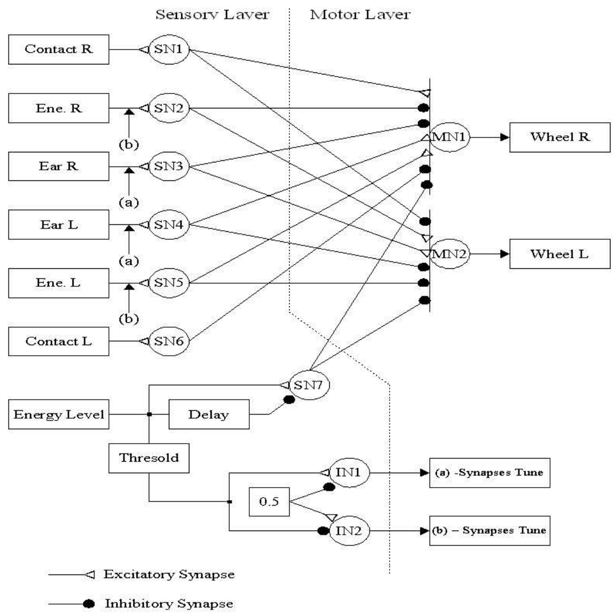

Fig. 1. The Neural Controller Architecture

A. *Collisions Avoidance*

This behavior involves the action of neurons SN1, SN6, MN1, and MN2 (Figure 1). In particular SN1 and SN2 have only an excitatory input that receives the signal directly from the sensors. The output of SN1 excites the motoneuron MN1 and inhibits the motoneuron MN2, making the robot to turn left when the right contact sensor (**Contact R**) is activated by the collision with an obstacle. The output of SN6 excites the motoneuron MN1, and permits the robot to turn right when an object is revealed by the left contact sensor. As in the schema, there is an asymmetry in the cross inhibition; this is necessary in order to force a left turning when an object is encountered exactly in front of the robot. Depending on the synapses value, the robot turn with less or more strength when it encounters the obstacle.

B. *Reaching the Sound Emitter*

The principal goal of our robot is to reach a sound source, mimicking the behavior of the cricket female in tracking the male position. This behavior is possible thanks to the neurons SN3, SN4, MN1, MN2. As we see from the schema (Figure1), SN3 realizes an inhibitory synapse with MN1 and an excitatory synapse with MN2, so the robot turns right if it receives n the right ear (EAR R) a signal stronger than the one received by the left ear (EAR L). The other two connections (SN4-MN1 and SN4-MN2) of this sub-network are completely symmetric, and permit the robot to turn left if the sound signal perceived by the left ear is stronger than that of the right ear.
In this network the symmetry in the direct inhibitions works because we want to reach the source, not to avoid it.
In reality it is possible to use this kind of architecture to develop other kinds of behaviors if we use also other kinds of sensors.

C. *Recharge Platforms Reaching*

The *Recharge Platforms Reaching* behavior, with the *Energy Level Monitoring*, is critical for the robot "life", to guarantee energy for some activity. The corresponding behavior in the animal behaviors is searching for food, that the animal can perceive using olfactory or chemical receptors.
The sub-network involved in this task is that one constituted by neurons: SN2, SN5, MN1, MN2. The architecture is similar to that one which permits the *Sound Emitter Reaching* behavior, but now only the energy-stations sensors are involved.

D. *Energy Level Monitoring*

This sub-network, located in the bottom part of figure 1, has a key role in the control system. It permits to regulate the priority of the concurrent behaviors: *Sound Emitter Reaching* and *Recharge Platforms Reaching*. They are concurrent because it is not possible to follow two different targets at the same time .
The neural circuit contains two different parts: one constituted by neurons SN7, MN1 and MN2, and the other by IN1 and IN2. Both these circuits receive as input the signal coming from the sensor that measures the available energy. When the energy level goes below a fixed threshold, a signal reaches both the excitatory synapse of neuron IN1 and the inhibitory synapse of the neuron IN2. Because of this, the neuron IN1 increases its membrane activity and IN2 decreases it. Their outputs go directly to influence the synapses values of neurons SN2, SN3, SN4 and SN5. When IN1 is activated, and therefore IN2 results deactivated, the *Sound Emitter Reaching* behavior is suppressed and the *Recharge Platforms Reaching* behavior takes control of the motoneurons. Note that this mechanism doesn't control the *Obstacles Avoiding* behavior, because it needs to be active also during the energy stations tracking.
When the robot needs energy it is attracted by the energy stations, the more the energy level is low the more the *Recharge Platforms Reaching* behavior takes control of the robot. When the robot reaches a recharge station, the changing level of energy is perceived by the neuron SN7 that becomes active and rises its output. This causes the motoneurons inhibition and therefore the robot remains motionless until the recharge is complete.

III. The Neurons Model

Each neuron in the neural controller is modelled using equations 1, where **P** is the membrane potential and **Y** the neuron's output. The potential changes depend on the excitatory inputs x_i and on the inhibitory inputs x_j, weighted by We_i and Wi_j respectively. The term kP performs a forgetting mechanism, regulated by the forgetting constant k.

This permits the neuron to avoid the saturation, and therefore to adapt to different stimulation patterns [11].

$$\begin{cases} \dot{P} = \sum_{i=1}^{n} We_i x_i - \sum_{j=1}^{m} Wi_j x_j - kP \\ Y = Th(P) \end{cases} \tag{1}$$

In this neuron model the activation function is a piecewise linear function (Equation 2), that bounds the output in the range 0 - 1, and at the same time keeps the system linear. Usually, in many neural networks architectures[12], a non linear activation function is introduced to improve the performance of the network in approximating non linear functions. But here what is important is to avoid the neuron saturation and therefore the network instability.

$$Th(P) = \begin{cases} 0 & P \leq 0 \\ P & 0 < P \leq 1 \\ 1 & P > 1 \end{cases} \tag{2}$$

In figure 2 we can see the potential and the output of the neuron when stimulated with one excitatory and two inhibitory signals.

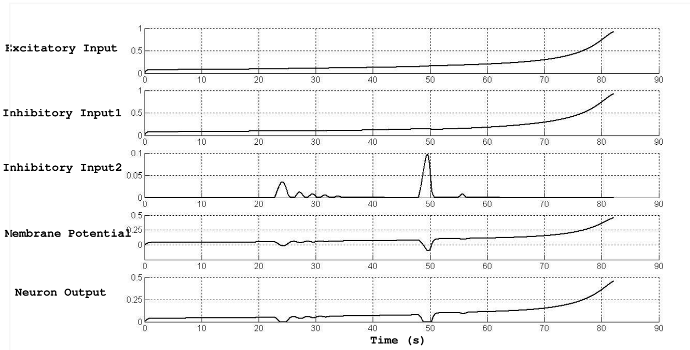

Fig. 2. The neuron signals.

A. Variable Synapse

Describing the network architecture, in paragraph II-D, we indicated the capability of IN1 and IN2 to change the input synapse value of the neurons SN2, SN3, SN4 and SN5. This is possible modelling the synapse with a first order differential equation (Eq.3).

$$\begin{cases} \dot{W}_s = W_c - k_d \\ W = Th(W_s) \end{cases} \tag{3}$$

where W_s is the synapse internal state, W_c is the tuning signal coming from the neuron IN_i, K_d a term that allows the depolarizing mechanism, necessary to decrease the synapse value when the tuning signal is low, and finally Th is the function described in equation 2. In figure 3 we see that, when the signal W_c decreases to zero also the weight decreases, and therefore the excitatory input of the neuron doesn't have influence on its potential.

This inhibition mechanism is very important to regulate and coordinate the robot behaviors. What is interesting here, is the possibility to modulate the behaviors in a continuous way, this means that it is possible to switch from a behavior to another with a smooth trend.

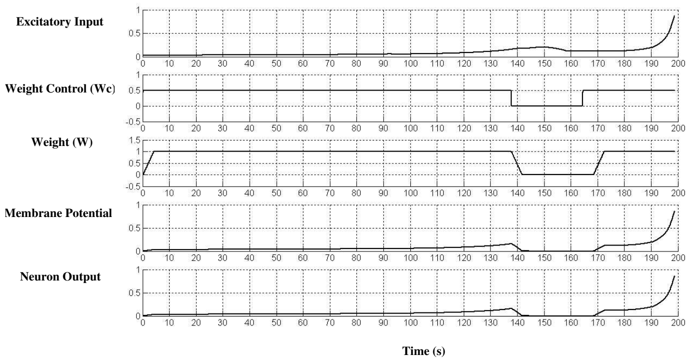

Fig. 3. The weight tuning mechanism.

IV. The Robot Model

In order to test our controller we developed a virtual world where the robot can move and interact with objects.
The arena (10x10 meters, see figure 4) contains obstacles represented by circles of different diameters, a sound source (the target position for the robot) and two recharge platforms. The mobile robot (0.6x0.4 meters) has two wheels in a differential drive configuration; controlling independently the velocity of the left and right wheels the robot can move forward, backward, turn left or turn right. The robot direct kinematic can be solved using the system of equations 4 :

$$\begin{cases} x(t) = \frac{1}{2} \int (v_l(t) + v_r(t)) \cos(\theta(t)) dt \\ y(t) = \frac{1}{2} \int (v_l(t) + v_r(t)) \sin(\theta(t)) dt \\ \theta(t) = \frac{1}{2} \int (v_l(t) - v_r(t)) dt \end{cases} \tag{4}$$

where $(x(t), y(t))$ is the robot position, $\theta(t)$ its orientation and v_l, v_r the linear velocities of the left and right wheel respectively, obtained directly from the wheel angular velocities. All of these quantities are respective of an inertial reference system. In this model we neglected the dynamics of the robot, therefore we do not considered mass and inertia. This simplification is plausible, especially if it is possible assume that the robot is very light, nevertheless future models may also include this aspect.
The robot is equipped with two sound sensors located at the right and left side in front of the robot, two energy station sensors located in the same positions, and two circular contact sensors (see figure 4).

The intensity of the sound signals received by the sound sensors is modelled by equation 5:

$$I_{Received} = I_{Source} \frac{1}{K_1 + K_2 d + K_3 d^2} \tag{5}$$

The intensity of the sound received ($I_{Received}$) by the sensors is directly proportional to the intensity of the sound

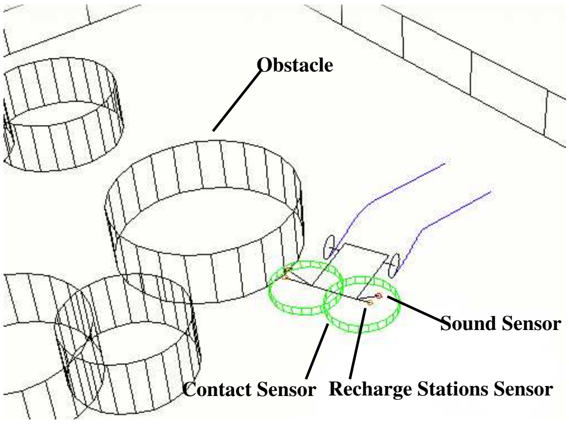

Fig. 4. The robot equipped with sensors inside the Arena.

source (I_{Source}) and inversely proportional to a quadratic polynomial of the source distance d. A similar equation can be used also to represent the signal level received by the recharge-station sensors.
The level of the signal generated by the contact sensor can be model by equations 6.

$$\begin{cases} I_{Received} = K_1 Com + K_2 Com^2 \\ Com = ObstacleDistance - SensorRadius \end{cases} \tag{6}$$

Here Com is the compression of the circular sensor when it encounters the obstacle.

V. Results in Simulation

All the simulations were done using Matlab; for the integration method of the differential equations we used the Runge-Kutta algorithm with an integration step of 0.001s.
The first simulation we performed, was done to test the *Sound Emitter Reaching* and the *Obstacles Avoiding* behaviors. As mentioned before this two behaviors work together to govern the robot movements. The robot moves from a **Start**(in the figure is the star symbol) position to the position of the sound source.

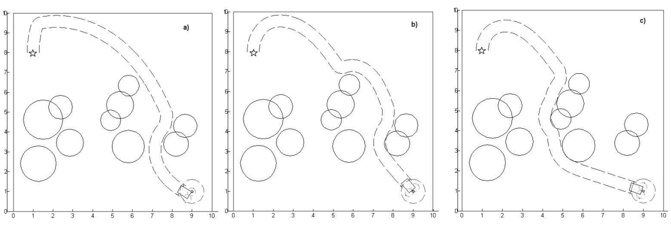

Fig. 5. Simulations of the *Sound Emitter Reaching* and the *Obstacles Avoiding* behaviors with different values for the inhibitory synapses.

In figure 5 we see the paths followed by the robot with three different values for the cross inhibition synapses (**a** 0.1, **b** 0.5, **c** 0.6), located in the sub-network that performs the *Sound Emitter Reaching* behavior. Increasing the values for these two synapses makes the robot to narrow the curves. This is useful to more precisely reach the target, however near the sound source a strong inhibition (quite similar in both the motoneurons) slows down the robot velocity.

Another experiment was for testing all the behaviors. Now the robot has a limited amount of energy that doesn't permit it to directly reach the target (sound source). In this experiment we located two recharge platforms at the two side of the upper part of the arena. As we see in figure 6 the robot, at the beginning, performs a trajectory quite similar to that one obtained without considering the *Recharge Platforms Reaching* behavior; however, because now the robot has a finite energy storage, it needs to refill.

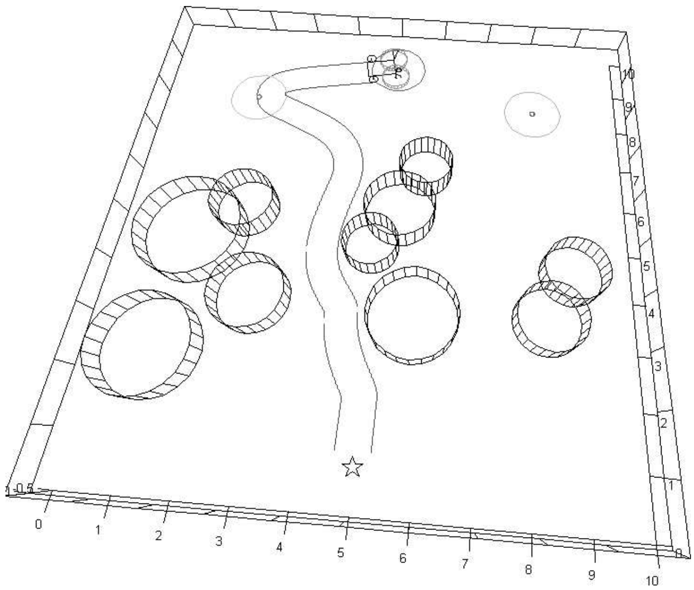

Fig. 6. Simulation with all the behavior active (prospect view).

When the energy level is under a certain value the *Recharge Platforms Reaching* behavior takes the control of the robot. Now the robot is more attracted by the energy stations than by the sound source. In the graphs of figure 7 we can see the progress of the energy level and the distance travelled by the robot.

After 137 seconds the energy reaches the bottom threshold and the robot changes the direction of movement. At the 158th second the recharge platform is attained and the robot stays for 8 seconds in recharging; after it moves around the platform for 7 seconds. This action is quite strange, it seems that the controller enters in a condition of instability. The phenomenon was interpreted considering that the station can supply a finite level of energy. When the energy is terminated, the robot is not anymore attracted by it and can go to the final target.

VI. Hardware Implementation of the neural Controller

In this section we illustrate a specific *hardware implementation* of the robot model and its benefits with respect to

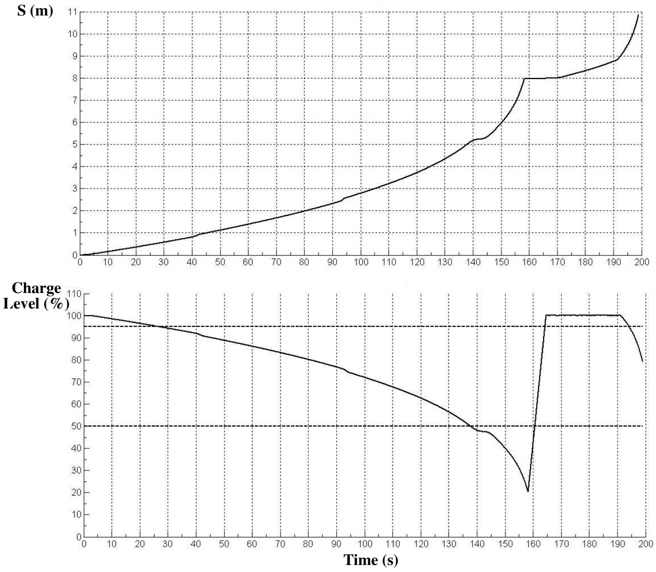

Fig. 7. The distance covered by the Robot and its energy level.

other possible implementations. There are different methodologies and strategies to develop the architecture presented in the previous paragraphs, for instance by using FPGA (Field Programmable Gate Array) based boards, reconfigurable devices, etc., but all these solutions have been shown in the robotic context to have disadvantages that make the hardware implementations in some cases to be worse than the software simulation. If we consider the FPGA-based boards, for instance, we often have to face with severe area and speed constraints, and the expressivity of hardware description languages like VHDL is, in some cases, too much limited for custom robotic applications, while we need more flexibility and better performances. Furthermore, because we are considering a mobile robot, we have to deal with energetic issues, that should be taken into account in order to minimize the number of recharges needed by the robot and maximize the space covered; that is, in most cases, a trade-off between board performances and board power consumption. By observing the architecture of the neural controller and the kind of computations involved (floating point multiplications, mantissa shifting, sums and threshold comparisons due to the logical activation functions) we focused on the possibility to develop a very fast hardware implementation of the system by exploiting a novel solution. We chosen a dedicated dual processor board equipped with an ARM7TDMI micro-controller for general purpose computation, and a floating-point VLIW digital signal processor core for hard computations like FFT (Fast Fourier Transform) and frequency domain phase-shift algorithms. The Diopsis D740 board by Atmel satisfies our needs, delivering 1 billion floating-point operations per second (1 GFLOPS). The board is equipped with two serial ports, two USARTS, an USB connection, a timer counter, watchdog, parallel I/O port (PIO), peripheral data controller, 8 ADC and 8 DAC (high quality, 24 bit precision) interfaces, clock generator and interrupt controller. Only some features of this board have been used to build the neural circuit (in a mobile setting, we would have the lightest robot). After the control architecture is encoded into a program, the system is able to operate in a completely standalone mode, this in order to give the robot a full autonomy.

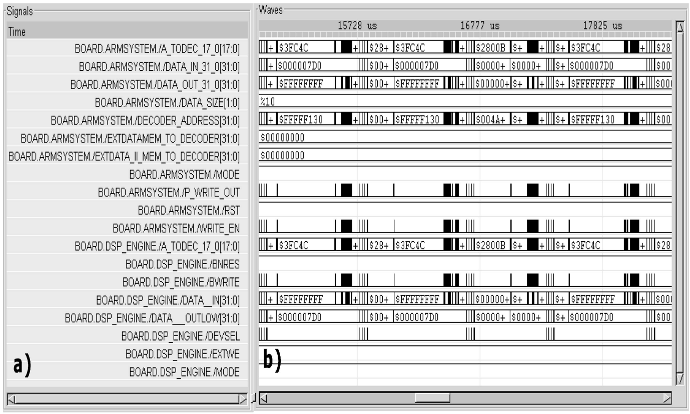

Fig. 8. a) Some of the signals exploited in the first hardware implementation b) Wave Forms of the signals.

We partitioned the tasks on the two board's processors. The job performed by the *ARM processor*, that constitutes the **vegetative system** of the cricket, is very simple: it defines the shared (between the two processors through a common interchange bus) memory space, parameters and constants and configures interrupts, timers and everything's needed to take advantage of the ADDA's (ADCs+DACs) interfaces and of the 7-segment display. It is intended to pass the stimuli receipt from the environment (for instance by the sensors located in the moustaches and in the ears) and then acquire the stimuli elaborated by the neural network in order to actuate the movement for autonomous navigation. The **neural controller** has been implemented through the *DSP Processor*: this means that all the heavy floating point computations required by the neural network and needed to obtain the new actuation signals are performed by this parallel-dedicated processor. More specifically, the DSP is really engaged managing all the audio samples and converting them in a numerical form, the sample time it operates is $t = 1/44100s$. In addition to the network computations (computational time is in average $9.2ms$), the whole audio elaboration (that represent the sensorial stimuli from the surrounding environment) takes place inside this processor.

The DSP effectively builds the neural network and its relative interconnections, this is replicating the effective job of each basic block with smart loops. The software running on the DSP when invoked by the ARM processor is much more complicated than ARM's: it represents the real kernel of the application. The exploitation of this kind of processor permits us to gain true real time rendering. As we can see in figure 8, thanks to a Win32 tool it is possible to supervise the signals of the two processors, this is critical especially for synchronization procedures between different tasks. We can observe here an interesting similarity with the biological activation and spikes signals in the biological neurons. This is however not easy to see at first sight, also because most of the signals are opportunely encoded to permit to the processors to exchange all the data with a single transfer per time, reducing the time required for data exchange. To implement the *audio management* part of the bio-cricket, we used the analogical to digital and viceversa interfaces (ADDA). It's possible to use only a single-input and single-output (exploiting a stereo-channel solution) or alternatively a more complex (but closer to the biologic configuration of a real cricket) solution using separated channels for the two ears (using two mono-aural microphones instead of one stereo), two outputs as actuation channels for the engines (tension control) and other two additional inputs as proximity sensors (the moustaches). About the **performances**, there are techniques that permit to evaluate at a glance where the application spends the major number of cycles (e.g. *profiling information*), and so it is possible to decide which routines must be optimized in order to speedup the overall system. Furthermore, it's possible to change the assembler-linker code produced by the DSP code compiler and implement optimizations by attempting to parallelize as much of the overall architecture as possible. The great part of optimizations can be done exploiting the 2-way parallel architecture of DSP processor, because in our application all operations are 2- way parallel due to the intrinsic characteristic of the bio-cricket brain, sensor system and actuation system (left and right for each sensor, neuron, actuation). Making a comparison with the software implementation, we obtained a speed-up of 100x with respect to the Matlab software. By inserting the assembler-like optimizations and simulating the same number of neural cycles (1000), the results have shown a second speed-up of 2.5x. This fact shows once again how dedicated hardware and optimized software can perform better than traditional software-only architectures. During this development of neural networks-based applications, the DSP dedicated boards performed very well, also for robotic applications. The application takes advantages from the features that give the Atmel board a higher place than other traditional architectures. Its single-chip high-performance ARM RISC and dedicated S.O.C. 1 Gflops DSP VLIW processor with dual ported shared memory architecture significantly decrease the computation time. The 24 bit ADDA interface brings highest precision in audio elaboration and with 8 total in/out connectors offers many possible sensorial choices for robotic applications.

VII. Conclusion and Future Work

In this paper we presented a bio-inspired neural controller for a mobile robot. The network architecture is organized in two neurons layers: the sensory layer receives the output signals coming from the robot sensors and feeds with excitatory and/or inhibitory connections the motoneurons, the motor layer combine its input signals to govern the robot wheels. Inside the architecture it is present also a sub-network, that using information about the energy level, regulates the robot behaviors. The regulation is based on an inhibition mechanism that acts directly on the synapses of the sensory-motor layer.

From the first results obtained in a simulated environment we have shown that the controller is able to govern the robot in its primary task, that is following a sound source. We changed the values for the inhibitory synapses that connect the Ear sensory neuron to the motoneuron and evaluated the robot performance.

Compared with other neural controllers [8] [10], we introduced a more complex architecture able to perform different kind of behaviors concurrently. This is possible thanks to an inhibition mechanism that modulates the synaptic strength of different sensory-neurons. Related to the subsumption architecture [13], we developed a control system that is more biomimetic, in the sense that the control layers here are represented by different dynamical neural networks that resemble parts of the neural circuits of the insects.

Experiments suggest us to consider and develop a mechanism to adjust the synapses in order to improve the robot performances. The synapse optimization may be done for example on the time needed by the robot to reach the target and on the level of energy consumed to perform this task. Or it is possible to think to use a learning paradigm [14], [15].

After the hardware implementation of the controller using a DSP processor, we can conclude that the time needed to actuate the robot using the neural architecture is absolutely acceptable.

A more realistic scenario to test the robot may be developed to contribute to the wide area of service robots.

References

[1] C. Ghez. *Principles of Neural Science*. Appleton and Lange, Norwalk, Conneticut, third edition, 1991.
[2] Cesare Casella and Vanni Taglietti. *Principi di Fisiologia*. La Guliardica, 1996.
[3] R.D. Beer, H.J. Chiel, R.D. Quinn, and R.E. Ritzmann. Biorobotic approaches to the study of motor systems. *Current Opinion in Neurobiology*, 8(6):777–782, 1998.
[4] Oswald Steward. *Functional Neuroscience*. Springer, 2000.
[5] D. Floreano, J.C. Zufferey, and J.D. Nicoud. From wheels to wings with evolutionary spiking neurons. *Artificial Life*, 2004.
[6] S. Nolfi and D. Floreano. Neural synthesis of artificial organisms through evolution. Trends in Cognitive Science, 6, 31-37, 2002.
[7] B. Mathayomchan and R.D Beer. Center-crossing recurrent neural networks for the evolution of rhythmic behavior. *Neural Computation*, 14:2043–2051, 2002.
[8] Barbara Webb and T. Scutt. A simple latency dependent spiking neuron model of cricket phonotaxis. *Biological Cybernetics*, 82(3):247–269, 2000.
[9] Valentino Braitemberg. *Vehicles: Experiments in Synthetic Psychology*. MIT Press, 1984.
[10] Reeve R. and Barbara Webb. New neural circuits for robot phonotaxis. *Philosophical Transactions of the Royal Society*, 361:2245–2266, 2002.
[11] Michele Folgheraiter and Giuseppina Gini. Human-like reflex control for an artificial hand. *BioSystem Journal, Elsevier Science*, 76(1-3):65–74, 2004.
[12] M. Scholles, B.J. Hosticka, M. Kesper, P. Richer, and M. Schwarz. Biologically-inspired artificial neurons modeling and applications. *International Joint Conference on Neural Networks*, 1993.
[13] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence Journal*, (47):139–159, 1991.
[14] S. Schaal. *Learning from demonstration*. MIT Press, 1997.
[15] D.O. Hebb. The organization of behavior: A neuropsychological theory. *New York:Wiley*, 1949.