

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

Journal:	<i>Transactions on Autonomous Mental Development</i>
Manuscript ID:	TAMD-2010-0057
Manuscript Type:	Regular
Date Submitted by the Author:	21-May-2010
Complete List of Authors:	Tikhanoff, Vadim; Italian Institute of Technology, Robotics, Brain and Cognitive Sciences Cangelosi, Angelo; University of Plymouth, Centre for Robotics and Neural Systems Metta, Giorgio; Italian Institute of Technology, Robotics, Brain and Cognitive Sciences
Keywords:	Artificial Intelligence, Cognitive Robotics, Manipulation, Speech Recognition



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

V. Tikhanoff, A. Cangelosi and G. 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 bio-inspired 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 [2-8]. 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.

Manuscript received October, 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 Behavior & Cognition research lab at the University of Plymouth, UK. He is now with the Italian Institute of Technology, Genoa, IT, (e-mail: vadim.tikhanoff@iit.it)

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

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

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 [9-14]. 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 [10, 15-18]. Amongst diverse solutions to the programming of robots' capabilities such as attention sharing, turn-taking and social regulation [19, 20], a major effort in developmental robotics has currently focused on imitation. A considerable amount of research has been conducted in order to achieve imitating intentional agents [21-25]. 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 [30].

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 [6, 26-31]. For example, Steels [28] 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 [6] and object manipulation tasks [29-31], 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 (multi-agent systems, artificial life, and robotics). There are many developmental robotic models involved in speech learning, such as the development of vocabulary and grammar [32, 33]. 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

1 language in sensorimotor categories [34-38]. Computational
2 models of language have, in the last few decades, focused on
3 the idea of a symbolic explanation of linguistic meaning [39-
4 41]. Using this symbolic approach, word meanings are
5 defined in terms of other symbols, leading to circular
6 definitions [42, 43]. However, there has recently been an
7 increased focus on symbol grounding approach, i.e. on the
8 important process of “grounding” the agent’s lexicon directly
9 to its own representation of the interaction with the world.
10 Agents learn to name entities, individuals and states, whilst
11 they interact with the world and build sensorimotor
12 representations of it. Language grounding models provide a
13 new route for modeling complex cross-modal phenomena
14 arising in situated and embodied language use. As early
15 language acquisition is overwhelmingly concerned with
16 objects and activities, which occur in a child’s immediate
17 surrounding environment, these models are of a significant
18 interest for understanding situated language acquisition in
19 developmental robotics.

22 As cognitive systems research is increasingly based on
23 robotics platforms, it is important to consider the contribution
24 of simulations in developmental robotics research. Robot
25 simulators have recently become an essential tool in the
26 design and programming of robotic platforms, whether for
27 industry or research [45-47]. Furthermore, these robotic
28 simulators have had a significant role in cognitive research,
29 where they have proven to be critical for the development and
30 demonstration of many algorithms and techniques (such as
31 path planning algorithms, grasp planning, and mobile robot
32 navigation).

34 This paper proposes a new approach to the design of a
35 robotic system that is able to take advantage of all the
36 functionalities that a humanoid robot such as the iCub robotic
37 platform [13, 44] provides. This work will focus on object
38 manipulation capabilities, where refined motor control is
39 integrated with speech “understanding” capabilities. The
40 paper describes cognitive experiments carried out on the iCub
41 simulator [45, 46]. More specifically, the research focuses on
42 a fully instantiated system integrating perception and
43 learning, capable of interacting and communicating in the
44 virtual (simulated) and real world and performing goal
45 directed tasks. This system allows a tighter integration
46 between the representation of the peripersonal space (tactile,
47 proprioceptive, visual and motor) and the ability to move
48 different effectors. In particular, the goal is to develop a
49 controller that learns to use the available effectors to solve
50 cognitive tasks, potentially by transferring and generalizing
51 already acquired skills. Cognitive experiments will focus on
52 the humanoid iCub robot with vision, touch, audition, and
53 proprioceptive sensorial abilities.

54 Section II provides a detailed description of the
55 development of the iCub simulator used for the experiments.

The cognitive experiments are then presented within the
following two sections. 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.

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 [47]. 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 [48]. 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 [49]. Another advantage is that robotic
simulations make it possible to apply particular algorithms
for creating robots’ controllers, such as evolutionary or
reinforcement learning algorithms [50]. 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 co-
evolution of the morphology and the control system [48]. 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.
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 [45-46]. The simulated iCub robot is composed of multiple rigid bodies connected via joint structures (see figure 1). 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.

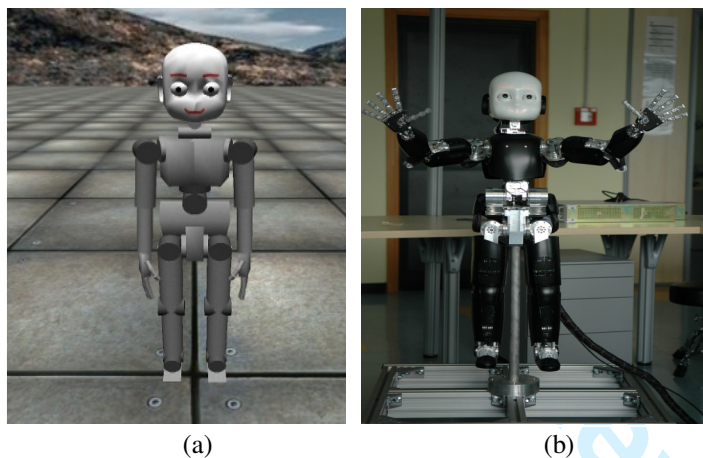


Fig 1. Photo the simulated iCub (a) and of the real iCub as of July 2009.

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 3D dynamic robot environment capable of recreating complex worlds and fully based on non-proprietary open source libraries.

B. Physics Engine

The iCub simulator uses ODE [51] (Open Dynamic Engine) 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...), terrains and meshes for complex object creation.

C. Communication Protocol

As the aim was to create an exact replica of the physical iCub robot, the same software infrastructure and inter-process

communication will have to be used as those used to control the physical robot. iCub uses YARP (Yet Another Robot Platform) [52, 53] 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 Figure 1. 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.

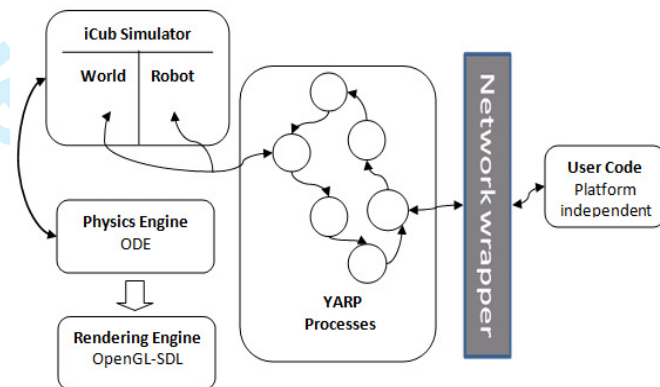


Fig. 2. Detail of the architecture of the simulator with YARP support.

E. The iCubBody 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 105cm, weighs approximately 20.3kg and has a total of 53 degrees of freedom (DoF). These include 12 controlled DoFs for the legs, 3 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.

1 However the simulated robot torque parameters and their
2 verification in static or motion are a good basis and can be
3 proven to be reliable [54].

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

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

14 III. MOTOR CONTROL LEARNING

15 A. Introduction

16 This section proposes a method for teaching a robot how to
17 reach for an object that is placed in front of it and then to
18 attempt to grasp the object. The first part of the work focuses
19 on solving the task of reaching for an object in the robot's
20 peripersonal environment. This employs a control system
21 consisting of an artificial neural network configured as a
22 feed-forward controller [55]. The second part of the motor
23 learning model incorporates the above reaching module
24 within an additional controller needed for the robot to
25 actually grasp the object. This employs another control system
26 consisting of a neural controller configured as a Jordan
27 Neural Network [56].

28 B. Learning to Reach

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

40 In computational neuroscience, research on reaching has
41 focused on the development of neuro-cognitive models of
42 human behavior, that can also be employed in humanoid
43 robots to achieve human-like reaching [57, 58]. Additionally,
44 neuroscience research considers the issue of pre-grasping as
45 defined by Arbib and colleagues [59]. This deals with the
46 configuration of the fingers for successful grasping, whilst
47 performing the reaching movement. These finger
48 configurations must satisfy some form of pre-defined
49 knowledge on the object affordances for appropriate grasping,
50 and pre-defined knowledge about the task to accomplish.
51 However, the model presented in this paper is not concerned
52 with generating a reaching system consistent with human

models of pre-grasping, but assumes that reaching and
grasping can be performed independently [60, 61].

Current research on humanoid robot manipulation [62] has
considered the reaching problem without dealing in depth
with the grasping issue. Issues such as grasping, friction and
the mechanics behind it are typically not taken into
consideration, and use reaching for pointing and touching
only. Other robotics models of reaching [63] have
implemented reaching by using a path planner with some
obstacle avoidance procedure. Kagami and colleagues [64]
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 [64], current models have simplified the problem of
reaching to a greater extent, 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 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
[65, 66]. Metta and colleagues [66] 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 [67]. Even with the addition
of the eye vergence, there were some limitations due to errors
in the hand positioning. In an earlier paper, Marjanovic and
colleagues [68] 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 [69-71]. 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. Unfortunately, in humanoid
robots it is not possible to assume that the hand will be
constantly visible during object manipulation. More recently,
Gaskett and colleagues [65] have successfully implemented a
system that uses 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 loses track of the arm, they used a three dimensional
self-organizing map (SOM) [72] 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 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. A feed-forward multilayer perceptron, with back propagation algorithm [73] was modeled to simulate reaching for diverse objects that reside within their surroundings. The neural network architecture as depicted in Figure 3 was used.

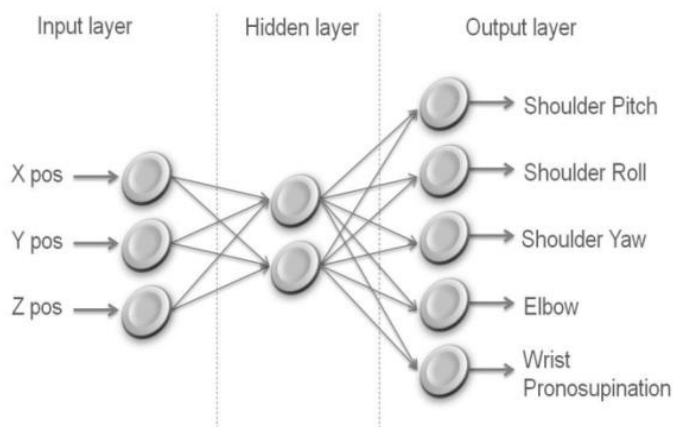


Fig. 3. The architecture of the employed feed-forward neural network

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 [74], and depth estimation [75, 76]. 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 4 shows an example of 150 positions of the endpoints of the robot hands used during training, by representing them as green squares.

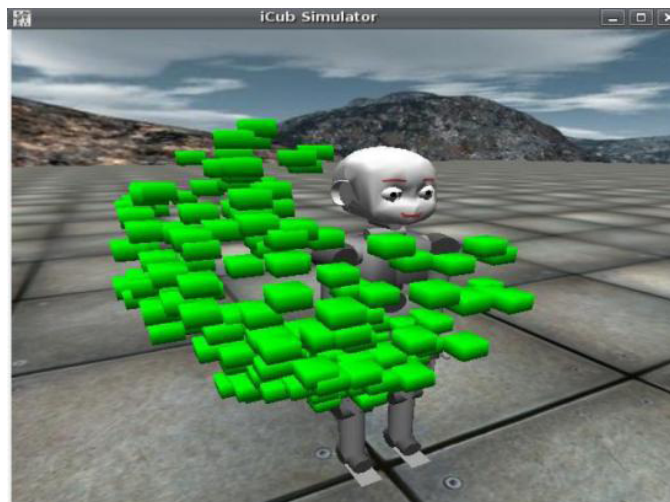


Fig. 4. Example of the 150 end positions of the robot arms during training.

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

Learn Size	Test Size	Total	Num Iterations	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 final RMSE (root mean squared error) ranged from 0.15 to 0.16 (e.g. sample training curve figure 5). Although low, an RMSE of 0.15 indicates that the neural network was not able to fully learn the task.

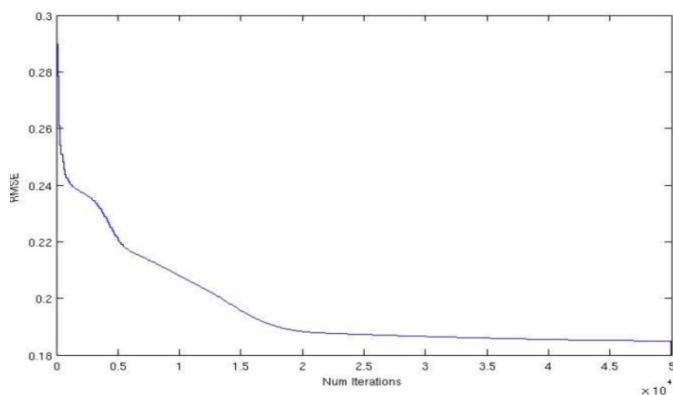


Fig. 5. RMSE value during training of the reaching module

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 6. Figure 6 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).

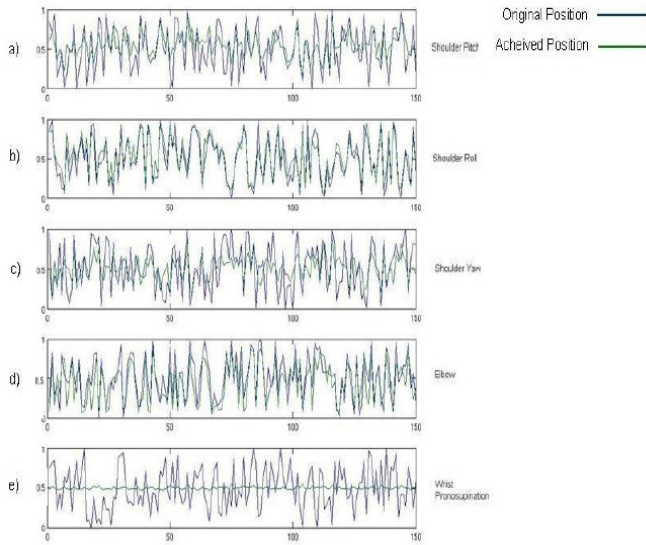


Fig. 6. The first 150 results of the 2,500 samples given to the network. Each graph represents the different joint degrees at each of the 150 positions. a) shoulder pitch – b) shoulder roll – c) shoulder yaw – d) elbow – e) wrist pronosupination

The reason for such a high RMSE is 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 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. The second factor contributing to the error 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 generalization tests showed that the model was capable of successfully locating and tracking the object in new positions, and finally reaching the target. Figure 7 is a collection of images taken after the detection of the object (by the vision system) and the attempt to reach the tracked object.

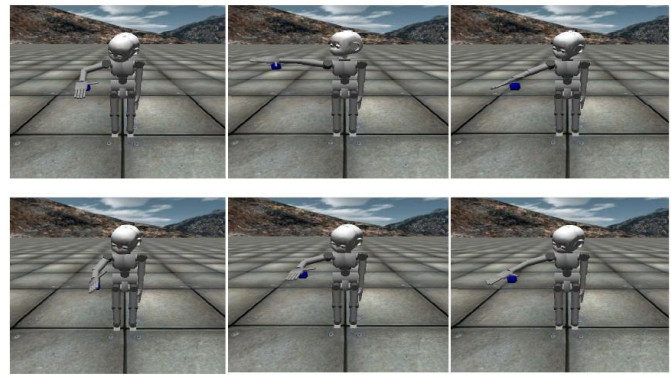


Fig. 7. Images taken from the robot during the testing of the reaching module

Figure 8 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.

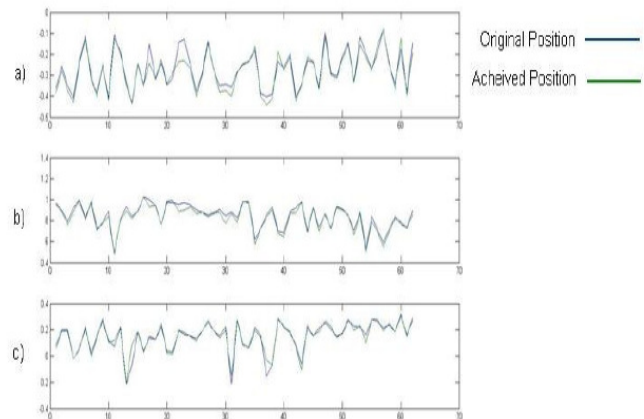


Fig. 8. Comparison of 62 random XYZ positions of objects, with the actual resulting position of the robot's hand

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, 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 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 ([77]):

- 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 [78-80].

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 pre-grasping, 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 [81, 82]. 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 easily achieved. Finally, the sensory driven grasping approach tries to solve the previously mentioned problems by using learning and task exploration [83, 84].

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 [85, 86]. More recently, Carezzi and colleagues [87] developed neural network models which are able to learn 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 model of the iCub grasping, 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 approach was proposed by Jordan [56], who 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 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 figure 9.

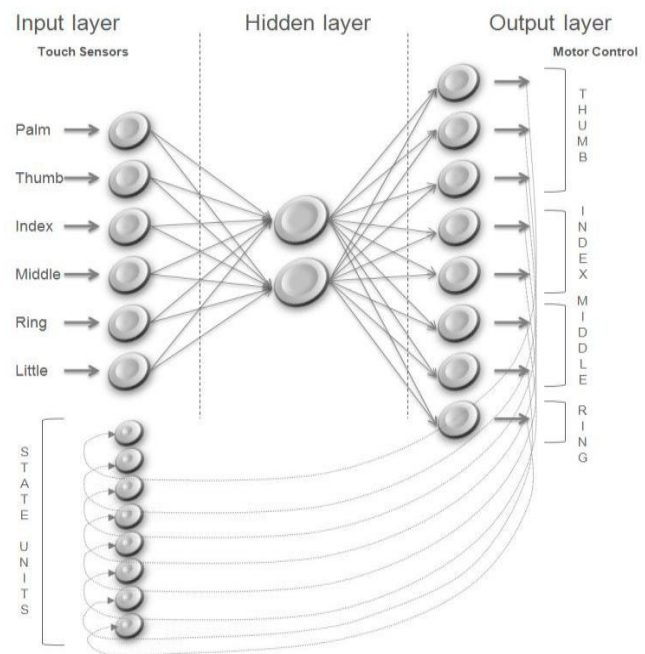


Fig. 9. The architecture of the employed Jordan Neural Network

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 8 finger joints, which are located on the hand of the robot. The hidden layer comprises 5 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 figure 10 and a detailed description of the hand joints used can be seen in table III.

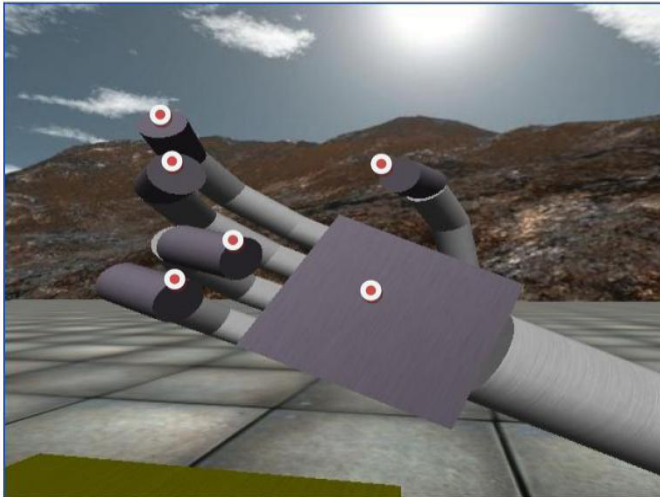


Fig. 10. Location of the six touch sensors on the iCub's simulator hand

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 (object) that triggers 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 front-back movement	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 Jordan neural network is achieved online and therefore no training patterns have been pre-defined 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 algorithm (ARP) is implemented in order to train the network connection weights. A description of this algorithm can be found in [88]. 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.

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 figures 11 and 12 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 figure 13. The object size parameters (in meters) are:

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

The object was then modified to a cube with parameters:

- 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 $z = 0.04$.

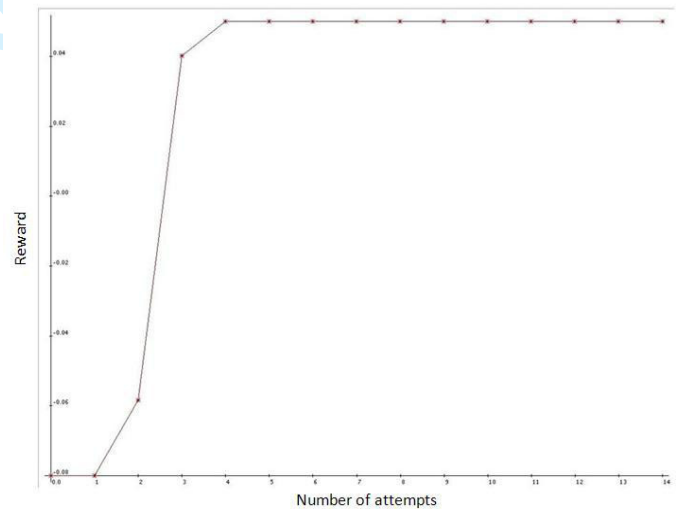


Fig. 11. The reward rate during the grasping neural network training phase

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 13 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).

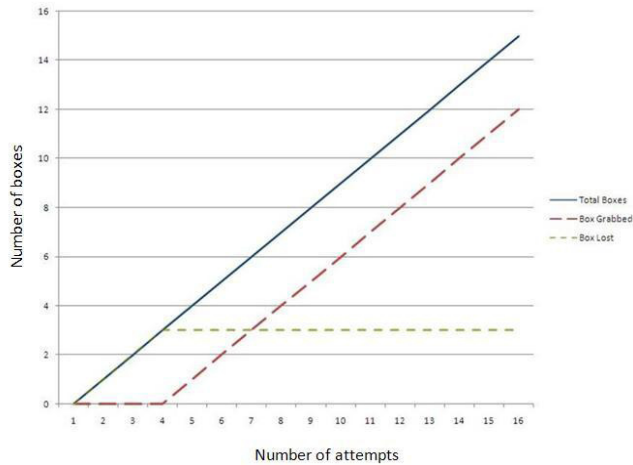


Fig. 12. Graph showing the total boxes used (Red), total boxes grabbed (Yellow), and total boxes lost (Green), during a simple grasping experiment with an object of specific size.

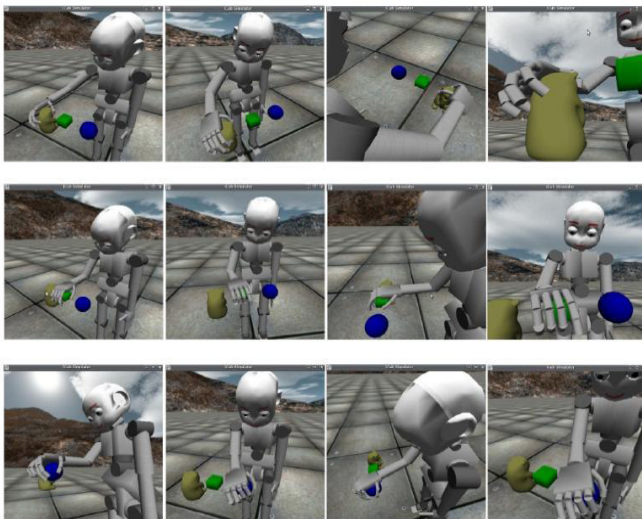


Fig. 13. Grasping of three different objects

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 [89-91]. 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 [92] 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 [93]. Brent and Siskind [94] 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.

B. Speech Recognition

The speech recognizer system developed at Carnegie Mellon University was used [95, 96]. 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 [96]).

- 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 sub-segment

- 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 sub segment

- 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.

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 pre-recorded samples. The left hand side of Figure 14 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 re-use of the module on different platforms and hardware, as well as allowing the reproduction of experiments from recordings. Figure 14 gives an overview of the global architecture entailing both the Sphinx module and the YARP audio interface.

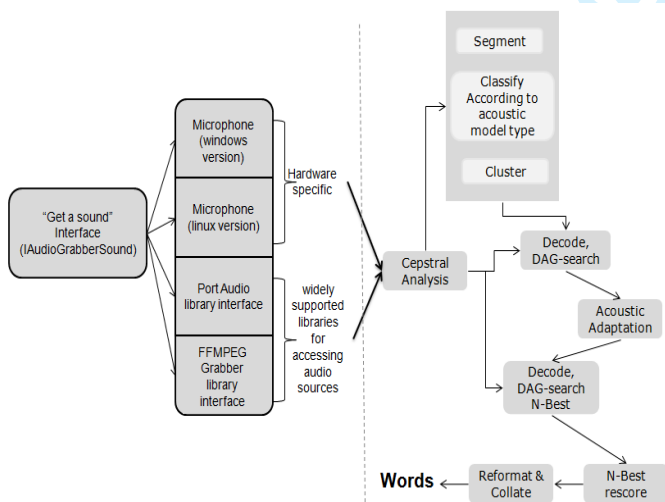


Fig. 14. The architecture of the integration of YARP and Sphinx

E. Learning to Integrate Speech and Action

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. 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. The hidden layer comprises fifteen units. The neural network's architecture can be seen in figure 15. 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.

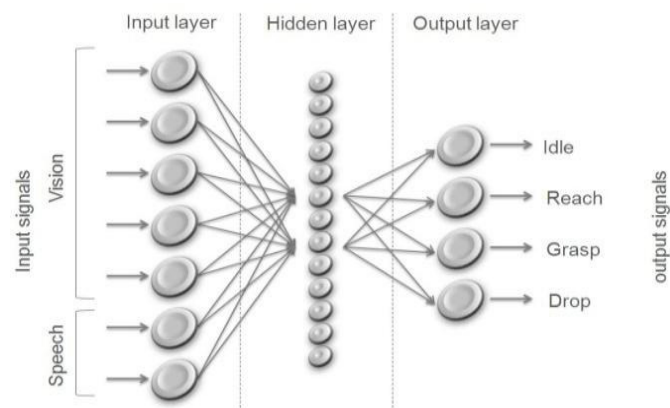


Fig. 15. The Goal Selection Neural Network architecture used

"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 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 (figure 16).

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

--	--	--	--	--	--

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

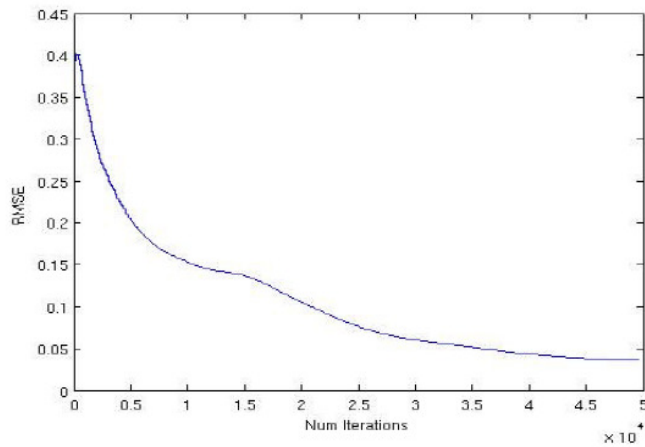


Fig 16. RMSE value during training of the goal selection module

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 17.

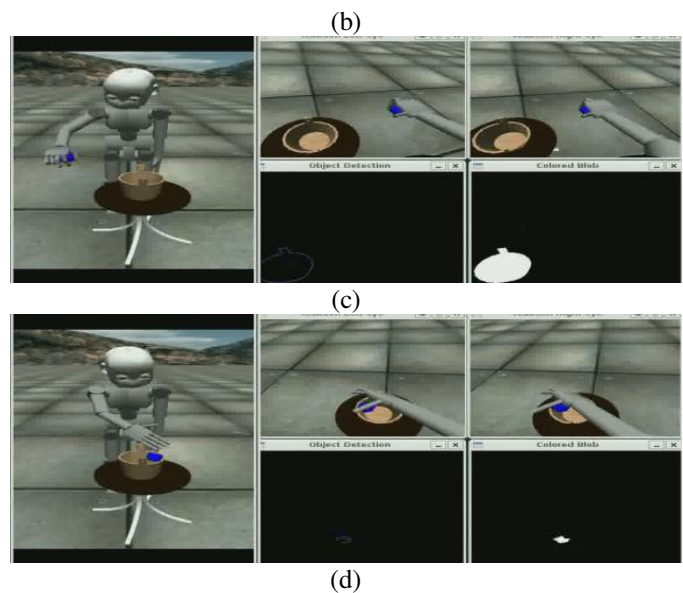
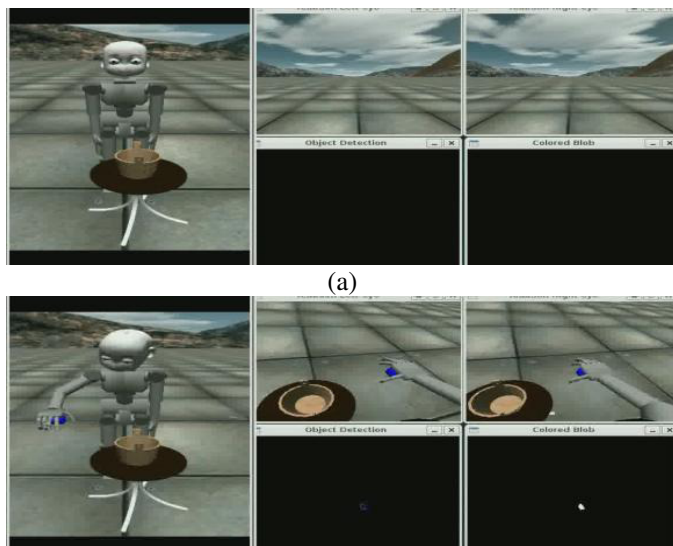


Fig. 17. Selection of images showing: (a) the setup of the cognitive experiment; (b) the input of the linguistic command; (c) the reaching and grasping of the blue box; (d) the dropping of the blue box

The 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 [97], 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 [98]. 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.

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 spatiotemporally coherent visual surface [99]. 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 spatiotemporal 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, re-configuring 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 [99].

The propensity to attend and segment spatio-temporally coherent visual surfaces (objects) yields significant benefits in terms of object classification. Classifying a pre-segmented 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 pre-scaling and auto-centering 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 [99, 100]. 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] Sperber, D. and L. Hirschfeld, *Culture Cognition and Evolution*, in *MIT Encyclopedia of the Cognitive Sciences*, R. Wilson and F. Keil, Editors. 1999, MIT Press: Cambridge Mass. p. cxi-cxxxii.
- [2] Breazeal, C. and B. Scassellati, *Infant-like Social Interactions Between a Robot and a Human Caretaker*. *Adaptive Behavior*, 2000. **8**(1): p. 49-74.
- [3] Breazeal, C. and B. Scassellati, *Robots that imitate humans* *Trends in Cognitive Sciences*, 2002. **6**(11): p. 481-487.
- [4] Brooks, R., et al. *Alternative essences of intelligence. Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*. 1998.
- [5] Brooks, R., *Robot: The Future of Flesh and Machines* 2002: Allen Lane/Penguin.
- [6] Cangelosi, A., *Evolution of communication and language using signals, symbols, and words*. *IEEE Transactions on Evolutionary Computation*, 2001. **5**(2): p. 93 - 101.
- [7] Fong, T., C. Thorpe, and C. Baur, *Robot, asker of questions* *Robotics and Autonomous Systems*, 2003. **42**(3-4): p. 235-243.
- [8] Dautenhahn, K. and A. Billard. *Studying robot social cognition within a developmental psychology framework*. in *Third European Workshop on Advanced Mobile Robots*. 1999.
- [9] Asada, M., K.F. MacDorman, H. Ishiguro and Y. Kuniyoshi, *Cognitive developmental robotics as a new paradigm for the design of humanoid robots* *Robotics and Autonomous Systems*, 2001. **37**(2): p. 185-193.
- [10] Lungarella, M. and R. Pfeifer. *Robot as a cognitive tool: an information theoretic analysis of sensory-motor data*. in *2nd IEEE-RAS International Conference on Humanoid Robotics*. 2001. Tokyo, Japan.
- [11] Lungarella, M., G. Metta, R. Pfeifer and G. Sandini, *Developmental robotics: a survey* *Connection Science*, 2003. **15**(4): p. 151-190.
- [12] Metta, G., G. Sandini, L. Natale and F. Panerai. *Development and robotics*. in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*. 2001.
- [13] Metta, G., et al. *The RobotCub project an open framework for research in embodied cognition*. in *International Conference of Humanoids Robotics. Workshop on Dynamic Intelligence*. 2005. Tokyo, Japan.
- [14] Pfeifer, R., *Robots as cognitive tools* *International Journal of Cognition and Technology*, 2002. **1**(1): p. 125-143.
- [15] Balkenius, C., et al. *Modeling Cognitive development in Robotics Systems. First International Workshop on Epigenetic Robotics*. 2001. Lund University.
- [16] Prince, G. and Y. Demiris, *Editorial: introduction to the special issue on Epigenetic robotics*. *Adaptive Behavior*, 2003. **11**(2): p. 75-77.
- [17] Weng, J., W.S. Hwang, Y. Zhang, C. Yang and R.J. Smith *Developmental Humanoids: Humanoids that develop skills automatically. Proceedings of the 1st IEEE-RAS Conference on Humanoid*. 2000. Beijing China.
- [18] Zlatev, J. and C. Balkenius. *Introduction: Why 'epigenetic robotics'*. in *Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*. 2001. Lund University.
- [19] Dautenhahn, K., *Socially intelligent robots: dimensions of human-robot interaction*. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 2007. **362**(1480): p. 679-704.
- [20] Fong, T., I. Nourbakhsh, and K. Dautenhahn, *A survey of socially interactive robots*. *Robotics and Autonomous Systems*, 2003. **42**(3-4): p. 143-166.
- [21] Bakker, P. and Y. Kuniyoshi. *Robot See, Robot Do : An Overview of Robot Imitation*. in *AISB Workshop on Learning in Robots and Animals*. 1996. Brighton, UK.

- [22] Jansen, B. and T. Belpaeme. *A model for inferring intention in imitation tasks. The 15th IEEE International Symposium on Robot and Human Interactive Communication*. 2006. Hatfield UK.
- [23] Meltzoff, A., *Elements of a developmental theory of imitation*, in *The Imitative Mind*, A. Meltzoff and W. Prinz, Editors. 2002, Cambridge University Press: New York. p. 19-141.
- [24] Nadel, J., *Imitation and Imitation Recognition: Functional use in Preverbal infants and Nonverbal children with autism*, in *The Imitative Mind*, A. Meltzoff and W. Prinz, Editors. 2000, Cambridge University Press: Cambridge. p. 42-62.
- [25] Scassellati, B. *Knowing What to Imitate and Knowing When You Succeed. Proceedings of the AISB'99 Symposium on Imitation in Animals and Artifacts* 1999. Edinburgh, Scotland.
- [26] Kirby, S., *Learning, Bottlenecks and the Evolution of Recursive Syntax*, in *Linguistic Evolution through Language Acquisition: Formal and Computational Models*, T. Briscoe, Editor. 2002, Cambridge University Press: Cambridge.
- [27] MacWhinney, R., *Models of the Emergence of Language*. Annual review of psychology, 1998. **49**: p. 199-227.
- [28] Steels, L., *Evolving grounded communication for robots*. Trends in Cognitive Sciences, 2003. **7**(7): p. 308-312.
- [29] Cangelosi, A., E. Hourdakis, and V. Tikhanoff. *Language acquisition and symbol grounding transfer with neural networks and cognitive*. in *International Joint Conference on Neural Networks (IJCNN06)*. 2006. Vancouver.
- [30] Cangelosi, A. and T. Riga, *An Embodied Model for Sensorimotor Grounding and Grounding Transfer*. Cognitive Science, 2006. **30**(4): p. 673-689.
- [31] Marocco, D., A. Cangelosi, and S. Nolfi, *The emergence of Communication in evolutionary robots*. Philosophical transactions of the Royal Society of London. Series A 2003. **361**: p. 2397-2421.
- [32] Roy, D., *Learning visually grounded words and syntax of natural spoken language*. Evolution of Communication, 2002. **4**(1): p. 33-56.
- [33] Steels, L., *Self-organising vocabularies*, in *Artificial Life V*, C.G. Langton and K. Shimohara, Editors. 1996, MIT Press. p. 179-184.
- [34] Cangelosi, A., *The emergence of language: Neural adaptive agent models*. Connection Science, 2005. **17**(3-4): p. 185-190.
- [35] Cangelosi, A., G. Bugmann, and R. Borisjuk, *Modeling Language, Cognition And Action: Proceedings of the Ninth Neural Computation and Psychology Workshop*. 2004, Plymouth UK: World Scientific.
- [36] Pecher, D. and R. Zwaan, *Grounding Cognition: The role of perception and action in memory, language and thinking*. 2005, Cambridge: Cambridge University Press.
- [37] Plunkett, K., C. Sinha, MF. Møller and O. Strandsby, *Symbol grounding or the emergence of symbols? Vocabulary growth in children and a connectionist net*. Cognitive Science, 1992. **4**: p. 293-312.
- [38] Roy, D. and N. Mukherjee, *Towards situated speech understanding: visual context priming of language models* Computer Speech and Language, 2005. **19**(2): p. 227-248.
- [39] Kintsch, W., *Comprehension: A paradigm for cognition*. 1998, Cambridge: Cambridge University Press.
- [40] Miller, G.A., *Wordnet: a lexical database for english community*. Communications of the ACM, 1995. **38**(11): p. 39 - 41.
- [41] Simon, H., *Physical symbol systems*. Cognitive Science, 1980. **4**: p. 135-183.
- [42] Harnad, S., *The Symbol Grounding Problem*. Physica D 1990. **42**: p. 335-346.
- [43] Roy, D., *Scemiotic Schemas: a framework for grounding language in action and perception*. Artificial Intelligence, 2005. **167**(1-2): p. 170-205.
- [44] Sandini, G., 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, et al., Editors. 2007, Springer-Verlag Berlin Heidelberg: Festschrift. p. 359-370.
- [45] Tikhanoff, V., P. Fitzpatrick, F. Nori, L. Natale, G. Metta and A. Cangelosi. *The iCub humanoid robot simulator. International Conference on Intelligent Robots and Systems (IROS)*. 2008. Nice, France.
- [46] Tikhanoff, V., 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 *IEEE Workshop on Performance Metrics for Intelligent Systems* 2008. Washington.
- [47] Ziemke, T., *On the role of robot simulations in embodied cognitive science*. AISB Journal, 2003. **1**(4): p. 389-399.
- [48] Bongard, J.C. and R. Pfeifer, *Evolving complete agents using artificial ontogeny*, in *Morpho-functional machines: The new species (designing embodied intelligence)*, F. Hara and R. Pfeifer, Editors. 2003, Springer-Verlag: Berlin. p. 237-258.
- [49] Kumar, S. and P.J. Bentley, *On growth, form and computers*. 2003: Elsevier Academic Press Amsterdam.
- [50] Nolfi, S., et al., *How to evolve autonomous robots: different approaches in evolutionary robotics*, in *Artificial Life IV*, R. Brooks and P. Maes, Editors. 2000, MIT Press: Cambridge.
- [51] Smith, R. *Open Dynamic Engine*. 2001 [cited October 2005 - present]; Available from: <http://www.ode.org/>.
- [52] Metta, G., P. Fitzpatrick, and L. Natale, *YARP: Yet Another Robot Platform*. International Journal of Advanced Robotics Systems, special issue on Software Development and Integration in Robotics, 2006. **3**(1): p. 43-48.
- [53] Fitzpatrick, P., G. Metta, and L. Natale, *Towards long lived genes*. Robotics and Autonomous systems, 2008. **56**(6): p. 29-45.
- [54] Nava, N., et al. *Kinematic and Dynamic Simulations for the design of RobotCub upper body Structure. Engineering systems design and analysis conference ESDA*. 2008. Israel.
- [55] Rumelhart, D.E., G.E. Hinton, and J.L. McClelland, *A General Framework for Parallel Distributed Processing*, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D.E. Rumelhart and J.L. McClelland, Editors. 1986, MIT Press: Cambridge MA.
- [56] Jordan, M., *Serial Order: A Parallel Distributed Processing Approach*. 1986, California University - Institute for Cognitive Science: San Diego. p.64.
- [57] Bullock, F., S. Grossberg, and F. Guenther, *A self-organizing neural model of motor equivalent reaching and tool use by a multi-joint arm*. Journal of Cognitive Neuroscience, 1993. **5**(4): p. 408-435.
- [58] Crowe, A., J. Porrill, and T. Prescott, *Kinematic coordination of reach and balance*. Journal of motor behavior, 1998. **30**(3): p. 217-233.
- [59] Arbib, M.A., T. Iberall, and D. Lyons, *Coordinated control programs for movements of the hand*. Experimental brain research, 1985. **10**: p. 111-129.
- [60] Mason, M., *Mechanics of robotic manipulation*. 2001, Cambridge Massachusetts: MIT Press.
- [61] Okada, K., A. Haneda, H. Nakai, M. Inaba and H. Inoue. *Environment manipulation planner for humanoid robots using task graph that generates action sequence. IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2004. Japan.
- [62] Lavelle, S.M., *Planning algorithms*. 2006: Cambridge University Press.
- [63] Brock, O. and O. Khatib, *Elastic strips: A framework for motion generation in human environments*. International Journal of Robotics Research, 2002. **21**(12): p. 1031-1052.
- [64] Kagami, S., et al. *Humanoid arm motion planning using stereo vision and RRT search. Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2003. Las Vegas USA.
- [65] Gaskett, C. and G. Cheng. *Online Learning of a Motor Map for Humanoid Robot Reaching. Proceedings of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003)*. 2003. Singapore.
- [66] Metta, G., G. Sandini, and J. Konczak, *A developmental approach to visually-guided reaching in artificial systems*. Neural Networks 1999. **12**(10): p. 1413-1427.
- [67] Metta, G., F. Panerai, R. Manzotti, and G. Sandini *Babybot: an artificial developing robotic agent. 6th International Conference on the simulation of Adaptive behaviour*, 2000. Paris.
- [68] Marjanovic, M., B. Scassellati, and M. Williamson. *Self taught visually guided pointing for a humanoid robot*. in *In Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB-96)*. 1996.
- [69] Cooperstock, J.R. and E.E. Milius, *Self-supervised learning for docking and target reaching*. Robotics and Autonomous Systems, 1993. **11**(3-4): p. 243-260.
- [70] Ritter, H., T. Martinetz, and K. Schulten, *Neural computation and self-organizing maps: an introduction*. 1992, New York: Addison-Wesley Longman Publishing.
- [71] Walter, J. and K. Schulten, *Implementation of self-organising neural networks for visuo-motor control of an industrial robots*. IEEE Transactions on Neural Networks, 1993. **4**(1): p. 86-95.
- [72] Kohonen, T., *Self-Organizing Maps*. Vol. 3. 1995: Springer.

- [73] Rumelhart, D.E., G.E. Hinton, and R. Williams, J. *Learning representation by back-propagating errors*. *Nature*, 1986. **323**: p. 533-536.
- [74] Lewis, J. P. (1995). Fast Template Matching. *Vision Interface*, 120-123.
- [75] Birchfield, S., & Tomasi, C. (1999). Depth Discontinuities by Pixel-to-Pixel Stereo. *International Journal of Computer Vision*, 35(3), 269-293.
- [76] Qian, N. (1997). Binocular disparity and the perception of depth. *Neuron*, 18, 359-368.
- [77] Rezzoug, N. and P. Gorce, *Robotic Grasping: A generic Neural Network Architecture*, in *Mobile Robots Towards New Applications*, A. Lazinica, Editor. 2006, Pro Literatur Verlag Robert Mayer-Scholz: Germany.
- [78] Bekey, G.A., et al., *Knowledge-based control of grasping in robot hands using heuristics from human motor skills*. *IEEE Transactions on Robotics and Automation*, 1993. **9**(6): p. 709-722.
- [79] Iberall, T., *Human prehension and dexterous robot hands*. *International Journal of Robotics Research*, 1997. **16**(3): p. 285-299.
- [80] Saito, F. and K. Nagata. *Interpretation of grasp and manipulation based on grasping surfaces*. *IEEE International Conference on Robotics and Automation*. 1999. Detroit, MI.
- [81] Gorce, P. and J. Fontaine, *Design methodology for flexible grippers*. *Journal of Intelligent and Robotic Systems*, 1996. **15**(3): p. 307-328.
- [82] Miller, A. and P. Allen. *Examples of 3D grasps quality measures*. in *Proceedings of the IEEE International Conference on Robotics and Automation*. 1999.
- [83] Grupen, R. and J. Coelho, *Acquiring State form Control Dynamics to learn grasping policies for Robot hands*. *International Journal on Advanced Robotics*, 2002. **15**(5): p. 427-444.
- [84] Wheeler, D.S., A.H. Fagg, and R.A. Grupen. *Learning prospective pick and place behavior*. *Proceedings of the 2nd International Conference on Developmental and Learning*. 2002.
- [85] Moussa, M. and M. kamel, *An Experimental approach to robotic grasping using a connectionist architecture and generic grasping functions*. *IEEE Transactions on System Man and Cybernetics*, 1998. **28**(2): p. 239 - 253.
- [86] Taha, Z., R. Brown, and D. Wright, *Modelling and simulation of the hand grasping using neural networks*. *Medical Engineering and Physics*, 1997. **19**(6): p. 536-538.
- [87] Carenzi, F., P. Gorce, Y. Burnod, and M. Maier. *Using generic neural networks in the control and prediction of grasp postures*. in *13th European Symposium on Artificial Neural Networks ESANN 2005*. 2005. Bruges, Belgium.
- [88] Barto, A.G. and M. Jordan. *Gradient following without back-propagation in layered networks*. in *IEEE First Annual Conference on Neural Networks*. 1987.
- [89] Perzanowski, D., A.C. Schultz, and W. Adams. *Integrating natural language and gesture in a robotics domain*. *Proceedings held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS) Intelligent Control (ISIC)*. 1998.
- [90] Spiliotopoulos, D., I. Androutopoulos, and C. Spyropoulos. *Human-robot Interaction based on spoken natural language dialogue*. *European Workshop on Service and Humanoid Robots*. 2001.
- [91] Varchacskaia, P., P. Fitzpatrick, and C. Breazeal. *Characterizing and Processing Robot-Directed Speech*. *Proceedings of the International IEEE/RSJ Conference on Humanoid Robotics*. 2001. Tokyo, Japan.
- [92] Zue, V., S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen and L. Hetherington, *Jupiter: A telephone based conversational interface for weather information*. *IEEE Transaction on Speech and Audio Processing*, 2000. **8**: p. 100-112.
- [93] Werker, J.F., 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, Editors. 1996, Lawrence Erlbaum Associates: Mahwah, NJ. p. 427-447.
- [94] Brent, M. and J. Siskind, *The role of exposure to isolated words in early vocabulary development*. *Cognition*, 2001. **81**: p. B33-B44.
- [95] Lee, K., H. Hon, and R. Reddy, *An overview of the SPHINX Speech Recognition System*. *IEEE Transaction on Acoustics, Speech and Signal Processing*, 1990: p. 35-45.
- [96] Placeway, P., et al. *The 1996 Hub-4 Sphinx-3 System*. in *ARPA Speech Recognition Workshop*. 1997.
- [97] Jusczyk, P.W., *How infants begin to extract words from speech* *Trends in Cognitive Sciences*, 1999. **3**(9): p. 323-328.
- [98] Meltzoff, A. and M. Moore, *Explaining facial imitation: a theoretical model*. *Early development and parenting*, 1997. **6**(3-4): p. 179-192.
- [99] Dankers, A., N. Barnes, and A. Zelinsky, *MAP ZDF segmentation and tacking using active stereo vision: hand tracking case study*. *CVIU*, 2008. **1-2**(October-November 2007): p. 74-86.
- [100] Grossberg, S., *How Does the Cerebral Cortex Work? Development, Learning, Attention, and 3D Vision by Laminar Circuits of Visual Cortex*. *Spatial Vision*, 2003. **12**: p. 163-187.