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

Vadim Tikhanoff, Angelo Cangelosi, and Giorgio Metta

***Abstract*—Building intelligent systems with human level competence is the ultimate grand challenge for science and technology in general, and especially for cognitive developmental robotics. This paper proposes a new approach to the design of cognitive skills in a robot able to interact with, and communicate about, the surrounding physical world and manipulate objects in an adaptive manner. The work is based on 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, cognitive robotics, manipulation, speech recognition.**

I. Introduction

THOUGH humanoid robots are becoming mechanically more sophisticated, they are still far from achieving human-like dexterous performance when manipulating objects. Cognitive systems research, including developmental robotics, focuses on the development of bioinspired information processing systems that are capable of perception, learning, decision-making, communication, and action. The main objective of cognitive systems research is to transform human–machine systems by enabling machines to engage human users in a human-like cognitive interaction [1]. A cognitive system is based on computational representations and processes of human behavior that replicate the cognitive abilities of natural cognitive systems such as humans and animals [2]–[4]. Using evidence from domains such as neuroscience, cognitive science, and developmental and cognitive psychology, it is possible to build artificial intelligence systems that can possess human-like cognitive abilities.

Developmental cognitive robotics is a growing area of cognitive systems research at the intersection of robotics and developmental sciences in psychology, biology, neuroscience, and artificial intelligence [5], [7], [8]. 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 [6], [10], [11]. Amongst diverse solutions to the programming of robots' capabilities such as attention sharing, turn-taking, and social regulation [12], a major effort in developmental robotics has focused on imitation. A considerable amount of research has been conducted in order to achieve imitating intentional agents [13]–[15]. More recently, researchers have used developmental robotics models in order to study other cognitive functions such as language and communication. Given the developmental approach, linguistic skills are designed in close integration with other sensorimotor and cognitive capabilities [19].

Research into language learning in robots has been significantly influenced over the last ten years by the development of numerous models of evolutionary and developmental emergence of language [3], [16]–[19]. For example, Steels [17] studied the emergence of shared languages in group of autonomous cognitive robotics, which learn categories of object shapes and colors. Cangelosi and collaborators analyzed the emergence of syntactic categories in lexicons that supported navigation [3] and object manipulation tasks [18], [19], in populations of simulated agents and robots.

The majority of these models are based on neural network architectures (e.g., connectionism and computational neuroscience simulations) and adaptive agent models (multiagent systems, artificial life, and robotics). There are many developmental robotic models involved in speech learning, such as the development of vocabulary and grammar [20], [21]. The goal of this section is to produce a real-time system of speech understanding in humanoid robots.

Within the research conducted on linguistic cognitive systems, the focus has been not uniquely on the linguistic element, but also on the close relationship between language and other cognitive capabilities, such as the grounding of language in sensorimotor categories [22]–[24]. Computational models of language have, in the last few decades, focused on the idea of a symbolic explanation of linguistic meaning [25]. Using this symbolic approach, word meanings are defined in terms of other symbols, leading to circular definitions [26], [27]. However, there has recently been an increased focus on symbol grounding approach, i.e., on the important process of

Manuscript received May 21, 2010; revised October 14, 2010; accepted November 26, 2010. Date of publication December 20, 2010; date of current version March 16, 2011. 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 Behavior and Cognition Research Laboratory, University of Plymouth, Plymouth, U.K. He is now with the Italian Institute of Technology, Genoa, 16163, Italy (e-mail: vadim.tikhanoff@iit.it).

A. Cangelosi is with the Adaptive Behavior and Cognition Research Laboratory, University of Plymouth, Plymouth, U.K. (e-mail: acangelosi@plymouth.ac.uk).

G. Metta is with the Robotics, Brain, and Cognitive Science Department, Italian Institute of Technology, Genoa, Italy (e-mail: giorgio.metta@iit.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAMD.2010.2100390

"grounding" the agent's lexicon directly to its own representation of the interaction with the world. Agents learn to name entities, individuals and states, while they interact with the world and build sensorimotor representations of it. Language grounding models provide a new route for modeling complex cross-modal phenomena arising in situated and embodied language use. As early language acquisition is overwhelmingly concerned with objects and activities, which occur in a child's immediate surrounding environment, these models are of a significant interest for understanding situated language acquisition in developmental robotics.

As cognitive systems research is increasingly based on robotics platforms, it is important to consider the contribution of simulations in developmental robotics research. Robot simulators have recently become an essential tool in the design and programming of robotic platforms, whether for industry or research [29], [30]. Furthermore, these robotic simulators have had a significant role in cognitive research, where they have proven to be critical for the development and demonstration of many algorithms and techniques (such as path planning algorithms, grasp planning, and mobile robot navigation).

This paper proposes a new approach to the design of a robotic system that is able to take advantage of all the functionalities that a humanoid robot such as the iCub robotic platform [28] provides. This work will focus on object manipulation capabilities, where refined motor control is integrated with speech "understanding" capabilities. The paper describes cognitive experiments carried out on the iCub simulator [29]. More specifically, the research focuses on a fully instantiated system integrating perception and learning, capable of interacting and communicating in the virtual (simulated) and real world and performing goal directed tasks. This system allows a tighter integration between the representation of the peripersonal space (tactile, proprioceptive, visual, and motor) and the ability to move different effectors. In particular, the goal is to develop a controller that learns to use the available effectors to solve cognitive tasks, potentially by transferring and generalizing already acquired skills. Cognitive experiments will focus on the humanoid iCub robot with vision, touch, audition, and proprioceptive sensorial abilities.

Section II provides a detailed description of the development of the iCub simulator used for the experiments. The cognitive experiments are then presented within the Sections II and III. Section III concentrates on the motor control system, which consists of a reaching and a grasping module. Section IV presents a description of the speech module, and reports simulation experiment results on speech understanding behavior. Both experimental Sections III and IV will also include introductory sections that review current progress in the robotics literature on motor and language learning. An overview of the different modules involved, in the iCub behavior tests, is presented in the following figure, Fig. 1.

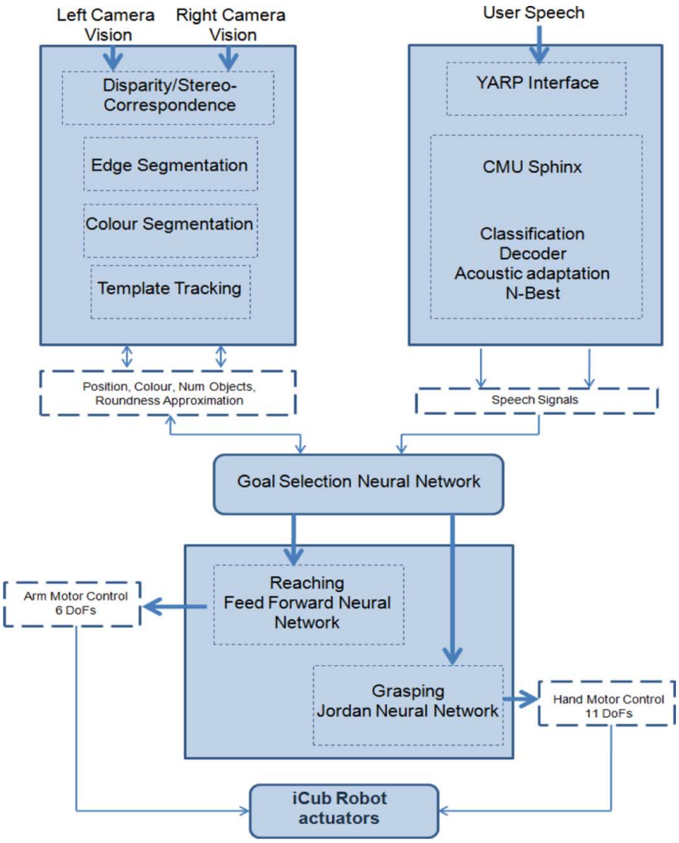


Fig. 1. Architecture of the iCub Cognitive system.

II. Methods

A. The iCub Simulator

Computer simulations play an important role in robotics research. Despite the fact that the use of a simulation might not provide a full model of the complexity present in the real environment and might not assure a fully reliable transferability of the controller from the simulation environment to the real one, robotic simulations are of great interest for cognitive scientists [30]. There are several advantages of robotics simulations for researchers in cognitive sciences. The first is that simulating robots with realistic physical interactions permit to study the behavior of several types of embodied agents without facing the problem of building in advance, and maintaining, a complex hardware device. The computer simulator can be used as a tool for testing algorithms in order to quickly check for any major problems prior to use of the physical robot. Moreover, simulators also allow researchers to experiment with robots with varying morphological characteristics without the need to necessarily develop the corresponding features in the hardware [31]. This advantage, in turn, permits the discovery of properties of the behavior of an agent that emerges from the interaction between the robot's controller, its body, and the environment [32]. Another advantage is that robotic simulations make it possible to apply particular algorithms for creating robots' controllers, such as evolutionary or reinforcement learning algorithms [33]. The use of robotics simulation permits to drastically reduce the time of the experiments such as in evolutionary robotics. In addition, it makes it possible to explore research topics like the coevolution of the morphology and the control system [31]. A simulator for the iCub robot magnifies the value a research group can extract from the physical robot, by making it more practical to share a single robot between several researchers.

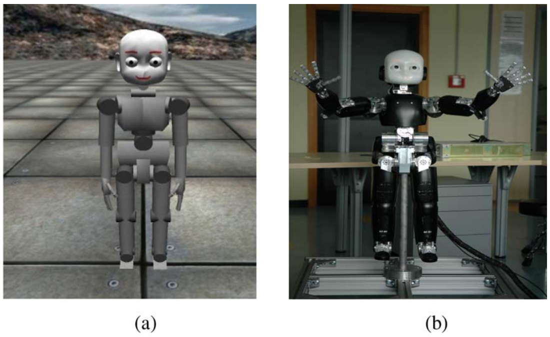


Fig. 2. Photograph of the simulated iCub (a) and of the real iCub as of July 2009.

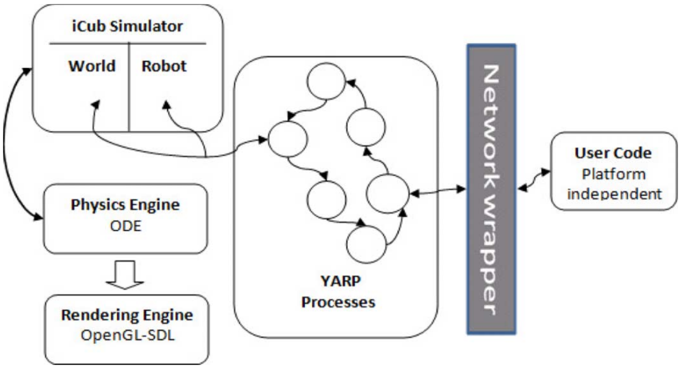


Fig. 3. Detail of the architecture of the simulator with YARP support.

The fact that the simulator is free and open makes it a simple way for people interested in the robot to begin learning about its capabilities and design, with an easy "upgrade" path to the actual robot due to the protocol-level compatibility of the simulator and the physical robot. And for those without the means to purchase or build a humanoid robot, such small laboratories or hobbyists, the simulator at least opens a door to participation in this area of research.

The iCub simulator has been designed to reproduce, as accurately as possible, the physics and the dynamics of the robot and its environment. The simulated iCub robot is composed of multiple rigid bodies connected via joint structures (see Fig. 2). It has been constructed collecting data directly from the robot design specifications in order to achieve an exact replication (e.g., height, mass, degrees of freedom) of the first iCub prototype developed at the Italian Institute of Technology in Genoa. The environment parameters on gravity, objects mass, friction, and joints are based on known environment conditions.

The iCub simulator presented here has been created using open source libraries in order to make it possible to distribute the simulator freely to any researcher without requesting the purchase of restricted or expensive proprietary licenses. Although the proposed iCub simulator is not the only open source robotics platform, it is one of the few that attempts to create a 3-D dynamic robot environment capable of recreating complex worlds and fully based on nonproprietary open source libraries.

B. Physics Engine

The iCub simulator uses open dynamic engine (ODE)[34] for simulating rigid bodies and the collision detection algorithms to compute the physical interaction with objects. The same physics library was used for the Gazebo project and the Webots commercial package. ODE is a widely used physics engine in the open source community, whether for research, authoring tools, gaming, etc. It consists of a high-performance library for simulating rigid body dynamics using a simple C/C++ API. ODE was selected as the preferred open source library for the iCub simulator because of the availability of many advanced joint types, rigid bodies (with many parameters such as mass, friction, sensors, etc.), terrains, and meshes for complex object creation.

C. Communication Protocol

As the aim was to create an exact model of the physical iCub robot, the same software infrastructure and interprocess communication will have to be used as those used to control the physical robot. iCub uses yet another robot platform (YARP) [35] as its software architecture. YARP is an open-source software tool for applications that are real-time, computation-intensive, and involve interfacing with diverse and changing hardware. The simulator and the actual robot have the same interface either when viewed via the device API or across network and are interchangeable from a user perspective. The simulator, like the real robot, can be controlled directly via sockets and a simple text-mode protocol; use of the YARP library is not a requirement. This can provide a starting point for integrating the simulator with existing controllers in esoteric languages or complicated environments.

D. Software Architecture

The architecture of the iCub simulator supporting YARP can be seen in Fig. 3. The User code can send and receive information to both the simulated robot itself (motors/sensors/cameras) and the world (manipulate the world). Network wrappers allow device remotization. The network wrapper exports the YARP interface so that it can be accessed remotely by another machine. This provides an easy path for any system to be extended to the physical robot.

E. The iCub Body Model

The iCub simulator has been created using the data from the physical robot in order to have an exact replica of it. As for the physical iCub, the total height is around 105 cm, weighs approximately 20.3 kg, and has a total of 53 degrees of freedom (DoF). These include 12 controlled DoFs for the legs, three controlled DoFs for the torso, 32 for the arms, and six for the head.

The robot body model consists of multiple rigid bodies attached through a number of different joints. All the sensors were implemented in the simulation on the actual body, such as touch sensors and force/torque sensors. As many factors impact on the torque values during manipulations, the simulator might not guarantee to be perfectly correct. However, the simulated robot torque parameters and their verification in static or motion are a good basis and can be proven to be reliable [36].

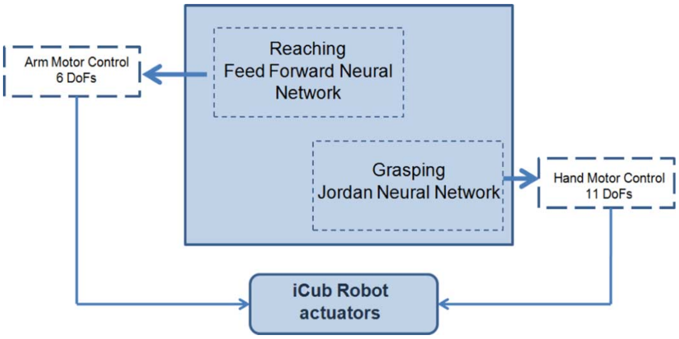


Fig. 4. Diagram of the motor control modules.

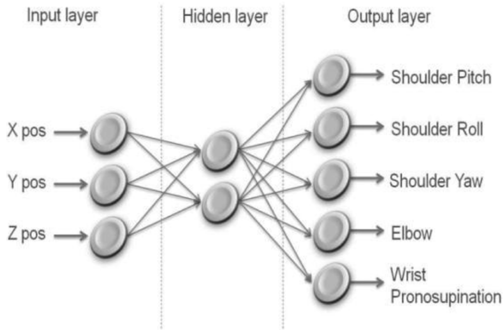


Fig. 5. Architecture of the employed feed-forward neural network.

All the commands sent to and from the robot are based on YARP instructions. For the vision, we use cameras located at the eyes of the robot which in turn can be sent to any workstation using YARP in order to do develop vision analysis algorithms.

The system has full interaction with the world/environment. The objects within this world can be dynamically created, modified, and queried by simple instruction resembling those that YARP uses in order to control the robot.

III. Motor Control Learning

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 to attempt 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 employs a control system consisting of an artificial neural network configured as a feed-forward controller [37]. The second part of the motor learning model incorporates the above reaching module within an additional controller needed for the robot to actually grasp the object. This employs another control system consisting of a neural controller configured as a Jordan neural network [38].

Fig. 4 provides a diagram overview of the motor control section that is described in this section.

B. Learning to Reach

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.

In computational neuroscience, research on reaching has focused on the development of neurocognitive models of human behavior, that can also be employed in humanoid robots to achieve human-like reaching [39]. Additionally, neuroscience research considers the issue of pregrasping as defined by Arbib *et al.* [40]. This deals with the configuration of the fingers for successful grasping, while performing the reaching movement. These finger configurations must satisfy some form of predefined knowledge on the object affordances for appropriate grasping, and predefined knowledge about the task to accomplish. However, the model presented in this paper is not concerned with generating a reaching system consistent with human models of pregrasping, but assumes that reaching and grasping can be performed independently [41].

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 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 [42]. Metta *et al.* [42] have developed a humanoid robot controller based on single motor mapping, where the mapping from the two eyes can 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 limitations due to errors in the hand positioning. In an earlier paper, Marjanovic *et al.* [44] proposed 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 45.

The reaching module developed in this work is based on the learning of 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 constraint in the 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. The robot will then be able to compute the required motor outputs to reach an object at a specific x, y, and z coordinate. A feed-forward multilayer perceptron, with back propagation algorithm [46] was modeled to simulate reaching for diverse objects that reside within their surroundings. The neural network architecture as depicted in Fig. 5 was used.

TABLE I
DESCRIPTION OF THE DIFFERENT JOINTS USED FOR THE REACHING MODULE

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

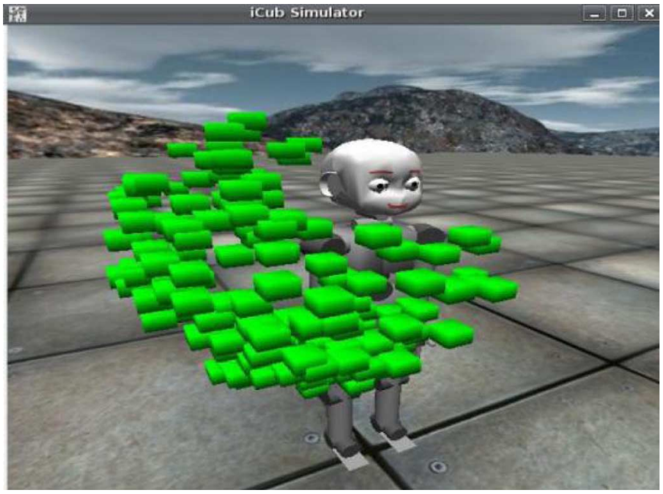


Fig. 6. Example of the 150 end positions of the robot arms during training.

TABLE II
TRAINING PARAMETERS OF THE REACHING FEED—FORWARD NETWORK MODULE

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

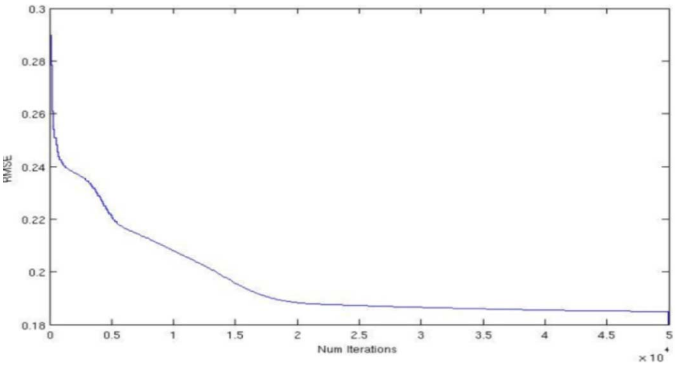


Fig. 7. RMSE value during training of the reaching module.

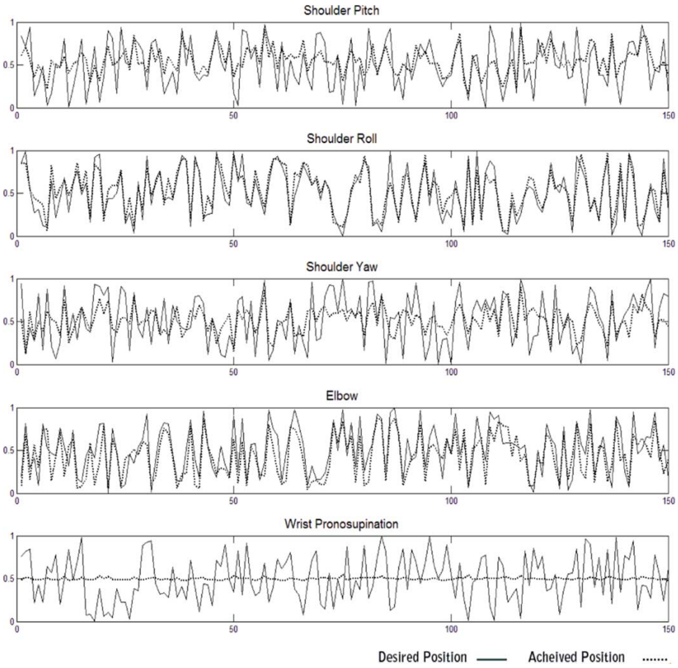


Fig. 8. First 150 results of the 2500 samples given to the network. Each graph represents the different joint degrees at each of the 150 positions.

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 [47], and depth estimation [48]. The output of the network is a vector of angular positions of five joints that are located on the arm of the robot. The joints used for the reaching module are described in Table I.

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 5000 random sequences, while 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. Fig. 6 shows an example of 150 positions of the endpoints of the robot hands used during training, by representing them as green squares.

The feed-forward neural network controller was trained with the parameters listed in Table II.

After multiple tests of 50 000 iterations, the final RMSE (root mean squared error) ranged from 0.15 to 0.16 (e.g., sample training curve Fig. 7). Although low, an RMSE of 0.15 indicates that the neural network did not achieve optimal performance.

By analyzing the results, in contrary to just base on the final RMSE, we can see that the network has actually been successful in learning to reach the specific position, with its joint configuration. But it has discarded the last joint completely, as shown in Fig. 8. Fig. 8 displays the first 150 results of the 2500 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).

The reason for such a high RMSE is due to several factors. The main 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, it is eventually discarded by the network when learning the training data. The desired mappings of the remaining joints of the iCub have been satisfied as much as possible without the use of this joint.

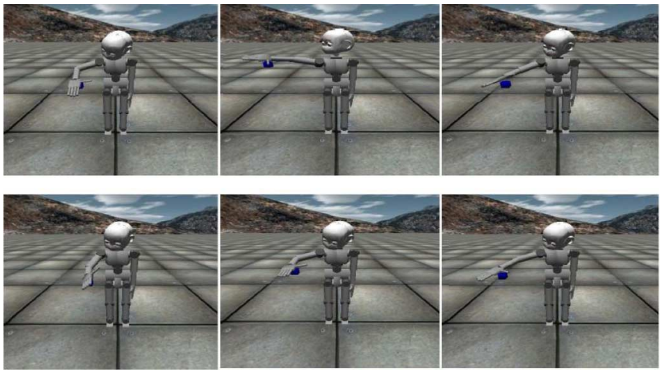


Fig. 9. Images taken from the robot during the testing of the reaching module.

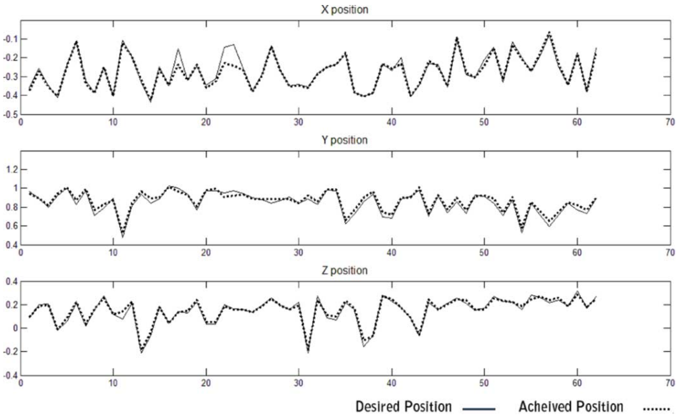


Fig. 10. Comparison of 62 random XYZ positions of objects, with the actual resulting position of the robot's hand.

In order to test the performance of the model, a pretrained reaching neural network was loaded onto the simulation, while random objects were placed in the vicinity of the iCub robot. The results of these generalization tests showed that the model was capable of successfully locating and tracking the object in new positions, and finally reaching the target. Fig. 9 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. 10 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 (desired position), and then compares them with the actual resulting position of the robot's hand (achieved position).

Overall, the experimental setup and results show a robotic system that is able to perform reaching using stereo cameras from the iCub simulator. 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, while 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 Section IV, after a brief discussion on recent work on grasping, we will describe the approach that was used to solve the well known grasping problem.

C. Learning to Grasp

One of the major challenges in humanoid 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. To take 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 the following methodological approaches [49]:

- knowledge-based grasping;
- geometric contact grasping;
- sensory driven and learning-based grasping.

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 [50], [51].

Although these methods are effective and produce good results, they have the requirement to require 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 have to deal with the issue of pregrasping, which requires anticipation of the grasp before reaching the object, and depends on the task and the object. 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 [52]. This is an optimal approach, as it can be applied to a large amount of dexterous robotic hands while 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 easily achieved. Finally, the sensory driven grasping approach tries to solve the previously mentioned problems by using learning and task exploration [53].

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 [54]. More recently, Carenzi *et al.* [55] 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 multijoint hand) dependent on the object size. Although this work

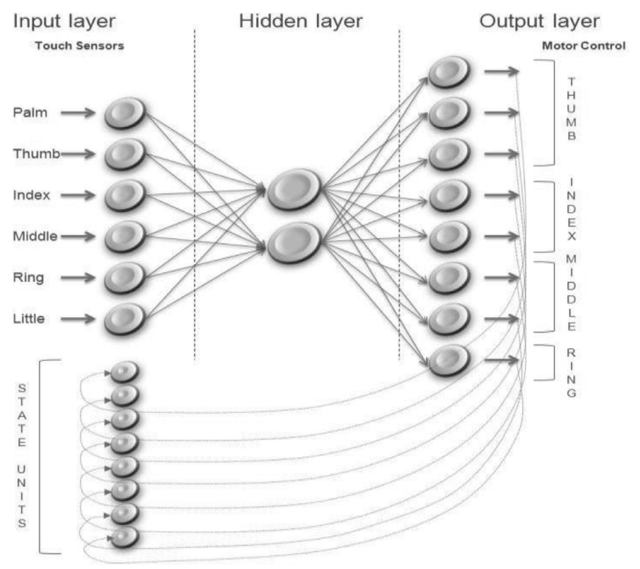


Fig. 11. Architecture of the employed jordan neural network.

is interesting, it is highly simplified and both wrist position and orientation need to be predefined.

In our model of the iCub grapsing, a new method 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 approaches was proposed by Jordan [38], who proposed a neural network with recurrent connections copying the output unit values and feeding them back to the context or state units. To briefly recap; in his paper, Jordan described a neural network as carrying recurrent connections, which are implemented to associate a stable pattern, are considered as a plan with a continual output pattern, and as a sequence of actions. The recurrent connections permit the neural network's hidden units to discover its own previous output. This is useful for the subsequent behaviors as they will be influenced by the previous responses.

A Jordan type neural network was implemented in this model to train the simulated iCub to learn to grasp diverse objects located in the robot's environment. The neural network architecture can be seen in Fig. 11.

The input layer of the Jordan neural network consists of the vector of the touch sensors information of the robot's hand (either 0 or 1). The output is a vector of normalized (0 to 1) angular positions of the eight finger joints, which are located on the hand of the robot. The hidden layer comprises five units. 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 Fig. 12 and a detailed description of the hand joints used can be seen in Table III.

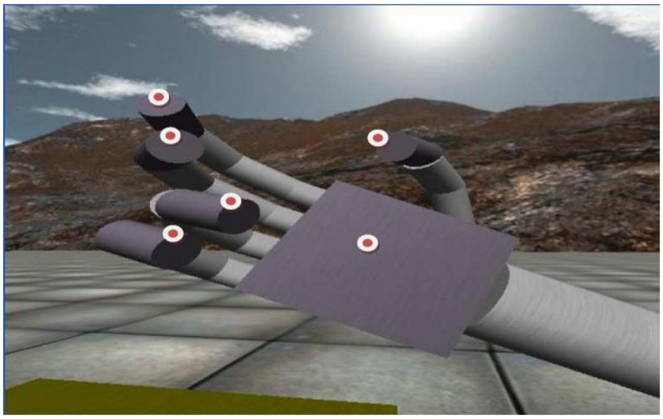


Fig. 12. Location of the six touch sensors on the iCub's simulator hand.

TABLE III
LIST OF FINGER JOINTS USED IN THE GRASPING MODULE

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 front-back movement	Middle closing
Ring and little finger flexion	Ring and little front-back movement and closing

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 (object) that triggers the activation of the sensor (1).

The training of the grasping Jordan neural network is achieved online and therefore no training patterns have been predefined to teach 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 (ARP) algorithm is implemented in order to train the network connection weights. A description of this algorithm can be found in [56]. This method is used for associative reinforcement learning, as the standard back-propagation algorithm is not able to perform such a task. The neural network needs 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 how gravity affects 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.

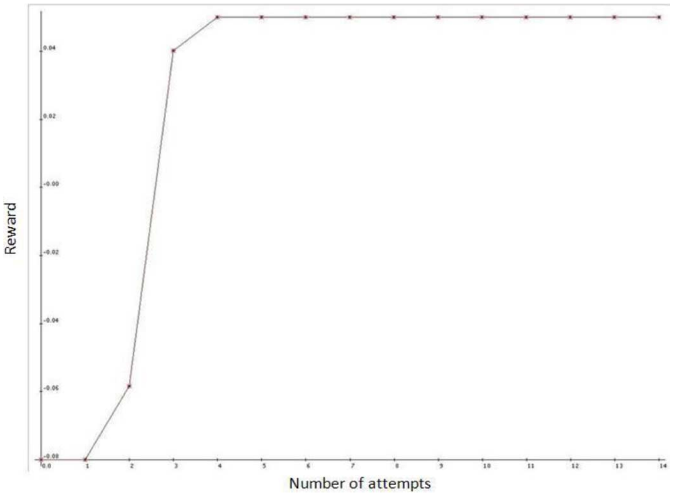


Fig. 13. Reward rate during the grasping neural network training phase.

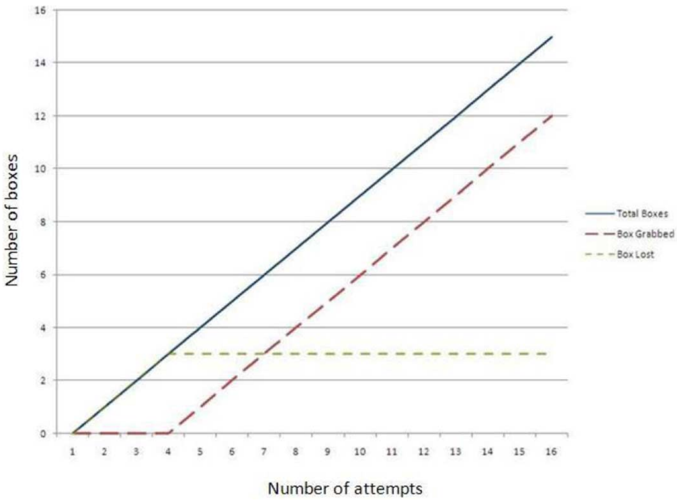


Fig. 14. Graph showing the total boxes used, total boxes grabbed, and total boxes lost, during a simple grasping experiment with an object of specific size.

A number of experiments were carried out in order to test the model 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 Figs. 13 and 14 show the results of an 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 Fig. 15. The object size parameters (in meters) are:

- $\mathrm{width} = 0.05$, $\mathrm{height} = 0.03$, $\mathrm{depth} = 0.04$.

The object was then modified to a cube with parameters:

- $\mathrm{width} = 0.04$, $\mathrm{height} = 0.04$, $\mathrm{depth} = 0.04$.

The object was placed at different coordinates in order to further test the system under simple conditions. Fig. 13 displays the reward rate of the grasping neural network during a training phase of 15 attempts; the maximum reward was obtained after four attempts. Fig. 14 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 $z = 0.04$.

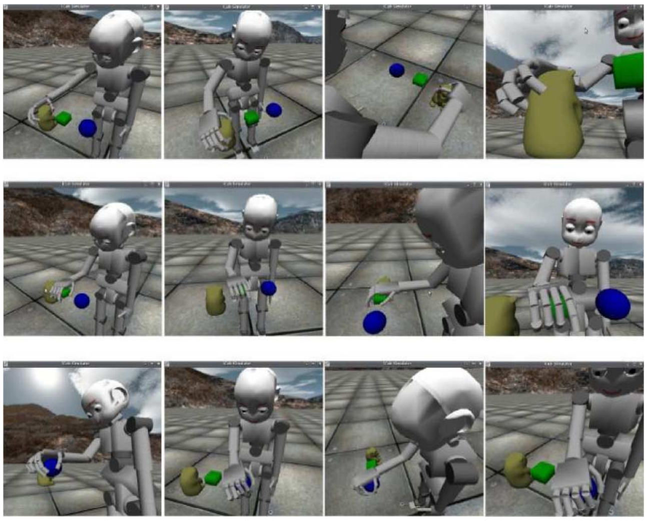


Fig. 15. Grasping of three different objects.

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 pretrained grasping neural network was then loaded onto the simulation to demonstrate that the system is able to generalize grasping with different objects.

Fig. 15 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).

IV. Working With Speech

A. Introduction

As mentioned in Section I, language and speech shape a large part of human–human and even human–machine interaction [57]. In speech, there is an immense potential for diversity, as speech is very flexible. This flexibility is apparent when interacting with children or pets; therefore, a similar approach would be ideal for robots. The goal of this section is to produce a real-time system of speech understanding.

Speech recognition can be applied successfully for a large user population across noisy conditions [58] such as basic vocabulary typically used for queries, or using a good quality headset and extensive user training typically used in dictations with a large grammar. At this stage it is important to establish where robot directed speech lies depending on the task given to the robot.

It has been shown that infant-directed words are usually kept short with large pauses between words [59]. Brent and Siskind [60] present evidence that isolated words are in fact a reliable feature of infant-directed speech, and that infants' early word acquisition may be facilitated by their presence. In particular, the authors find that the frequency of exposure to a word in isolation is a better predictor of whether the word will be learned, than the total frequency of exposure. This suggests that isolated words may be easier for infants to process and learn.

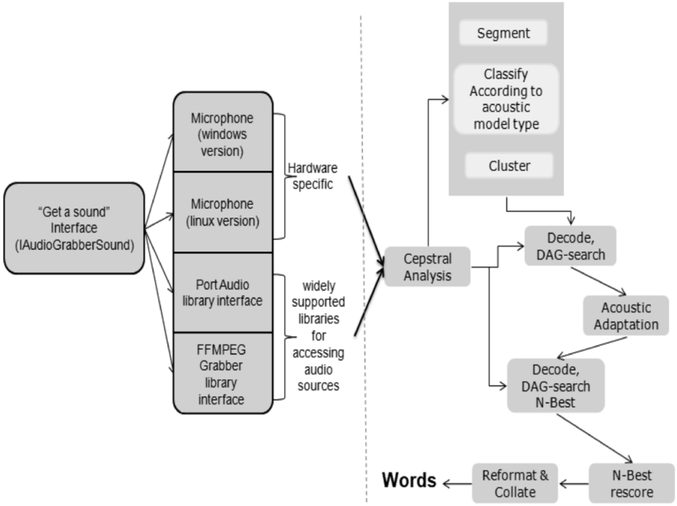

"Get a sound" Interface (IAudioGrabberSound)
Microphone (windows version)
Microphone (linux version)
Port Audio library interface
FFMPEG Grabber library interface
Hardware specific
widely supported libraries for accessing audio sources
Cepstral Analysis
Segment
Classify According to acoustic model type
Cluster
Decode, DAG-search
Acoustic Adaptation
Decode, DAG-search N-Best
N-Best rescore
Reformat & Collate
Words

Fig. 16. Architecture of the integration of YARP and Sphinx.

B. Speech Recognition

The speech recognizer system developed at Carnegie Mellon University was used [61]. The Sphinx-3 system is a flexible hidden Markov model-based speech recognition system. Its components can be configured at run-time along the spectrum of semi-to-fully-continuous operation. These include a series of speech recognizers (Sphinx 2–4) and an acoustic model trainer (SphinxTrain). CMU Sphinx is perhaps the only open source, large vocabulary, continuous speech recognition project that consistently releases its work under the liberal BSD-license.

C. CMU Sphinx Recognition Structure

The sphinx recognition system is composed of number of sequential stages. In particular, we can identify the following seven stages (adapted from [61]).

- **Segmentation, classification, and clustering**:
  Initially, the long audio streams are chunked into smaller segments. The segmentation points are chosen such that these coincide with acoustic boundaries.
- **Initial-pass recognition**:
  Preliminary recognition is done with a straight-forward continuous-density Viterbi beam search producing a word lattice for each subsegment.
- **Initial-pass best-path search**:
  These lattices are then searched for the global best path according to the trigram grammar.
- **Acoustic adaptation**:
  The HMM means is then adapted using maximum likelihood linear regression (MLLR). This adaptation is performed with a single regression matrix.
- **Second-pass recognition**:
  Each sub segment is then decoded again, using the acoustic models adapted in the previous step. Again a lattice is produced for each subsegment.
- **Second-pass best-path search**:
  The lattice is searched for the global best path and an N-best search over the lattice is also done.
- **N-best rescoring**:
  The N-best lists generated using the supplemented vocabulary were processed to convert the phrases and acronyms into their constituent words and letters.

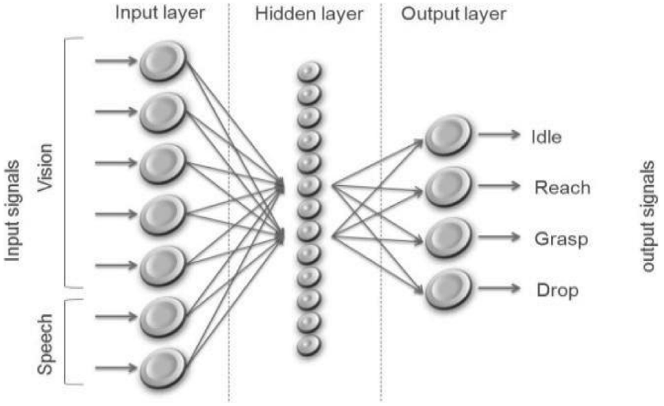


Fig. 17. Goal selection neural network architecture used.

D. YARP and CMU Sphinx Integration Architecture

YARP includes an abstract interface, named *IAudioGrabberSound*, that decouples streaming audio functionality from the underlying hardware, platform, or format. This interface may be used either live using the robot's microphones, or alternatively using prerecorded samples. The left hand side of Fig. 16 shows the *IAudioGrabberSound* interface and some concrete implementations thereof.

The Sphinx module is designed such that it only depends on this interface in order to obtain streaming audio input. This design provides implementation independence and facilitates reuse of the module on different platforms and hardware, as well as allowing the reproduction of experiments from recordings. Fig. 16 gives an overview of the global architecture entailing both the Sphinx module and the YARP audio interface.

E. Learning to Integrate Speech and Action

An essential skill, of any type of cognitive system, is the ability to acquire and generate a variety of actions, and to exhibit the behavior that corresponds to social and environmental conditions. This requires that the robot is endowed with a certain amount of knowledge concerning the past and present, or even future events, which will permit it to perform precise motor controls, while also communicating using a speech understanding module.

The integration of the speech signals, visual input, and motor control abilities was based on a goal selection neural network, a feed forward neural network. This is able to learn and categorize speech signals so that it can form corresponding visual categories and finally determine the appropriate action to perform.

The input to the network consists of seven parameters from the vision acquisition system (e.g., object size and location) and the output of the Sphinx speech signals. The output consists of four units corresponding to the following action: idle, reach, grasp, and drop. For example, the output "reach" consists of just one sequence, whereas "grasp" and "drop" consist of multisequences that are composed of different actions; "grasp" is composed of reaching and then grasping, and "drop" is a sequence

TABLE IV
LIST OF SPEECH SIGNALS USED IN THE COGNITIVE EXPERIMENT

“Blue ball”	“Reach blue ball”	“Grasp blue ball”	“Drop blue ball into basket”
“Red ball”	“Reach red ball”	“Reach 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 V
TRAINING PARAMETERS OF THE GOAL SELECTION NEURAL NETWORK MODULE

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

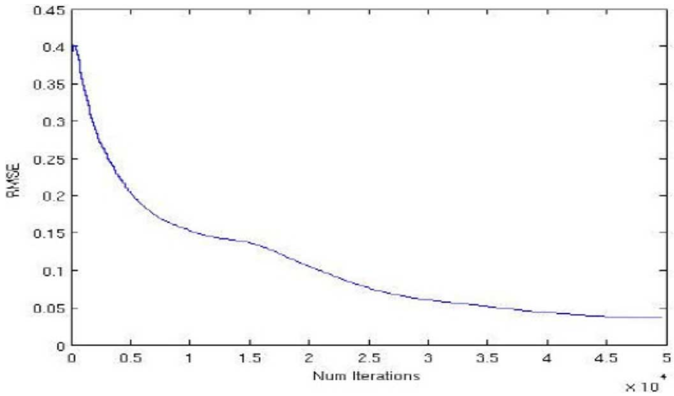
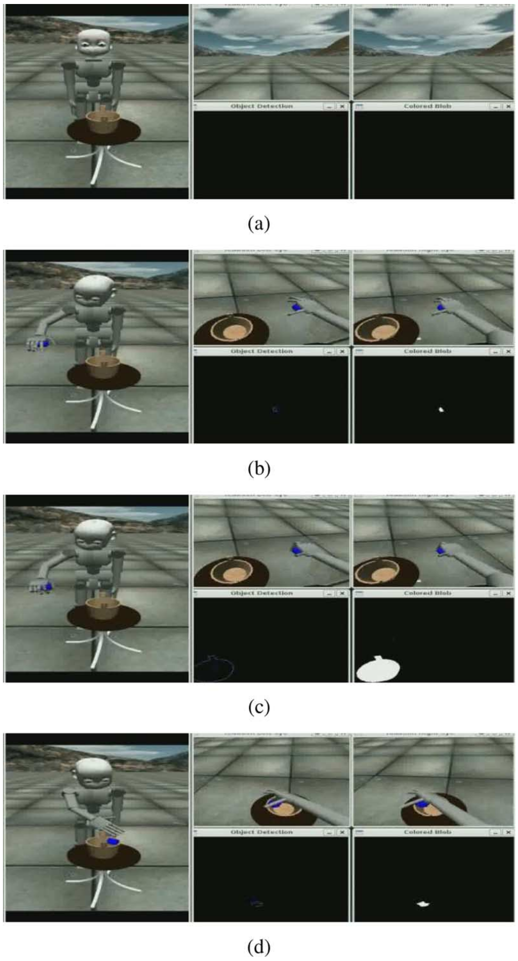


Fig. 18. RMSE value during training of the goal selection module.



(a)

(b)

(c)

(d)

Fig. 19. 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.

of reaching the object, grasping it, reaching a position, and then releasing the object by inverting the grasping module to return to the original joint configuration of the hand. The hidden layer comprises 15 units. The neural network’s architecture can be seen in Fig. 17. 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.

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 root mean squared error (RMSE) was ranging at 0.0368, which indicates a successful learning of the neural network (see Fig. 18).

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 Fig. 19.

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

V. 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 system developed here was influenced by the way infants tend to learn speech from sounds [62], 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 [63]. 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 and linguistic skills of the humanoid robot. The proposed cognitive control architecture reported here has been based on the iCub simulator, but has also been transferred to the physical iCub robot with comparable results.

VI. Future Work

Current work is now focusing on modeling of visual attention, with particular focus on how a robotic visual attention system can develop in an autonomous manner, through interacting with its environment. An object, in terms of computer vision, is often defined in terms of restricted sets of visual cue responses or abstractions thereof. Instead, we generalize the notion of an object as a visual surface at fixation exhibiting spatiotemporal coherence, regardless of its cue responses. A spatiotemporal zero disparity filter (SpTZDF) encodes the likelihood that an image coordinate projects to a spatio–temporally coherent visual surface [64]. Subsequently, a Markov random field refinement step converts the generated probability maps into image segmentations. A tracking algorithm is instantiated such that the visual surface remains at fixation by detecting spatio–temporal coherence rather than explicitly encoding permitted motion models. The approach elicits real-time active monocular and/or coordinated stereo fixation upon arbitrarily translating, scaling, rotating, reconfiguring visual surfaces, and marker-less pixel-wise segmentation thereof. Segmentation and tracking is shown to be robust to lighting conditions, defocus, foreground and background clutter, and partial or gross occlusions of the visual surface at fixation [64].

The propensity to attend and segment spatio–tempotrally coherent visual surfaces (objects) yields significant benefits in terms of object classification. Classifying a presegmented object removes background regions that could induce error in the classification. Moreover, such segmentations can be used to significantly improve the training stage of classifier development. Training images can be acquired autonomously by the same apparatus that uses query stage. Prior segmentation additionally allows segmentation prescaling and autocentering such that additional constancy is induced before training. To further induce constancy, the segmentations (both training and query images) are processed with a difference-of- Gaussian filter that imposes intensity invariance. This system takes inspiration from biology. Primates train and query using the same visual apparatus [64], [65]. Primates have the propensity to attend and discern spatiotemporally coherent objects from backgrounds. Mechanisms to induce constancy, including ganglion responses similar to that of a difference-of- Gaussian filter, are known to exist in the primate visual system.

The development of the visuo–attentional system described above, and its integration with the speech-action model presented in this paper, provides a novel and useful approach for the development of integrated cognitive systems for developmental robotics.

References

[1] D. Sperber and L. Hirschfeld, "Culture cognition and evolution," in *MIT Encyclopedia of the Cognitive Sciences*, R. Wilson and F. Keil, Eds. Cambridge, MA: MIT Press, 1999, pp. cxi–cxxxii.

[2] C. Breazeal and B. Scassellati, "Infant-like social interactions between a robot and a human caretaker," *Adapt. Behav.*, vol. 8, no. 1, pp. 49–74, 2000.

[3] A. Cangelosi, "Evolution of communication and language using signals, symbols, and words," *IEEE Trans. Evol. Comput.*, vol. 5, no. 2, pp. 93–101, Apr. 2001.

[4] T. Fong, C. Thorpe, and C. Baur, "Robot, asker of questions," *Robot. Autonom. Syst.*, vol. 42, no. 3–4, pp. 235–243, 2003.

[5] M. Asada, K. F. MacDorman, H. Ishiguro, and Y. Kuniyoshi, "Cognitive developmental robotics as a new paradigm for the design of humanoid robots," *Robot. Autonom. Syst.*, vol. 37, no. 2, pp. 185–193, 2001.

[6] M. Lungarella and R. Pfeifer, "Robot as a cognitive tool: An information theoretic analysis of sensory-motor data," in *Proc. 2nd IEEE-RAS Int. Conf. Human. Robot.*, Tokyo, Japan, 2001, pp. 245–252.

[7] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: A survey," *Connect. Sci.*, vol. 15, no. 4, pp. 151–190, 2003.

[8] G. Metta, G. Sandini, L. Natale, and F. Panerai, "Development and robotics," in *Proc. 2nd IEEE-RAS Int. Conf. Human. Robot.*, Tokyo, Japan, Nov. 2001, pp. 33–42.

[9] R. Pfeifer, "Robots as cognitive tools," *Int. J. Cogn. Technol.*, vol. 1, no. 1, pp. 125–143, 2002.

[10] C. Balkenius *et al.*, "Modeling cognitive development in robotics systems," in *Proc. 1st Int. Workshop Epig. Robot.*, Lund, Sweden, 2001, pp. 1–4.

[11] J. Weng, W. S. Hwang, Y. Zhang, C. Yang, and R. J. Smith, "Developmental humanoids: Humanoids that develop skills automatically," in *Proc. 1st IEEE-RAS Conf. Human.*, Beijing, China, 2001, pp. 123–132.

[12] K. Dautenhahn, "Socially intelligent robots: Dimensions of human-robot interaction," *Philos. Trans. Roy. Soc. London-Series B*, vol. 362, no. 1480, pp. 679–704, 2007.

[13] P. Bakker and Y. Kuniyoshi, "Robot see, robot do: An overview of robot imitation," in *Proc. AISB Workshop Learn. Robot. Animals*, Brighton, U.K., 1996, pp. 3–11.

[14] A. Meltzoff, “Elements of a developmental theory of imitation,” in *The Imitative Mind*, A. Meltzoff and W. Prinz, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2002, pp. 19–141.

[15] B. Scassellati, “Knowing what to imitate and knowing when you succeed,” in *Proc. AISB’99 Symp. Imitation Animals Artifacts*, Edinburgh, Scotland, Apr. 1999, pp. 105–113.

[16] R. MacWhinney, “Models of the emergence of language,” *Annu. Rev. Psychol.*, vol. 49, pp. 199–227, 1998.

[17] L. Steels, “Evolving grounded communication for robots,” *Trends Cogn. Sci.*, vol. 7, no. 7, pp. 308–312, 2003.

[18] A. Cangelosi, E. Hourdakis, and V. Tikhanoff, “Language acquisition and symbol grounding transfer with neural networks and cognitive,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN06)*, Vancouver, BC, Canada, 2006, pp. 1576–1582.

[19] A. Cangelosi and T. Riga, “An embodied model for sensorimotor grounding and grounding transfer,” *Cogn. Sci.*, vol. 30, no. 4, pp. 673–689, 2006.

[20] D. Roy, “Learning visually grounded words and syntax of natural spoken language,” *Evol. Commun.*, vol. 4, no. 1, pp. 33–56, 2002.

[21] L. Steels, “Self-organising vocabularies,” in *Artificial Life V*, C. Langton and K. Shimohara, Eds. Cambridge, MA: MIT Press, 1996, pp. 179–184.

[22] A. Cangelosi, “The emergence of language: Neural adaptive agent models,” *Connect. Sci.*, vol. 17, no. 3–4, pp. 185–190, 2005.

[23] K. Plunkett, C. Sinha, M. F. Møller, and O. Strandsby, “Symbol grounding or the emergence of symbols? vocabulary growth in children and a connectionist net,” *Cogn. Sci.*, vol. 4, pp. 293–312, 1992.

[24] D. Roy and N. Mukherjee, “Towards situated speech understanding: Visual context priming of language models,” *Comput. Speech Lang.*, vol. 19, no. 2, pp. 227–248, 2005.

[25] W. Kintsch, *Comprehension: A Paradigm for Cognition*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[26] S. Harnad, “The symbol grounding problem,” *Physica D*, vol. 42, pp. 335–346, 1990.

[27] D. Roy, “Scemiotic schemas: A framework for grounding language in action and perception,” *Artif. Intell.*, vol. 167, no. 1–2, pp. 170–205, 2005.

[28] G. Sandini, G. Metta, and D. Vernon, “The iCub cognitive humanoid robot: An open-system research platform for enactive cognition,” in *50 Years of AI*, M. Lungarella, Ed. *et al.* Berlin, Germany: Springer-Verlag, 2007, pp. 359–370.

[29] V. Tikhanoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale, and F. Nori, “An open-source simulator for cognitive robotics research: The prototype of the iCub humanoid robot simulator,” in *Proc. IEEE Workshop Perform. Metrics Intell. Syst.*, Washington, DC, 2008.

[30] T. Ziemke, “On the role of robot simulations in embodied cognitive science,” *AISB J.*, vol. 1, no. 4, pp. 389–399, 2003.

[31] J. C. Bongard and R. Pfeifer, “Evolving complete agents using artificial ontogeny,” in *Morpho-Functional Machines: The New Species (Designing Embodied Intelligence)*, F. Hara and R. Pfeifer, Eds. Berlin, Germany: Springer-Verlag, 2003, pp. 237–258.

[32] S. Kumar and P. J. Bentley, *On Growth, Form and Computers*. Amsterdam, The Netherlands: Elsevier , 2003.

[33] S. Nolfi *et al.*, “How to evolve autonomous robots: Different approaches in evolutionary robotics,” in *Artificial Life IV*, R. Brooks and P. Maes, Eds. Cambridge, MA: MIT Press, 2000.

[34] R. Smith, Open Dynamic Engine 2001 [Online]. Available: http://www.ode.org/

[35] G. Metta, P. Fitzpatrick, and L. Natale, “YARP: Yet another robot platform,” *Int. J. Adv. Robot. Syst., Special Issue Software Develop. Integr. Robot.*, vol. 3, no. 1, pp. 43–48, 2006.

[36] N. Nava *et al.*, “Kinematic and dynamic simulations for the design of robotcub upper body structure,” in *Proc. Eng. Syst. Design Anal. Conf. (ESDA)*, Haifa, Israel, 2008, pp. 413–421.

[37] D. E. Rumelhart, G. E. Hinton, J. L. McClelland, and A. General, “Framework for parallel distributed processing,” in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClell, Eds. Cambridge, MA: MIT Press, 1986.

[38] M. Jordan, *Serial Order: A Parallel Distributed Processing Approach*. San Diego, CA: California Univ. Press, 1986, pp. 64–64.

[39] A. Crowe, J. Porrill, and T. Prescott, “Kinematic coordination of reach and balance,” *J. Motor Behav.*, vol. 30, no. 3, pp. 217–233, 1998.

[40] M. A. Arbib, T. Iberall, and D. Lyons, “Coordinated control programs for movements of the hand,” *Exp. Brain Res.*, vol. 10, pp. 111–129, 1985.

[41] K. Okada, A. Haneda, H. Nakai, M. Inaba, and H. Inoue, “Environment manipulation planner for humanoid robots using task graph that generates action sequence,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Tokyo, Japan, 2004, pp. 3553–3558.

[42] G. Metta, G. Sandini, and J. Konczak, “A developmental approach to visually-guided reaching in artificial systems,” *Neural Netw.*, vol. 12, no. 10, pp. 1413–1427, 1999.

[43] G. Metta, F. Panerai, R. Manzotti, and G. Sandini, “Babybot: An artificial developing robotic agent,” in *Proc. 6th Int. Conf. Simulation Adapt. Behav.*, Paris, France, 2000.

[44] M. Marjanovic, B. Scassellati, and M. Williamson, “Self taught visually guided pointing for a humanoid robot,” in *Proc. 4th Int.l Conf. Simulation Adapt. Behav. (SAB-96)*, Cape Cod, MA, 1996.

[45] J. R. Cooperstock and E. E. Milios, “Self-supervised learning for docking and target reaching,” *Robot. Autonom. Syst.ems*, vol. 11, no. 3–4, pp. 243–260, 1993.

[46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representation by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986.

[47] J. P. Lewis, “Fast template matching,” *Vis. Interface*, pp. 120–123, 1995.

[48] S. Birchfield and C. Tomasi, “Depth discontinuities by Pixel-to-Pixel stereo,” *Int. J. Computer Vis.*, vol. 35, no. 3, pp. 269–293, 1999.

[49] N. Rezzoug and P. Gorce, “Robotic grasping: A generic neural network architecture,” in *Mobile Robots Towards New Applications*, A. Lazinica, Ed. Berlin, Germany: Pro Literatur Verlag Robert Mayer-Scholz, 2006.

[50] G. A. Bekey *et al.*, “Knowledge-based control of grasping in robot hands using heuristics from human motor skills,” *IEEE Trans. Robot. Autom.*, vol. 9, no. 6, pp. 709–722, Dec. 1993.

[51] T. Iberall, “Human prehension and dexterous robot hands,” *Int. J. Robot. Res.*, vol. 16, no. 3, pp. 285–299, 1997.

[52] P. Gorce and J. Fontaine, “Design methodology for flexible grippers,” *J. Intell. Robot. Syst.*, vol. 15, no. 3, pp. 307–328, 1996.

[53] R. Grupen and J. Coelho, “Acquiring state form control dynamics to learn grasping policies for robot hands,” *Int. J. Adv. Robot.*, vol. 15, no. 5, pp. 427–444, 2002.

[54] M. Moussa and M. kamel, “An experimental approach to robotic grasping using a connectionist architecture and generic grasping functions,” *IEEE Trans. Syst., Man, Cybernet.*, vol. 28, no. 2, pp. 239–253, 1998.

[55] F. Carenzi, P. Gorce, Y. Burnod, and M. Maier, “Using generic neural networks in the control and prediction of grasp postures,” in *Proc. 13th Eur. Symp. Artif. Neural Netw. (ESANN 2005)*, Bruges, Belgium, 2005.

[56] A. G. Barto and M. Jordan, “Gradient following without back-propagation in layered networks,” in *Proc. IEEE 1st Annu. Conf. Neural Netw.*, San Diego, CA, 1987, pp. II629–II636.

[57] P. Varchacskaia, P. Fitzpatrick, and C. Breazeal, “Characterizing and processing robot-directed speech,” in *Proc. Int. IEEE/RSJ Conf. Human. Robot.*, Tokyo, Japan, 2001, pp. 229–237.

[58] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen, and L. Hetherington, “Jupiter: A telephone based conversational interface for weather information,” *IEEE Trans. Speech Audio Process.*, vol. 8, no. 1, pp. 100–112, Jan. 2000.

[59] J. F. Werker, V. L. Lloyd, and J. E. Pegg, “Putting the baby in the bootstraps: Towards a more complete understanding of the role of the input in the infant speech processing,” in *Signal to Syntax: Bootstrapping From Speech to Grammar in Early Acquisition*, J. Morgan and K. Demuth, Eds. Mahwah, NJ: Erlbaum, 1996, pp. 427–447.

[60] M. Brent and J. Siskind, “The role of exposure to isolated words in early vocabulary development,” *Cognition*, vol. 81, pp. B33–B44, 2001.

[61] P. Placeway *et al.*, “The 1996 hub-4 sphinx-3 system,” in *Proc. ARPA Speech Recogn. Workshop*, Gaithersburg, MD, 1997, pp. 85–89.

[62] P. W. Jusczyk, “How infants begin to extract words from speech,” *Trends Cogn. Sci.*, vol. 3, no. 9, pp. 323–328, 1999.

[63] A. Meltzoff and M. Moore, “Explaining facial imitation: A theoretical model,” *Early Develop. Parent.*, vol. 6, no. 3–4, pp. 179–192, 1997.

[64] A. Dankers, N. Barnes, and A. Zelinsky, “MAP ZDF segmentation and tacking using active stereo vision: Hand tracking case study,” *CVIU, 2008*, vol. 1–2, pp. 74–86, Oct.–Nov. 2007.

[65] S. Grossberg, “How does the cerebral cortex work? development, learning, attention, and 3D vision by laminar circuits of visual cortex,” *Spatial Vis.*, vol. 12, pp. 163–187, 2003.

Vadim Tikhanoff received the B.Sc. degree in computer science from the University of Essex, Essex, U.K., in 2004, presenting a study based on artificial intelligence and, in particular, multiagent systems. He received the M.Sc. degree in interactive intelligent systems in 2005, presenting a work in interactive entertainment robotics, and the Ph.D. degree in 2009, both from the University of Plymouth, Plymouth, U.K. His thesis focused on the development of cognitive capabilities in humanoid robots.

He currently holds a Postdoctoral position at the Italian Institute of Technology, Genoa, Italy, working in the Robotics, Brain, and Cognitive Sciences Laboratory. He has a background in artificial intelligence and robotic systems, and is now focusing on the development of innovative techniques and approaches for the design of skills in a robot to interact with the surrounding physical world and manipulate objects in an adaptive, productive, and efficient manner.

Angelo Cangelosi received the M.Sc. degree in experimental psychology from the University of Rome, Rome, Italy, in 1991, and received the Ph.D. degree in psychology and cognitive modeling from the University of Genoa, Genoa, Italy, in 1997, while also working as a visiting scholar at the National Research Council, Rome, the University of California San Diego, CA, and the University of Southampton, U.K.

He is currently a Professor of Artificial Intelligence and Cognition at the University of Plymouth, U.K., where he leads the Centre for Robotics and Neural Systems. He has produced more than 140 scientific publications, and has been awarded numerous research grants from the United Kingdom and international funding agencies for a total of over 10 million Euros. The books he has edited include *Simulating the Evolution of Language* (2002, Springer, coedited with D. Parisi) and *Emergence of Communication and Language* (2007, Springer; coedited with C. Lyon and C. L. Nehaniv). He is coordinator of the EU FP7 project ITALK: Integration and Transfer of Action and Language Knowledge in Robot, a multipartner research project on action and language learning on the humanoid robot iCub. He also coordinates the RobotDoC Marie Curie doctoral network in developmental robotics, and the U.K. Cognitive Systems Foresight project VALUE, cofunded by EPSRC, ESRC, and BBSRC.

Dr. Cangelosi is Co-Editor-in-Chief of the *Interaction Studies Journal* and serves on the editorial board of the *Journal of Neurolinguistics*, *Connection Science*, and *Frontiers in Neurorobotics*. He has chaired various international conferences, including the 6th International Conference on the Evolution of Language (Rome, Italy, 2006) and the 9th Neural Computation and Psychology Workshop (Plymouth, U.K., 2004). He has been invited/keynote speaker at numerous conferences and workshops.

Giorgio Metta received the M.Sc. degree (Honors) in 1994, and the Ph.D. degree in 2000, both in electronic engineering, from the University of Genoa, Genoa, Italy.

From 2001 to 2002, he was a Postdoctoral Associate with the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, where he worked on various humanoid robotic platforms. He has been an Assistant Professor with the Dipartimento di Informatica Sistemistica e Telematica, University of Genoa, since 2005, where he has been teaching courses on anthropomorphic robotics and intelligent systems. Since 2006, he has also been a Senior Scientist with the Robotics, Brain, and Cognitive Sciences Department, Italian Institute of Technology, Genoa, Italy. His current research interests include biologically motivated humanoid robotics and, in particular, the development of artificial systems that show some of the abilities of natural systems. His research has been developed in collaboration with leading European and international scientists from different disciplines like neuroscience, psychology, and robotics. He is the author of more than 100 publications. He has also been engaged as a Principal Investigator and Research Scientist in several international and national funded projects. He has been reviewer of international journals and the journals of European Commission. He is also one of the leaders of the ROBOTCUB project, which resulted in the design and production of the humanoid robot iCub.