Speech and Action Integration in Humanoid Robots: Simulation Experiments with iCub

V. Tikhanoff, A. Cangelosi, G. Metta

***Abstract*— Building intelligent systems with human level of competence is the ultimate grand challenge for science and technology in general, and especially for the computational intelligence community. Recent theories in autonomous cognitive systems have focused on the close integration (grounding) of communication with perception, categorization and action. Cognitive systems are essential for integrated multi-platform systems that are capable of sensing and communicating. This paper presents a cognitive system for a humanoid robot that integrates abilities such as object detection and recognition, which are merged with natural language understanding and refined motor control. The work includes robotic simulation experiments showing that a humanoid robot (iCub platform) is able to acquire behavioral, cognitive, and linguistic skills through individual and social learning. The robot is able to learn to handle and manipulate objects autonomously, to understand basic instructions, and to adapt its abilities to changes in internal and environmental conditions.**

***Index Terms*—Artificial intelligence, Manipulation, Cognitive Robotics, Speech recognition, iCub.**

I. INTRODUCTION

COGNITIVE systems research , including developmental cognitive robotics, focuses on the development of artificial information processing systems that are capable of perception, learning, decision-making, communication and action. The main objective of cognitive systems is to transform human-machine systems by enabling machines to engage human users in a human like cognitive interaction [58]. A cognitive system, such as a robot or a simulated agent, is based and designed upon human cognitive processes that enable the system to interact or engage human users in a human like cognitive process. A cognitive system is based on computational representations and processes of human behavior that aim to replicate the cognitive abilities of humans [8], [9], [11], [12], [14], [18], [20], [22]. Using evidence from domains such as neuroscience and cognitive science, it is possible to build artificial intelligence systems that are capable of human cognitive abilities.

Developmental cognitive robotics is an emergent area of cognitive systems, which is at the intersection of robotics and developmental sciences in psychology, biology, neuroscience and artificial intelligence [2], [35], [36], [41], [42]. Developmental robotics is based on methodologies such as embodied cognition, evolutionary robotics and machine learning. New methodologies for the continued development of cognitive robotics are constantly being sought by researchers, who wish to promote the use of robots as a cognitive tool [4], [7], [35], [50], [65], [66], [68]. Amongst diverse solutions to the programming of robots such as attention sharing, turn-taking behavior and social regulation [20], [23], a major part of the research focus in developmental cognitive robotics is imitation. A considerable amount of research has been conducted in order to achieve imitating/intentional agents [3], [29], [39], [48], [57]. More recently, researchers have used developmental robotics model in order to study other cognitive functions such as language and communication.

This paper proposes a new approach to the design of a robust system that is able to take advantage of all the functionalities that a humanoid robot such as the iCub robotic platform [45], [56] provides. This will focus on object manipulation with refined motor control integrated with language "understanding" capabilities. The paper describes cognitive experiments carried out on the iCub simulator [62], [63]. These experiments are divided into three main sections: (1) vision, (2) motor learning, and (3) natural language understanding. The scope of this paper is limited to the latter two due to spatial constraints. Section II concentrates on the motor control system, which consist of a reaching and grasping module. Section III which provides a detailed description of the speech module, and finally describes a complete experiment on cognitive behavior.

II. REFINED MOTOR CONTROL

A. Introduction

This section proposes a method for teaching a robot how to reach for an object that is placed in front of it and then, attempting to grasp the object. The first part of the work focuses on solving the task of reaching for an object in the robot's peripersonal environment. This work employs a control system configuration, consisting of a neural network that is configured as a feed-forward controller. The second part

Manuscript received April 20, 2009. This work was supported in part by grants from the EuCognition NA097-4 and FP7 project ITALK ICT-214668.

V. Tikhanoff was with the Adaptive Behaviour & Cognition research lab at the University of Plymouth, Plymouth PL4 8AA, UK. He is now with the Italian Institute of Technology IT, (e-mail: vadim.tikhanoff@plymouth.ac.uk , vtikha@gmail.com).

A. Cangelosi, is with the Adaptive Behaviour & Cognition Research lab at the University of Plymouth, Plymouth PL4 8AA, UK (e-mail: acangelosi@plymouth.ac.uk).

G. Metta is with the Robotics, brain and cognitive sciences department at the Italian Institute of Technology, 16163 Genoa (e-mail: giorgio.metta@iit.it).

incorporates the above reaching module with an extra controller that is needed for the robot to actually grasp the object. This employs another control system configuration, which consists of a neural network that is configured as a Jordan Neural Network [30].

B. Reaching

In recent years, humanoid research has focused on the potential for efficient interaction with the environment through motor controls and manipulation. Reaching is one of the most important assignments for a humanoid robot, as it provides the robot with the ability to interact with the surrounding environment, and permits the robot to discover and learn through the task of manipulation. However, this task is not a simple problem. Significant progress has been made to solve these problems and this section will briefly explain some of the past applications that have been used towards the reaching problem.

Reaching in neuroscience has focused on the development of human models of reaching, in order for a humanoid robot to achieve human-like reaching [13], [17]. Additionally, neuroscience considers the issue of pre-grasping as defined by Arbib and colleagues [1]. This deals with the configuration of the fingers for successful grasping, whilst performing the reaching movement. These finger configurations must, therefore, satisfy some types of pre-defined knowledge covering the object in order to grasp it, and also some type of pre-defined knowledge about the task to accomplish. This document is not concerned with generating a reaching system, which is similar to human models of reaching by using pre-grasping, but assumes that reaching and grasping can be performed independently. Recent works that are directly applicable to humanoid reaching have considered this manipulation planning problem [38], [49].

Current research on humanoid robot manipulation [34] has considered the reaching problem without dealing in depth with the grasping issue. Issues such as grasping, friction and the mechanics behind it are not taken into consideration, and use reaching for pointing and touching. Other recent work [10] has implemented reaching by using a path planner with some obstacle avoidance procedure. Kagami and colleagues [32] use an interesting approach by taking into account the humanoid stereo vision, in order to construct a virtual model of an environment. This includes the use of inverse kinematics to perform a reaching and grasping task. Apart from [32], these few works have simplified the problem of reaching to a greater extend, for example by not involving vision and other sensory inputs from the humanoid robots.

This work considers reaching as a hand-eye coordination task, which greatly depends on vision for tracking of objects, whether static or moving, and their depth estimation. The control system that has been designed for reaching does not depend on heavy camera calibration and an extensive analysis of the robot's kinematics. The reaching system uses the uncalibrated stereo vision system, to determine the depths of the objects. A suitable system for a humanoid robot must take into consideration the movement of the robot's head and eyes [24]. Metta and colleagues [44] have developed a humanoid robot controller based on single motor mapping. They developed the mapping from the two eyes of the robot to control two joints in the arms. They then added the eye vergence in order to determine the depth of an object [43]. Even with the addition of the eye vergence, there were some imperfections due to errors in the hand positioning. In an earlier paper, Marjanovic and colleagues [37] described a system that was able to correct mapping errors by redirecting the robot's eyes to focus on its hand, after looking at the object. This permitted, to some extent, an improvement in the results by using simple motor mapping. There have also been several systems that have used learning with endpoint closed loop controls [16], [52], [64]. These systems use fixed cameras and can perform several types of error corrections, which permit adjustment of the learned mappings and the end position of the hand. Although the different systems were reliable, they failed when the hand was not visible by the vision system. However, it is not possible to assume that the hand will be constantly visible during object manipulation in a humanoid robot. More recently, Gaskett and colleagues [24] have successful implemented a system that used stereo vision to view a target. This involved moving the hand towards the end position, whilst also assisting the eyes, so that the object could be tracked by moving the head and torso of the humanoid robot. When the vision lost track of the arm, they used a three dimensional self-organizing map (SOM) [33] in order to map the three dimensional movements of the robotic arm. By knowing the state of the eye, head and arm joints, they used the learned SOM to make the robot find the hand and look at it. Although the design of the system is reliable, the controllers cannot be refined online and are based on non-learning networks of proportional derivatives. The system used for the reaching module, implemented in this work, uses the knowledge of previous findings and adapts them to use an improved mapping.

The reaching module is based on learning motor-motor relationships between the vision system of the head/eyes and the iCub's arm joints. This is represented by a feed-forward neural network trained with a back propagation algorithm. The only initial condition is that the hand is positioned in the visual space of the robot to initiate the tracking of the visual system. This will then calculate the three-dimensional coordinates of the hand itself, and consequently move the head accordingly. A feed-forward multilayer perceptron, with back propagation algorithm [54] was modeled to simulate reaching for diverse objects that reside within their surroundings. The following architecture has been used to model the feed-forward neural network:

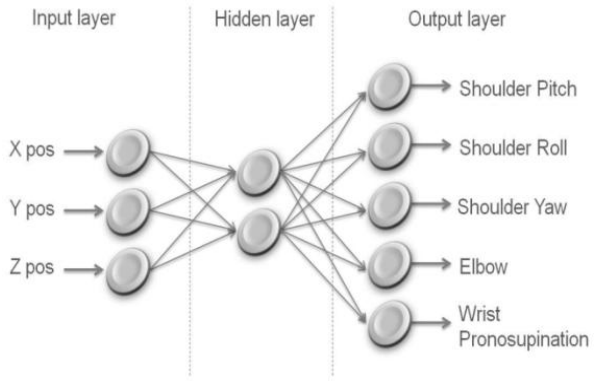

Fig. 1. The architecture of the employed feed-forward neural network.

The input to the feed-forward neural network is a vector of three dimensional coordinates (X, Y and Z) of the robot's hand, normalized from 0 to 1. These coordinates were determined by the vision system, by means of the template matching method, and depth estimation. The output of the network is a vector of angular positions of 5 joints that are located on the arm of the robot. The joints used for the reaching module are described in Table I.

Joint	**Description**
Shoulder Pitch	Front and back movement
Shoulder Roll	Adduction-abduction movement
Shoulder Yaw	Yaw movement when the arm axis is aligned with gravity
Elbow	Elbow movement
Wrist Pronosupination	Forearm rotation along the arm axis

Table I. Description of the different joints used for the reaching module.

The hidden layer comprises of 10 units. This is the optimal number of hidden units identified after preliminary experiments. During the training phase, the robot generates 5,000 random sequences, whilst performing motor babbling within each joint's spatial configuration/limits. When the sequence is finished, the robot determines the coordinates of its hand and what joint configuration was used to reach this position. Figure 2 shows 150 positions of the endpoints of the robot hands, by representing them as green squares.

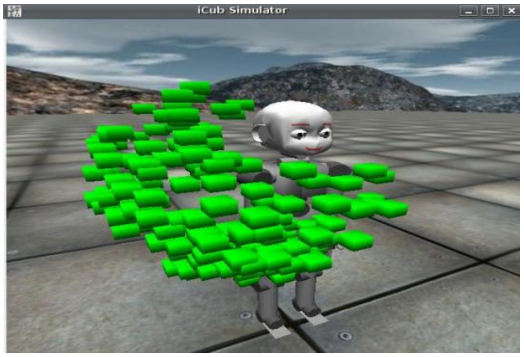

Fig. 2. Example of the 150 end positions of the robot arms during training.

This feed-forward neural network was trained with the parameters listed in the following table:

Learn Size	Test Size	Total	NumIterations	Learn Rate	RMSE
2,500	2,500	5,000	50,000	0.05	0.156

Table II. Training parameters of the reaching feed-forward network module.

After multiple tests of 50,000 iterations, the RMSE (root mean squared error) ranged from 0.15 to 0.16, which indicates that the neural network was not able to fully learn the task (see figure 3).

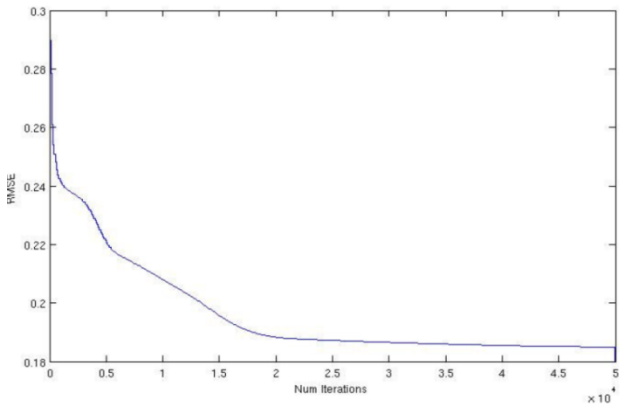

Fig. 3. RMSE value during training of the reaching module

By analyzing the results, we can see that the network has been successful in learning to reach the specific position, with its joint configuration. But it has discarded the last joint completely, as shown in figure 4. Figure 4 displays the first 150 results of the 2,500 testing samples provided to the network. Each graph represents the different normalized (from 0 to 1) joint degrees (Y axis) at each of the 150 positions (X axis). Starting from the top:

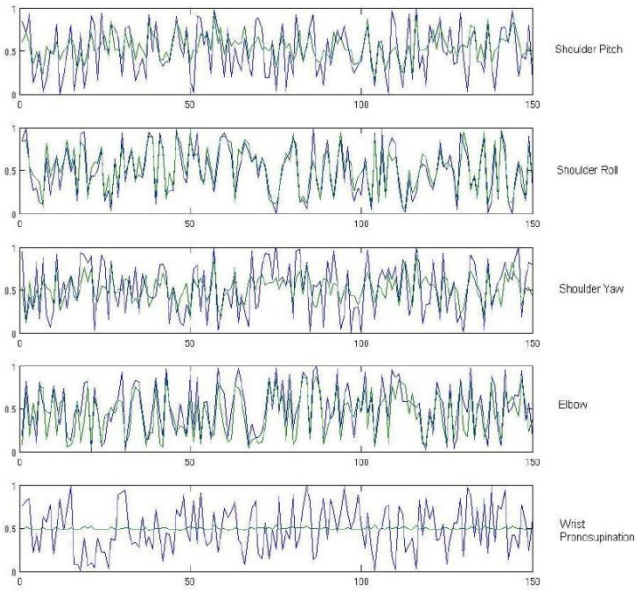

Fig. 4. The first 150 results of the 2,500 samples given to the network. Each graph represents the different joint degrees at each of the 150 positions. Starting from the top: shoulder pitch – shoulder roll – shoulder yaw – elbow – wrist pronosupination

The reason for such a high root mean square error is believed to be due to several factors. The first one is the fact that the wrist pronosupination (forearm rotation along the arm principal axis) is not needed for the robot to reach a specific position and therefore is eventually discarded by the network when learning the training data. The desired mappings of the joints of the iCub simulator have been satisfied as much as possible without the use of this joint. The second factor is due to the fact that the hand would never reach the center of gravity of the object itself (detected from the vision module)

as collisions from the hand and the object would not allow it to reach this point. In order to test the performance of the model, a pre-trained reaching neural network was loaded onto the simulation, whilst random objects were placed in the vicinity of the iCub robot. The results of these tests showed that the model was capable of successfully locating and tracking the object, and finally reaching the target (see figure 5). Figure 5 is a collection of images taken after the detection of the object (by the vision system) and the attempt to reach the tracked object.

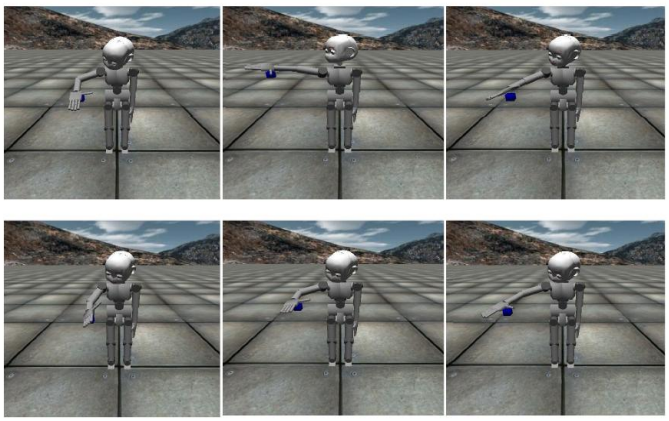

Fig. 5. Images taken from the robot during the testing of the reaching module

Figure 6 supports the previous argument, by showing the X, Y and Z coordinates of 62 random objects that were placed within the vicinity of the iCub, and then compares them with the actual resulting position of the robot's hand.

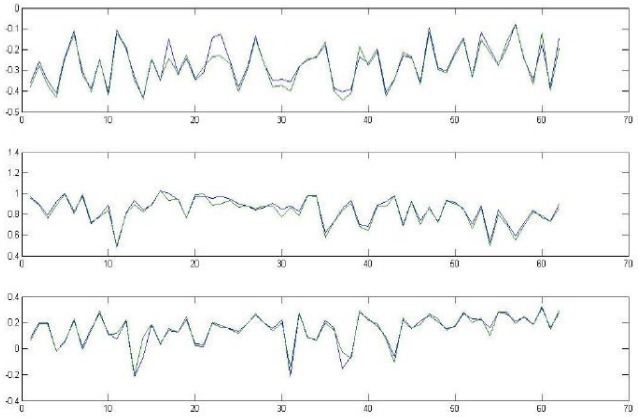

Fig. 6. Comparison of 62 random XYZ positions of objects, with the actual resulting position of the robot's hand

Overall, the experimental setup and results deal with a system that is able to perform reaching, using stereo cameras from the iCub simulator. The only input units are the three dimensional coordinates. Between the vision module and the reaching module, eleven degrees of freedom were used: six for the head and eyes, and five for the arm joints. The reaching module was able to learn an approximation of the randomly placed object in its vicinity, whilst autonomously discarding unnecessary joint motion to achieve its goal.

The next step will be to attempt to grasp the object that the robot has successfully reached. In the next section, after a brief discussion on recent work on grasping, we will describe our approach that was used to solve the well known grasping problem.

C. Grasping

One of the major challenges in robotics is to reproduce human dexterity in unknown situations or environments. Most of the humanoid robotic platforms have artificial hands with varying complexity. Attempting to define their configuration, when seeking to grasp an object in its environment, is one of the most difficult tasks. Many parameters must be accounted for, such as the structure of the hand itself, the parameters of the object, and the specification of the assignment. Taking these parameters into consideration, the ability to receive sensing information from the robot is crucial when implementing an efficient robotic grasp. The quality of the sensing information must also be taken into consideration, as signals may limit precision and can potentially be noisy. In recent years, there have been several models implemented to perform a grasping behavior. The different models can be divided into categories such as:

• Knowledge based grasping,
• Geometric contact grasping,
• Sensory driven and learning based grasping.
(adapted from [51])

Knowledge based grasping takes into account techniques where the hand parameters are adjusted according to the knowledge and experience behind human grasping, therefore taking advantage of the human dexterity capabilities. This approach is based on diverse studies on human grasping. These have been classified depending on parameters, such as the hand shape, the world and the tasks requirements, and have been used to suggest solutions in the robotic field [6], [28], [55].
Although these methods are effective and produce good results, there is the requirement to have sophisticated equipment, such as data gloves, to utilize motion sensors. Furthermore, there is a significant drawback: the ability of the robot to generalize grasping in different conditions, as the robot can only learn what has been demonstrated. Additionally, knowledge based grasping uses the issue of pre-grasping, which requires anticipation of the grasp before reaching the object, and depends on the task and the object. On the other hand, geometric contact grasping is used in conjunction with algorithms to find an optimal set of contact points, according to the requirements, such as feedback from forces and torques [25], [46]. This is an optimal approach, as it can be applied to a large amount of dexterous robotic hands whilst finding a suitable hand configuration. The main issue with the geometric contact grasping is that there must be a predefined scenario to be performed and therefore, generalization cannot be performed. Finally, the sensory driven grasping approach tries to solve the previously mentioned problems by using learning and task exploration [27], [67].

The approach proposed here relies on artificial neural networks in order for the humanoid robot to learn the principles of grasping. Sensory driven models have been previously utilized to perform grasping with a robotic hand, using a limited amount of degrees of freedom for circular and rectangular shaped objects [47], [60]. More recently, Carenzi and colleagues [15] developed neural network models which are able to lean the inverse kinematics of the robotic arm, to reach an object, depending on information such as size, location and orientation. The model is then able to learn the appropriate grasping configurations (using a multi-joint hand) dependent on the object size. Although this work is interesting, it is highly simplified and both wrist position and orientation need to be pre-defined.

In our approach, a new method that is based on the sensory driven grasping approach is proposed. This is achieved by modeling an additional artificial neural network that is able to learn how to grasp the different objects in its environment, by feeding it with the sensory information of the hand itself. There are many ways in which this can be accomplished, and a number of interesting proposals have appeared in the literature. One of the most promising was suggested by Jordan [30]. Jordan proposed a neural network with recurrent connections copying the output unit values and feeding them back to the hidden layer. A Jordan type neural network was implemented in this research to simulate the grasping of diverse objects that reside in the robot's environment. The neural network's architecture can be seen in figure 7.

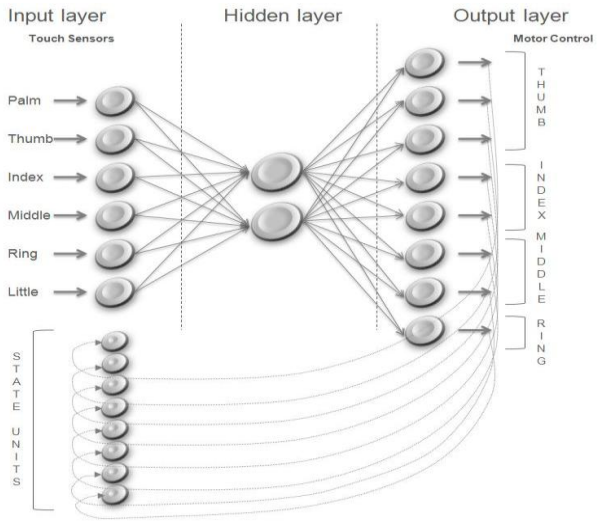

Fig. 7. The architecture of the employed Jordan Neural Network

The input layer of the Jordan neural network is a vector of the touch sensory information of the robot's hand (either 0 or 1), and the output is a vector of normalized (0 to 1) angular positions of the 8 finger joints, which are located on the hand of the robot. The hidden layer comprises of 5 units placed in parallel. This is the optimal number of hidden units that have been identified after preliminary experiments. The output activation values (normalized joint angular positions) are fed back to the input layer, to a set of extra neurons called the state units (memory). An image, showing the location of the hand sensor, can be seen in figure 8 and a detailed description of the hand joints used can be seen in table III.

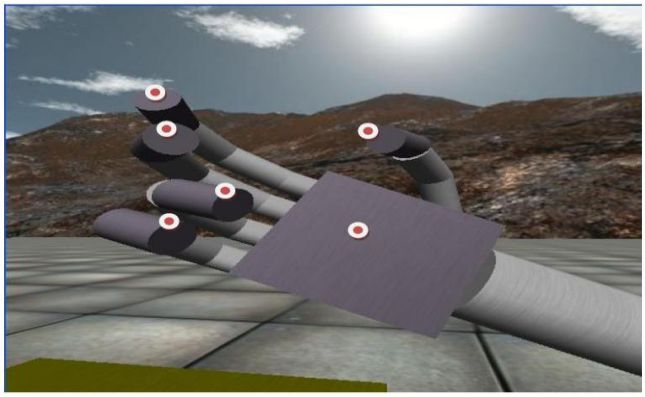

Fig. 8. Location of the six touch sensors on the iCub's simulator hand

The touch sensors work in an "off and on mode". Meaning that the touch sensor is always off (0), unless there is a collision with a foreign body that would trigger the activation of the sensor (1).

Joint	**Description**
Thumb opposition	Thumb lateral movement
Thumb proximal flexion/extension	Thumb front-back movement
Thumb distal flexion	Thumb closing
Index proximal flexion/extension	Index front-back movement
Index distal flexion	Index closing
Middle proximal flexion/extension	Middle front-back movement
Middle distal flexion	Middle closing
Ring and little finger flexion	Ring and little front-back movement and closing

Table III. List of finger joints used in the grasping module

The training of the grasping neural network is achieved online and therefore no training patterns have been pre-defined to learn grasping; hence no data acquisition is required. A reward mechanism has been implemented in the network to adjust the finger positions. The Associative Reward Penalty algorithm (ARP) is implemented in order to train the network connection weights. A description of this algorithm can be found in [5]. The error is determined for each of the output units and their weights are updated in the back propagation algorithm. This method is used for associative reinforcement learning, as the back-propagation algorithm is not able to perform such a task. The neural network would, therefore, need to adapt to maximize the reward rate over time.

During training, a static object is placed under the hand of the iCub simulator and the network at first randomly initiates joint activations. When the finger motions have been achieved, or stopped by a sensor activation trigger, the grasping is tested by allowing gravity to affect the behavior of the object. The longer the object stays in the hand (max 250 time steps) the

higher the reward becomes. If the object falls off the hand, then the grasping attempt was not achieved and therefore a negative reward is given to the network.

A number of experiments were carried out in order to test the model's ability to learn to grasp an object that was shown, and also to ultimately learn how to differentiate between objects by grasping them in different ways (object affordance and finding a solution in order to accomplish its task).

The charts in figures 9 and 10 show the results of a simple experiment, where the iCub robot's goal was to attempt to successfully grasp an object (cube) that was placed under its hand, as seen in figure 11. The object size parameters (in meters) are:

• width = 0.05, height = 0.03, depth = 0.04.

The object was then modified to a cube such that:

• width = 0.04, height = 0.04, depth = 0.04.

The object was placed at different coordinates in order to further test the system under simple conditions. Figure 9 displays the reward rate of the grasping neural network during a training phase of 15 attempts; figure 10 shows the number of total boxes used, grabbed, and the total number lost during a simple grasping experiment, with the object of size x= 0.04, y = 0.04, and z = 0.04.

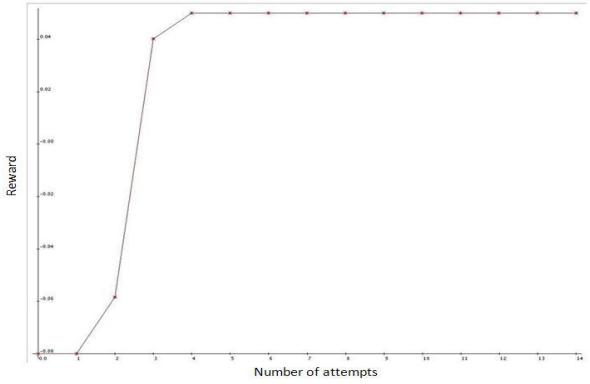

Fig. 9. The reward rate during the grasping neural network training phase

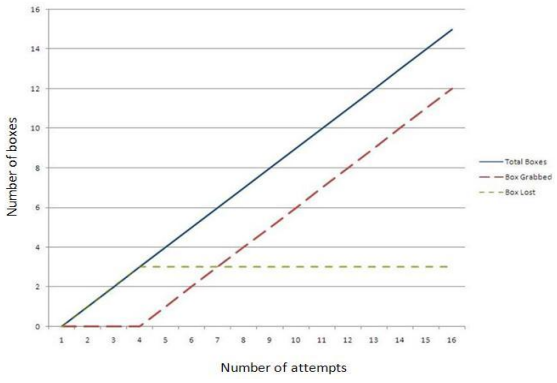

Fig. 10. Graph showing the total boxes used (Red), total boxes grabbed (Yellow), and total boxes lost (Green), during a simple grasping experiment with an object of specific size.

A further experiment was conducted which aimed to test the potential of the grasping module by placing different static sized and shaped objects in the vicinity of the iCub simulator. A pre-trained grasping neural network was then loaded onto the simulation to demonstrate that the system is able to generalize grasping with different objects.

Figure 11 shows an example of the learned grasping module that was performed on three different objects: a small cube, a ball, and a complex object (teddy bear).

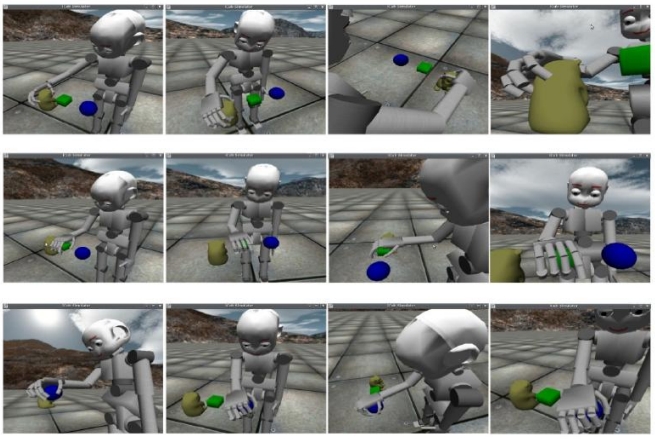

Fig. 11. Grasping of three different objects

III. WORKING WITH SPEECH

A. Introduction

Speech shapes a large part of human-human and even human-machine interaction. There are many cognitive robotic models which have focused on speech learning [53], [59]. The goal of this section is to produce a real-time system of speech understanding.

B. Feature Extraction

The human ear can detect and analyze vibration frequencies that originate from a sound and then distribute the sound to different nerve cells in the auditory portion of the central nervous system. In order to replicate this process on a humanoid robotic platform, a speech analysis software module has been built which receives raw input from a microphone, whilst also performing filtering and categorizing speech sounds. The filtering that is performed on the raw speech data is a Fast Fourier Transform (FFT). The FFT calculates a Fourier Transform of a digital signal using the divide and conquer method [21]. It takes the 1,024 samples per call and calculates the frequency spectrum for the intensities of various sine waves that are the components of that particular sound. In our model, the sampling was carried out at a rate of 8000Hz, more specifically 8,000 audio samples per second, thus requesting 1,024 samples per call; the received packets are, therefore, about 8 packets per second, equal to 8Hz. The mean frequency of calls is 8,000/1,024 = 7.8125 Hz.

In order for the system to be able to learn from the sound analysis that is produced by the speech processing module, we constructed a self organizing map (SOM) [33], and trained it using unsupervised learning. The SOM is a single layered

feed-forward neural network, where the output units are arranged in a topological two dimensional grid. The purpose of learning in the SOM is to associate various parts of the SOM lattice to respond to different input patterns. This is partially inspired by the auditory/vision and other sensor information, and how they are handled in distinct parts of the cerebral cortex within the human mind [26]. Another reason for the use of such a system is that: the learning process is competitive and unsupervised, meaning that no teacher is needed to define the correct output, or to specify which cell of the output should be mapped for any input. Only one map node (winner) at a time is activated, corresponding to each input. The locations of the responses in the array tend to become ordered in the learning process when a meaningful nonlinear coordinate system is created over the network for the different input stimuli [33]. The SOM consists of a 10x10 topological two-dimensional grid, represented by a set of weight vectors in the output space, and a continuous input space of 1,024 units for the speech input corresponding to the FFT. Each input stimuli (spoken word) is divided in a sequence of 20 temporal patterns. The self organizing map, in real time, analyses all of the data provided by the filtering of the inputs via the microphone. In the first stage, called the initialization phase, all the self-organizing map's weights are assigned to a small randomly generated value, ranging from 0 to 1. After initialization, the network goes through three learning cycles so that it can form the self-organizing map: (1) the competition, (2) the cooperation, and (3) the synaptic adaptation. In the competition stage, the neurons use a discriminant function for each input pattern, which provides the basis for the competition amongst the neurons. Therefore, the neuron with the largest value is declared the winner of the competition. The second cycle is the cooperation step. Here, the winning neuron of the previous part will determine the spatial location of the topological neighborhood, of excited neurons, to enable cooperation among such neighboring neurons. The third and final cycle is the synaptic adaptation. This consists of enabling the exited neurons to increase the individual values of the discriminant function, in relation to the input pattern, through adjustments made to the synaptic weights.

The modeled self organizing map has been independently trained using data collected from various sources. The data comprises of 112 English words from two different speakers (words spoken in isolation) and 544 syllable utterances from two different speakers, for determining the ability of the system to distinguish between substantially small differences. The data was collected using different sources, such as an "off the shelf" microphone, sound files gathered from various users and samples of syllables and utterances (with and without noise).

The training of the self organizing map consisted of 50 cycles per independent word (word pattern). Each training cycle consisted of a variable number of activations between 3 and 10, and depended on the length of the sound. At the end of the cycles all the words were correctly discriminated. The average feature learning performance for the feature map was 92%, which was obtained by calculating the average performances of the most activated neurons, in respect to the initial input provided to the network. The learning performance will slightly increase if the number of training cycles is prolonged. We chose this training setup for its balance between duration and quality of results. Figure 12 shows the sum of the most activated neurons over the 50 cycles, for the words "ball" and "call". The results of the self organizing maps provide a well suited sound feature extraction, outputting a sequence of X and Y coordinates of the most activated neurons.

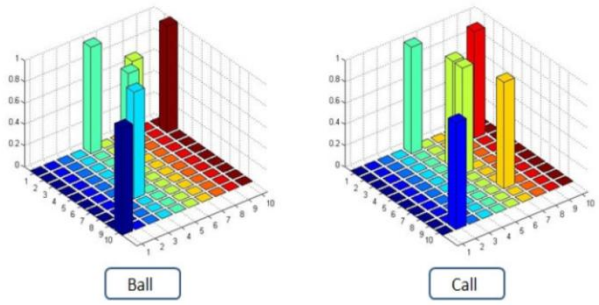

Fig. 12. Graph displaying the most activated neurons over 50 cycles, for the words "ball" and "call"

C. Word Classification

In order to extract some constructive information from the speech module, there is a need to classify and therefore, recognize such sequences. The classification and recognition was achieved by constructing a Recurrent Neural Network with Parametric Biases (RNNPB) [61].

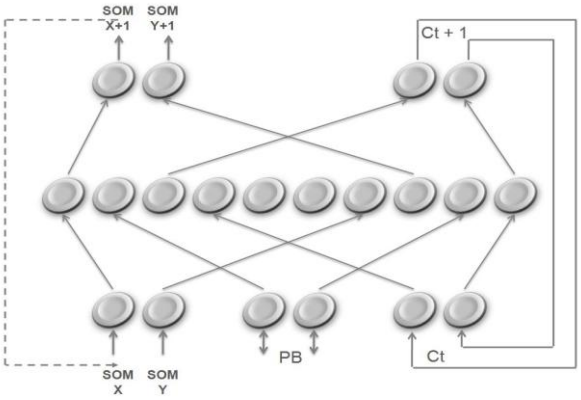

Fig. 13. The RNNPB architecture used

The main feature of the recurrent neural network with parametric biases is that the blocks of temporal patterns can be represented by a vector of small dimensions (parametric biases), which then acts as a bifurcation parameter of nonlinear dynamical systems [61]. Thus different vector values are given and the system will be able to produce different dynamic patterns. A further advantage of using such a neural network is that the RNNPB is able to encode an infinite number of dynamic patterns with the values of the parametric biases vector. Furthermore, the use of the parametric biases is ideal in their ability to both generate and recognize sequence patterns as a mirror system after learning is complete [61]. It is important to mention that the parametric biases vector for each pattern is self-determined in a non

supervised fashion. The prediction of the recurrent neural network with parametric biases is executed depending on the context. Therefore, the sequences at each step are anticipated simply by using the previous step and its past history. This neural architecture is therefore optimal for a classification and recognition module, as it can learn multiple patterns by extracting relational structures that are shared among them.

The vector containing such sequences of x/y coordinates is then passed on to the recurrent neural network with parametric biases input for the classification generation and recognition. The total amount of the training lasted for 1,000 epochs, with each epoch consisting of the presentation of each of the 28 individual speech signal patterns. The final square error of the output nodes was 0.059 over all of the learning results.

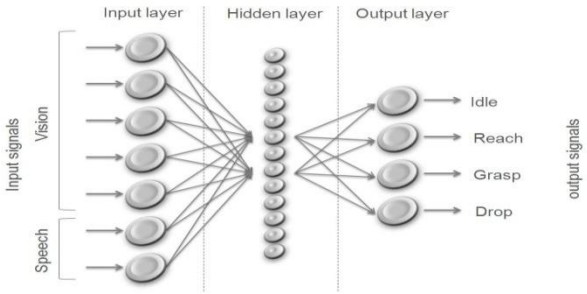

Fig. 14. The Goal Selection Neural Network architecture used

The initial two dimensional parametric bias vectors are all initialized to 0.5; PB1 = [0.5], PB2 = [0.5]. The application of the learning model to the above list of words reveals the formation of 28 distinct parametric biases. Figure 15 plots the parametric biases vectors in a three dimensional graph, where X is the first dimension of the PB vector, Y the second dimension, and Z is time.

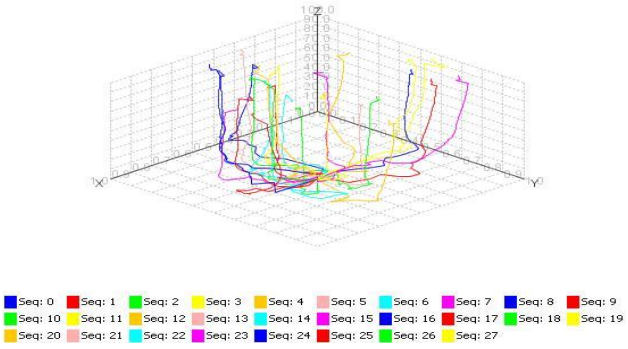

Fig. 15. The Evolution of the parametric biases for each of the sequences

D. Integrating Language and Action

The integration of the speech signals, visual input, and motor control abilities was based on a Goal Selection Neural Network, a feedforward neural network. The input to the network consists of seven parameters from the vision acquisition system (e.g. object size and location) and two dimensional parametric biases of the speech signals. The output consists of four units corresponding to the following action: idle, reach, grasp, and drop. The hidden layer comprises fifteen units. The neural network's architecture can be seen in figure 14. During the training phase, the robot is shown an object along with a speech signal. The list of objects and speech signals, used in this experiment, can be seen in table IV.

"Blue ball"	"Reach blue ball"	"Grasp blue ball"	"Drop blue ball into basket"
"Red ball"	"Reach red ball"	"Grasp red ball"	"Drop red ball into basket"
"Green ball"	"Reach green ball"	"Grasp green ball"	"Drop green ball into basket"
"Blue cube"	"Reach blue cube"	"Grasp blue cube"	"Drop blue cube into basket"
"Red cube"	"Reach red cube"	"Grasp red cube"	"Drop red cube into basket"
"Green cube"	"Reach green cube"	"Grasp green cube"	"Drop green cube into basket"
"Teddy bear"	"Reach teddy bear"	"Grasp teddy bear"	"Drop teddy bear into basket"

Table IV. List of speech signals used in the cognitive experiment.

The Goal Selection feed-forward neural network was trained with the above data, using the parameters in table V. After multiple tests of 50,000 iterations, the RMSE (root mean squared error) was ranging at 0.0368, which indicates a successful learning of the neural network.

Learn Size	Test Size	Total	Num Iterations	Learn Rate	RMSE
28	28	56	50,000	0.07	0.0368

Table V. Training parameters of the goal selection neural network module.

The testing phase, reported in this section, consisted of the presentation of a simple object (blue cube) to the iCub simulator. At first, the object presented was not selected as the system did not know what to do with it, since it was expecting an extra feature (the speech signal). Initially, the hand was positioned in the visual space of the robot, so that it would initiate tracking of the visual system, calculate the three dimensional coordinates of the hand itself, and consequently move the head accordingly. The most complex behavior sequence is then sounded out "drop blue cube into basket" and the robot would now focus its attention to the complex object by means of head tracking. The robot will then attempt to reach the object and grasp it in sequence. When the grasping is achieved, it will then look visually for the bucket. It will then move its arm towards the object by means of retrieving its X, Y, Z coordinate and then feeding it into the reaching module and attempting to release the object into the bucket. This sequence of actions can be seen in figure 16.

IV. CONCLUSION

This experiment described a system which focuses on the learning of action manipulation skills, in order to develop object-action knowledge, combined with action-object-name.

The results demonstrate that the cognitive model is capable of understanding continuous speech, to form visual categories that correspond to part of the speech signals, and thus develop action manipulation capabilities.

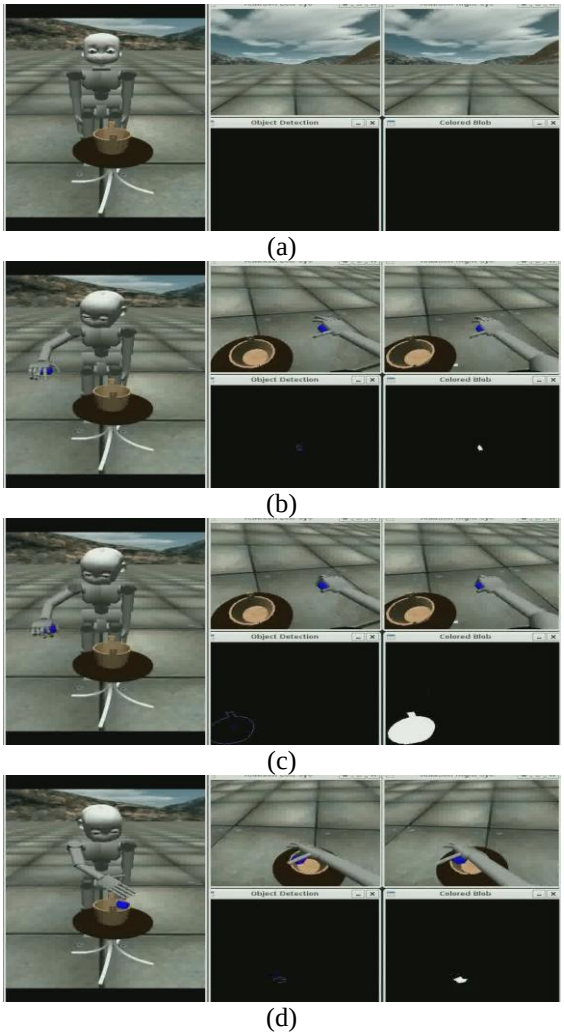

Fig. 16. Selection of images showing: (a) the setup of the cognitive experiment; (b) the input of the linguistic command; (c) the reaching and grasping of the blue box; (d) the dropping of the blue box

The system developed here was influenced by the way infants tend to learn speech from sounds [32], and then associate them with what is happening in their neighboring world. This work assumes that, for a robot to understand and categorize what is being said, its vocabulary initially needs to be limited and focused. Therefore, by providing a robot with such a system it will be able to quickly learn the vocabulary that is needed for the appropriate task. In addition to the visual perception and speech understanding system, the robot is able to receive tactile information and feedback from its own body. Neural network modules are used to permit the robot to learn and develop behaviors, so that it may acquire embodied representation of the objects and actions. Furthermore, a novel merging of active perception, understanding of language, and precise motor controls, has been described. This will enable the robot to learn how to reach and manipulate any object within the joint's spatial configuration, based on motor babbling, which again has been influenced by how infants tend to discover joint configurations [41]. New experiments used the complete embodied cognitive model that has been endowed with a connection between speech signals understood by the robot, its own cognitive representations of its visual perception, and sensorimotor interaction with its environment. The detailed analysis of the neural network controllers can be used to increasingly understand such behavior that occurs in humans, and then deduct new predictions about how vision, action and language interact between them.

This work provides some useful insights towards the building a reliable cognitive system for the iCub humanoid robot, so it can interact and understand its environment. Further research will aim to enhance and expand the cognitive skills of the humanoid robot. The model has only been based on the iCub simulator, but current work is focusing on the testing of the proposed cognitive control architecture for the physical iCub robot. Although the simulator is a faithful representation of the real robot, some functionalities such as vision will require further attention in order to deal with impurities of real cameras and the surrounding environment.

REFERENCES

[1] Arbib, M., Iberall, T., & Lyons, D. (1985). Coordinated control programs for movements of the hand. *Experimental Brain Research* , 111-129.

[2] Asada, M. (2001). Cognitive developmental robotics as a new paradigm for the design of humanoid robots. *Robotics and Autonomous systems , 37*, 135-193.

[3] Bakker, P., & Kuniyoshi, Y. (1996). Robot see, robot do: An overview of robot imitation. *AISB workshop on learning in robots and animals*, (pp. 3-11). Brighton.

[4] Balkenius, C., Zlatev, J., Kozima, H., Dautenhahn, K., & Breazeal, C. (2001). Modeling Cognitive development in Robotics Systems. *Proceedings of the First International Workshop on Epigenetic Robotics, 85.* Lund University.

[5] Barto, A., & Jordan, M. (1987). Gradient following without back-propagation in layered networks. *Proceedings of the IEEE First Annual International Conference on Neural Networks, 2*, pp. 629-636.

[6] Bekey, G., Liu, H., Tomovic, R., & Karplus, W. Knowledge-based control of grasping in robot hands using heuristics from human motor skills. *IEEE Transactions on Robotics and Automation , 9* (6), 709-722.

[7] Berthouze, L., & Prince, C. G. (2003). Introduction: The Third International Workshop on Epigenetic Robotics. In C. G. Prince, L. Berthouze, H. Kozima, D. Bullock, G. Stojanov, & C. Balkenius (Ed.), *Proceedings of the Third International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, 101*. Lund, Sweden: Lund University Cognitive Studies.

[8] Breazeal, C., & Scassellati, B. (2000). Infant-like social interactions between a robot and a human caretaker. *Adaptive Behaviour , 8* (1), 49-74.

[9] Breazeal, C., & Scassellati, B. (2002). Robots that imitate humans. *Trends in Cognitive Sciences , 6* (11), 481-487.

[10] Brock, O., & Khatib, O. (2002). Elastic strips: A framework for motion generation in human environments. *International journal of robotics research , 21* (12), 1031-1052.

[11] Brooks, R. (1998). Alternative essences of intelligence. *15th National Conference on Artificial Intelligence (AAAI-98)* (pp. 961-976). WI: Madison.

[12] Brooks, R. (2003). *The Future of flesh and machines.* London: Penguin Books.

[13] Bullock, D., Grossberg, S., & Guenther, F. (1993). A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm. *Journal of Cognitive Neuroscience , 5* (4), 408-435.

[14] Cangelosi, A. (2001). Evolution of communication and language using signals, symbols and words. *IEEE Transactions on Evolutionary Computation , 5* (2), 93-101.

[15] Carenzi, F., Gorce, P., Burnod, Y., & Maier, M. (2005). Using Generic neural networks in the control and prediction of grasp postures. *ESANN05, European Symposium on Artificial Neural Networks*, (pp. 61-66). Bruges.

[16] Cooperstock, J., & Milios, E. (1993). Self-supervised learning for docking and target reaching. *Robotics and Autonomous Systems , 11* (3-4), 243-260.
[17] Crowe, A., Porrill, J., & Prescott, T. (1998). Kinematic coordination of reach and balance. *Journal of Motor Behavior , 30* (3), 217-233.
[18] Dautenhahn, K. (2007). Socially Intelligent robots: dimensions of human robot interaction. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences. , 362* (1480), 679-704.
[19] Dautenhahn, K., & Billard, A. (1999). Studying robot social cognition within a developmental psychology framework. *Proceedings of the 3rd International Workshop on Advanced Mobile Robots*, (pp. 187-194). Zurich.
[20] Dautenhahn, K., Bond, A., Canamero, L., & Edmonds, B. (2002). Socially intelligent agents: creating relationships with computer and robots. In K. Dautenhahn, A. Bond, L. Canamero, & B. Edmonds, *Socially Intelligent agents: creating relationships with computers and robots* (Vol. 3, pp. 1-20). Norwell: Kluwer Academic.
[21] Elliot, D., & Rao, K. (1982). *Fast Transforms Algorithms, Analyses, Applications.* New York, USA: Academic Press.
[22] Fong, T., Nourbakhsh, I., & Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and Autonomous Systems , 42* (3-4), 143-166.
[23]Fong, T., Thorpe, C., & Baur, C. (2003). Robot, asker of questions. *Robotics and Automation systems , 42* (3-4), 235-243 .
[24] Gaskett, C., & Cheng, G. (2003). Online Learning of a motor map for humanoid robot reaching. *Proceedings of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems.* Singapore.
[25] Gorce, P., & Fontaine, J. (1996). Design methodology approach for flexible grippers. *Journal of Intelligent and Robotic Systems , 15* (3), 307-328.
[26] Grossberg, S. (2003). How Does the Cerebral Cortex Work? Development, Learning Attention, and 3-D Vision by Laminar Circuits of Visual Cortex. *Behavioral and Cognitive Neuroscience Reviews , 2* (1), 47-76.
[27] Grupen, R., & Coelho, J. (2002). Acquiring State form Control Dynamics to learn grasping policies for Robot hands. *International Journal on Advanced Robotics , 16* (5), 427–444.
[28] Iberall, T. (1997). Human prehension and dexterous robot hands. *International Journal of Robotics Research , 16* (3), 285-299.
[29]Jansen, B., & Belpaeme, T. (2006). A model inferring the intention in imitation tasks. *RO-MAN: Robot and Human Interactive Communication.* Hatfield.
[30] Jordan, M. (1986). *Serial Order: A Parallel Distributed Processing Approach.* San Diego.
[31] Jusczyk, P. (1995). Infants detection of the sound patterns of words in fluent speech. *Cognitive Psychology , 29*, 1-23.
[32] Kagami, S., Kuffner, J., Nishiwaki, K., Inaba, M., & Inoue, H. (2003). Humanoid arm motion planning using stereo vision and RRT search. *Journal of Robotics and Mechatronics , 15* (2), 200-207.
[33] Kohonen, T. (1995). *Self-organizing maps* (3rd Edition ed.). Springer.
[34] LaValle, S. (2006). *Planning algorithms.* Cambridge University Press.
[35] Lungarella, M., & Pfeifer, R. (2001). Robot as a cognitive tools: an information-theoretic analysis of sensory-motor data. *2nd IEEE-RAS International Conference on Humanoid Robotics*, (pp. 245-252). Tokyo.
[36] Lungarella, M., Giorgio, M., & Sandini, G. (2004). Developmental robotics: a survey. *Connection Science , 15* (4), 151-190.
[37] Marjanovic, M., Scassellati, B., & Williamson, M. (1996). Self taught visually guided pointing for a humanoid robot. *Fourth International Conference on Simulation of Adaptive Behavior.* MA.
[38] Mason, M. (2001). *Mechanics of robotic manipulation.* Cambridge: MIT Press.
[39] Meltzoff, A. (2002). Elements of a developmental theory of imitation. In A. Meltzoff, & W. Prinz, *The Imitative Mind* (pp. 19-141). New York: Cambridge University Press.
[40] Meltzoff, A., & Moore, M. (1997). Explaining facial imitation: a theoretical model. *Early development and parenting , 6* (3-4), 179-192.
[41] Metta, G. (2001). Development and robotics. *IEEE-RAS International Conference on Humanoid Robots.* Tokyo.
[42] Metta, G., Fitzpatrick, P., & Natale, L. (2006). YARP, Yet Another Robotic Platform. *International Journal on Advanced Robotics Systems* .
[43] Metta, G., Panerai, F., Manzotti, R., & Sandini, G. (2000). Babybot: an artificial developing robotic agent. *6th International Conference on the simulation of Adaptive behaviour*, (pp. 42-53). Paris.
[44] Metta, G., Sandini, G., & Konczak, J. (1999). A developmental approach to visually-guided reaching in artificial systems. *Neural Networks , 12* (10), 1413-1427.
[45] Metta, G., Sandini, G., & Vernon, D. (2005). The RobotCub Project: an open framework for research in embodied cognition. *International Conference of Humanoids Robotics.*
[46] Miller, A., & Allen, P. (1999). Examples of 3D grasps quality computations. *Proceedings of the 1999 IEEE Int. Conf. on Robotics and Automation, 2*, pp. 1240 - 1246.
[47] Moussa, M., & Kamel, M. (1998). An Experimental approach to robotic grasping using a connectionist architecture and generic grasping functions. *IEEE Transactions on System Man and Cybernetics , 28* (2), 239 - 253.
[48] Nadel, J. (2000). Imitation and Imitation Recognition: Functional use in Preverbal infants and Nonverbal children with autism. In A. Meltzoff, & W. Prinz, *The Imitative Mind* (pp. 42-62). Cambridge: Cambridge University Press.
[49] Okada, K., Haneda, A., Nakai, H., Inaba, M., & Inoue, H. (2004). Environment manipulation planner for humanoid robots using task graph that generates action sequences. *international conference on intelligent robots and systems, 2*, pp. 1174 - 1179. Japan.
[50] Pfeifer, R. (2004). Robots as cognitive tools. In B. Gorayska, & J. Mey, *Cognition and Technology* (pp. 109–126). Cambridge University Press.
[51] Rezzoug, N., & Gorce, P. (2006). Upper-limb posture definition during grasping with task and environment constraints. In *Gestures in Human-Computer interaction and simulation* (pp. 212-223). Berlin: Springer.
[52] Ritter, H., Martinetz, T., & Schulten, K. (1992). *Neural computation and self-organizing maps: An introduction.* Boston: Addison Wesley Longman Publishing.
[53] Roy, D., & Pentland, A. (2002). Learning words from sights and sounds:A computational model. *Cognitive Science , 26* (11), 113-146.
[54] Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning representations by back-propagating errors. *Nature , 323*, 533 - 536.
[55]Saito, F., & Nagata, K. (1999). Interpretation of grasp and manipulation based on grasping surfaces. *Proceedings of the 1999 IEEE International Conference on Robotics and Automation, 2*, pp. 1247 - 1254.
[56] Sandini, G., Metta, G., & Vernon, D. (2007). The iCub Cognitive Humanoid Robot: An Open-System Research Platform for Enactive Cognition. In M. Lungarella, F. Iida, J. C. Bongard, & R. Pfeifer, *in in 50 Years of AI* (pp. 358-369). Heidelberg: Springer-Verlag.
[57] Scassellati, B. (1999). Knowing what to imitate and knowing when you succeed. *AISV99 Symposium on Imitation in Animal and Srtefacts*, (pp. 105-113). Edinburg.
[58] Sperber, D., & Hirschfeld, L. (1999). *Culture Cognition and Evolution.* Cambridge Mass: MIT Press.
[59] Steels, L. (1996). Self-organising vocabularies. In C. G. Langton, & K. Shimohara, *Artificial Life V* (pp. 179-184). MIT Press.
[60] Taha, Z., Brown, R., & Wright, D. (1997). Modelling and simulation of the hand grasping using neural networks. *Medical Engineering and Physiology , 19* (6), 536-538.
[61] Tani, J. (2003). Learning to generate articulated behavior through the bottom-up and the top-down interaction processes. *Neural Networks , 16* (1), 11-23.
[62] Tikhanoff V, Fitzpatrick P, Nori F, Natale L, Metta G, Cangelosi A. The iCub Humanoid Robot Simulator. *International Conference on Intelligent Robots and Systems IROS'08*, Nice France 2008
[63] Tikhanoff V, Fitzpatrick P, Nori F, Metta G, Natale L, Cangelosi A, An Open Source Simulator for Cognitive Robotics Research: The Prototype of the iCub Humanoid Robot Simulator. *Performance Metrics for Intelligent Systems Workshop*
[64] Walter, J., & Schulten, K. (1993). Implementation of self-organising neural networks for visuo-motor control of an industrial robot. *IEEE Transactions on Neural Networks , 4* (1), 86 - 96.
[65] Weng, J. (2002). A Theory for Mentally Developing Robots. *2nd international Conference on Cognitive Development and Learning.* MIT Cambridge: IEEE Computer Society Press.
[66] Weng, J., Hwang, W. S., Zhang, Y., Yang, C., & Smith, R. J. (2000). Developmental Humanoids: Humanoids that develop skills automatically. *In Proceedings of the 1st IEEE-RAS Conference on Humanoid.*
[67] Wheeler, D., Fagg, A., & Grupen, R. (2002). Learning prospective Pick and place behavior. *Proceedings of the 2002 IEEE International Conference on Development and Learning*, (pp. 197- 202).
[68] Zlatev, J., & Balkenius, C. (2001). Introduction: Why 'epigenetic robotics'. In C. Balkenius, J. Zlatev, H. Kozima, K. Dautenhahn, & C. Breazeal (Ed.), *Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, 85*, pp. 1-4. Lund University.