

UNIVERSITÀ DEGLI STUDI DI NAPOLI  
FEDERICO II



DOTTORATO DI RICERCA IN  
SCIENZE COMPUTAZIONALI ED INFORMATICHE  
CICLO XXVI

A hierarchical visuo-motor  
organization for computational models  
of the mirror system

PH.D. DIRECTOR:  
**G. MOSCARIELLO**

PH.D. STUDENT:  
**GUGLIELMO MONTONE**

SCIENTIFIC ADVISOR:  
**R. PREVETE**

TUTOR:  
**P. FESTA**

**Acronymus index:**

<b>CNS</b>	Central Nervous System
<b>DOF</b>	Degrees of Freedom
<b>EEG</b>	Electroencephalography
<b>EMG</b>	Electromyography
<b>FFNN</b>	Feed-Forward Network
<b>fMRI</b>	functional Magnetic Resonance Imaging
<b>HVM</b>	Hierarchical Visuo-Motor architecture
<b>IPL</b>	Inferior Parietal Lobe
<b>MN</b>	Mirror Neuron
<b>MDN</b>	Mixture Density Network
<b>MEP</b>	Motor-Evoked Potentials
<b>PCA</b>	Principal Component Analysis
<b>RMS</b>	Root-Mean-Square error
<b>STS</b>	Superior Temporal Sulcus
<b>TPS</b>	Temporal Postural Synergy
<b>TSSM</b>	Tree-Structured Synergies Method

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Background and motivations . . . . .	3
1.2	The proposed approach . . . . .	5
1.3	Overview . . . . .	7
<b>2</b>	<b>Object-directed action representation in the motor cortex</b>	<b>9</b>
2.1	Biological findings . . . . .	10
2.1.1	The motor cortex . . . . .	10
2.1.2	Mirror neurons . . . . .	11
2.1.3	Speculations on mirror neurons functionality . . . . .	19
2.2	Action representation and control . . . . .	24
2.2.1	Motor synergies in the brain . . . . .	24
<b>3</b>	<b>Computational models of mirror neurons and action representation</b>	<b>27</b>
3.1	Mirror Models . . . . .	27
3.2	Action representation . . . . .	31
3.2.1	Motor synergies in the hand and grasping actions . . . . .	32
3.2.2	Indirect synergy studies on hand action . . . . .	34
<b>4</b>	<b>Hierarchical temporal postural synergies</b>	<b>39</b>
4.1	Tree-structured synergies method . . . . .	40
4.2	Dataset collection . . . . .	42
4.3	Tree-structure and PCA representation . . . . .	44
4.4	Recruiting the original dictionary . . . . .	44
4.5	TSSM representation capacity . . . . .	48
4.6	Usage, Commonality, Selectivity . . . . .	50
<b>5</b>	<b>Hierarchical Visuo-Motor architecture</b>	<b>55</b>
5.1	Biological systems and HVM architecture . . . . .	55
5.2	Visual and motor representation . . . . .	57

5.3	General overview of the architecture . . . . .	58
5.3.1	When performing an action . . . . .	58
5.3.2	When observing an action . . . . .	59
5.4	Non-functional mapping block . . . . .	62
5.4.1	Mixture density network . . . . .	62
5.4.2	Mixture density network module . . . . .	64
5.4.3	Spatial congruence module . . . . .	66
5.5	Temporal congruence module . . . . .	67
5.6	Strictly and broadly neurons . . . . .	68
<b>6</b>	<b>Tests and results</b>	<b>71</b>
6.1	Details of the dataset . . . . .	71
6.2	Motor data codify . . . . .	73
6.3	PCA and Tree motor representations . . . . .	74
6.4	Non-functional visuo-motor mapping . . . . .	75
6.4.1	K-means test . . . . .	76
6.4.2	Feed Forward mapping . . . . .	77
6.4.3	The Mixture density network . . . . .	83
6.4.4	The ambiguity is described but not solved . . . . .	88
6.5	Motor involvement . . . . .	89
6.6	Broadly and strictly neurons . . . . .	92
<b>7</b>	<b>Conclutions and future work</b>	<b>95</b>
7.1	Contribution of this work . . . . .	95
7.2	Open questions and future work . . . . .	97
<b>A</b>	<b>Pricipal Component Analysis</b>	<b>99</b>
A.1	Theoretical background . . . . .	99
A.2	PCA motor representaion . . . . .	100
<b>B</b>	<b>Tree-Structured Synergy Method</b>	<b>103</b>
B.1	Tree-Structured Stage . . . . .	104
B.2	Synergy Dictionary Stage . . . . .	105
<b>C</b>	<b>Mixture Density Network</b>	<b>107</b>
C.1	Mixture Density Network . . . . .	108

# Chapter 1

## Introduction

### 1.1 Background and motivations

Mirror neurons (MN)s were first discovered in the motor cortex of macaque monkeys. It was a serendipity discovery, Rizzolatti and his group during the '90 were studying the *motor areas* of monkeys, i.e. cortex areas mainly involved in controlling monkey movement, by realizing single cell recording on a monkey performing grip actions. It happens that, during the experiments, one of the scientist grasped one of the objects in front of the monkeys eliciting activity of the recorded cell. Thus MNs are motor neurons that present the intriguing property of activate not only when the monkey is performing an action, such as grasping an object using a power grip, but also when the same or a similar action is just observed.

MNs were originally found in the ventral premotor cortex and in the intraparietal sulcus of monkey brains(Di Pellegrino et al., 1992; Gallese et al., 1996), but there is now a big evidence that these neurons are even present in the human brain (Molenberghs et al., 2012). Since their discovery a large number of speculations have been done on the possible functional role for these neurons in a lot of cognitive processes ranging from manual communication (Rizzolatti et al., 1996) to language processing (Rizzolatti and Arbib, 1998) to intention reading (Iacoboni et al., 2005) and empathy (Avenanti et al., 2005). All these studies were even associated with a big number of researches intended to characterize the mirror neurons and, more in general, all the areas of the brain involved in visual processing of motor acts(Murata et al., 2000; Matelli and Luppino, 2001). Despite these studies some fundamental questions about mirror neurons still remain unanswered. It is not clear, in fact, if MNs and, more in general, motor cortex actually take part to the process of visual elaboration and eventually what could be their func-

tional role in this process.

The main target of this thesis is realizing a computational model of visuo-motor interaction in the mirror neurons system. We decide to tackle this main problem by pursuing two subgoals that are:

- investigating a possible action representation in the motor cortex;
- investigating the role of the motor representation in the process of visual analysis and interpretation.

The topic of our first sub-goal has been bringing a lot of interest in the scientific community. In particular different studies were developed in order to investigate the hypothesis of a synergy action representation in the motor cortex (d'Avella et al., 2006). The computational approach to this problem has shown to be particularly profitable. As in fact different algorithms for data representation were successfully applied to represent actions in terms of specific patterns in muscles activities or movement kinematics/dynamics (Thakur et al., 2008). Some works have shown that hand actions can be efficiently represented by linear combinations of postural synergies, with the coefficients varying overtime (Mason et al., 2001). Some other proposed to efficiently approximate action with linear over-positions of temporal postural synergies (Santello et al., 2002; Vinjamuri et al., 2010a).

More recently another interesting aspect of action representation seem to appear. Has been suggested (Iacoboni et al., 2005; Hamilton and Grafton, 2008) that in the brain there would be areas coding action at different levels of detail. With some area responding according to action kinematic details, areas grabbing more general characteristics of the action, and even areas that seem to codify action according to its goal/outcome. These hierarchical organization of action representation seem present even for the codify of observed actions. In fact, as reported in some works by Rizzolatti and colleagues (Rizzolatti and Sinigaglia, 2010) MNs differently respond to an observed action: different groups of neurons spikes each according to different action details, form very specific details to more general ones. The plausibility of a hierarchical organized synergy representation of action was investigated for the first time in some works realized in our laboratory, bringing promising results (Tessitore et al., 2013; Amico, 2011). The first part of this thesis consisted in improving these studies on a much bigger set of data, developing new tests on a bench of actions composed of hand grasping actions chosen among those used for mirror neurons experiments (Gallese et al., 1996).

The second subgoal of the thesis, investigating the role of motor cortex in the visual process, aroused interest in the scientific community since the discovery of MNs and the consequent statement of the Rizzolatti's *direct matching*

*hypothesis* (Rizzolatti et al., 2001). This hypothesis in fact suggests a strong, if not a central, role for the motor cortex in the process of visual elaboration of actions. According to this view the recognition of an action would consist in the capacity of the motor system of the observer to "resonate" when looking at an action present in the observer motor repertoire. Using the words of Rizzolatti: "the motor knowledge of the observer is used to understand the observed action. In other words, we understand an action because the motor representation of that action is activated in our brain" (Rizzolatti et al., 2001). The possibility that MNs would develop an important part in the visual elaboration of observed actions appear plausible but is not convincing the whole scientific community. In fact in a recent paper Cook and colleagues (Cook et al., 2013) suggested that MNs would not be a system realized in order to facilitate visual processing, but instead would be the result of general-domain processes of sensorimotor associative learning. Therefore not necessary involved in visual elaboration. Cook and colleagues stress in particular the high non-specificity response of the MNs (some MNs are view dependent, some are not, some spikes depending on whether the action is executed with the left or the right hand, if the action take place in the peri-personal or extra-personal space, etc.), that would be unconvincing for a system specifically intended for action recognition. The current computational models of MNs (Haruno et al., 2001; Ito and Tani, 2004; Oztop and Arbib, 2002) are frequently functionally uninformative in describing the possible role of motor cortex in the process of visual elaboration. A lot of models in fact develop architecture that receive the same input when an action is observed or executed, where these inputs are the results of processes that do not involve the motor system. Other models seem instead descriptively inadequate, in particular not taking in any account the big non-specificity of the MNs. For example in these works are realized architectures where MNs activate in the same way during action execution and observation.

## 1.2 The proposed approach

The first part of the work described in this thesis was intended to realize a particular kind of action representation. In our representation action was described as a linear combination of temporal postural synergies hierarchically organized. Temporal postural synergies (TPS)s are specific patterns in the space of motor configurations varying over time. The novelty of our approach consists in that we used an algorithm for action representation that induce a structure in the TPSs. In particular the algorithm associated each of the TPSs with the nodes of a tree structure. The representation obtained was

such that the synergies associated with nodes near to the root of the tree resulted to be used to represent more than one action, thus codifying more general action properties. While the synergies more near to the leafs, used to represent few kind of actions, were codifying actions details. This research, to the best of our knowledge, represent one of the first indirect prove for a hierarchical synergy representation of action in the brain.

In the second part of the thesis the realization of the Hierarchical Visuo-Motor (HVM) architecture is described. HVM, modeling some characteristics of the MNs, would propose a mechanisms through which the motor knowledge, represented according to the previous action representation, could be part of the process of visual elaboration. MNs activate both during action execution and observation. To reproduce this behaviour the HVM architecture was equipped with an its own motor repertoire and the motor representation of action was used for codify both an executed and an observed action. This was realized by projecting the visual representation of action into the relative motor representation. This visuo-motor mapping presents non common difficulties due to the non-functional relation among the visual input and the motor representation. This problem was overcome by using a particular kind of artificial neural network, the mixture density network, specifically intended to model non-functional relations. Another important characteristics of the architecture is that the visuo-motor mapping is organized according to different levels of abstraction. In this way, modeling another characteristic of the biological system, in our architecture the same action is codified according to different levels of detail both when the action is executed and when is observed.

As said before one of the principal target of this thesis was investigating the role of the motor representation in the process of visual analysis and interpretation. In other words, one of the contribution of this work was constituting a clue in favor of the Rizzolatti's direct matching hypothesis. To this end, in our architecture, we realized a biologically plausible mechanisms that provides for an important role of the motor representation in the process of visual elaboration. In the HVM architecture the exploitation of some characteristics of the motor representation resulted in an improvement of the quality of the visuo-motor mapping. Our system, besides giving a functional role to the motor cortex in the visual process, result to be even a descriptively adequate model of the MNs. In fact our architecture, as we will show, is able to explain one of the main characteristics of the MNs, namely their classification in strictly and broadly neurons.



## 1.3 Overview

In *Chapter 2* we will propose a detailed review of the biological findings about the mirror neurons. We will first give a general overview of the main motor areas involved in the viso-motor mapping, then we will characterize the three neural circuits where mirror neurons have been found. We will moreover review the recent debate on the mirror neurons functionality, in particular focusing on the direct matching hypothesis and the role of these neurons in goal action codify. We will end the chapter by presenting some recent results of biological studies on the way in which action is codified in the motor cortex. Some of these results suggesting a multiple level details action representation. Some other instead underlining the possible synergy codify of actions.

In *Chapter 3* we will propose a review of the main computational model on the mirror systems. We will discuss the main strength and weakness of the first models of the mirror neurons, like the ones from Arbib and Oztop, Ito and Tani (Ito and Tani, 2004; Oztop and Arbib, 2002). We will then presents some very new models like the ones of Friston and Chersi (Kilner et al., 2007; Chersi et al., 2011). We will then give an insight into the main works that investigate, from a computational point of view, the plausibility of a synergy representation of action in the central nervous system. In these work a precise definition of synergy is proposed and the efficiency of a synergy action representation is proven.

In *Chapter 4* we will introduce our representation of action in terms of temporal postural synergies hierarchically organized. We will first introduce the algorithm that we used in order to find such a hierarchical representation. The algorithm was first tested on a synthetic dataset, the kind of test and the results are shown in this chapter. We will then present our action dataset, describing in detail how data were collected and pre-processed. Once collected these data were represented as an overposition of temporal postural synergies, the validity of these representation was tested and the results are presented in this chapter. Finally, in the last paragraph of the chapter, we develop some tests to show that the temporal postural synergy used for representing actions are actually hierarchically organized.

In *Chapter 5* we will introduce our HVM architecture. The first paragraphs are intended to stress how some biological results on mirror neurons and action representation are modeled in our system. In these paragraphs we will in particular focus on how action codify is used in order to represent

an executed or an observed action. In the central paragraphs of the chapter we will present in detail all the modules the architecture is constituted of. The first module, the mixture density module, actually solves the problem of describing the non-functional visuo-motor mapping among visual and motor representation of action. In the second and the third module, the spatial and temporal congruence modules, is implemented the mechanisms that using the architecture motor knowledge actually improve the ability to associate to a visual representation of an action its relative motor representation. In the last paragraph of the chapter is described the capacity of our architecture to model the different behaviour of the strictly and broadly neurons.

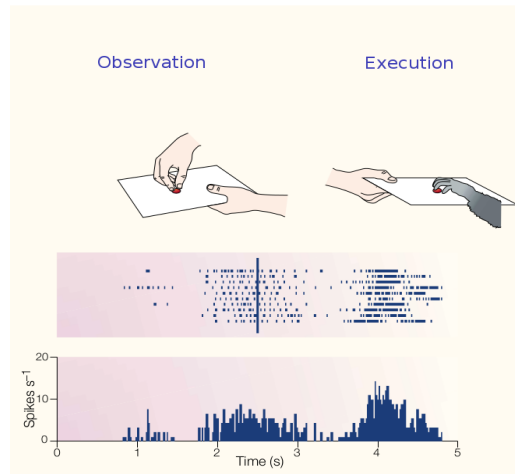
In *Chapter 6* all the tests realized on HVM architecture and the relative results are presented. In the first paragraph we will describe the dataset used to train and test the different modules of our architecture. Different test were made to actually prove the non-functional character of the visuo-motor mapping. These tests and the results are illustrated in the fourth paragraph of the chapter. In the fifth paragraph a test is realized in order to show that using its motor repertoire, HVM architecture, can actually improve the visuo-motor mapping. In the last paragraph of the chapter we realize a test to show the ability of the HVM to model strictly and broadly neurons.

In *Chapter 7* we summarize the results obtained and propose possible future developments.

## Chapter 2

# Object-directed action representation in the motor cortex

In this chapter we will review the main biological findings relative to mirror neurons and, more in general, to object directed action representation in the motor cortex. Mirror neurons were discovered in the '80 by Rizzolatti and his colleagues. These neurons, found in the monkeys motor cortex, activate both when the monkey is executing an action or when is observing someone else developing the same action. In figure 2.1 the typical response of a mirror neuron is shown. In the first part of the chapter we will propose a review of the main characteristics of these neurons and of the cortex areas strickly related. More reacent research has shown as mirror neurons are present even in other species and possibly in humans. The particular behaviour of these neurons has brought a lot of speculations on their role in the process of visual elaboration of action. In fact is not clear if mirror neurons, and more general motor cortex, could be part of system used in the visual processing of action and in what this involvement could actually consist. Always in the first of the chapter we will review some of the most debated issues in this respect. In the second part of the chapter we summurize some studies investigating the way in which action is coded in the motor cortex. We will in particular show some results suggesting a synergy representation of action in the motor cortex. These synergies are patter muscles activities or movement kinematics/dynamics. Thier use in action codify could help the Central Nervous System (CNS) in contolling motor effectors like limbs or hand, that are frequently compex system with a lof of degrees of freedom.

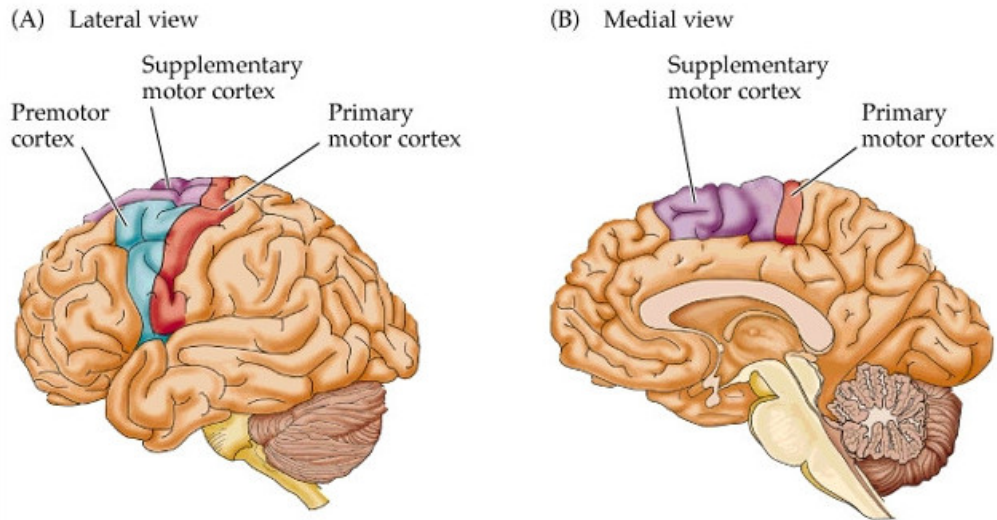


**Figure 2.1:** Recording on a mirror neuron when the monkey is observing an action executed by the experimenter or is itself developing the action. Picture taken from (Rizzolatti et al., 2001)

## 2.1 Biological findings

### 2.1.1 The motor cortex

Pianification, execution, control and recognition of actions are functions that involve a very big portion of the brain. Among the areas involved, the ones that are more used can be identified in the motor cortex, the associative motor cortex, the dorsal and ventral paths. The motor cortex is composed of the *primary motor cortex*, the *supplementary motor area* and the *premotor cortex*. The primary motor cortex is located in the precentral gyrus, see Figure 2.2. Has been shown (Penfield Rasmussen 1950) that the primary motor cortex is somatotopically organized, i.e. the activation of specific areas of this cortex bring to the contraction of specific muscles. The main input to the primary motor cortex come from the supplementary motor area and the premotor cortex. The supplementary motor area is located in the medial surface of the cortex, rostrally located respect to the primary cortex (Figure 2.2). It mainly receives inputs from the parietal and temporal cortexes. It is involved in the execution of simple motion plans like pushing and then turning a bar, or executing a sequence of dance movements. Some studies on the human seem to suggest that this area is involved in storing the future movement in a sequence. The premotor cortex is located in the lateral surface of the cortex, rostrally located respect to the primary motor cortex (Figure 2.2). It seems mainly involved in the elaboration of arbitrary stimuli associated to



**Figure 2.2:** Motor cortex areas.

the execution of a particular action. For arbitrary stimuli we mean stimuli that are not directly correlated to the movement they suggest. This area is for example strongly involved when monkeys are trained to move their hands in accordance with the color of a light. Interestingly one of the areas that shows mirror characteristics, named area F5, is located in the rostral part of the premotor cortex. Even this area receives input from the temporal and parietal cortex. In particular the inferior temporal cortex includes some zones of the cortex that are frequently referred to as the "what" system or the ventral path. The visual information initially received in the occipital cortex is elaborated in a path that ends in the temporal cortex. This path seems to be related to recognition of shapes and objects. A second path starting in the occipital cortex ends in the posterior parietal cortex, this is known as the dorsal path or the "where" system. In this path the visual information is elaborated in order to extract information on the spatial location of an object.

### 2.1.2 Mirror neurons

The mirror neurons were originally found in the monkey brain (*Macaca nemestrina* and *Macaca mulatta*). The seminal work of Giacomo Rizzolatti and

co-workers (Rizzolatti et al., 1996; Gallese et al., 1996) bring to the discovery of these neurons in area F5 of the ventral premotor cortex (PMC) and in the inferior parietal lobule (IPL). More recent studies have found neurons with mirror characteristics even in the primary motor cortex and dorsal premotor cortex (Dushanova and Donoghue, 1996; Tkach et al., 2007).

Evidence of the presence of mirror neurons were found even in humans. The areas that presents strongly sensorimotor matching properties are the 'classical' ones, i.e. inferior frontal gyrus (the human homologue of monkey F5 area) (Kilner et al., 2009) and the inferior parietal cortex (Chong et al., 2008), but even the dorsal primary motor cortex, supplementary motor area, media temporal lobe and superior parietal lobule (Mukamel et al., 2010).

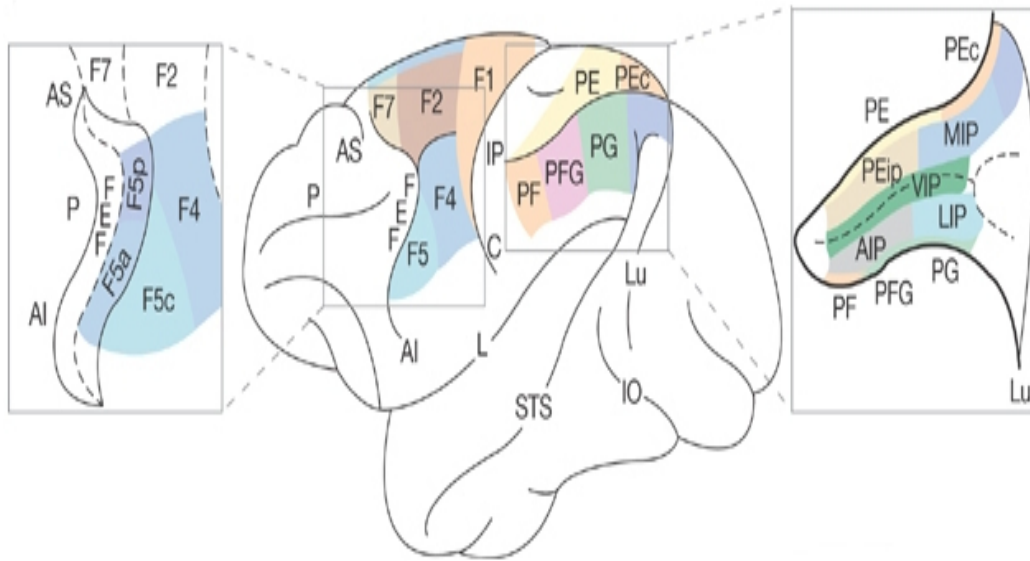
In the next two subsections we will describe the results of the studies on mirror neurons for monkeys and humans focusing mainly on the classical mirror zone. The studies on monkeys allowed to precisely characterize the different zones involved in the visuo-motor transformations describing the behaviour of the single neurons in these zones and hypothesizing their functional role. The results of these studies are reported in the first subsection. The studies on human were mainly directed to verify with different techniques the presence of zones with mirror properties similar to the ones found in monkeys. The results of these studies are reported in the second subsection.

## **Mirror neurons in monkeys**

The possibility to realize invasive measures on monkeys has allowed a deep investigation of the behaviour of mirror neurons and of the areas strictly related. As we said in the previous paragraphs the regions more involved in the visuo-motor transformations are the ventral premotor area and the parietal lobule. Different studies (Matelli and Luppino, 2001) suggest the presence of a *parieto-frontal circuits*. Three of these circuits seem more involved in the development of object directed actions and were well identified and studied: the *VIP-F4*, the *AIP-F5ab* and the *PF-F5c* circuits, see Figure 2.3. The areas F1, F4 and F5 are motor areas, PF and PFG are in the posterior parietal cortex, VIP and AIP in the intra-parietal sulcus. The F5 area is composed of two main regions, F5c is located in the dorsal convexity and the F5ab located on the posterior bank of the inferior arcuate sulcus.

## **Parieto-frontal circuits**

Area VIP is a zone of the intraparietal sulcus and receives visual and sensory motor information. The neurons of this area can be classified as *purely visual* (unimodal) and *visual and tactile* (bimodal). Neurons of the first class are

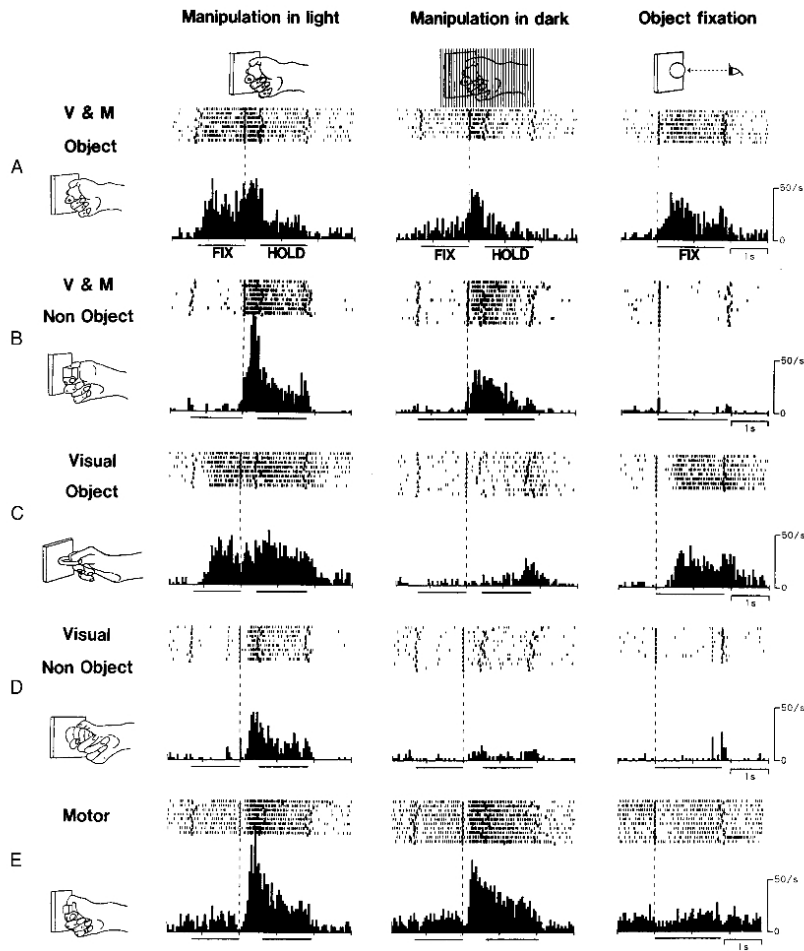


**Figure 2.3:** Main areas involved in the mirror system. Picture taken from (Matelli and Luppino, 2001).

selective to visual stimuli mainly presented in the peri-personal space, the receptive field of these neurons is usually egocentric and not in retinal coordinate. Bimodal neurons respond to visual and tactile stimuli. Interestingly the visual and tactile receptive fields of these neurons are overlapped. For example there are neurons that spike both when the arm of the monkeys is touched by the experimenter or when the monkey see something near its arm. VIP area is strongly connected to F4 motor area. The neurons of these area codify movements of arms, neck, face, mouth. Many of the neurons of this area spike during movement directed towards or away from the body and do not respond to distal movements (Rizzolatti et al., 1998). The area AIP belonging to the AIP-F5ab circuit is located in the intraparietal sulcus. Neurons of this area are mainly responsive to actions of reaching and grasping objects and their activity is mainly due to hands and fingers movement. The neurons of this regions can be divided in three classes: *visual-dominant*, *motor-dominant* and *visual and motor*. The visual-dominant are so called because they activate when the monkey observe a grasping action, while they remain silent when the monkey performs the grasping. The motor-dominant neurons present the opposite behaviour, responding when the action is executed and not responding when is observed. Finally the visuo-motor dominant respond

in both situations. In all of these three classes of neurons there are cells that spike just to the sight of a graspable object, even when this is not followed by a grasping action. Interestingly the neurons that spike when an object is presented are the ones that spike even in accordance to actions that are possible actions to interact with the object. This means that, for example, a neuron that responds when the monkey is gazing a small object will discharge also during precision grip actions, which are action used to grasp small object. In Figure 2.4 are reported the results of measures realized on the AIP neurons where the just described behaviours are well represented. The F5ab area receives afferent connections from area AIP. The neurons of these area are mainly associated to movements of hands and mouth. Most of them are selective for one of the most common grip types of the monkey independently on how the action is specifically performed, some other instead codify a specific way of performing one of the class action. Rizzolatti and colleagues (Rizzolatti et al., 1988) performed a study on the temporal relation of this neurons discharge with the action execution, showing that some neurons fire during the last part of grasping, others start to fire at finger aperture and continue during finger closure, other are activated in advance of the onset of the finger movement. Rizzolatti and Gentilucci (Rizzolatti and Gentilucci, 1988) suggested that functional properties of F5ab neurons could be a way to store a 'vocabulary' of motor acts, where some neuron populations would indicate just the general action category and some other would specify with bigger details the movement of finger and arm to perform a particular grip. More recently a study of Fogassi (Fogassi et al., 2005) focusing on area F5 and PF investigated how this motor vocabulary are then organized in order to code a whole action. The experiment were organized so that the monkey has to perform two grasping acts that share some parts of the movement and differ in some other parts and in the goal of the grasping (grasping for eating and grasping for placing). Interestingly the results showed that during grasping execution the discharge of the majority of these motor neurons was modulated by the final goal of the action. Some neurons discharged stronger during grasping for eating, others during grasping for placing even if the two actions were kinematically very similar. The remaining neurons did not show any modulation. Some researchers proposed that actions in the motor cortex are coded by neuronal chains. (Rizzolatti et al., 2006). The same motor act involved in two different actions could be in fact represented in the motor cortex by two different neuronal populations belonging to two different motor chains.





**Figure 2.4:** AIP neurons behaviour (picture taken from (Murata et al., 2000))  
 The *visual and motor* neurons shown in A and B panels respond when object manipulation is executed in light or in dark condition, some of them shown in A panel respond even to the sight of the object. *Visual-dominant* in panels C and D respond when object manipulation is realized in light even visual-dominant can be divided into some that respond to object sight and some that do not. Finally *motor-dominant* respond exclusively during object manipulation.

## F5 visuo-motor neurons

Some of the F5 neurons have also visual property. These visuo-motor neurons are of two types: *canonical neurons* and *mirror neurons*. Canonical neurons are mainly located in F5ab while mirror neurons mainly in F5c. The canonical neurons spikes both when the monkey performs a grasp action or when a graspable object is shown. Often there is a congruence between the action coded by a given canonical neuron and the observed object that elicit an activity. Frequently, in fact the object that stimulates a canonical neuron is an object on which the action codified by the neuron can be applied to. The mirror neurons respond both when the monkey executes an object-directed action and when the monkey observes another individual (monkey or human) executing a similar action. Usually this neurons do not show any response when the action is mimicked by the experiment or when the action is an intransitive one (non-object directed). Mirror neurons can be divided in different classes depending on the degree of congruence between the observed and executed actions that elicit activity. In particular these neurons are divided in three classes: *strictly congruent*, *broadly congruent* and *non-congruent*. Strictly congruent neurons usually spikes when the executed and observed actions are of the same general kind (i.e. grasp) or are executed in the same way (i.e. precision grip). The broadly congruent are those neurons that spike when there is a similarity, but not a congruence between the executed and the observed action. These neurons can be further divided into three different kinds. The response of the first kind is very selective to motor execution, these neurons spikes just when an action belonging to a particular class is executed in a specific way. They are not so selective when observing. In fact they respond to any action of the same class of the action that elicits a spike when executed, independently of the precise way the action is performed. The second type of broadly is constituted of neurons less specific in term of motor activity then the previous ones. These neurons spike when any action of a specific class is executed. Their visual behaviour is even less selective in fact they respond to more then one class type of action. The last type of broadly congruent neurons seem to be selective to the 'goal' of the action observed, independently by the kind of action and if the action is executed or observed by the monkey. We will come back to this argument and will be more precise on the notion of 'goal' in the next paragraphs. The last kind of mirror neurons are the non-congruent. These exhibit a response with a no clear-cut relationship between the observed action and the executed action movement (Gallese et al., 1996). More recent studies on F5 have reveled two other important characteristics of the mirror neurons. The first study showed that the neurons of this zone are selective

Classification	Percentage
View Invariant	25% (51/201)
Multiple view preference	45% (89/201)
View preference at 180°	8% (15/201)
View preference at 90°	9% (18/201)
View preference at 0°	13% (27/201)

**Table 2.1:** Results of the measures of view invariance in F5 mirror neurons as reported in (Caggiano et al., 2011)

to the prospective under which a grasp action is observed, the second that these neurons are selective to peri-personal and extra-personal space. The first study mentioned (Caggiano et al., 2011) discovered that some of the neurons present in the F5 region are selective to the view point of action observation. Caggiano and colleagues measured the response of various mirror neurons when videos representing the same action under three different angles were shown to the monkey. They found that there are neurons which spike just when the action is observed under a particular angle, that the majority of the neurons analyzed were not view independent but use to spike for two of the three view angles and finally that, not the majority, but a big part of the neurons observed remained instead view invariant. Their results are presented in the table 2.1. The second study of Caggiano and colleagues found that (Caggiano et al., 2009) mirror neurons activity is "differentially modulated by the location in space of the observed motor acts relative to the monkey, with about half of them preferring either the monkeys peri-personal or extra-personal space. A portion of these spatially selective mirror neurons encode space according to a metric representation, whereas other neurons encode space in operational terms, changing their properties according to the possibility that the monkey will interact with the object"<sup>1</sup>.

### Mirror neurons in humans

Although no direct measures were made on human a lot of studies report indirect evidence of the presence of mirror neurons or of at least areas with mirror properties. The main results were obtained with the use of neuro-imaging and transcranial magnetic stimulation, but even with behavioral studies. functional Magnetic Resonance Imaging (fMRI)<sup>2</sup> studies identified

---

<sup>1</sup>Taken from: (Caggiano et al., 2009)

<sup>2</sup>fMRI is a neuro-imaging technique that measures brain activity by detecting associated changes in blood flow. It consists in measuring the electromagnetic waves emitted by the

premotor cortex and inferior parietal areas active during action execution and observation. These kind of experiments were proposed by different researchers, sometimes with different protocols all confirming this activity (Iacoboni et al., 1999; Vogt et al., 2007; Leslie et al., 2004). Positron emission tomography (PET)<sup>3</sup> studies shown that in superior temporal sulcus (STS) and in the inferior parietal cortex there was activity when the patients were observing actions. This activation was stronger when the action observed was a transitive one (Grafton et al., 1996). All this neuro-imaging studies seem to indicate a good consistency between the area of the monkey brain in which are present mirror neurons and the area of the human brain that activate during action execution and observation. One of the first experiment using transcranial magnetic stimulation (TMS)<sup>4</sup> was developed by Fadigà and his colleagues (Fadigà et al., 1995). They stimulate the left motor cortex of normal subject while were observing both transitive and intransitive actions of the arm. In the mean time the patients Motor-Evoked Potentials (MEP)<sup>5</sup> were recorded. The recorder values of MEP were compared with the ones measured while the patient were observing three-dimensional objects or performing a dimming-detection task, which is known to be particularly demanding on subject attention. The results clearly shown a selective increase in MEPs during the observation of goal-directed movements and of intransitive, meaningless arm movements. A more recent experiment shown that when TMS is applied to M1 area during passive action observation, the amplitude of the MEPs recorded from the muscles required to execute that action is greater than the amplitude of the MEPs recorded when observing a different action (Catmur et al., 2011).

The behavioral studies consisted mainly in observing the variations in the ability of imitating or executing an action after the observation of similar action performed by others. A typical experiment of this kind was develop by Brass and colleagues and is reported in (Brass et al., 2001). In this experiment the subject was instructed to perform two different index finger movement while watching a video. In the video was presented casually one

---

oxygen atoms present in the blood that were previously excited by a strong magnetic external field.

<sup>3</sup>PET is a nuclear imaging technique that measures the metabolic activity of a brain area by virtue of regional glucose uptake. The system detects pairs of gamma rays emitted indirectly by a positron-emitting radionuclide, which is introduced into the body on a biologically active molecule analogue of glucose

<sup>4</sup>TMS is a noninvasive method to cause depolarization or hyper-polarization in the neurons of the brain through the application of an external magnetic field

<sup>5</sup>MEP or motor-evoked potential is a measure of the impulse sent from the motor cortex to the muscles in order to elicit a contraction. It can be recorded with surface electrodes, which are placed over small hand muscles

of the two fingers action. The results clearly shown a pronounced reaction time advantage when the action performed and the observed one were compatible as compared to incompatible trials. This kinds of results together with some strong automatic imitation capacity when observing hand, arm and mouth movements were also considered by many researchers as evidence of a human mirror mechanisms.

### 2.1.3 Speculations on mirror neurons functionality

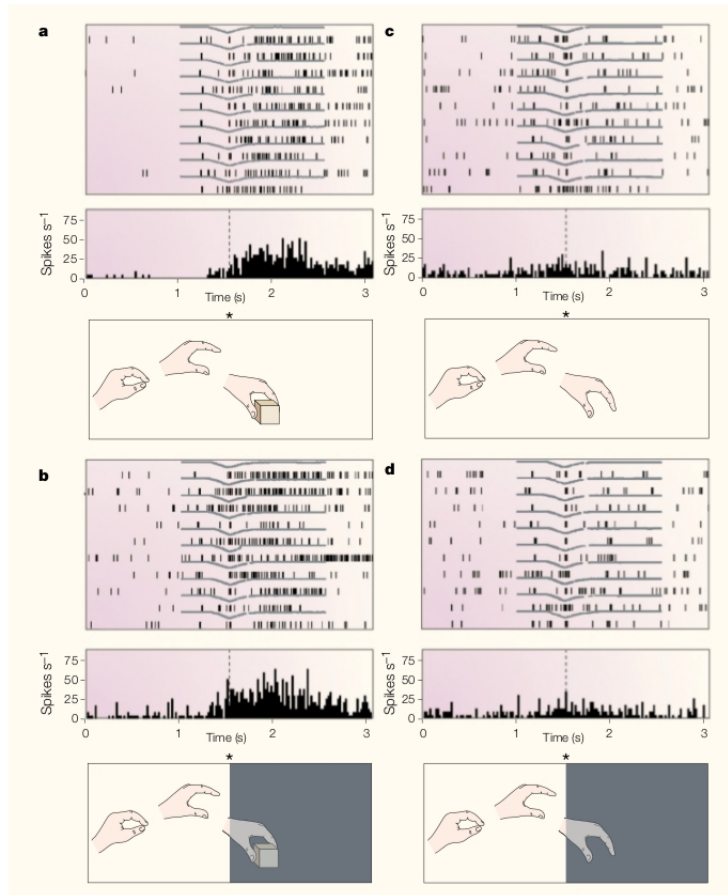
Since I dedicated all this chapter to describing the biological experimental results about the mirror neurons I would like to end by describing some more general ideas and implications that came from these finding and what kind of new questions these ideas implies. The discovery of mirror neurons system was accepted with great enthusiasm from the scientific community, since it was suddenly clear that it would have had a deep impact in the study of the brain. The mirror neurons discovery has in fact repercussions on many different problems in neuroscience, from language developing to imitation, from social behaviour to mind reading. In this paragraph I will focus on some of these repercussions, that I think are more significative for the study of the brain and more related to the scope of this thesis work. I will first focus on the Rizzolatti *direct matching hypothesis*, I will describe some of the biological results in support of this hypothesis and I will speak about the big repercussions this idea has on the development of computational models of mirror neurons.

Another very interesting aspect of the mirror neuron I will go through is their apparent involvement in the codify of action goals. I will account for some experiments suggesting this hypothesis, but I will also show that this is a quite debated point that has already not found its answer. Even this debate has strong implication in the construction of a biologically plausible computational model of the mirror system.

#### Direct matching hypothesis

Before the discovery of mirror neurons was believed that the process of visual elaboration of action, starting in the occipital lobe was completed in the two visual paths occurring in the temporal and parietal lobes were some internal description of the action was achieved. In fact, as we previously said, these areas are responsive to objects presentation, biological motion, interaction between hands and objects. I will refer to this hypothesis as the *visual hypothesis*. The *direct matching hypothesis* (Rizzolatti et al., 2001) states instead that achieving the internal representation of an observed action is

based on a process in which the observer maps the visual representation of the observed action onto his own motor representation of the same action. In this view the recognition of an action is more like the capacity of the motor system of the observer to *resonate* when the observer is looking at an action present in his own motor repertoire. The principal argument in favor of the visual hypothesis is the big degree of complexity and generality of the action representation as observed in some neurons of the superior temporal sulcus and in the inferior parietal lobe. In this respect the works of Perret and colleagues (Jellema et al., 2000; Perrett et al., 1990) showed the presence in STS of neurons that respond to goal directed hand action and even neurons that combine information about the direction of gaze of an agent with the action performed by that agent. These neurons become active when the monkey sees the reaching action, but only if the action is performed with the agent gaze directed to the intended target of reaching. Although these results some recent studies seem to suggest that the visual description is not a necessary condition to have a representation of action in the brain. A famous experiment by Umiltà and colleagues (Umiltà et al., 2001) was so realized. The F5 mirror neurons of the monkey were recorded in four different conditions. In the first condition the monkey could see the experimenter performing a grasping action, while in the second one the experiment was just mimicking the grasp. In the third condition the monkey was first shown the experiment beginning a grasping through an object, but then the object and the last part of the grasping were occluded to the monkey. In the fourth condition the monkey can see the experimenter starting a mimicking grasping action (the monkey could see that there was no object in front of the experimenter), but even this time could not see the last part of the movement. As expected in the first condition there was a big response of the mirror neuron, instead no response was observed in the second condition. Interestingly in the third condition a strong activation of the mirror was also recorded while no activity was observed in the last (see figure 2.5). These results seem to stress the involvement of motor area in the codify of an action independently if it is realized, observed or inferred. Even if the precise role of the mirror system in action codify is still object of debate, as I will quickly explain in a while, the direct matching hypothesis implies for the motor cortex a strong or even a predominant role in the process of visual action recognition. The idea that in the brain the motor control and the visual elaboration could be realized independently in two well defined areas is not in agreement with the recent experiment results. With this I mean that although the representation of the visual input achieve a high level of generalization in the visual area, the representation of an action is obtained only with an important computational help of the motor area were maybe the action representation itself is finally



**Figure 2.5:** Activity of a mirror neuron in the F5 in response to action observation in full vision and hidden conditions. The lower part of the four panels illustrate the action as seen from the point of view of the monkey. The panels (a) and (c) refer to the full vision conditions, respectively when the experimenter is grasping an object or is mimicking the action. The panels (b) and (d) describe the experiment in hidden condition. In this second condition the monkey at the beginning of the experiment can see the whole scene, then the right part of the scene is covered with a opaque sliding screen so that the monkey can not see the last part of the action. This is represented by the grey zones in the pictures. In the upper part of the four panels are reported the raster plots of 10 consecutive trials and the histograms reporting the number of spikes per second. Picture taken from (Rizzolatti et al., 2001)

physically realized. As I just said anyway the precise role of the mirror neurons in the process of action recognition is still debated as well as the importance of their role in the visual process of action recognition.

### **Genetic account vs associative account**

The involvement of mirror neuron in action recognition, and all the amount of cognitive functions that descent from that, has for a long period suggested for the mirror mechanisms a so called *genetic account*. This hypothesis asserts that the genetic predisposition to the development of mirror neuron is the result of a big positive evolutionary pressure due to the important rule the mirror neurons play in the process of action recognition. In contrast recently some scientists (Cook et al., 2013) proposed for the mirror neurons what they called an *associative account*. This hypothesis suggests that "mirror neurons acquire their capacity to match observed with executed actions through domain-general processes of sensorimotor associative learning"<sup>6</sup>. Actually the associative account doesn't make any assumption on the functional role of the mirror neurons, it allows but not assume that the mirror neuron could have a role in action recognition and all the cognitive functions related to that one. One of the most debated evidence is about the capacity of mirror neurons to encode action 'goals'. Many supporters of the genetic account assert that the receptive field of the mirror neurons is tuned in accordance to the 'goal' of the observed action making them important in the process of action understanding and a possible target of evolutive pressure. At this point we have first to clarify what this authors mean when they refer to action goal. In literature two definitions are commonly adopted, the first one assumes that mirror neurons encode 'goals' if they encode object-directed action, the second one if they encode high-level action intentions. As we said in the previous sections have been found mirror neurons that discharge in accordance just with the observation of transitive actions and not pantomimed ones, but it is even important to stress that more recent studies (Kraskov et al., 2009) reported that the majority of the mirror neurons observed shown similar responses during observation of transitive or intransitive actions. Regarding the ability of mirror neurons to codify intentions, indeed seem that among the mirror neurons have been found neurons with this high level of generality that for examples spikes when the monkey grasp in order to eat or grasp in order to place, independently of the way the action is performed(I will go through these interesting results in the next paragraph). However, the single cell data again suggest that relatively few

---

<sup>6</sup>taken from (Cook et al., 2013)



mirror neurons present a specific response to object directed actions or action goals for a system designed by genetic evolution to this end. Most of the mirror neurons respond instead differently depending on whether the action is executed with the left or the right hand, if the action take place in the peri-personal or extra-personal space, moreover as I said before the majority of them is view-dependent, finally some are tuned according to the distance at which the action is taking place. This strong non-specificity in the mirror neuron responses can be more easily explained by the associative account. The computational models that aim for depict mirror mechanism as a system that is useful in the action recognition must face the hard challenge to explain how the action recognition functionality can emerge from a system with a strong non-specific response.

### **Hierarchical motor representation**

Another debated issue on mirror neurons and on motor codify in the CNS, is the possibility that action could be represented with different levels of detail in different cortex areas. Iacoboni and colleagues (Iacoboni et al., 2005) developed an experiment on human recording the fMRI activity while subjects were observing different grasping actions in different contexts. Interestingly they could find some areas in the ventral premotor cortex that respond according to the action goals and not to the action kinematic-dynamic characteristics. In a recent paper Grafton and Hamilton (Grafton and Hamilton, 2007; Hamilton and Grafton, 2008) identified through the use of fMRI different areas of the brain that seem to represent the same action with a different degree of complexity. They could identify three main degrees of action complexity representations codified by different zones: kinematics representation, goal-object representation and outcome representation. The areas coding the kinematic representation activate according to the reaching trajectory, the grip configuration and the kind of dynamical interaction required, these were mainly localized in the inferior frontal gyrus. The goal-object action representation codify the identity and function of the grasped object, these area according to the authors are in the intraparietal sulcus. Finally the physical consequences of an action were coded in the outcome representation area localized in the intra parietal sulcus and in the inferior frontal gyrus. The works of Rizzolatti and colleagues (Rizzolatti and Sinigaglia, 2010; Rizzolatti et al., 2001) on monkey motor areas seem to confirm the hypothesis of a hierarchical action representation. In these works single-cell recording in the ventral premotor cortex and inferior parietal lobe were realized while the monkeys were performing some grasping actions. Even in this case they could find neurons that respond according to different degree

of details of action description. In fact they could find neurons that spike when a specific kind of action is executed in a specific way, like neurons that spikes when a grasping with a whole hand prehention is executed, and other neurons that spike whenever a kind of action is executed independently of the specific way, like neurons that spikes for every grasping action provided that the action is executed with the right hand. Another interesting findings of this work was the presence in the motor areas of neurons that codify action even in a more general way, such as neurons that are sensitive to grasping a piece of food whether the action is executed with the left hand, the right hand or the mouth.

## **2.2 Action representation and control**

In this paragraph we depart from describing results connected to mirror neurons to analyze some studies regarding more generally the codify of action in the CNS. In the last years a lot of work has been done in this direction, in particular in investigating the possibility that action could be codified in the CNS through the use of synergies. The concept of synergies in motor action representation in the brain was for the first time introduced by Bernstein in 1967. Bernstein defines synergies as specific patterns in muscles activities or movement kinematics/dynamics as building block for representing and controlling actions. Since that time a lot of works have been develop in order to prove the existence in the brain of a representation of actions though synergies. Some works have develop direct invasive measure in the monkey motor area, some non-invasive measure on the human brain, some other looked for an indirect proof of the existence of synergies, recording, through the use of movement sensors, different kind of actions and then trying to show that the data obtained could all be represented using few action patterns differently combined together according to the different actions.

### **2.2.1 Motor synergies in the brain**

One of the recent work that more then the other bears strong evidences for the existence of synergies is the one of d'Avella and colleagues (d'Avella et al., 2006). In this work they analyzed different fast-reaching movements of the arm, hand and shoulder. The subjects in their experiment were in standing position holding a load in their hand and were asked to move the load from a central location in front of them to some other locations either in the sagittal plane or in the frontal plane. These locations were easily reachable with fast movements and moving just hand, arm and shoulder. The Electromy-

graphy (EMG) activity signals of up to 19 muscles was recorded during the experiments. An algorithm was then developed in order to represent muscle patters as an over-position of synergies according to the following formula:

$$\mathbf{m}(t) = \sum_{i=1}^N c_i \mathbf{w}_i(t - t_i) \quad (2.1)$$

In the previous formula  $\mathbf{m}(t)$  is a vector of real numbers, each component of which represents activation of a specific muscle at time  $t$ ;  $\mathbf{w}_i$  describe the muscles activity patterns and  $t_i$  are the corresponding onset times. The algorithm was initialized by choosing  $N$  random synergies and minimizing the least squares reconstruction error iterating the following steps:

- given a set of synergies find synergies onset times by a matching pursuit procedure;
- given a set of synergies and their onset times, find the coefficients  $c_i$  by linear least square;
- given onset times and scaling coefficients update the synergies according to the gradient descent on the least squares reconstruction error.

Clearly this algorithm assumes that each synergies cannot be used more then once in a movement. d’Avella assumes that this could be the case for fast reaching movements. The results obtained offer big clues for a synergy representation of action. The first interesting result consists in finding that more or less the same number of synergies, between 7 and 9, was sufficient for explaining a big percentage of the data variances, between 93% and 96%, for each of the nine subjects analyzed. Interestingly d’Avella and colleagues create a synthetic dataset with the same mean and variance of the original one but where the components of the vector representing the muscle activation were shuffled in order to destroy any synergy eventually present in the data. Applying the same algorithm to the synthetic dataset they found that with 7-9 vectors  $\mathbf{w}_i$  they were able to explain just the 30% of the variance of the data. This result confirmed that the high percentage of explained variance using relatively few vectors  $\mathbf{w}_i$  in the original dataset was probably due to the presence of synergies. Finally they tried to associate synergies usage with movement directions finding for most of the synergies a directional tuning, in other words some synergies seem to be related to the direction through which the arm and hand were directed.

In a more recent work developed by Overduin and colleagues (Overduin et al.,

2012) direct measures on monkeys performing hand movements were developed looking for a synergy representation in the motor cortex. They compared the EMG forelimb muscles signals and the hand joint patterns in two different experimental conditions. In the first experimental condition some sites in the monkey primary motor cortex and premotor cortex were electrically microstimulated. In the second experimental condition the monkey was free to develop some naturalistic movements. In the first experiment they notice that, regardless of the initial hand posture, the artificial stimulation for different sites of the motor cortex evoked convergent motion of one or more joints, moreover very similar EMG muscles patterns were associated to the same area stimulation. These patterns were analyzed and represented using an algorithm very similar to the one previously described obtaining a set of synergies relative to the artificially induced actions. The same algorithm was applied when EMG were recorded while the monkey was developing natural actions obtaining even for the signal a synergy representation. The comparison of the two sets of synergies returned a strong correspondence of elements of the two sets. The authors concluded that the synergies observed by directly stimulating the motor area are not a trivial biomechanical result of imposing artificial patterns of tonic muscle contraction. The authors could at this point develop a study looking for correlations between areas stimulated and synergies recorded. They found that the same synergy could be represented in different areas of the motor cortex. These results seem coherent with some results described in the previous paragraphs.

## Chapter 3

# Computational models of mirror neurons and action representation

In the first paragraph of this chapter we will have a fast review of the works present in literature that realize computational models of mirror neurons. We will stress their strengths and weaknesses, in particular underlining the necessity for a model that would be more descriptively adequate and functionally informative. In fact as we will show many of the models present in literature do not describe any of the many different mirror neurons behaviour (strictly and broadly MNs, view-dependent MNs, etc.). Another criticism we address to models present in literature is their inability to depict a precise functional role to the motor cortex in the elaboration of an observed action.

In the second part of the chapter we will come back to the possibility of representing action through the use of synergies. Differently from the studies presented in the first chapter, that are mainly neurophysiological studies, here we will present some indirect proofs of a synergy organization in the brain. These studies mainly consist in collecting data recording the joints of the hands or of the limbs while performing an action, and then analyzing these data with machine learning techniques in order to show that the data collected could be well represented in terms of synergies.

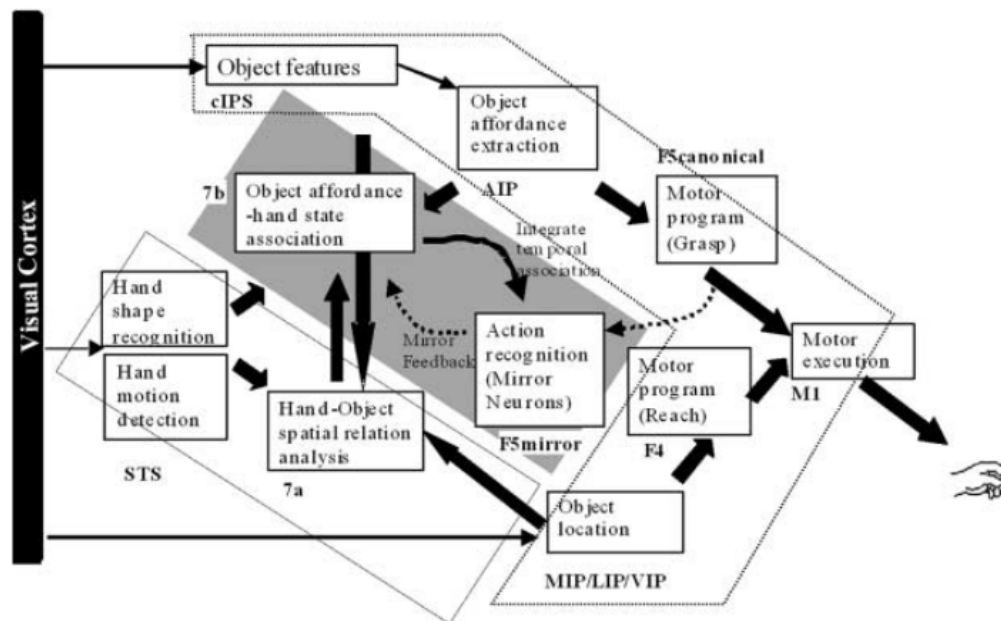
### 3.1 Mirror Models

The works of Oztop and Arbib (Oztop and Arbib, 2002; Oztop et al., 2004, 2006) have the big merit of realizing a global model that includes many of the biological circuits involved in the mirror neurons system. In particular the MNS1 model (which stands for Mirror Neuron System 1 and whose block schema is presented in figure 3.1) is constituted of three main "grand

schemas":

1. *Reach and grasp schema.* This schema model some characteristics of the VIP-F5 and AIP-F5 biological circuits. It evaluates a motor program as output of two computational paths. The first one computing the affordances of the object to grasp, the second one evaluating the relative position of object and hand.
2. *Visual analysis of hand state.* This schema processes visual input concerning hand/target-object pairs and codes these into a vector, called hand-state, which holds high-level, observer-independent features of hand-object configurations, such as hand-object distance and grip size compared with object size.
3. *Mirror circuits* contains the module that actually models mirror neurons activity, named in figure as "action recognition". This receives two inputs, one is the motor program selected, and the other is an object affordance-hand state association. This module is implemented by means of a feed-forward neural network and can work in two modes: learning and recognition. In self-observation condition this network is trained to associate hand-state sequence to encoded hand programs. In the recognition mode it works as a classifier of the ongoing actions.

One of the main weaknesses of this model is to be not so much descriptively adequate, associating to the mirror neurons the same behaviour, when observing or executing an action. Clearly, as vastly explained in the previous chapter, this is not the case for mirror neurons that show very different behaviours in observation and execution. The same criticism can be addressed to the model of Ito and Tani (Ito and Tani, 2004). Even if it is to say that this work has the big merit of realizing an architecture strongly biologically inspired. Using in their model a Continuous Time Recurrent Neural Network (CTRNN) the authors depict a mechanism for storing and re-using a motor action in order to recognize an observed action. Another criticism common to the previous and others computational models (Keyser and Perrett, 2004; Haruno et al., 2001) is that they usually endorse the hypothesis that perspective-free perceptual information is fed into the mirror mechanisms. In the model of Oztop and Arbib, for example, the hand-state vector is calculated from perspective-free information on the hand and the target object. These vectors are then the input for the action classifier. Therefore in this model a substantial preprocessing of the visual input is realized without considering any motor involvement. Even this aspect results quite in disagreement with the experimental data that show mirror neurons with a

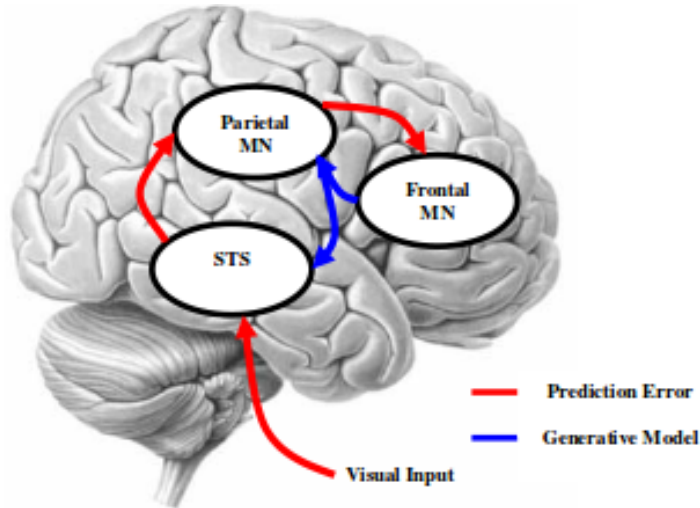


**Figure 3.1:** The MNS1 model is composed of several functional blocks which are related to the computation of different brain areas. The picture is taken from (Oztop and Arbib, 2002).

view-dependent behaviour (Caggiano et al., 2011).

The problem of associating a view representation of action to the relative motor representation entails several modeling challenges. Some of these can be addressed to the ill-posed nature of the problem. Let us consider our case. Suppose that we wish to associate to a set of configurations that a hand takes on during a grasping action the relative motor representation of the hand. Clearly could be that different hand configurations, when observed from a particular point of view, could result in the same visual image. In this way the same image should be associated to more than one input. Friston and colleagues describe a system that solve the previous problem involving mirror neurons (Kilner et al., 2007; Friston, 2005). The authors suggest that MNs together with other brain area (like superior temporal sulcus and inferior parietal lobule) constitute both a forward and an inverse model for action representation. In their model (see figure 3.2) the previous areas of the brain are hierarchically organized, realizing an action representation according to different levels of detail. Each level of the hierarchy constitute a generative model to predict representation in the lower level. This prediction is sent to the lower level in the hierarchy via the backward connections, in blue

in the figure. The prediction is then compared with the representation in this subordinate level to produce a prediction error, this is sent back to the higher level via forward connections, in red in the figure. This model, as the

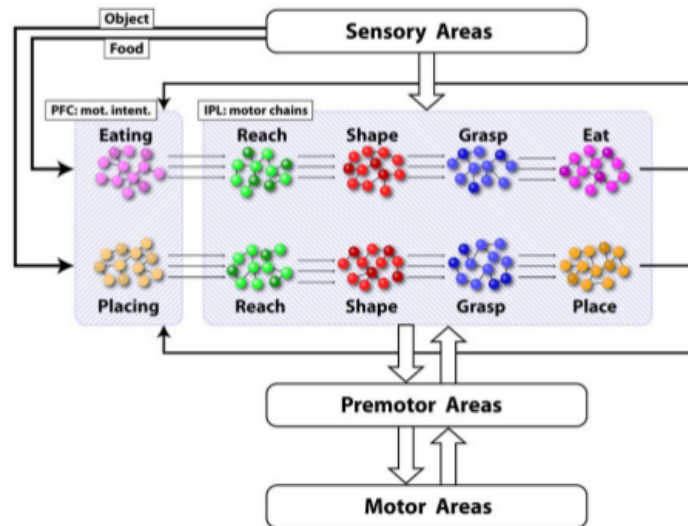


**Figure 3.2:** The hierarchical model of Friston and colleagues. The blue arrows represent channels through which the prediction of the higher level of the hierarchy is sent to the lower level. The red arrow indicating the channels through which prediction error is sent from lower to upper areas in the hierarchy. The picture is taken from (Kilner et al., 2007)

authors suggest, is formally equivalent to the empirical Bayesian inference, and for this reason has the value of showing how this inference can be implemented in a biologically plausible system like a neural network. On the other hand it is to say, that, at least in the case of mirror neurons, the model is not explicative of the functional role of the different areas involved in the visuo-motor mapping and for this reason do not allow for a comparison of the model with the experimental data. The idea of an hierarchical model of the mirror neurons system is even present in a recent work by Chersi and colleagues (Chersi et al., 2011). In this model the inferior parietal lobe (IPL), the premotor areas and the motor areas are hierarchically organized and, according to the experimental findings, represent action at multiple level of details both when observed or executed. This architecture also proposes a mechanism for explaining the particular behaviour observed in the inferior parietal lobule (IPL). In particular has been shown () that neurons in these area seem to codify different grasping motor act. These neurons seem to respond differently according the final goal of the action sequence in which the



act is embedded. The architecture proposed explain the observed behaviour of the IPL neurons hypothesizing a goal-specific neuronal chain organization in this area. The architecture is presented in the following picture 3.3. The



**Figure 3.3:** In the model proposed by Chersi and colleagues IPL, premotor and motor areas are hierarchically organized. Moreover the motor acts in the IPL are disposed in terms of goal-specific neuronal chains. The picture is taken form (Chersi et al., 2011)

IPL area is constituted of different chains of neuron pools, each chain corresponding to a different action goal. As the picture shows, the same action, like "reach" for example, is represented by more then one neuronal pool. We will repropes this idea of representing the same action multiple times according to the particular motor act it is embedded in, showing even how this action organization can improve the process of visual elaboration of the observed actions.

## 3.2 Action representation

In this section we will come back to the problem of action representation in the CNS. In the first chapter we showed some results that, developing measures on brain signals, present clues in favor of a synergies representation of action. In this section we will bring other clues in this direction, this time analyzing indirect studies on action codify. In these kind of studies usually

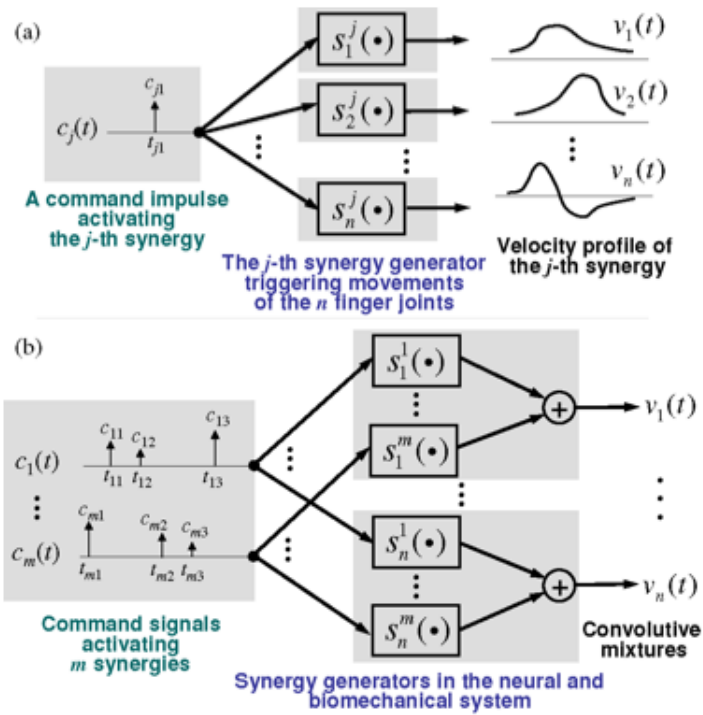
the data analyzed are collected through the use of devices that record the values of the angles among the joints of the hand or more generally of the limbs while performing an action. A synergy representation is then evaluated for these data.

### 3.2.1 Motor synergies in the hand and grasping actions

A lot of the literature facing the problem of synergy has been applied to the study of hand actions. The hand is in fact a very complex system with more than twenty Degrees of Freedom (DOF) allowing to dexterously perform actions. This complexity has provided a key challenge for research in representing action, for it seems unlikely that all DOF are individually represented and controlled during the execution of hand action such as grasping, tearing, holding. Several studies have highlighted the need for a simplified way of representing/controlling hand actions (Iberall et al., 1986; Santello et al., 1998; Mason et al., 2001). Most of the studies on synergy representation of hand actions are indirect. With this I mean that frequently the following protocol is developed. Hand actions are recorded by using some sensor, frequently applied to a glove, able to measure the joint angles of the hand. The data collected are then represented through the use of a linear overposition of vectors frequently referred as synergies as in equation 2.1. Clearly assessing that the hand actions can be expressed as an overposition of synergies is just the result of a simplification that yet seem to work properly. Let us see where this simplification came from. Following Vinjamuri (Vinjamuri et al., 2010a) we will develop in this paragraph some calculations as we were considering kinematic synergies, i.e. characteristic patterns in the space of the velocities of the hand joint angles, but the same reasoning can be applied equally well when synergies are searched in the space of hand joints. Let's assume that in the brain the spike frequency of a specific neuron  $c_j(t)$  is able to activate at time  $t$  a given synergy  $\mathbf{s}^j$  involving, at least in principle, all the joints of the hand for a certain amount of time. The synergy is in fact a vector  $\mathbf{s}^j(t) \equiv [s_1^j(t), \dots, s_n^j(t)]$  where each component of the vector could represent the velocity or the joint angles values of the hand. Assuming that the frequency of the spikes is proportionally related to the amount to which the relative synergy is involved in defining the values of the joint angle velocities, we could write:

$$\mathbf{v}(t) = c_{j1} \mathbf{s}^j(t - t_{j1}) \quad (3.1)$$

Where the vector  $\mathbf{v}(t)$  is the one representing angular velocities of the joint angles of the hand,  $c_{j1}$  is the spike frequency at time  $t_j$ . Clearly if we consider that the spike frequency can change in time the previous formula can be



**Figure 3.4:** (a) A synergy can be considered as the boxes in the figure that transform the spike frequency at time  $t_{j1}$  into  $n$  different velocity profiles, one for each joint of the hand. (b) Multiple neurons can activate multiple synergies, the velocity profile of the single joint is the result of this multiple activation. This image has been taken from (Vinjamuri et al., 2010a)

rewritten as the convolution of spikes frequency and synergy temporal profile.

$$\mathbf{v}(t) = (c_j * \mathbf{s}^j(t)) \quad (3.2)$$

Dividing the time into small intervals we could consider the frequency as being constant in this interval, transforming the previous convolution into a summation:

$$\mathbf{v}(t) = (c_j * \mathbf{s}^j(t)) = \sum_k c_{jk} \mathbf{s}^j(t - t_{jk}) \quad (3.3)$$

When more than one synergy is activated, the final velocity could be considered as the result of the linear overposition of the different synergies. This is clearly a quite strong assumption and we will see in a while how it can be verified. By now let's assume it :

$$\mathbf{v}(t) = \sum_j \sum_k c_{jk} \mathbf{s}^j(t - t_{jk}) \quad (3.4)$$

This formula allows for repetitive uses of synergies in a single movement. This assumption even if physiologically plausible, involves a much difficult computational problem. In fact an algorithm should determine not only the shapes of the synergies, but also their onset time, amplitudes and number of recruitments in the movement. A possible solution to the previous problem could consist in considering the synergies as all combining almost instantaneously. In this case the previous formula would be more simply reduced to:

$$\mathbf{v}(t) = \sum_j c_{j0} \mathbf{s}^j(t - t_0) \quad (3.5)$$

In deriving this final simple equation we have supposed that when more than one synergy is activated by some neurons spikes, then the final velocity profile of the hand joints would be the linear overposition of the different synergies and that we could consider all synergies as synchronous active. In the next paragraph we will rapidly look at some indirect experiments that confirm the plausibility of the just mentioned assumption and give some important clues for a synergy representation of hand action in the brain.

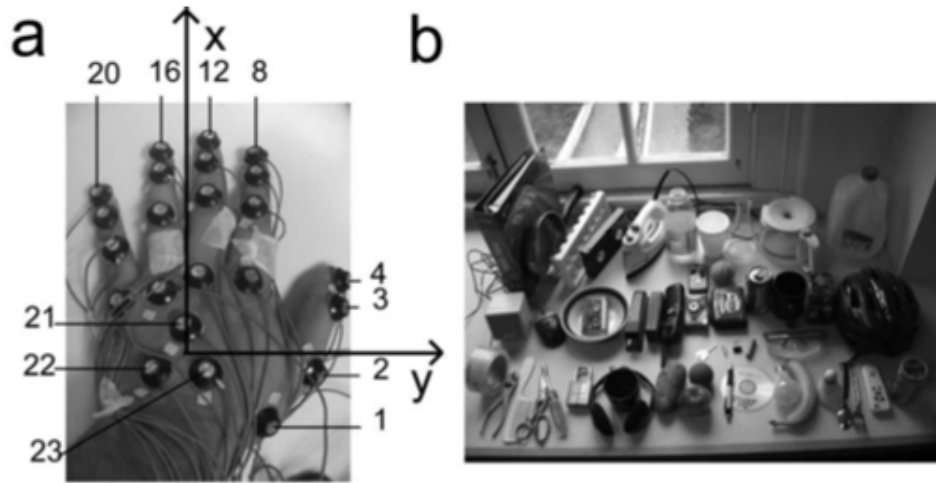
### 3.2.2 Indirect synergy studies on hand action

In this paragraph we will first give a very rapid overview of the literature on synergy representations of hand grasping actions. Then we will analyze more closely some particular works, that are emblematic of many of the works on synergies, both for the way data are collected, and the way are analyzed. Finally we will discuss what are the main results obtained by this work and

other.

The works present in literature that look for synergies mainly differ in the kind of data analyzed and in the definition of synergy. In fact in most of the works the data recorded are the values of the joint angles of the hand (Thakur et al., 2008; Todorov and Ghahramani, 2004), but in other works the joint angle velocities are recorded, an example is the work of Vinjamuri (Vinjamuri et al., 2010a), some other works record the forces exerted by fingers (Santello and Soechting, 2000). Even the definition of synergy adopted clearly slightly change among these works. Two synergy definitions are more common than other: postural synergies and temporal-postural synergies. In a postural synergy representation the action can be expressed as a linear combination of vectors (the synergies), where the coefficients of the overposition change over time (Mason et al., 2001). On the other hand, it has been proposed that hand actions, expressed as a temporal sequences of hand-joint configurations, should be represented by linear combinations of a small number of temporal-postural synergies, that is, of specific patterns in the space of hand-joint configurations varying over time (Santello et al., 2002; Vinjamuri et al., 2010a).

The experiments frequently consists of recording hand movement from different subjects. The hand joint angles of the subjects are frequently recorded using a glove equipped with motion sensors, that the subjects wear during the experiments. The kind of experiments are usually reach-to-grasp tasks, but could even be haptic exploration tasks or developing common hand gesture. In the first case the subject is comfortably seated in front of a table where some common objects of different shape, size and weight are present, the subjects are asked to reach and grasp the objects. In an haptic exploration task the subject is blindfolded and asked to identify through haptic exploration common objects placed in front of them. Finally in the last kind of task the subject is asked to perform common hand actions like flipping through the pages of a book or crumple a sheet of paper. In a reach-to-grasp task, frequently the subject is asked to develop the task in the shortest possible time. In fact, as we stressed in the previous paragraph, in this case possibly we could consider synergies as all combining almost instantaneously and represent action as a linear combination of synergies all starting at the same time, like expressed by the formula 3.5. The assumption of instantaneous synergies activation was proved by Vinjamuri and colleagues in (Vinjamuri et al., 2010a), they asked to some subjects to perform fast reaching grasp action (an action last for about 1.6s). To each action was associated a vector in which the first  $N$  values were the values returned from the motion sensor at the first time instant, the second  $N$  values were the values of the motion sensor for the second time instant and so on for the whole action duration. The vector



**Figure 3.5:** This picture is taken from a work of Thakur (Thakur et al., 2008). (a) The hand of a subject covered with motion sensors. (b) The object used in order to develop an haptic exploration task.

obtained were expressed as an overposition of temporal-postural synergies through the use of a Principal Component Analysis (PCA) techniques. They could find that a set of 7/ 9 synergies could well represent all the grasp actions analyzed, by explaining a big percentage of the data variance<sup>1</sup>(around 93% / 96%). Thakur and colleagues (Thakur et al., 2008) analyzed different hand actions looking for postural synergies, i.e for synergies in the space of configuration. The data they analyzed were collected asking subjects to haptic explore some common object and performing reach to grasp actions. Even in this case PCA was used to find a representation of the data as linear overposition of some synergies. Interestingly even in this case the number of synergies to explain a large percentage of the variance was around 7/9 synergies. Very similar result are even shown in (Todorov and Ghahramani, 2004). Even in this case a PCA analysis is developed on hand joint configurations collected when the subject was performing common hand actions. Interestingly these works suggest even that most of the synergies found are task dependent. This result is even confirmed in a work by Ciocarlie and Allen (Ciocarlie and Allen, 2009). The authors, used the synergies found by Santello (Santello et al., 1998) in order to control a robotic hand in a variety

<sup>1</sup>The percentage of explained variance of a bench of data measures the quality of a representation of the data. It comes from comparing the variance of the original data with the variance of the reconstructed data. See appendix PCA for better explanation.

of grasping tasks. They found that the best way for controlling the robot hand in the was by using just few of the Santello synergies. The use of more synergies do not improve significantly the performance of the robot unless a much bigger computational time was admitted for the robot in order to find the best synergies combination. The authors suggest that these results are due to the fact that just few of the synergies found by Santello are involved in all grasping tasks, while the most of them are grasp specific.





## Chapter 4

# Hierarchical temporal postural synergies

As we said in the previous chapter a lot of results in literature show the plausibility for a synergy representation of actions in the CNS. Some other results, we shown in the first chapter, instead stress that in the brain the same action could be represented in different areas, with different levels of detail. In this chapter we will describe the experiments done in order to obtain an action representation that would model both these biological findings. The actions representation we will describe has been realized using synergies hierarchically organized. We will show that, considering any action  $\mathbf{x}$  in our motor dataset, is possible to find a set of vectors  $\mathbf{V}_i$  such that the action can be obtained as a linear overposition of the vectors.

$$\mathbf{x} = \sum_i c_i \mathbf{V}_i. \quad (4.1)$$

In our algorithm the vectors  $\mathbf{V}_i$  represent temporal postural synergies (TPSs), i.e. patterns in the space of motor configurations varying over time. According to Vinjamuri (Vinjamuri et al., 2010a) we considered actions developed in a short time interval, in this case in fact considering actions as a linear overposition of synergies is a plausible approximation. Our algorithm induced a hierarchical organization among the synergies. This means that we were able to find synergies, in the upper part of the hierarchy, that are used to represent different kinds of actions, so describing action in a more general fashion. We could even find synergies, in the lower part of the hierarchy, that are used to represent just a particular kind of action, describing actions at a more detailed level.

## 4.1 Tree-structured synergies method

The algorithm we used, originally proposed by Jenatton (Jenatton et al., 2010), allows for representing action  $\mathbf{x}$  in terms of a linear combination of vectors  $\mathbf{V}_i$ . The main characteristics of our representation are that:

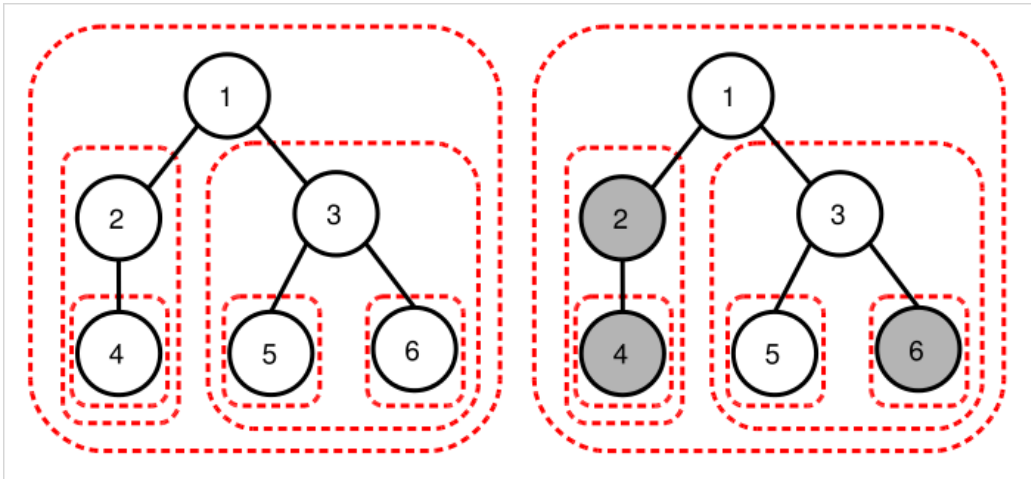
- the vectors involved in the representation are hierarchically organized;
- the representation is sparse.

The algorithm realizes a hierarchy among the TPSs by imposing a tree-constraint on the representation in the following way. The user can arbitrarily select a tree structure, the tree can have any kind of shape, but must have a number of nodes equal to the number of synergies  $\mathbf{V}_i$  in which the user wish to decompose the action. Each node of the tree will be associated with a vector  $\mathbf{V}_i$ . Then the algorithm will look for the coefficients and the vectors that better approximate the vector  $\mathbf{x}$ , trying to force a representation such that the vector  $\mathbf{V}_i$  will be used to represent action  $\mathbf{x}$  only if even its ancestors on the tree  $\mathbf{V}_j$  are used in the representation of the action.

This method, we called Tree-Structured Synergies Method (TSSM), is framed within the larger class of *sparse coding* problems (Aharon et al., 2006; Engan et al., 1999). These are unsupervised algorithms that try to find a set of vectors constituting an overcomplete basis for the space of data. Each data being represented as a linear combination of the basis vectors in which a small set of coefficients are different from zero. We will see in fact that our algorithm in addition to realize a hierarchical representation of action will even use few Temporal Postural Synergy (TPS)s  $\mathbf{V}_i$  to represent each action. Let's see more in detail how the algorithm works. Let's define the matrix  $\mathbf{X} \in \mathbb{R}^{n \times p}$ . This matrix is obtained organizing row-wise the vectors  $\mathbf{x}$  describing the action. The  $n$  rows of the matrix corresponding to the  $n$  actions in our dataset, while  $p$  is the dimensionality of the vector describing the action  $\mathbf{x} \in \mathbb{R}^p$ . The problem we addressed can be solved by finding a matrix  $\mathbf{V} \in \mathbb{R}^{p \times r}$  such that each row of  $\mathbf{X}$  can be approximated by a linear combination of the  $r$  columns of  $\mathbf{V}$ , i.e.,  $\mathbf{X}_i = \sum_{j=1}^r u_{ij} \mathbf{V}^j$ .  $\mathbf{V}$  is the matrix of TPSs which are disposed column-wise.  $\mathbf{V}$  is also called *dictionary* in machine learning context. Let us call  $\mathbf{U} \in \mathbb{R}^{n \times r}$  the matrix of the linear combination coefficients, i.e., the  $i$ -th row of  $\mathbf{U}$  corresponds to the  $r$  coefficients of the linear combination of the  $r$  columns of  $\mathbf{V}$  in order to approximate the  $i$ -th row of  $\mathbf{X}$ . Consequently,  $\mathbf{UV}^T$  is an approximation of  $\mathbf{X}$ . Following (Jenatton et al., 2010) the problem can be formulated as the following minimization problem:

$$\min_{\mathbf{U}, \mathbf{V}} \frac{1}{2np} \|\mathbf{X} - \mathbf{UV}^T\|_F^2 + \lambda \sum_{i=1}^n \sum_{j=1}^r \|\mathbf{D}_j \circ \mathbf{U}_i\|_\infty \quad (4.2)$$

where  $\mathbf{U}_i$  is the  $i$ -th row of  $\mathbf{U}$ ,  $\mathbf{V}^j$  is the  $j$ -column of  $\mathbf{V}$ , and the matrix  $\mathbf{D} \in \mathbb{R}^{r \times p}$ , encoding the tree  $T$ , is defined so that  $d_{ij}$  is equal to 1 if the  $j$ -th node in  $T$  is a descendant of the node  $i$ , and 0 otherwise. The vector  $\mathbf{D}_j \circ \mathbf{U}_i$  is the element-wise product of  $\mathbf{D}_j$  and  $\mathbf{U}_i$ . The first term in the equation B.1, is forcing our algorithm to reconstruct the original data. The second term is instead forcing the hierarchical structure and the sparsity. To enforce the tree hierarchical decomposition we may want that the decomposition of any vector  $\mathbf{X}_i$  could involve a dictionary element  $\mathbf{V}^j$  only if the ancestor of  $\mathbf{V}^j$  in the tree are themselves involved in the representation of  $\mathbf{X}_i$ . This statement could be equivalently formulated as: when a dictionary element  $\mathbf{V}^j$  is not involved in the representation of  $\mathbf{X}_i$ , none of its descendants should be involved in the representation, i.e. the representation should avoid as much as possible using subtrees. Each vector  $\mathbf{D}_i$  is selecting a node  $i$  and specifying its subtree, in the sense that the  $j$ -th components of the vector will be equal to zero if the node  $j$  is not in the subtree that has  $i$  as root and will be equal to one otherwise. Now the term  $\|\mathbf{D}_j \circ \mathbf{U}_i\|_\infty$  in equation B.1 is penalizing a representation if this is using the subtree described by the vector  $\mathbf{D}_j$ . On the other hand if we extract form the second summation



**Figure 4.1:** *On the left.* The dashed square indicate the subtrees. To each subtree corresponding a vector  $\mathbf{D}_i$ . *On the right.* Example of a sparsity pattern. The subtrees  $\{2, 4\}$ ,  $\{4\}$  and  $\{6\}$  are set to zero, so the corresponding node (in gray) are removed from the representation. The remain that constitute the representation,  $\{1, 3, 5\}$ , respect the hierarchical constraint. Taken form ((Jenatton et al., 2010)).

in equation B.1 the coefficients  $\mathbf{U}_i$  relative to the single data, we have that this can be interpreted as the  $l_1$  norm applied over the  $r$ -dimensional vector  $\alpha = [\|\mathbf{D}_1 \circ \mathbf{U}_i\|_\infty, \dots, \|\mathbf{D}_1 \circ \mathbf{U}_i\|_\infty]$ . We know that the  $l_1$  norm enforce sparsity so just few component of the vector  $\alpha$  will be different from zero, i.e. the algorithm will try to represent the data using the less possible number of subtrees (see figure 4.1).

In order to solve the problem (B.1), we followed the usual approach of finding the minimum by alternating optimizations with respect to the coefficients  $\mathbf{U}$  and to the dictionary  $\mathbf{V}$ . Most methods are based on this alternating scheme of optimization (Basso et al., 2011). Therefore the algorithm used here is composed of two alternate stages: 1) *Tree-Structured Coding Stage*. In this stage the dictionary  $\mathbf{V}$  is fixed and the matrix  $\mathbf{U}$  is updated. 2) *Synergy Dictionary Stage*. Here the matrix  $\mathbf{V}$  is updated while keeping the  $\mathbf{U}$ 's values fixed. For more details see Appendix (Section ??). Note that, even if the algorithm convergence is not theoretically guaranteed, from an experimental point of view we found an enough stable algorithm convergence: there was an encouraging independence of results from multiple runs of the algorithm on the same data. It is worth to note that different choices of the tree  $T$  and the sparsity parameter  $\lambda$  induce different solutions of the minimization problem. This dependence implies the need for a careful choice of  $T$  and  $\lambda$ .

## 4.2 Dataset collection

Ten right-handed subjects (five men and five women, age ranging from 24 to 30 years) took part in the experiments. Subjects were instructed to reach, grasp and hold different objects several times. Following Gallese's studies on monkeys (Gallese et al., 1996), three different classes of grasping actions were considered: Precision Grip, Finger Prehension and Whole Hand Prehension. Precision Grip is the grasping action done putting in opposition the index and the thumb of the hand; Finger Prehension is done putting in opposition the thumb to the other fingers; Whole Hand Prehension is done executing a flexion of all the fingers around the object. Three objects with different size and shape were used according to the type of grasp used. Nine different grasping action types were selected (see Table 4.1), and each action type was executed by all the subjects. The subject were seated at a table with two clearly visible surface marks ( $m_1$  and  $m_2$ ) placed at a distance of roughly 40 cm from each other. Each trial started with the subject having his hand closed on the mark  $m_1$ . Then, the subject had to reach and grasp the target object. This was in the position  $m_2$  placed on the table or on an appropriated sustain in order to facilitate the grasping. The subjects were asked to

perform the action in an accurate way, but even in the shortest possible time and to hold the object after grasping for some time instants. Participants had been clearly instructed about each type of grasping action. Before recording hand movements, there was a training phase where they familiarized with the different types of grasping actions. Subjects performed 50 trials for each type of grasping actions. Thus, each subject performed a total of 450 trials. A HumanGlove (Humanware S.r.l., Pontedera (Pisa), Italy) endowed with 16 sensors (see Figure 4.2) was used to record joint angles. Wrist related sensors were not considered for this work whereas 10 hand related sensors are considered according to (Vinjamuri et al., 2010b). In particular sensors which measure angles of the carpometacarpal(CMC) and metacarpophalangeal (MCP) joints of the thumb and the metacarpophalangeal (MCP) and proximal interphalangeal (PIP) joints of the other four fingers were considered, for a total of  $d = 10$  sensors. Once we recorded all the actions, we truncated them in order to preserve only the their relevant parts where the hand was actually moving. We then resampled each action in order to have the same length  $T = 30$ . To represent an action we first arranged the  $d$  sensor values recorded at time instant  $t$  into a vector we named hand-joint configuration  $\mathbf{hc}(t) \in \mathbb{R}^d$ , then the vector  $\mathbf{x}$ , describing the action was obtained by concatenating the  $T$  vectors  $\mathbf{hc}(t)$ , i.e.,  $\mathbf{x} = [\mathbf{hc}(1), \mathbf{hc}(2), \dots, \mathbf{hc}(T)]$ .



**Figure 4.2:** *DataGlove*. HumanGlove (Humanware S.r.l., Pontedera (Pisa), Italy) endowed with 16 sensors was used to record all the grasping actions.

Summarizing, ten datasets  $DS_1, DS_2 \dots, DS_{10}$ , one for each subject, were constructed. Each dataset contains a total of 450 grasping actions, 50 trials for each one of the 9 action types previously described. In order to develop the tests we will describe in the following sections we needed to create a






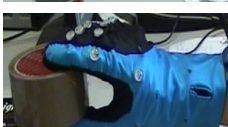
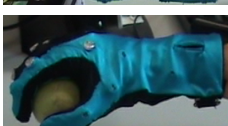


training and a validation set. We split each dataset  $DS_i$ , consisting of 450 actions, into two subsets consisting of 225 actions. These were obtained choosing in a random way 25 actions for each one of the 9 action types. For each  $DS_i$ , the first subset obtained was used as training set and we will call it  $TS_i$ , whereas from the second subset we built three noisy test sets  $NT_i^1$ ,  $NT_i^2$  and  $NT_i^3$  adding to the recorded actions a zero-mean Gaussian noise with standard deviation respectively of  $\sigma = 0.2$ ,  $\sigma = 0.4$  and  $\sigma = 0.6$ . Note that the data we collected were values of the angles among the hand joint expressed in radians, more or less varying in the range  $[-0.4, 1.8]$ .

### 4.3 Tree-structure and PCA representation

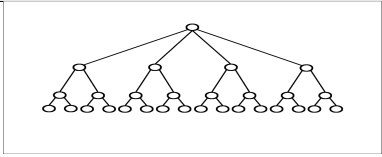
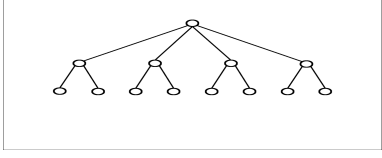
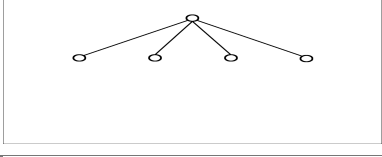
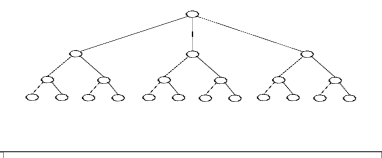
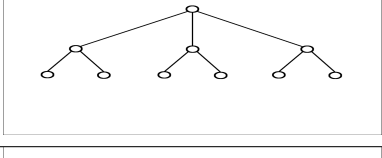
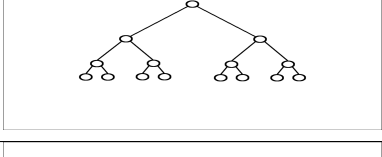
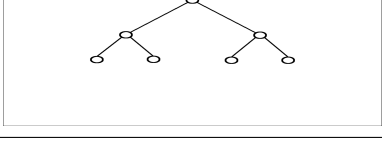
In our analysis of tree-structured TPSs representation we chose 7 kinds of different structures having different number of layers and different number of splits for each layers. The trees we used are depicted in table 4.2. To compare our representation with a more standard one we chose to compute TPSs with a PCA approach. This, as we saw in the previous chapter, is a widely used approach (Todorov and Ghahramani, 2004; Santello et al., 2002, 1998; Mason et al., 2001; Vinjamuri et al., 2010a). For each tree structure  $T_1, T_2, \dots, T_7$ , we realized a PCA action representation considering a number of principal components equal to the number of nodes in the tree. In this way we realized PCA representation with a number of principal components varying in the set  $\{29, 13, 5, 22, 10, 15, 7\}$ . It is worth to note that in a PCA-based action representation any action is typically represented using all the selected TPSs, consequently there is no specific TPSs organization.

### 4.4 Recruiting the original dictionary

Before starting analyzing our dataset we develop a test on the TSSM algorithm. We created some synthetic datasets obtained as linear combination of dictionary vectors that could be or not hierarchically organized. We expected our algorithm would be able to recruit the dictionary vectors when the synthetic data would be obtained from a hierarchically structured dictionary. For this kind of test we considered two of the tree structures in table 4.2, the tree  $T_1$  and  $T_2$ . For each of these trees we generated a synthetic  $p \times r_i$  dictionary  $\mathbf{V}$ , with  $r_i$  equal to the number of nodes in the tree  $T_i$  with  $i = 1, 2$  and  $p = 200$ . The dictionary was constructed so that atoms at the higher levels of the tree would represent more "global" aspects of the signals generated, whereas atoms at lower levels of the tree would capture more specific aspects

Grasp name	Object	Grasp Picture
Precision Grip	Usb pendrive cap	
Precision Grip	Ping-pong ball	
Precision Grip	Marking pen cap	
Prehension	Book	
Prehension	Compact-disk	
Prehension	Scotch tape	
Whole Hand	Tennis ball	
Whole Hand	Cup	
Whole Hand	Scotch tape	

**Table 4.1:** *Grasping action types.* The table shows the nine grasping action types used in our experiments.

Tree		#Levels	Level Split	Total nodes
$T_1$		3	4,2,2	29
$T_2$		2	4,2	13
$T_3$		1	4	5
$T_4$		3	3,2,2	22
$T_5$		2	3,2	10
$T_6$		3	2,2,2	15
$T_7$		2	2,2	7

**Table 4.2:** *Tree-structured synergies.* The Tree-Structured Synergy Method (TSSM) has been applied using 7 different rooted-trees  $T_1, T_2, \dots, T_7$ , these have several heights, from 1 to 3, and different number of split per level from 2 to 4.

of the signals. Accordingly, we defined the  $i$ -th atoms  $\mathbf{V}^i$  as follow (we will ignore the superscript for simplicity):

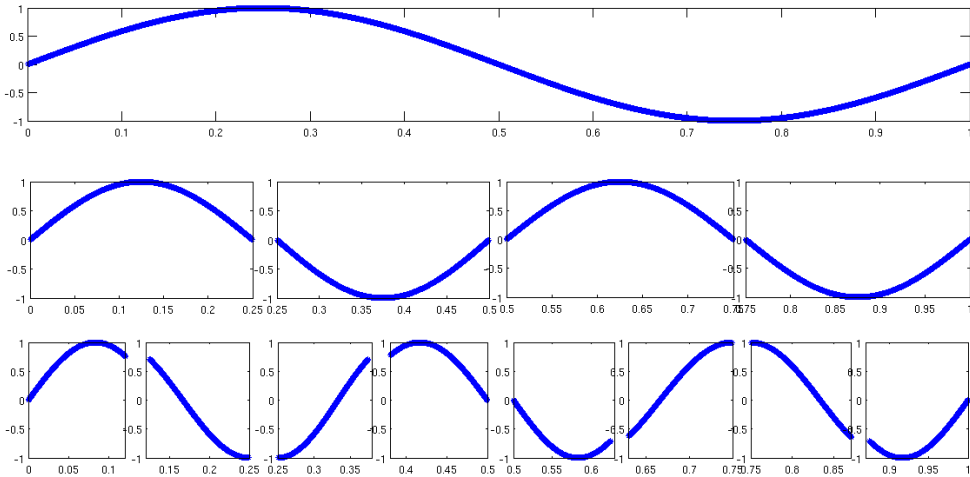


$$\mathbf{V} = \Phi(\sin(2\pi l\mathbf{x}));$$

$$\Phi(a) = \begin{cases} a & a \in [(k-1)p/Num_l, kp/Num_l]; \\ 0 & \text{otherwise}; \end{cases} \quad (4.3)$$

where  $\mathbf{x} \in \mathcal{R}^p$  assumes  $p$  equally spaced values in the interval  $[0, 1]$ ,  $l$  is the level to which the atom belongs (the root is considered at level 1),  $k$  is the position of the atom on the level  $l$  (1 correspond to the position of the most left atom in the level),  $Num_l$  equal to the number of atoms in the level  $l$ . A representation of the atoms is given in picture 4.3 for the tree  $T_2$ .

Using this dictionary two kind of synthetic datasets were generated *tree* –



**Figure 4.3:** The synthetic dictionary created using the tree structure  $T_2$ . The upper plot refers to the root atom, the plot on the second row represent the atoms in the second layer of the tree, while the ones on the third row represent the atoms of the third level. The atoms are all defined in the interval  $[0, 1]$ , but in the picture are plotted just the intervals where the atoms are different from zero.

$DS_i$  and *rand* –  $DS_i$ . More specifically for each of the tree  $T_i$  we generated 10 datasets *tree* –  $DS_i$  each composed of  $N$   $p$ -dimensional elements with  $N = 500$ . Each element was obtained as a linear combination of the atoms  $\mathbf{V}^i$  previously computed. When calculating the dataset *tree* –  $DS_i$  tree constraints were imposed on the selection of the linear combination of coefficients, moreover a value of coefficient sparsity around the 70% were forced. Likewise, we generated other 10 datasets, *rand* –  $DS_i$ , each composed of  $N$   $p$ -dimensional elements with each element computed as a linear

combination of the atom belonging to the synthetic dictionary without imposing the tree constraints and using a sparsity value about equal to 70% as before. The absolute value of all the non-zero coefficients was chosen in a random way, according to a uniform distribution, in the range  $[0.2, 1]$ . In order to compare the dictionary computed by our algorithm *TSSM* with the original dictionary we followed a procedure proposed by Aharon and colleagues (Aharon et al., 2006). For each atoms computed by our algorithm  $\mathbf{V}^k$ , we found the closest dictionary in the synthetic dataset according to the distance  $1 - |(\mathbf{V}^j)^T \mathbf{V}^k|$ . Two dictionary vectors were considered to be the same if their distance was less than 0.01. The TSSM algorithm was applied to all the 10 *tree - DS* and 10 *rand - DS* dataset for different values of the regularization parameter  $\lambda$  (tree values in the range  $[0.01, 0.1]$ ). We chose the computed dictionary having the minimum reconstruction error and a sparsity ranging in the interval  $[0.65, 0.75]$ . In the table 4.3 the mean and standard deviation of the percentage of retrieved dictionary vector is reported. It is evident in the results in the table

	TSSM mean and Std (%)
<i>Tree - DS</i>	
$T_1$	$82 \pm 10(82 \pm 10)$
$T_2$	$83 \pm 9(82 \pm 9)$
<i>Rand - DS</i>	
$T_1$	$25 \pm 8$
$T_2$	$14 \pm 6$

**Table 4.3:** The percentage of retrieved dictionary in the *Tree-DS* and *Rand-DS* datasets.

that the TSSM algorithm is able to retrieve the majority of the dictionary vectors when in the dataset is actually present a tree hierarchical organization. This is not the case in the dataset *Rand - DS* where in fact the percentage of retrieved dictionary is strongly reduced.

## 4.5 TSSM representation capacity

In this paragraph we describe a test we realized in order to compare the representation capacity of TPSs when these are computed with TSSM algorithm or with a PCA algorithm. In a first part of the test we represented the actions in our dataset according to different tree-structure and PCA representations. For each of the action representation we tested the performance of a linear

multi-class classifier in associating each action to the relative class.

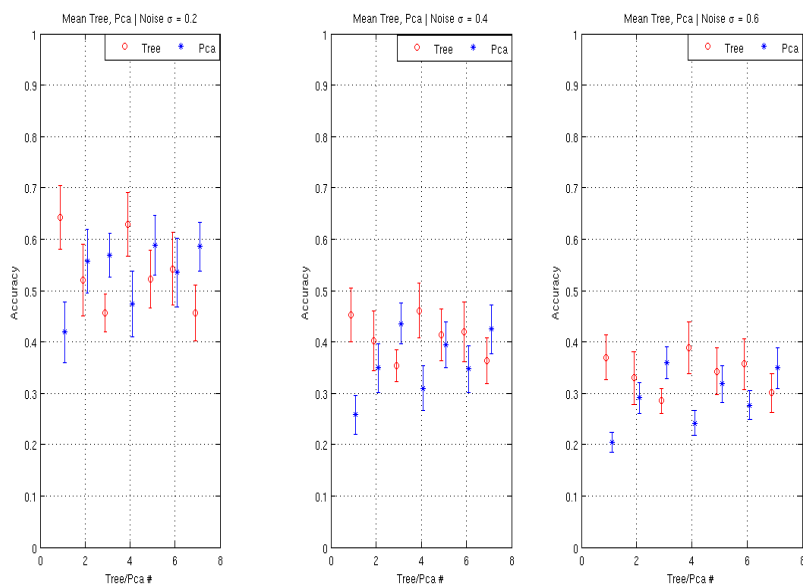
We consider to represent our data with all the trees  $T_1, T_2, \dots, T_7$ , each tree representation was compared with a PCA representation obtained respectively with 29, 13, 5, 22, 10, 15, 7, principal components.

For each training set  $TS_i$ , with  $i = 1, 2, 3, \dots, 10$ , we solved the minimization problem as expressed in equation B.1 by TSSM using in turn each of the seven trees  $T_1, T_2, \dots, T_7$ , and a regularization parameter  $\lambda$  (see eq. B.1) ranging in  $[0.01, 0.1]$  at step 0.005. The regularization parameter  $\lambda$  was chosen to obtain an high sparsity value. We consider high sparsity a mean sparsity value roughly equal to 30% of the total number of TPSs of each tree. More specifically the mean sparsity of the tree-based action representations belonging to each training set was computed as  $1 - \frac{1}{pm} \sum_{j=1}^n \| \mathbf{U}_j \|_0$ , where  $\mathbf{U}_j$  are the coefficients of the tree-based action representation for  $j$ -th action of the training set. Note that the mean value of the sparsity multiplied by the number of synergies gives the mean number of synergies used to represent each action. For the corresponding  $\lambda$  we found that the reconstruction error  $\frac{1}{2np} \| \mathbf{X} - \mathbf{U}\mathbf{V}^T \|_F^2$  was always lower than  $4 \times 10^{-3}$ .

This choice enabled us to obtain both compact and meaningful action representations. Thus, for each training set  $TS_i$  we found seven different sets of TPSs,  $S_{T_1}^i, S_{T_2}^i \dots S_{T_7}^i$ . Each action belonging to the training set  $TS_i$  was represented by seven different tree-based action representations, one for each tree. Similarly, for each action belonging to the training set  $TS_i$  we obtained seven different PCA-based action representations, one for each choice of the maximum number of principal components ( $\{29, 13, 5, 22, 10, 15, 7\}$ ). A linear SVM multi-class classifier (Hastie et al., 2009) was used to classify the action representations according to the grasping types. The SVM was trained with the coefficients of the obtained action representations. Thus, at the end of the training phase, the TPSs computed by TSSM and PCA, and the classifier’s parameters were determined.

In the test phase the actions belonging to the noisy test sets  $NT_i^1, NT_i^2$  and  $NT_i^3$  were used. Here, while leaving unchanged the TPSs computed previously in the training phase, for each action belonging to each  $NT_i^j$  we computed, similarly to the training phase, seven different tree-based action representations and seven PCA-based action representation. The classifier as determined in the training phase was fed, for each action belonging to each noisy test set, with the corresponding seven tree-based action representations and the corresponding PCA-based action representations. Finally, the performance of the multi-class classifier was measured by *classification accuracy* which is defined as the ratio between correctly classified actions over the total number of actions. Thus, for each noisy test set, we obtained: 1) for the tree-based action representations, seven classification accuracy val-

ues corresponding to the seven different trees, 2) for the PCA-based action representations, seven classification accuracy values. In figure 4.4, for each

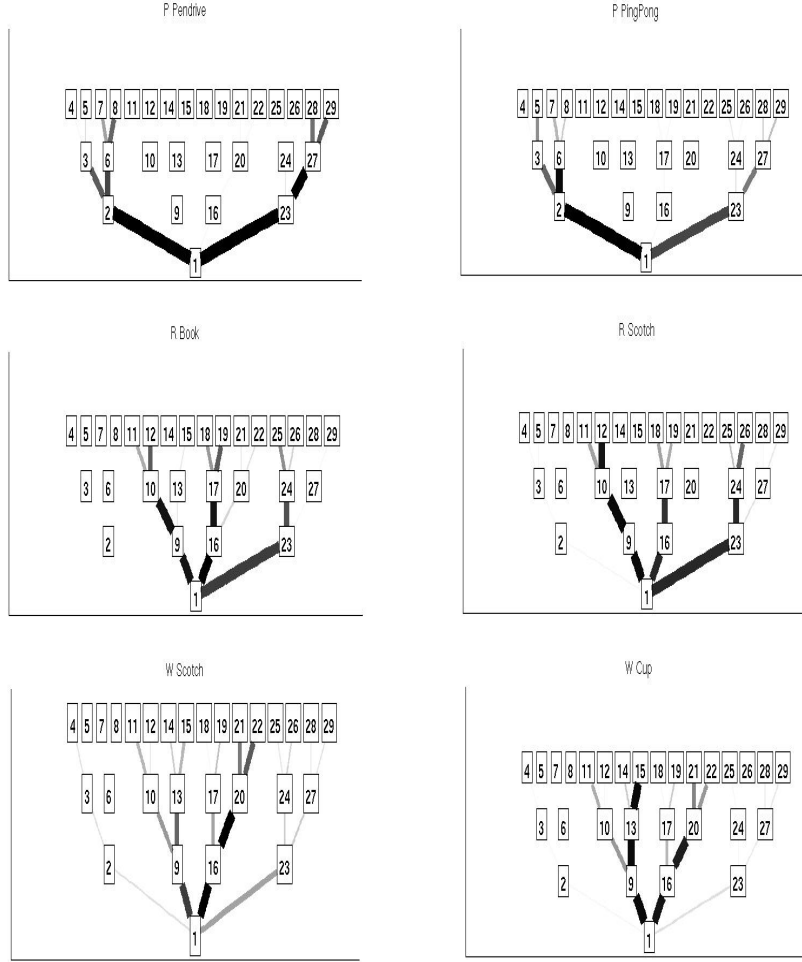


**Figure 4.4:** Plot of accuracy values (y axis), mediate on all the ten subjects, for the two methods (TSSM, in red dots, and PCA, in blue dots), for every tree structure  $T_1, T_2, \dots, T_7$

of the noisy test sets  $NT_i^1$ ,  $NT_i^2$  and  $NT_i^3$ , the seven classification accuracy values, mediated on all the ten subjects for both tree-structured and PCA action representation, are shown. As it is evident from the plots, the accuracy for the two kind of representations are nearly the same. This can be assumed as a prove that the tree structure methods is representing data at least as good as the PCA representation. Moreover in the case of the structures  $T_1$  and  $T_4$  the tree representations result to perform better then all the other representations. In the following analysis we will consider only TPSs organized according to the tree structures  $T_1$  and  $T_4$ .

## 4.6 Usage, Commonality, Selectivity

In order to verify that our algorithm is actually forcing a hierarchical structure among the synergies we developed the following analysis. At first we defined the *Usage* of a synergy. This is a measure of how much a given TPS,



**Figure 4.5:** *Subject 1: synergy usage for the trees  $T_1$ .* The figure on the top of the table shows the usage of the synergies in a tree-based action representation for six of the nine types of actions when the trees  $T_1$  is used for the subject 1. The numbered white squares organized in a tree refer to the computed TPSs. The gray level of a edge going from  $i$  to  $j$ , with  $j > i$ , represents the usage of the TPS  $j$ . Black level indicates the maximum value. If the edge is absent the synergy is not used.

$\mathbf{V}^k \in S_T^i$ , is used to represent a kind of action  $A_h$ , and is defined as follow:

$$a_k^h = usage(\mathbf{V}^k, A_h) = \frac{1}{card(h)} \sum_{j \in A_i} \|u_{jk}\|_0 \quad (4.4)$$

where  $card(h)$  is the number of actions belonging to the  $h$ -th type of actions. The *Usage* of a synergy actually correspond to the percentage of actions belonging to the class  $A_h$  that use the synergy  $\mathbf{V}^k$ . We evaluated the usage for each subject for the tree structure  $T_1$  and  $T_4$ . In order to represent the usage for an action we depicted the tree structure with the edge going for node  $i$  to node  $j$  with  $j > i$  that is more dark and thick as much the  $j$ -th TPS is used. In figure 4.5 we represent the usage of the synergies for the tree structure  $T_1$  for 6 of the 9 kind of action in the dataset. It is quite evident from the figure that different branches of the tree are used to represent different actions. Some synergies are common to more the one action class, this mainly happens for the synergies of the first level, for example synergy number 16 is used for representing whole hand actions but even for representing prehention actions. This effect is evident even in the other tree structure  $T_4$  (see figure 4.6) The representation of the *usage* seem to suggest that there are synergies used for representing more kind of actions, the synergies more near to the root of the tree, while other synergies, more near to the leafs, are specific for action type. In order to quantify this impression we defined two measures named *commonality* and *selectivity* of a TPS. The commonality of the synergy  $\mathbf{V}^k$  is defined according to the following formula:

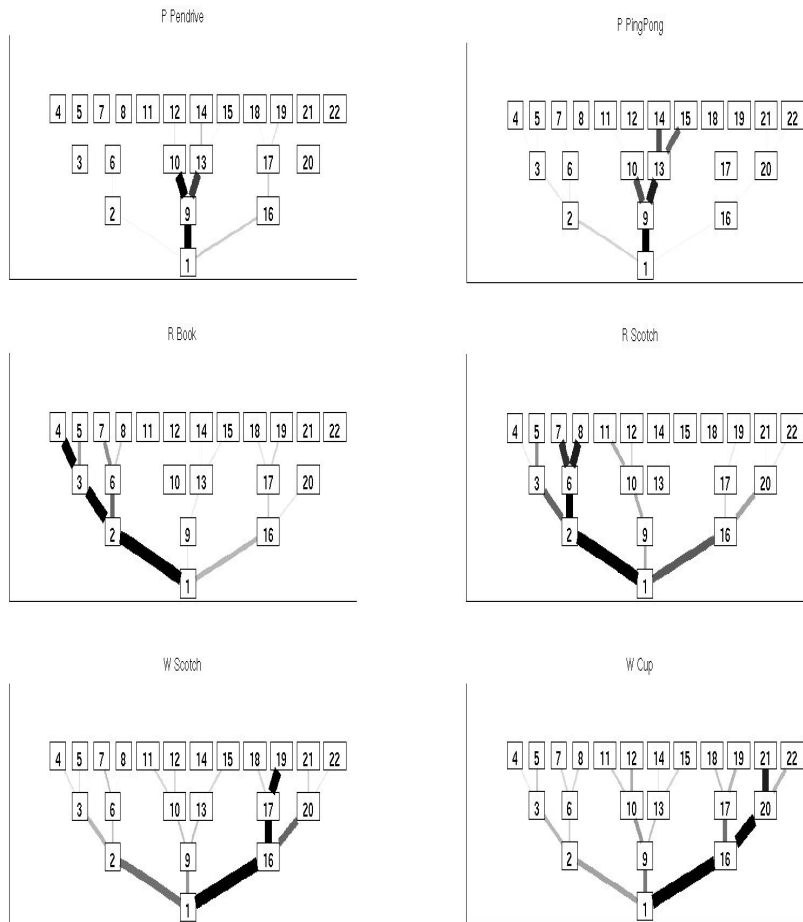
$$commonality(\mathbf{V}^k) = \frac{M_{\mathbf{V}^k}}{1 + S_{\mathbf{V}^k}} \quad (4.5)$$

where  $M_{\mathbf{V}^k}$  and  $S_{\mathbf{V}^k}$  are the mean and standard deviation of the usage values for the  $k$ -th synergy over all action types. Clearly a synergy with an high value of commonality is strongly used in more then one type of action. Since the usage takes values in the range  $[0, 1]$ , the commonality could as well take on values in the same set. A value of the commonality near to 1 means that the corresponding synergy is widely used by almost all the action types. The *selectivity* for a synergy  $\mathbf{V}^k$  is:

$$selectivity(\mathbf{V}^k) = \max_i usage(\mathbf{V}^k, A_i) - \frac{1}{C-1} \sum_{j \neq i_k} usage(\mathbf{V}^k, A_j) \quad (4.6)$$

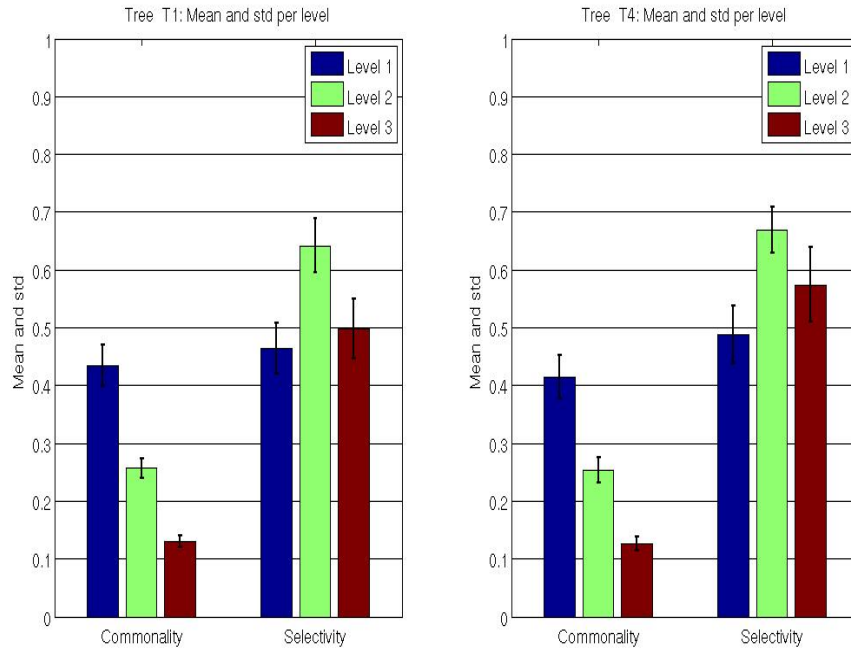
where  $i_k$  is the index of the action type for which  $usage(\mathbf{V}^k, A_i)$  assumes the maximum value. Note that also *selectivity* ( $\mathbf{V}^k$ ) lies between 0 and 1. The maximum value 1 is reached when a given synergy is used by all the actions belonging to just one type of actions.

We evaluated the commonality and selectivity for all the ten subjects for both the tree structures  $T_1$  and  $T_4$ . At first we evaluated the two measures for each single synergy, then we evaluated their mean values for each of the level of the trees. The two graphs in figure 4.7 respectively refers to the tree



**Figure 4.6:** *Subject 4: synergy usage for the trees  $T_4$ .* The figure on the top of the table shows the usage of the synergies in a tree-based action representation for six of the nine types of actions when the trees  $T_4$  is used for the subject 4. The numbered white squares organized in a tree refer to the computed TPSs. The gray level of an edge going from  $i$  to  $j$ , with  $j > i$ , represents the usage of the TPS  $j$ . Black level indicates the maximum value. If the edge is absent the synergy is not used.

$T_1$  and  $T_4$ . The three bars for each histogram represent the levels of the tree, we are not considering the commonality and selectivity of the root of the tree, since this is used in all the action representations. It is worth noting that the commonality value decreases over the tree layers, indicating that the synergies more near to the root are more common to different kinds of action



**Figure 4.7:** *Commonality and selectivity using the trees  $T_1$  and  $T_4$ .* The left (right) graph shows selectivity and commonality mean values computed for each level of the rooted-tree  $T_1$  ( $T_4$ ) used in the tree-based action representation.

possibly meaning that these synergies represent more general characteristics of action. The opposite can be said for the synergies on the leaves of the tree that should represent characteristics specific of the singular action. The selectivity values present an increasing trends only for the first two levels of the tree. This is because some synergies of the last level are used to be specific for a subset of the grasping actions of a given action type.



# Chapter 5

## Hierarchical Visuo-Motor architecture

The architecture we built, named Hierarchical Visuo-Motor (HVM) architecture, is intended to realize a mapping among the visual and motor representations of grasping actions. The architecture, modeling some aspects of the mirror system, will analyze and provide possible solutions to the problems related to the visuo-motor mapping in a biologically plausible system. In the first paragraph of the chapter we will give a precise description of the mirror system characteristics that our architecture models. We will moreover discuss of the main difficulties, from a computational point of view, of associating to an action presented in a video the relative motor representation. In the second paragraph we will quickly present the dataset we used, a more detailed presentation of the data would be given in the next chapter. In the third paragraph we will describe some possible biologically plausible mechanisms through which the computational problems previously analyzed could be overcome. In the same paragraph we will present the main computational modules that constitute the HVM architecture. Each of these modules would be described in a subsection of the paragraph. Finally in the last paragraph we will show that one of the main characteristics of the mirror neuron, the different behaviour of the broadly and strictly neurons, can be modeled by our architecture.

### 5.1 Biological systems and HVM architecture

In realizing the HVM architecture we wished to:

- model some characteristics of the mirror neuron system;

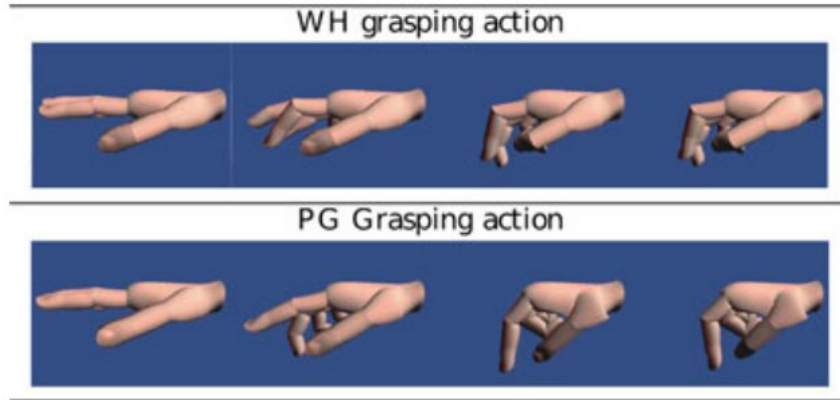
- solve the problem of mapping an action visual representation into the relative motor representation in a biologically plausible fashion.

According to the *direct matching hypothesis* (Rizzolatti et al., 2001), the motor system plays a fundamental role in the process of action recognition. Many of the experimental findings show that action observation activates a wide zone of the motor area of the observer. Notably two important things happens. The first one is that the areas used to perform an action are reactivated when the same action is observed. The second one is that during both, execution and observation, the actions are codified with different degrees of detail (from kinematic to goal codify). So there is a big evidence that in the motor area a detailed description of the action, both executed or observed is realized. To this respect we built an architecture equipped with an its own motor repertoire. This would resemble the motor codify of actions in the motor area of the brain. The motor repertoire of the HVM architecture would deeply enter in the process of visual elaboration. In particular we realized a mapping among the visual and motor representation of actions. The mapping as well as the motor representation would be hierarchically organized. From a computational point of view this kind of mapping would be particularly difficult because of two reasons: very different visual inputs are associated to the same motor act and very similar visual inputs are associated to different motor acts. The first problem rises from considering for example that the same action can be observed from totally different points of view. The second problem instead is particularly evident when considering that our dataset is constituted of grasping actions, where, due to hand self occlusion, two very similar hand images can in fact correspond to quite different grasping acts. So the brain and any architecture that wishes to model the mirror system must solve these two problems. The solution to the first problem is suggested by the *associative account* for the mirror neurons (Cook et al., 2013). This, as we explained in the first chapter, asserts that mirror neurons would be the result of an associative learning process that takes place among areas of the visual and motor cortexes. In fact there are different occasions where the same action is executed and in the mean time observed according to different perspectives. Let us think about self-observation when performing an action or imitation of an action preformed by someone else. The contemporary activation of areas encoding the visual characteristics and the motor characteristics of the same action, would bring to strengthen the connections among these areas. In the HVM architecture the problem of associating different images to the same motor act was faced faced by using an artificial neural network to realize the mapping. The network, trained on a big set of data, will possibly learn the right visuo-motor associations.

The second problem (similar images correspond to different motor acts) is solved in our architecture in the following way. The same hand image in input is associated, at least in a first computational step, to more than one motor representation. This would be actually realized in a module named *non-functional module*, using a particular kind of artificial neural network that can describe non-functional mapping. In the next paragraph a detailed description of this network would be given as well as a precise description of the use of this network in the HVM architecture. Once a visual input has been associated to more than one motor representation how can the architecture actually try to select one action representation among the multiple selected? The architecture will use its motor repertoire to develop this task, this would be realized through the use of two computational models we will refer as *spatial congruence* module and *temporal congruence* module. The spatial congruence module will check which of the action representation selected is more similar to one of the actions present in the architecture motor repertoire. As we will show more precisely later, the first two modules receive in input a video-frame for each time instant, associating to it a small group of plausible motor acts according to the architecture motor repertoire. The last module, the temporal congruence module, will collect the outputs of the previous two modules for different time instants, and will check which motor representation is more frequently present.

## 5.2 Visual and motor representation

The dataset we used for training and testing our architecture is just a subset of the one used for the experiments on the hierarchical synergy representation of chapter 3. To each action previously recorded a video of the hand as observed from a particular angle was associated. In the figure 6.1 are depicted the representation of the hand at different time instants while the subject was executing two different kind of grasping actions. In this way to each grasping action was associated: a set of motor data(i.e. the angle of all the joints of the hand for all the duration of the action) and a video of the action as observed from a specific prospective(see figure 5.1). The motor codify is the one described in the previous chapter. Each motor act was represented as an linear combination of temporal postural synergies hierarchically organized. We will use for the motor action representation, the symbol  $\mathbf{m}$ . More technical details about the tree-structured representations used will be given in the next chapter. In this chapter, in order to facilitate the explanation of the modules of the HVM architecture we will suppose that the motor representation would be obtained according to a 3 levels binary tree (see figure



**Figure 5.1:** Four frames extracted while two different actions were performed. The first set of frames refer to a whole hand (WH) grasping. The second set of frames refer to a precision grip (PG) grasp action.

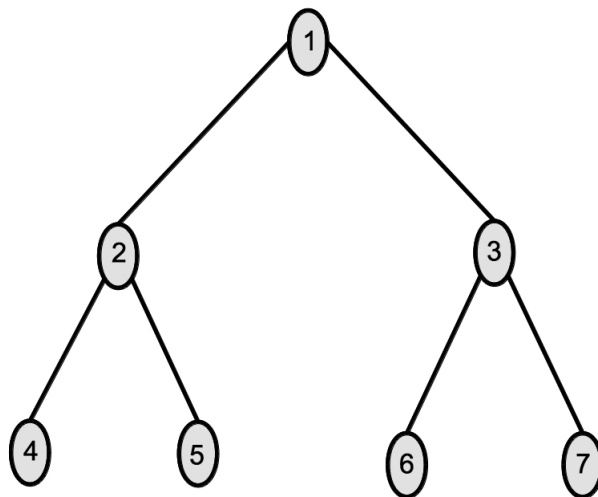
5.2), but all the modules of our architecture were in fact projected to work with any kind of tree, with any number of layers. As I said to each action was possible to associate a video.  $T$  frames at regular time intervals were extracted from each video, codified and stored in vectors  $\mathbf{v}$ . These vectors  $\mathbf{v}$  will constitute the input of the HVM architecture. Summarizing to each action we will associate:

- a motor representation vector  $\mathbf{m}$ . In the case of the tree in figure 5.2, this vector will have 7 components  $\mathbf{m} = (c_1, \dots, c_7)$ ;
- a set of  $T$  vectors, one for each video-frame, codifying the image represented in the frame:  $\mathbf{v}(1), \dots, \mathbf{v}(T)$ .

## 5.3 General overview of the architecture

### 5.3.1 When performing an action

In our motor representation each action can be described by a weighted summation of some atoms, the synergies, each atoms could be considered as describing a piece of action at a certain level of detail. Our representation of data induces a correlation among the use of the synergies. In other words there are groups of synergies that are frequently used together to represent some kind of actions and some other groups of synergies that are used to represent some other actions. In the following we will depict a way through



**Figure 5.2:** In this chapter we will consider the tree hierarchical motor representation, as obtained according to a 3 layers binary tree.

which this correlation can be exploited by our architecture in order to recognize an action shown in a video. Instead in this paragraph we wish to stress how this correlation among atoms could correspond to a correlation in the activity of some areas of the motor cortex. In fact, as stressed in the previous chapter, having found a synergies hierarchical representation of actions is an indirect prove that in the motor cortex actually the action could be codified according to synergies. This could mean that, as happen for the atoms of our representation, even among the activities of the motor cortex areas that codify synergies could exist a strong correlation. This correlation among motor areas in a subject could be established in an associative learning fashion when the subject is engaged in learning a new action or in performing an already learned one.

### 5.3.2 When observing an action

HVM architecture can be divided into two blocks, the first one, named *non-functional mapping* block is composed of two submodules, the *mixture density*

*network* module and the *spatial congruence* module (see figure 5.3). This non-functional mapping block receives in input a frame of a video and associates to it a set of motor action representations. The second block of the architecture is composed of a singular module named the *temporal congruence* module. This module collects the outputs of the first block for different time instants and returns the final output of the whole architecture. The three modules account for three different computational aspects of the HVM architecture that are enumerated in this paragraph. The first one is:

1. building a mapping among visual representation and motor representation.

The *mixture density network* module receives at each time instant a frame of a video representing a grasping act and returns the possible motor configurations associated to that video. Since as was stressed the same image can in fact be associated to different actions, we used for realizing the visuo-motor mapping a particular kind of artificial neural network, known in literature as Mixture Density Network (MDN) (Jacobs et al., 1991; Bishop, 1994). This network can account for modeling non-functional mapping. As well as action representation, even the visuo-motor mapping is developed in the HVM architecture in an hierarchical fashion. In fact it is realized through different computational modules (different MDNs), each projecting the same visual input into subsets of the action representation, each subset describing the action at different levels of detail. We will refer to this subset of the motor representation as *motor partial-representations*. This particular kind of mapping would be described in paragraph 5.4.

The output of the *mixture density network* module would be constituted by a sets of data for each level of the tree representation. Each set storing motor partial-representations. All these motor partial-representations can in fact be associated to the same video-frame. How can the architecture actually try to select one action representation among the multiple partial-representations selected? The architecture would need a way to choose/disambiguate among the different possible motor partial-representations returned as output of the MDNs. We hypothesized that the motor knowledge, helps in disambiguate by:

2. choosing the configuration that are more similar to an action present in the architecture motor repertoire;
3. choosing the motor configuration that are more frequently returned as output of the non-functional mapping block.

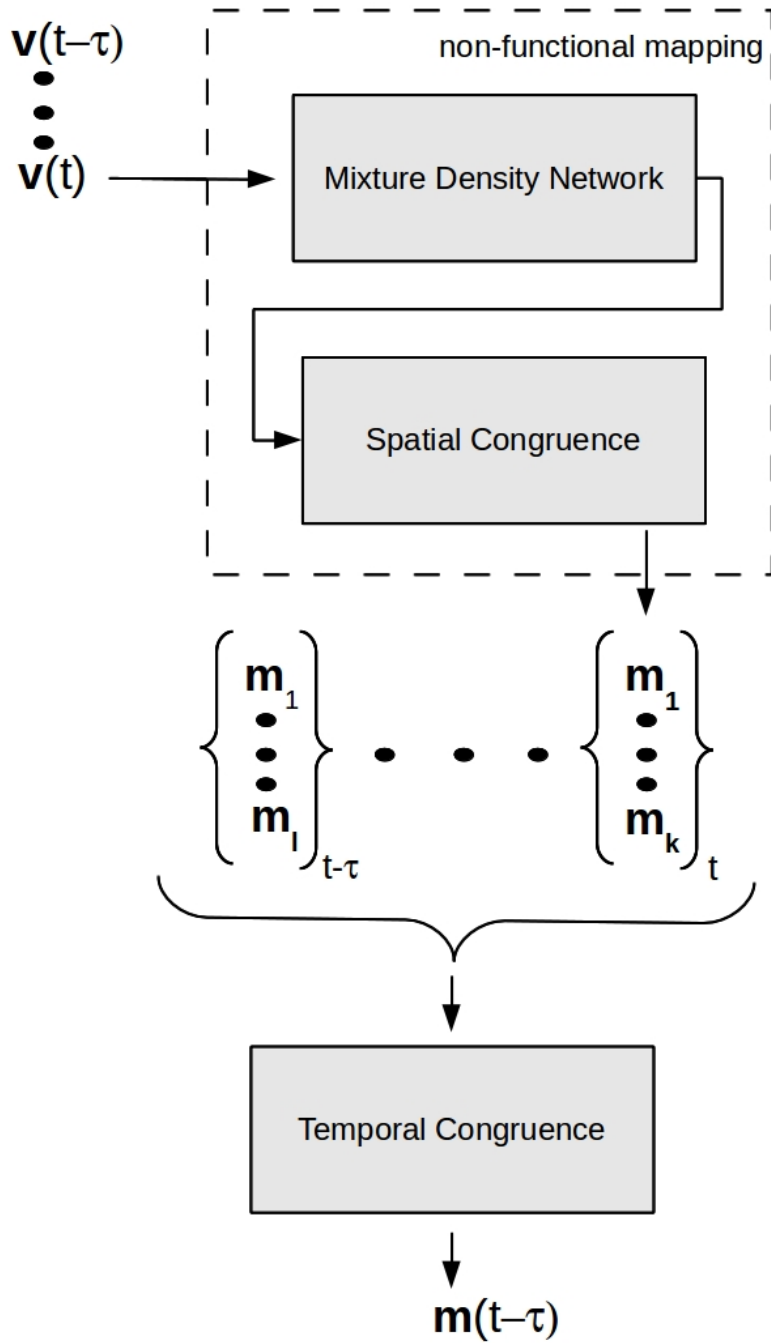


Figure 5.3: The HVM architecture with the three functional blocks

The point number 2 is realized in HVM architecture by the module named *spatial congruence* (see figure 5.3). This part of the architecture is equipped with an its own motor repertoire. It analyzes one by one the output of the *mixture density network* module and checks which of the motor partial-representations can be composed together to realize a motor configuration that actually resemble a motor configuration in its motor repertoire. The motor configuration returned as output of the spatial congruence module are indicated in figure 5.3 as  $\{\mathbf{m}_1, \dots, \mathbf{m}_l\}$ . We will describe in paragraph 5.4.3 how exactly the *spatial congruence* module works.

The last part of the architecture, named in the figure *temporal congruence* module, would check which of the motor configuration is more frequently returned as output of the previous architecture modules during the observation of an action. This part of the architecture will be described in the paragraph 5.5.

## 5.4 Non-functional mapping block

The *non-functional mapping* block is composed of the *mixture density network* module and the *spatial congruence* module. The main component of the *mixture density network* module is a MDN, this is a network that associate to an input the parameters of a mixture of gaussian as output. Details on these network could be found in the first subsection of this paragraph. In the second subsection of this paragraph we will describe how the non-functional mapping has been actually realized using more then one MDN. Finally in the last subsection we will describe the *spatial congruence* module.

### 5.4.1 Mixture density network

As we previously explained the problem of mapping the visual input in the corresponding motor data is an inverse problem, where the same or very similar inputs are associated with different outputs. One possibility to cope with a non-functional mapping could be to realize a structure that associates to an input data a probability distribution on the output set. More specifically, given a visual input  $\mathbf{v}$ , the probability of the possible output can be approximated by the probability density function  $p_{\mathbf{v}}(\mathbf{m})$ . Thus, in general, the problem of modeling a non-functional mapping can be viewed in terms of estimating the conditional probability distribution  $p(\mathbf{m}|\mathbf{v})$ . According to Bishop (Bishop and Nasrabadi, 2006), one can deal with the problem of estimating the previous probability distribution, by adopting a MDN approach. In this approach the probability distribution estimated is realized by a mix-



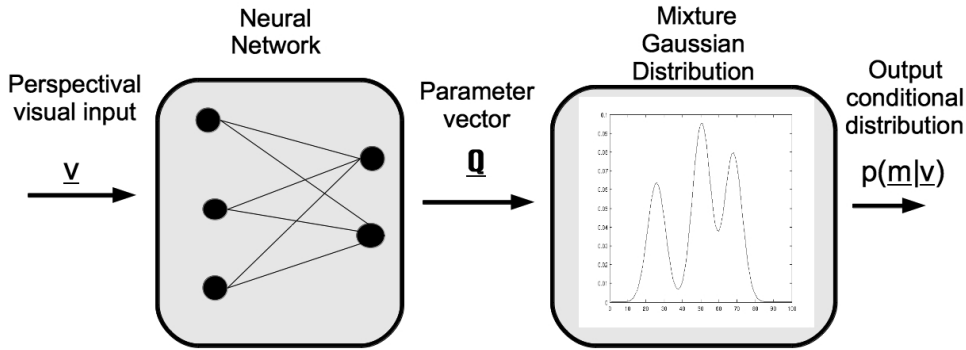
ture of gaussians:

$$p(\mathbf{m}|\mathbf{v}) = \sum_{i=1}^M \alpha_i(\mathbf{v}) \phi_i(\mathbf{m}|\mathbf{v}) \quad (5.1)$$

where the  $\phi_i(\mathbf{m}|\mathbf{v})$  are kernel functions identified with Gaussian functions of the form:

$$\phi_i(\mathbf{m}|\mathbf{v}) = \frac{1}{(2\pi)^{c/2} \sigma_i^c(\mathbf{v})} \exp\left(-\frac{\|\mathbf{m} - \mu_i(\mathbf{v})\|^2}{2\sigma_i(\mathbf{v})}\right) \quad (5.2)$$

The coefficients of the mixture,  $\alpha_i(\mathbf{v})$ , and the parameters of the kernel functions,  $\mu_i(\mathbf{v})$  and  $\sigma_i(\mathbf{v})$ , depend on the sensory input  $\mathbf{v}$ . Each gaussian is characterized by one parameter  $\sigma_i$  for the covariance, plus one for the mixture coefficient  $\alpha_i$ , plus the parameters defining the components of the mean vector  $\mu_i$ . A two layers, feed-forward neural network can be used to model the relationship between visual inputs  $\mathbf{v}$  and corresponding mixture parameters. Given a dataset of input-target couples  $\{(\mathbf{v}^1, \mathbf{m}^1), \dots, (\mathbf{v}^N, \mathbf{m}^N)\}$  the



**Figure 5.4:** Mixture density network.

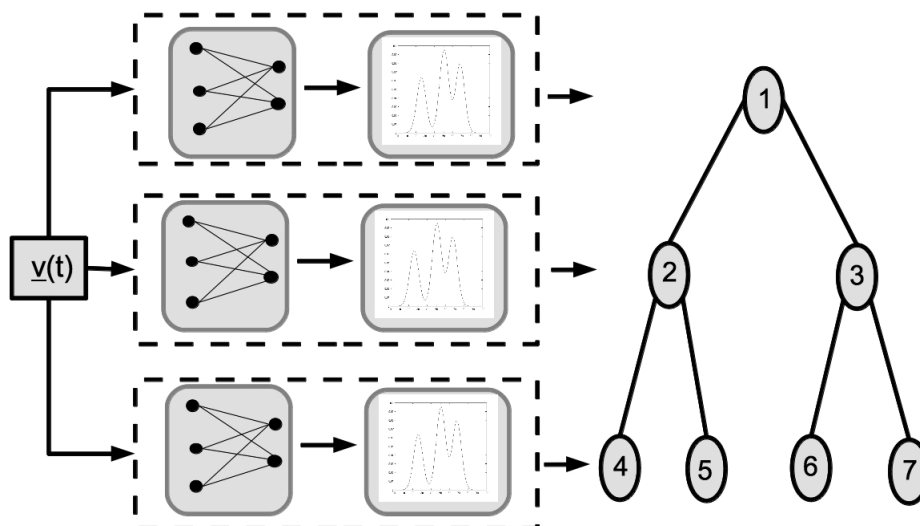
network is trained in order to minimize the negative log-likelihood of the data:

$$E = - \sum_{n=1}^N \ln \left\{ \sum_{i=1}^M \alpha_i(\mathbf{v}^n) \phi_i(\mathbf{m}^n|\mathbf{v}^n) \right\} \quad (5.3)$$

The derivative of the error function respect to the parameters of the mixture can be easily calculated (Bishop and Nasrabadi, 2006) and any gradient descent algorithm can be applied to learn the network weights.

### 5.4.2 Mixture density network module

The *mixture density network* module was realized mapping the visual input into the coefficients of the motor representation using one MDN per layer of the tree representation. In other words a first MDN was trained to realize the non-functional mapping between the visual input and the coefficient of the first layer of the tree (root layer). A second MDN mapped the visual input to the coefficients of the atoms of the second layer of the tree, and so on for all the tree layers. Considering the tree in figure 5.5. The first MDN



**Figure 5.5:** In the *mixture density network* module the viso-motor mapping is realized using a mixture density network for each layer of the tree.

receiving in input a frame of the video showing a grasping action, would return the parameter of a gaussian distribution in the one-dimensional space of the possible coefficients of the root. The second MDN will receive the same input as the first one and will return the parameter of a gaussian in a two

dimensional space, this is the space of the coefficients relative to the atoms of the second layer. Finally the last network would return the parameters of a gaussian in a four-dimensional space.

The output of the *mixture density network* module was built according to the following steps. At first we sampled  $n$  data from each of the distributions returned by the three networks, obtaining one set of data for each layer. In the case of the tree in figure 5.5 we obtained three sets of data:

$$\begin{aligned} L_1 &= \{c_1^1, \dots, c_1^n\}; & L_2 &= \{(c_2^1, c_3^1), \dots, (c_2^n, c_3^n)\}; \\ L_3 &= \{(c_4^1, c_5^1, c_6^1, c_7^1), \dots, (c_4^n, c_5^n, c_6^n, c_7^n)\}. \end{aligned}$$

$L_1$  are the  $n$  coefficients of the root sampled from the first distribution.  $L_2$  are the  $n$  two-dimensional vectors coefficient of the atoms of the second layer of the tree, in figure 5.5 the atoms are indicated with the number 2 and 3.  $L_3$  are the data sampled from the third distribution. The output of the *mixture density network* module was then obtained by considering all the possible vectors formed concatenating an element of the first set with one of the second and with one of the third. In other words by realizing the Kronecker product of the three sets, we will refer to this set as  $O_1$ :

$$O_1 \equiv L_1 \otimes L_2 \otimes L_3 = \{(c_1^i, c_2^j, c_3^j, c_4^k, c_5^k, c_6^k, c_7^k)\}_{i,j,k=1}^n.$$

The idea of realizing a visuo-motor map using a MDN for each layer of the tree is two folds. The first reason is a strictly practical one. In fact we could have used just one MDN for mapping the visual input into the motor representations, but training this network would have been more difficult since the quite big dimensionality of the output space. Even if this could not seem the case for the tree we are considering in these paragraph, the reader must envisage that in realizing a motor codify we used even quite big tree to represent the motor data, with about 30 nodes. In the next chapter we will develop some test to compare the visuo-motor mapping when realized with a single MDN or with a MDN for layer.

The second reason is the most important one and it deals with some assumptions we made about the computational characteristics of the mirror neuron system. We hypothesized that the projections from the visual area to the motor cortex are multiples and can be grouped, at least in first approximation, into groups that are mutually independent. Here in particular, considering one MDN for layer, we assumed different, parallel and independent computation that associate to the visual input a different motor representation at multiple degree of detail.

### 5.4.3 Spatial congruence module

The *spatial congruence* module has to check which of the motor representations returned as output of the mixture density module would in fact resemble actions in the architecture motor repertoire. We will give in the next chapter a lot of details on how we trained and validate the performance of HVM architecture. By now it would be sufficient to know that our dataset was mainly divided into two sets containing approximately the same number of data: a training and a validation set. The data in the training set were used to train the MDNs and to realize the architecture motor repertoire, the validation was used to test the performances of the MDNs and of the whole architecture. In order to equip our HVM architecture with a motor repertoire we proceed in the following way. We realized a probability distribution over the motor action representation in the training set. We considered all the motor representations of the data present in the training set and evaluated a probability distribution  $P_m$  over this data. In the case of our tree (see figure 5.2) we were looking for a probability distribution over the coefficients of the 7 atoms:

$$P_m = P_m(c^1, \dots, c^7)$$

The previous probability distribution, using the probability product rule, could even be rewritten as:

$$P_m(c^1, \dots, c^7) = P_3(c_4, \dots, c_7) P_2(c_2, c_3 | c_4, \dots, c_7) P_1(c_1 | c_2, \dots, c_7);$$

Where the three probability distributions  $P_1, P_2, P_3$  were so called to stress the fact that are defined respectively over the coefficients of the first, second and third layer of the tree. The probability  $P_3$  on the third level was chosen to be a mixture of gaussians.

$$P_3(c_4, \dots, c_7) = \sum_{i=1}^N \pi_i \mathcal{N}(c_4, \dots, c_7 | \mu_i, \Sigma_i);$$

The gaussians in the previous summation were the more general one in an  $\mathbb{R}^D$  space( where  $D$  is equal to 4 in this case) in other words no restriction was assumed on the covariance matrices  $\Sigma_i$ . The analytic representation of the previous gaussian is:

$$\mathcal{N}(\mathbf{x} | \mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) \right\}.$$

Where  $\mu$  is the  $D$ -dimensional mean vector,  $\Sigma$  is a  $D \times D$  covariance matrix and  $|\Sigma|$  denotes the determinant of  $\Sigma$ . In order to find the best parameter for

the  $P_3$  distribution we maximized the likelihood of the distribution over the training set using an expectation maximization algorithm (Dempster et al., 1977; McLachlan and Krishnan, 2007).

The other two probability distributions,  $P_1$  and  $P_2$ , were obtained using two MDNs. So these distributions are mixture of gaussians, where the parameters of the gaussians are function of the coefficients of the lower level of the tree. Once found the best parameters for the distributions  $P_1, P_2, P_3$  and hence having defined the distribution  $P_m$ , we could use this distribution in the *spatial congruence* module. This module received in input the set of vectors present in the output of the previous *mixture density network* module, named  $O_1$ . To each of the vector in the set  $O_1$ , let's call it  $o_i$ , was associated its probability according to the density probability distribution  $P_m(o_i)$ . Then:

- a number  $x_i$  was extracted form a uniform distribution in the interval  $[0, 1]$ ;
- if it happens that  $P_m(o_i) > x_i$  then the element  $o_i$  was included in the output set of the *spatial congruence* module, otherwise it was discarded.

After applying this procedure to all the vectors present in the set  $O_1$  we obtained a new set of motor action codify, we will call it  $O_2$  that differently form  $O_1$  contains just motor representation that are similar to the motor representation present in the architecture motor repertoire.

## 5.5 Temporal congruence module

The temporal congruence module exploits another characteristic of our motor representation. In fact as was shown in chapter 3, we are codifying actions in terms of temporal postural synergies. This means that a whole grasping act is codified with a vector of coefficients that do not change during the action development. On the other hand the first module of the architecture receives in input a video-frame of an action that is time dependent and associate to it a set of actions  $O_2$ . While  $O_2$  set is time dependent its elements, the action codifies are not. Let us consider the architecture while receiving in input the video-frames of a certain action at different time instants  $\mathbf{v}(t - \tau), \dots, \mathbf{v}(t)$ , and let us say that these are associated to an action whom motor codify is the vector  $\mathbf{m}$ . We know that HVM architecture will associate to each input a set of action representations for each time instant. We will refer to these sets as  $O_2(t - \tau), \dots, O_2(t)$ . It is plausible that the action  $\mathbf{m}$  would be present in more then one of this set, possibly in all of them (see figure 5.6). Our architecture will try to recognize the action by looking for which

action representation is more present in the different sets  $O_2$ . The *temporal congruence* module works in the following way. It evaluates the Euclidean distance between all the action representations at time  $t$  and time  $t - 1$ .

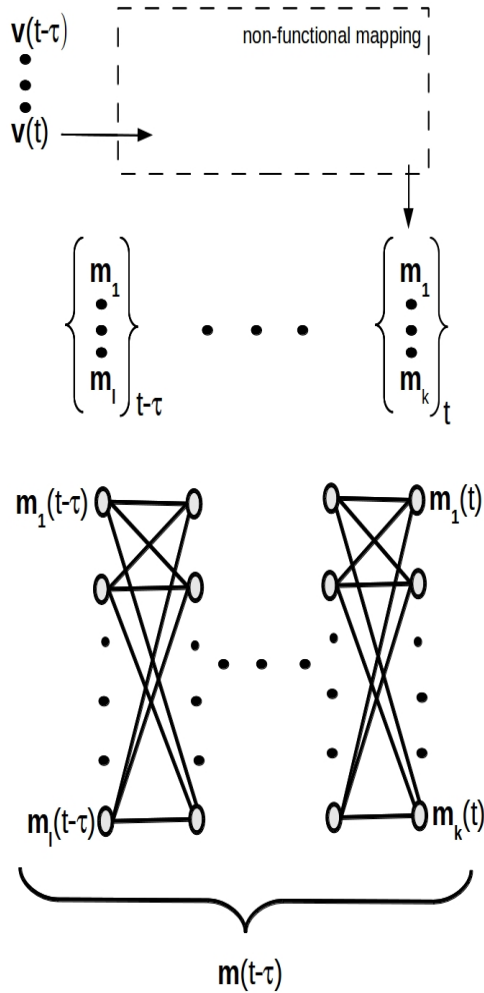
$$d_{i,j}(t) = \|\mathbf{m}_i(t) - \mathbf{m}_j(t - 1)\| \quad \forall \mathbf{m}_i \in O_2(t), \mathbf{m}_j \in O_2(t - 1)$$

It will repeat this operation for  $\tau$  time instants back in time, evaluating  $d_{i,j}(t - \tau), \dots, d_{i,j}(t)$ . In order to obtain the action representation that is more present in the different datasets we considered the graph in figure 5.6. The graph has a node for each action representation in the sets  $O_2$  for all the time instant from  $t - \tau$  to  $t$ . So it has  $l$  nodes relative to the  $l$  actions in the set  $O_2(t - \tau)$  and  $k$  nodes for the  $k$  actions in the set  $O_2(t)$  and so on for all the time instants. All the nodes of the graph relative to a given time instant are connected to all the nodes of the previous and the next time instant. The graph is a wighted one and the weights of the connections are equal to the distances previously calculated. We evaluated, using well known algorithms, the path with the minimum cost on the graph starting form a node relative to the first set,  $O_2(t - \tau)$ , to a node of the last set  $O_2(t)$ . The node of the first set  $O_2(t - \tau)$  belonging to the minimum path was assumed to be the output of the architecture at time  $t - \tau$ .

## 5.6 Strictly and broadly neurons

As we explained in the first chapter, mirror neurons are located in the motor cortex and are activated when an action is performed or when the same or very similar actions are observed. In the first chapter we analyzed different characteristics of the mirror neuron, the main one was the categorization in strictly and broadly neurons. The first ones spike when exactly the same action is observed or executed. The activity of the broadly neurons is instead associated to a bigger set of actions when these actions are observed and just to a subset of these, when the actions are executed. In the rest of this paragraph we will develop a very simple similarity between the biological system and the architecture we realized. In this way we could be able to show how our system can in fact give a possible explanation to the experimentally observed behaviour of the broadly and strictly neurons.

We could imagine that each of the atoms in the hierarchical representation could be associated to a cortical area in the motor cortex. Developing an action could consist in the contemporary activation of these areas. The coefficients of the atoms could in some way quantify the activation of the motor area associated to the relative atom. HVM architecture can be considered as having an its own motor repertoire, the training set, moreover it associates



**Figure 5.6:** In this picture there is a representation of the whole architecture with a focus on the *temporal congruence* module. In the lower part of the figure we see that to each of the  $O_2$  sets is associated a bunch of node in a graph. The weight of the connection between the nodes are given by the variables  $d_{i,j}(t)$ .

to a video of an action a motor representation for each time instant. In the chapter 3 we defined on the hierarchical action representation a *usage* measure. The *usage* quantifies how much an atom is used to represent just one kind of action or more then one action. In the next chapter we will calculate the *usage* of each atom when it is used to represent the action in the architecture motor repertoire and when is involved in the representation of

an observed action. Comparing this measures we will show that HVM architecture is able to account for the experimental observation of the broadly and strictly neurons.



# Chapter 6

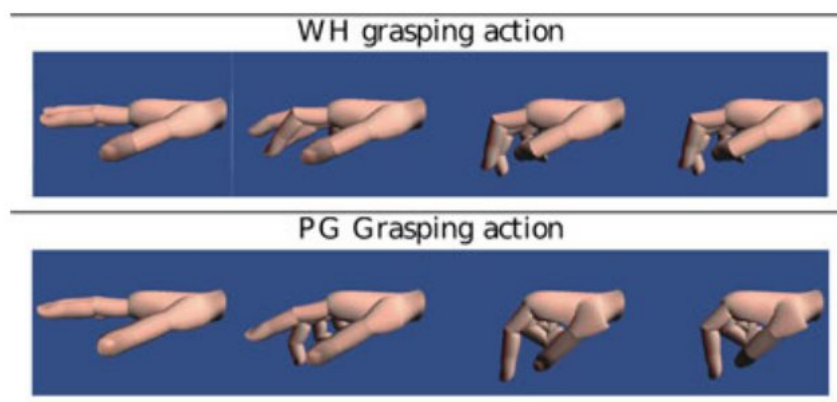
## Tests and results

In this chapter we describe the tests we did in order to verify the architecture performance and some of the assumptions we made in the previous chapter. In the first paragraphs of the chapter we describe how we collected the dataset and what kind of tree structures we used to represent the data. As we explained in the previous chapter our architecture strongly exploits a specific characteristic of the tree structure action representation, that is the strong correlation among the coefficients of this representation. In the third paragraph we describe how we tested that this correlation actually occurs. Another hypothesis we suggested in chapter 4 is that the visuo-motor mapping is a non-functional mapping. More than one test were developed for verifying this assumption. The kind of tests and the results are illustrated in the fourth paragraphs. In the last paragraphs the performances of the whole architecture are evaluated, showing that the motor involvement can actually facilitate the process of visual elaboration and that our model accounts for some characteristics of the mirror neurons, namely the different physiological behaviour between strictly and broadly neurons.

### 6.1 Details of the dataset

Precision grip (PG) and whole hand (WH) grasping actions executed by a human being were recorded by means of the HumanGlove (*HumanGlove, Humanware S.r.l., Pontedera (Pisa), Italy*) endowed with 16 sensors. This dataglove feeds data into a 3D rendering software which reads sensors values and constantly updates a 3D human hand model. Thus, this experimental setting enables one to collect pairs hand-joints configurations, hand images. Twenty PG actions and twenty WH actions were recorded, in the same experimental conditions as the ones described in chapter 3. During the execution

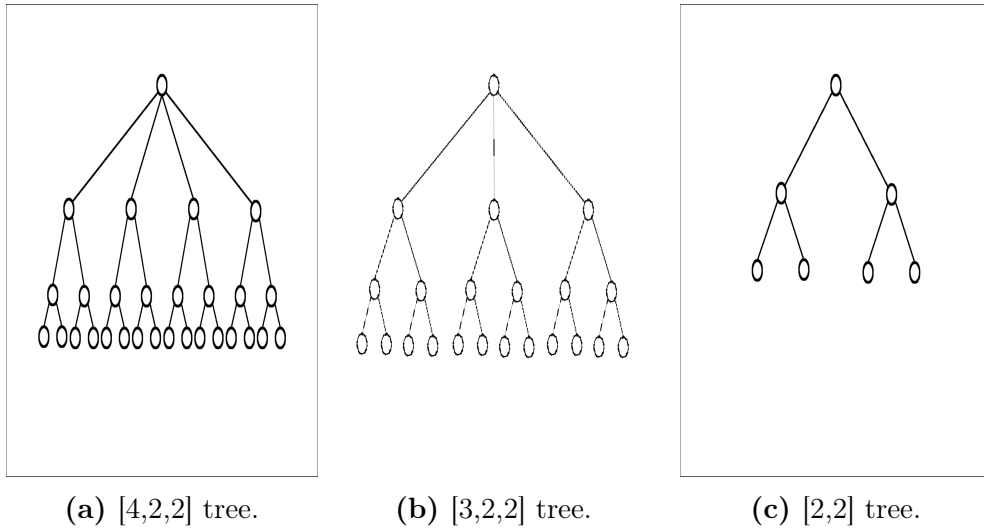
of the PG actions the subject was asked to grasp a pen cup, while during the WH action the subject was asked to grasp a tennis ball. The Human-Glove was able to record data with a frequency of  $100Hz$ . In this way to every grasping act (during about 3-4 seconds), we could associate about 400 motor-features vectors and the same number of visual-features vectors. Once we recorded all the actions, we truncated them in order to preserve only their relevant part where the hand was actually moving. We then resampled each action in both its visual and motor-features vectors in order to have the same length for every action. A sample of  $T = 30$  values was found to be sufficiently accurate to take count of the visual and motor features changes during the grasping act. The 3D simulator was set to synthesize hand configurations from just one fixed point of view. Figure 6.1 shows sample pictures extracted from the two different class actions. In order to extract vectors



**Figure 6.1:** A sample of hand configurations for the two different classes of actions.

of visual features, each image of size  $670 \times 490$  pixels was converted into a grayscale picture, subsampled at size  $151 \times 112$  pixels and linearized into a single vector of size  $1 \times 16912$ . A PCA algorithm (Bishop and Nasrabadi, 2006; Hotelling, 1933) was applied over the dataset of collected hand images and the first five principal components were computed. Each image was projected in the space of the 5 principal components resulting in a vector of  $p = 5$  visual features. The first 5 principal components were found to well represent our video-frame since explained more then the 98% of the variance of our data.

The motor-feature vectors were obtained in the same way as was described in the chapter 3, we rapidly recap here the main steps. As we said the glove is equipped with 16 sensors we preferred to not consider the wrist re-



**Figure 6.2:** The 3 structures used to represent the motor data

lated sensors reducing to 10 the number of sensors recorded. In particular sensors which measure angles of the carpometacarpal(CMC) and metacarpophalangeal(MCP) joints of the thumb and the metacarpophalangeal and proximal interphalangeal(PIP) joints of the other four fingers were considered, for a total of  $d = 10$  sensors. In this way we obtained for each action a set of  $T = 30$ ,  $d$ -dimensional motor vectors  $\{\mathbf{hc}(t)\}_{t=1}^T$ . In order to represent actions in a temporal synergy fashion we disposed the  $T$  vectors  $\mathbf{hc}(t) \in \mathbb{R}^d$  relative to the same action in sequence in the same vector, those obtaining  $\underline{m} = [\mathbf{hc}(1), \dots, \mathbf{hc}(T)]$ . Finally in this way, we associated to each action:

- a single motor vector:  $\mathbf{x} = [\mathbf{hc}(1), \dots, \mathbf{hc}(T)]$ ;
- 30  $p$ -dimensional visual vectors  $\{\mathbf{v}(t)\}_{t=1}^T$ .

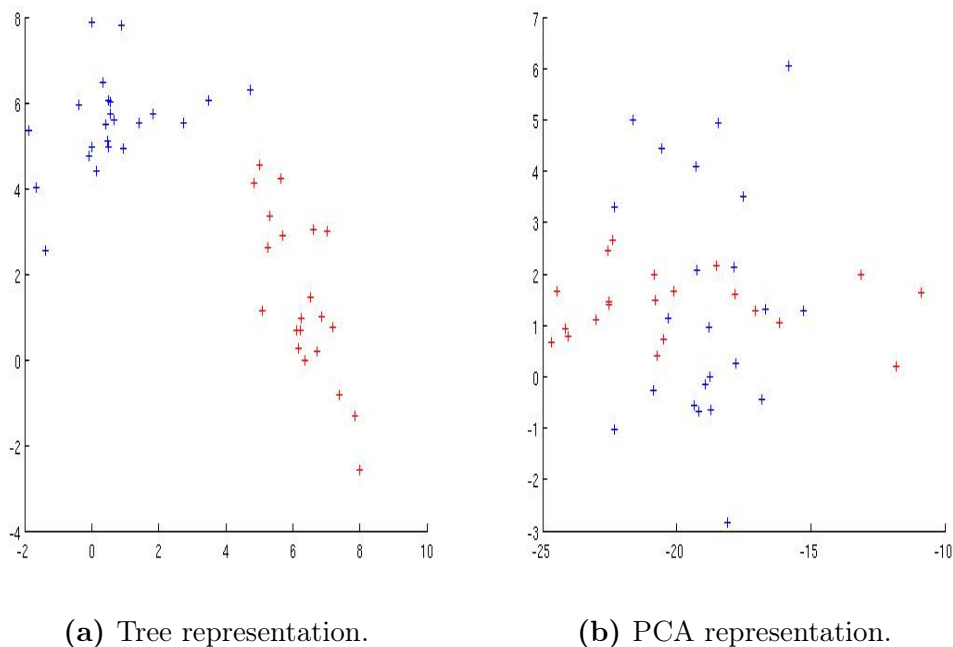
## 6.2 Motor data codify

The motor data were codified according to the tree hierarchical synergies representation described in the chapter 3. The tree structures selected to represent our data were the ones that result to perform better in the data representation according to the results shown in chapter 3. In particular we chose 3 tree structures:  $[4, 2, 2]$ ,  $[3, 2, 2]$ ,  $[2, 2]$ , where the components of the vectors indicate the number of splits for each tree level (see figure 6.2). We will refer to the motor datasets obtained with this representation respectively as:  $Tree_{29}$ ,  $Tree_{22}$ ,  $Tree_7$ . The subscript specifying the

number of total nodes in the relative tree structure. To compare the tree representations with the more common PCA representation we realized even three motor datasets obtained representing the motor data with their first 29, 22 and 10 principal components. We will refer to the datasets so obtained as:  $PCA_{29}$ ,  $PCA_{22}$ ,  $PCA_7$ . In the next paragraphs we will refer to the  $p$ -dimensional vector of visual input as  $\mathbf{v}$  and to the motor representation vector, whom size changes according to the dataset, as  $\mathbf{m}$ .

### 6.3 PCA and Tree motor representations

As we described in the previous chapter our architecture selects, among the outputs of the *mixture density network* module, the ones that are more similar to elements in its motor repertoire. To develop this task the architecture exploits a property of the tree codify, that is the strong correlation among the coefficients of the action representation. This property induced by the sparsity and tree constraints of the TSSM algorithm (see chapter 3 and appendix) is totally absent in the PCA representation. As a first prove of what

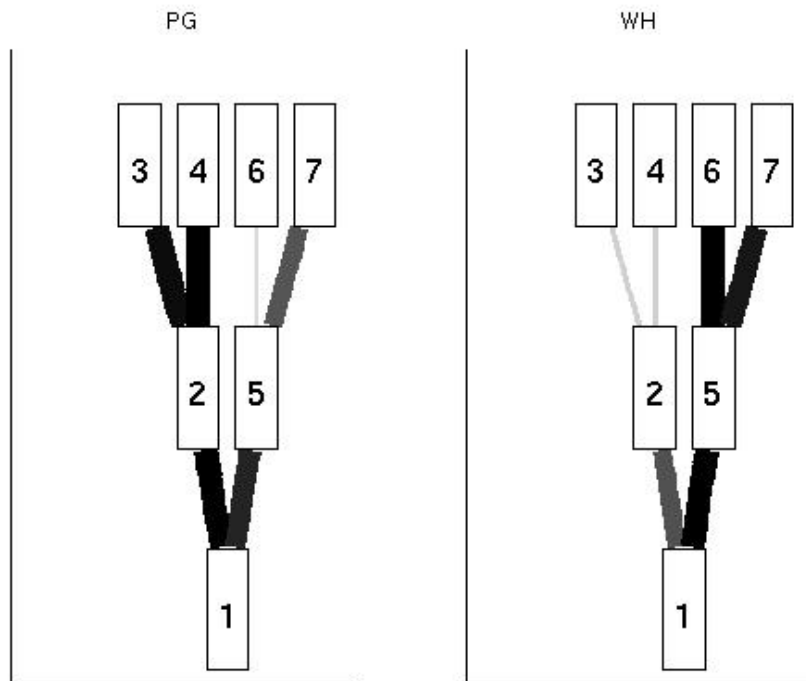


**Figure 6.3:** On the three axis the values of the second and third atom’s coefficients in the Tree and PCA representation. To each red point is associated a PG actions to the blue points the WH actions

we are claiming, we give a visual representation of our data codify in the following way. To each point in the figure 6.3a is associated one action  $\mathbf{m}$  of our dataset. The coordinates of each point are the second, third and fourth components of the action vector  $\mathbf{m}$  as represented in the tree representation [3, 2]. Those are the coefficients of the atoms of the second layer of the tree. In figure 6.3b, instead, are plotted the second, third and fourth coefficients of the actions relative to the PCA representation. The blue dots refer to precision grip actions while the red dots are relative to the whole hand actions. In the figure 6.3a our data are mainly disposed on the three axes indicating that in this representation usually when one of the coefficient is different from zero the other two are zero or very little. At the same time it's evident in figure 6.3b as in the PCA representation the coefficients are much less correlated. Another prove of the strong correlation between the coefficients of the tree representation can be obtained by representing the *usage* of each atom. This measure was presented in the chapter 3, and it quantifies how much an atom is used to represent actions belonging to a specific action class (whole hand and precision grip in this case). A good representation of the *usage* can be obtained by representing the atoms of the dictionary in a graph as in figure 6.4 where the numbered white squares organized in a tree refer to the atoms of the representation. The grey level of an edge going from  $i$  to  $j$ , with  $j > i$ , represents the usage of the atoms  $j$ . Black level indicates the maximum value. If the edge is absent, the atoms is not used. Analyzing the figure 6.4 it quite clear that the atoms 2, 3, 4 are mainly used for representing the precision grip actions, while the atoms 5, 6, 70 are used for representing whole hand actions. This results can be found even for the other representations we used: [3, 2, 2] and [4, 2, 2]. The usage representation for those are depicted in figure 6.5 and figure 6.6 respectively. Even for these other tree representations is evident that some atoms are mainly used to represent precision grip actions, while some other are used to represent whole hand actions. These clearly reflect a correlation among the coefficients of the action representation.

## 6.4 Non-functional visuo-motor mapping

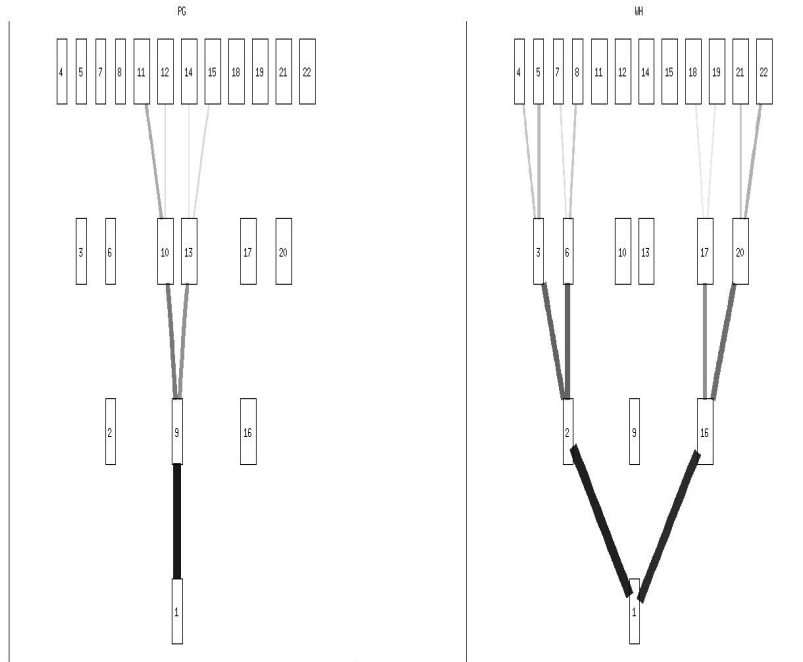
As we explained in the previous chapter one of the problem of mapping a video-frame of the hand to the relative motor representation consists in that very similar visual input are associated to different motor representation. In the next paragraphs we develop a series of test to show that this problem actually exists.



**Figure 6.4:** *Usage* of the atoms of the tree  $[2, 2]$  on the dataset composed of whole hand and precision grip actions. On the left the *usage* of the synergies relative to precision grip actions, on the right the *usage* relative to the whole hand actions.

### 6.4.1 K-means test

In this first test we wish to know if, given two similar visual data, the relative motor data are similar too or not. In a first part of the test we apply a K-means clustering algorithm (Lloyd, 1982) to both the visual and motor dataset. So obtaining a partition of both the sets into clusters. Then for each cluster in the visual dataset, we consider all the visual data belonging to the same cluster and calculate to how many different motor clusters the relative motor data belonged to. We repeat this calculus for all the clusters in the visual dataset. In this way we were able to evaluate the mean number of motor clusters associated to a single visual cluster. We repeated this test for different values of the clusters number used to partition the motor and visual datasets. In the figure 6.7 are depicted the results obtained. In the graph the mean number of different motor clusters associated to a single visual cluster are plotted vs the number of clusters used to partition the visual and motor dataset (we used the same number of clusters for both

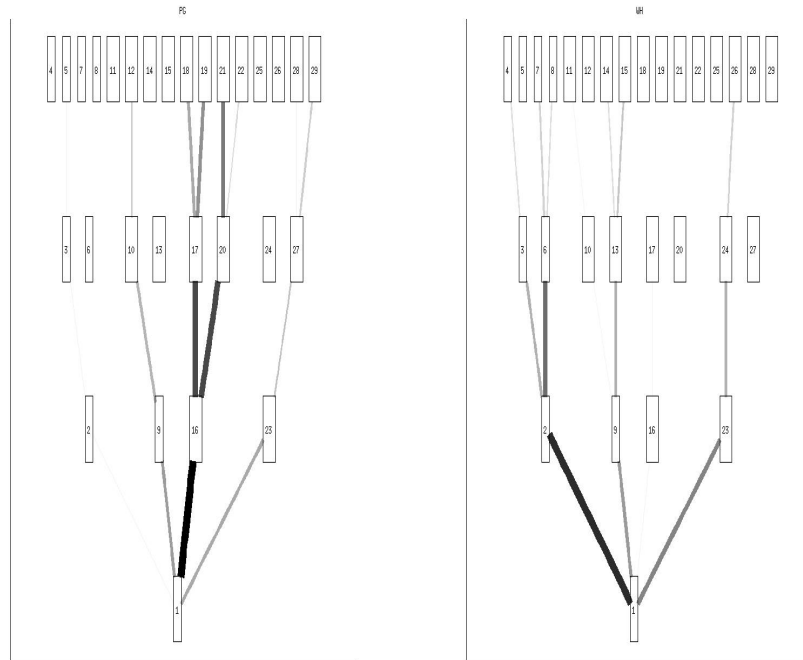


**Figure 6.5:** Usage of the atoms of the tree  $[3, 2, 2]$  on the dataset composed of whole hand and precision grip actions.

sets). If the visual and motor data would have been associated according to a functional mapping we would have obtained that the mean number of clusters associated to one visual cluster would have been always one with very few exceptions. This is clearly not the case, in fact here the mean number of clusters associated to a single visual cluster increases with the number of clusters used for partitioning the datasets.

### 6.4.2 Feed Forward mapping

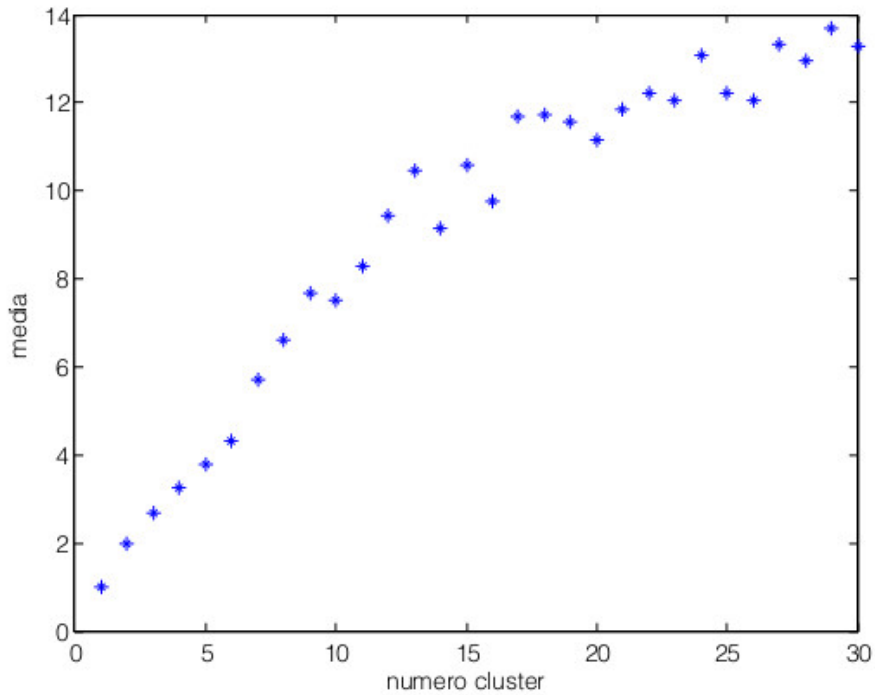
A second test to verify the non-functional relation among the visual and motor representations was developed using a Feed-Forward Network (FFNN). Here we describe the training and testing of a FFNN to realize a mapping between our visual and motor datasets. We know that those networks are, at least in principle, able to model any kind of functional relation between an input and a target, but that are not structurally intended to model multi-values functions and non-functional mapping. For this reasons we are not



**Figure 6.6:** Usage of the atoms of the tree  $[4, 2, 2]$  on the dataset composed of whole hand and precision grip actions.

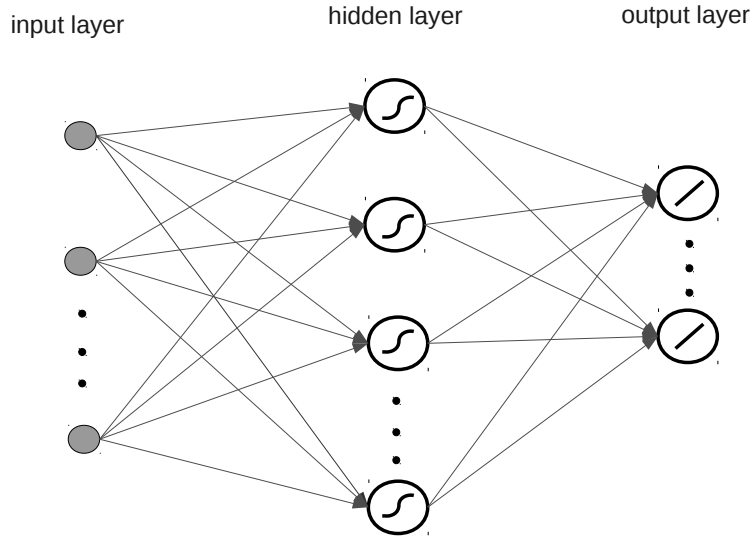
expecting a good performance of the network, and a bad result could be intended as another clue of the non-functional visuo-motor mapping. The FFNN we used is a two layers network, the hidden nodes having a sigmoidal activation function, while the output nodes have a linear activation function (figure 6.8). Since our dataset is not so big, we choose to train and test the network with a 5 fold cross-validation (Stone, 1974; Wahba and Wold, 1975). We partitioned our dataset into five subsets then we trained the network using in turn four of the five sets. The remaining set was used as validation set. The Root-Mean-Square error (RMS) (Bishop, 1995) on the training and validation sets were calculated each time, the mean and the variance were finally evaluated. The training of the network was realized using a Resilient Backpropagation algorithm (Riedmiller and Braun, 1993). This is a gradient descent algorithm where the step size is defined by the values of the gradient calculated in the previous iterations, and by two parameters chosen by the user. The RMS was evaluated for different FFNN, changing the number of hidden units of the network, and for different values of the resilient





**Figure 6.7:** In this graph are plotted the mean number of motor clusters associated to the single visual cluster versus the number of clusters with which we partitioned the motor and visual datasets.

parameters. The plot in figure 6.9 shows the best RMS values obtained for different numbers of internal nodes. The results in the plot are relative to the FFNN trained using the motor dataset *Tree<sub>22</sub>*. In red are the values of the RMS on the training set, in blue on the validation set. The values of the RMS on the training data follows the typical theoretical behaviour. That is decreasing as the complexity of the model (number of hidden node of the network) is increased. This happen because the network solves the problem of associating to similar visual input different motor configurations by having big oscillations in the output for small changes in the input. On the other hand the RMS on the validation does not follow the theoretical behaviour according to which there should be a minimum of the RMS for some optimal FFNN architecture. The high values of the RMS for the architectures with few nodes could be addressed to the excessive simplicity of the architecture,



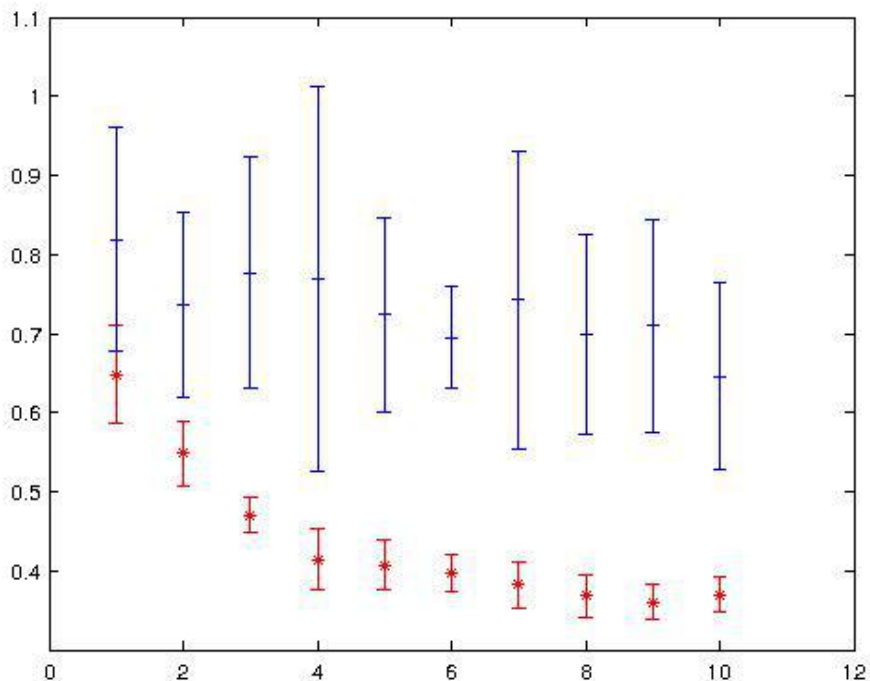
**Figure 6.8:** Structure of the FFNN we used. This is a two layers network. The activation function of the hidden nodes is a sigmoidal function. The activation function of the output node is a linear function.

but this could not be the case for the architectures with a bigger number of hidden nodes. Although this the values of the RMS on the validation set do not seem to show any decreasing or increasing trend changing the number of hidden units. We repeated the previous training for all the different datasets, namely:  $Tree_{29}$ ,  $Tree_7$ ,  $PCA_{29}$ ,  $PCA_{22}$ ,  $PCA_7$ . In the following table are indicated the best values obtained of the RMS (varying the number of hidden nodes and the Resilient parameters) for each of the datasets. As is clear form

**Table 6.1:** The best values obtained for the RMS error (varying the number of hidden nodes and the Resilient parameters)

$Tree_{29}$	$Tree_{22}$	$Tree_7$	$PCA_{29}$	$PCA_{22}$	$PCA_7$
$0.98 \pm 0.02$	$0.85 \pm 0.04$	$0.72 \pm 0.06$	$0.83 \pm 0.07$	$0.83 \pm 0.04$	$0.75 \pm 0.06$

the table, not substantial changes in the RMS are observed for the different motor representations.



**Figure 6.9:** The value of the RMS versus the number of hidden nodes of the FFW network. In red the RMS on the training set, in blue on the validation.

### One network for each layer

The test described in the previous paragraph was even repeated when the motor data were represented by considering not all the coefficients of motor representation, but just a part of these coefficients according to which layer of the tree they belong. To be more specific: we have seen previously that the motor codify is defined by dictionary vectors and their relative coefficients organized in a tree structure. In this way we could consider to group the coefficient according to the layer of the tree they refer to. Let us for example focus on the tree structure  $[4, 2, 2]$ . This is a four level tree with a number of nodes per level of: 1, 4, 8, 16. From the dataset  $Tree_{29}$  we realized four different datasets:  $TreeL1_{29}$ ,  $TreeL2_{29}$ ,  $TreeL3_{29}$ ,  $TreeL4_{29}$ . The first dataset was obtained considering for each data just the coefficient of the root in the motor representation, i.e. the coefficient of the first layer of the tree. The second dataset,  $TreeL2_{29}$ , was obtained considering just the coefficient relative two the atoms of the second layer of the tree, and so on for all the tree levels. Having realized these datasets we could try to realize a map-

**Table 6.2:** The best values obtained for the RMS error when the mapping is executed layer by layer. The motor data are codified with the hierarchical tree representation.

Level/Motor Code	$Tree_{29}$	$Tree_{22}$	$Tree_7$
First Layer	$1.2 \pm 0.3$	$1.1 \pm 0.1$	$1.30 \pm 0.17$
Second Layer	$0.90 \pm 0.08$	$0.82 \pm 0.11$	$0.71 \pm 0.17$
Third Layer	$0.97 \pm 0.04$	$0.87 \pm 0.04$	$0.76 \pm 0.03$
Fourth Layer	$1.02 \pm 0.02$	$1.01 \pm 0.02$	-

**Table 6.3:** The best values obtained for the RMS error when the mapping is executed layer by layer. The motor data are codified with the PCA representation.

Level/Motor Code	$PCA_{29}$	$PCA_{22}$	$PCA_7$
First Layer	$0.54 \pm 0.02$	$0.57 \pm 0.03$	$0.57 \pm 0.08$
Second Layer	$1.03 \pm 0.04$	$1.01 \pm 0.02$	$1.00 \pm 0.05$
Third Layer	$1.10 \pm 0.05$	$1.02 \pm 0.03$	$1.07 \pm 0.05$
Fourth Layer	$1.06 \pm 0.02$	$0.96 \pm 0.02$	-

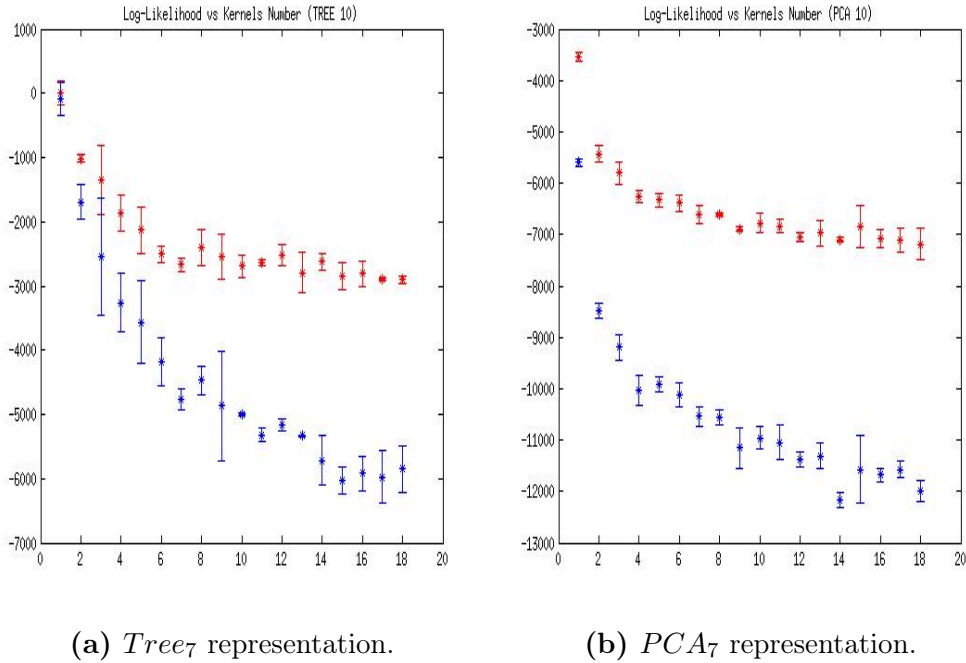
ping among the visual input and the datasets. A first FFNN, was trained to associate the visual inputs with the relative data in  $TreeL1_{29}$ . A second network was trained to associate to the visual input the four coefficients of the second layer of the motor tree representation( $TreeL2_{29}$ ), and so on till the last layer. To compare the tree representation with the PCA representation we realized different datasets by grouping the PCA components in the same way as we grouped the tree representation coefficients, obtaining even for the PCA four different datasets: $PCAL1_{29}$ ,  $PCAL2_{29}$ ,  $PCAL3_{29}$ ,  $PCAL4_{29}$ . A first FFNN was trained to associate to the visual input the first principal components(dataset  $PCAL1_{29}$ ), the second network was trained to associate to the visual input the 2-nd, 3-th, 4-th and 5-th principal components(dataset  $PCAL2_{29}$ ) and so on for the other levels. We realized each of this test for all the tree representations and the relative PCA representations. The networks were trained with a 5 fold cross-validation. The algorithm adopted was the resilient backpropagation algorithm. The training procedure was repeated for different values of the resilient parameters and the number of hidden nodes. The number of hidden layers takes on all the values in the set  $\{2, \dots, 15\}$ . In the tables 6.2 and 6.3 are shown the best values obtained of the RMS errors (by changing the number of hidden node and the resilient parameters) for each level and each representation.

As we expected the PCA performs a bit better on the first layer then the tree representations, this is because the first principal component is very

different for the two kinds of actions (PG and WH). Anyway the results remain not so good due to the non-functional mapping problems that still hold. Is interesting noting that the tree representation performs better on the second and third level, this is because as we shown in paragraph 6.3 the coefficients of PG and WH actions are better spread in the tree representation then in the PCA representation. Even here anyway the results are not so good due to the non-functional mapping.

### 6.4.3 The Mixture density network

As a first test on the MDN we just proved the ability of the network to realize the non-functional visuo-motor mapping. We trained and tested the network to realize a mapping between the visual input and the six different motor datasets:  $Tree_{29}$ ,  $Tree_{22}$ ,  $Tree_7$ ,  $PCA_{29}$ ,  $PCA_{22}$ ,  $PCA_7$ . The training was executed with a 5-fold CrossValidation. Previously we saw that the number of hidden nodes for a FFNN is associated with the capacity of mapping a more complex relation between input and output. In the case of a MDN the parameter that can be set to enhance the capacity of the network to described more complex mapping are two: the number of nodes of the network and the number of gaussian kernels. The latter are the number of gaussians that constitute the mixture of gaussians distribution returned as output of the network. We could observe, that varying the number of gaussians was much more effective to improving the performance of the MDN, then varying the number of hidden nodes. Hence for all the previously listed datasets the trainings were repeated fixing to 5 the number of hidden nodes, and changing the number of gaussian kernels of the MDN. The number of kernels was varied from 1 to 18. In figure 6.10 are reported the values of the negative log-likelihood changing the number of the gaussian kernels when the mapping is executed on the motor dataset  $Tree_7$  (plot on the left in figure 6.10) or on the motor dataset  $PCA_7$  (plot on the right in figure 6.10). In both the plots is evident that the error on the training set (blue points) continuously decreases by increasing the complexity of the model, but here differently from the FFNN, the error on the validation set (red points) first decreases until the number of kernels is less then 6, then rises or remains approximately the same when the number of kernels exceed that value. This shape, in agreement with the theoretically expected one, suggests that the architecture is consistently modeling the data. The MDN were trained and tested even on the datasets:  $Tree_{29}$ ,  $Tree_{22}$ ,  $PCA_{29}$ ,  $PCA_{22}$ , obtaining results very similar to the ones just described.



**Figure 6.10:** In the two graphs the negative log-likelihood is plotted versus the number of kernels of the MDN. The plot on the left is relative to the motor dataset  $Tree_7$ . The plot on the right to the motor dataset  $PCA_7$ . In blue the values of the training set, in red of the validation.

### One network for each layer

As I said in the previous chapter we were interested in realizing a visuo-motor mapping layer by layer. So we trained the MDN to model these mappings. In particular we tested the results of the MDN when mapping from the visual input into the motor datasets:

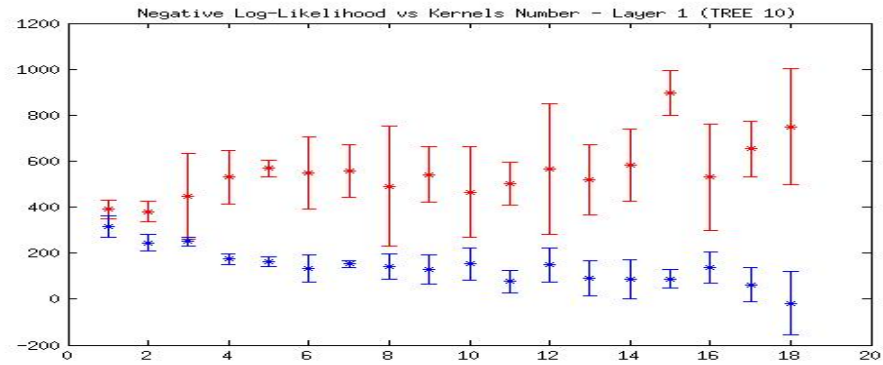
- $TreeL1_{29}$ ,  $TreeL2_{29}$ ,  $TreeL3_{29}$ ,  $TreeL4_{29}$  and  $TreeL1_7$ ,  $TreeL2_7$ ,  $TreeL3_7$ . These are the datasets obtained using respectively the tree representations  $Tree_{29}$  and  $Tree_7$  were the coefficient of the representation are partitioned according to the level of the tree they belong.
- $PCAL1_{29}$ ,  $PCAL2_{29}$ ,  $PCAL3_{29}$ ,  $PCAL4_{29}$  and  $PCAL1_7$ ,  $PCAL2_7$ ,  $PCAL3_7$ . These are the datasets obtained partitioning by layer the representations  $PCA_{29}$  and  $PCA_7$ .

As previously we realized the training using a 5-fold cross-validation and evaluating the performance of the MDN changing the number of gaussian kernels. In the plots shown in figure 6.11 and figure 6.12 are reported the values of the

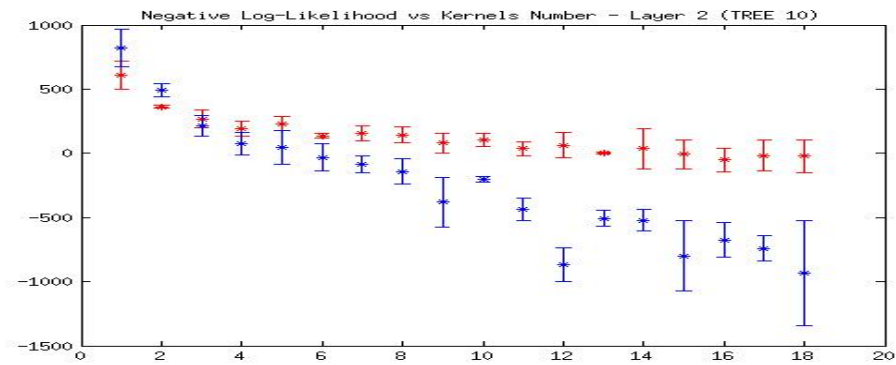
**Table 6.4:** Number of kernels for which the MDN have the smaller values of the negative log-likelihood on the validation set.

Level/Motor Code	$PCA_{29}$	$PCA_7$	$Tree_{29}$	$Tree_7$
First Layer	2	2	2	2
Second Layer	8	6	6	6
Third Layer	10	8	9	8
Fourth Layer	12	—	10	—

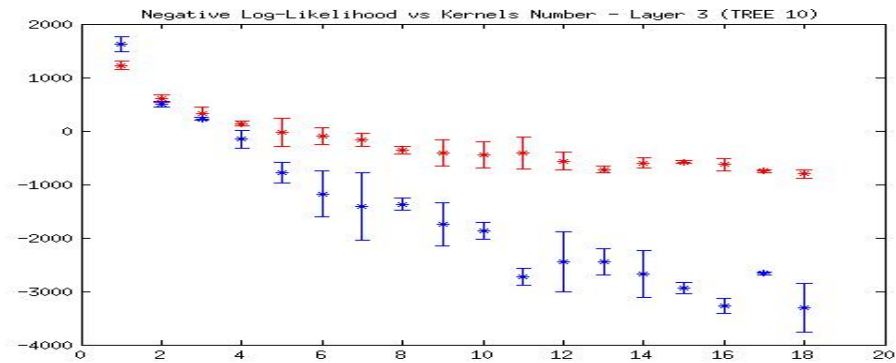
negative log-likelihood versus the number of kernels for the training and the validation set. The plots suggest that the MDN is succeeding in realizing the viso-motor mapping. This is particularly evident for the mapping on the second and third level for both *Tree* and *PCA* representations. In fact let us consider the pannels **(b)** and **(c)** of both the figures 6.11 and 6.12. In the plots when the number of gaussian kernels is increased the negative log-likelihood at first decreases for both the training and the validation set, then remains approximately constant. In particular the negative log-likelihood stops decreasing when number of kernels exceeds a value of about 6 for the mapping relative to the second layer (plots **(b)** on both figures) and about 8 for the mapping on the third layer (plots **(b)** on both figures). The plot relative to the first layer should be considered a bit more carefully. In the case of the *Tree* representation, plot **(a)** figure 6.11, the values of the validation seem to have more or less a continuous growing trend. This would probability means that a very simple MDN with one or two kernel could be sufficient to represent our data. In the case of the *PCA* representation, plot **(a)** figure 6.12, the negative log-likelihood at first has a strong decrease, passing form a MDN with one kernel to a MDN with two kernels, then start increasing. The strong initial decrease observed could be explained considering that our motor data are relative to two motor classes, precision grip actions and whole hand actions. Probably this two kinds of actions can be very well partitioned according to the first *PCA* component into two sets. Similar tests to the ones just described were represented even for the other *Tree* and *PCA* representations:  $TreeL1_{29}$ ,  $TreeL2_{29}$ ,  $TreeL3_{29}$ ,  $TreeL4_{29}$  and  $PCAL1_{29}$ ,  $PCAL2_{29}$ ,  $PCAL3_{29}$ ,  $PCAL4_{29}$ . The results obtained were quite similar to those represented in the just described plots. These results gave as the possibility to select the MDN that better performs in the mapping. In other word the MDN with the smallest values of the negative log-likelihood on the validation set and the smallest number of kernels. In the table 6.4 are reported the best architectures for each mapping.



(a)  $TreeL1_7$  representation.



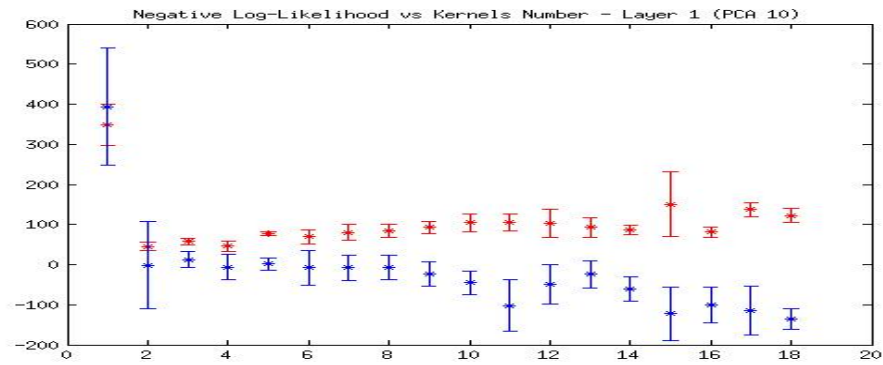
(b)  $TreeL2_7$  representation.



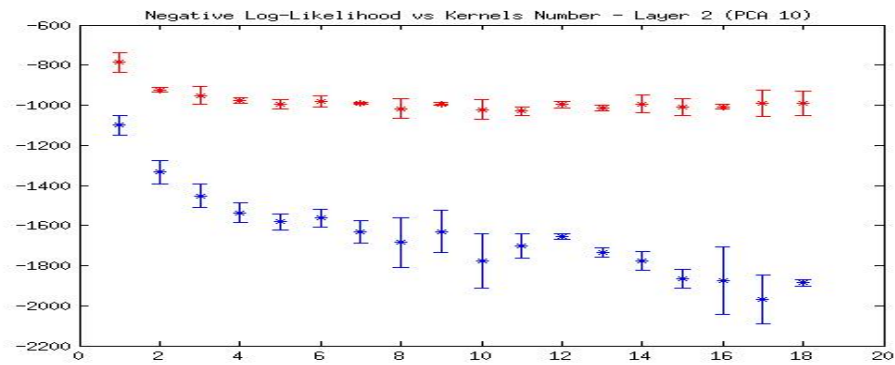
(c)  $TreeL3_7$  representation.

**Figure 6.11:** The values of the negative log-likelihood versus the number of kernels of the MDNs. In red the values relative to the validation set, in blue to the training set. The plots refers to the motor representation  $Tree_7$ . In particular the plots (a), (b), (c) respectively refers to the mapping of the visual input into the coefficient of the first, second and third level of the tree.

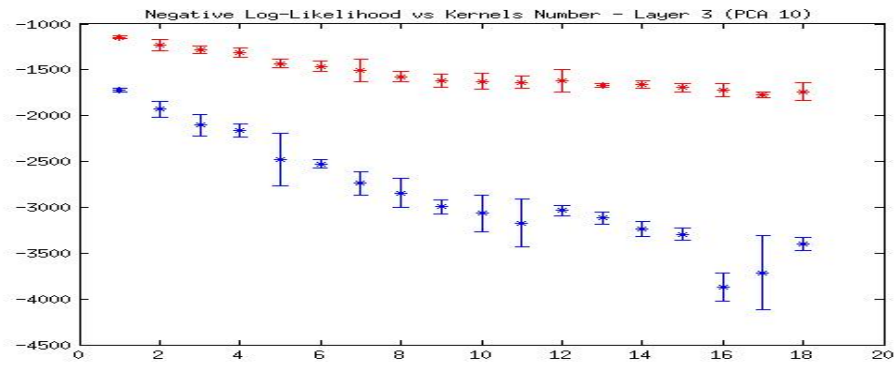




(a)  $PCAL_7$  representation.



(b)  $PCAL_2_7$  representation.



(c)  $PCAL_3_7$  representation.

**Figure 6.12:** The values of the negative log-likelihood versus the number of kernels of the MDNs. In red the values relative to the validation set, in blue to the training set. The plots refers to the motor representation  $PCA_7$ . In particular the plots (a), (b), (c) respectively refers to the mapping of the visual input into the coefficient of the first, second and third partition of the  $PCA$  representation.

**Table 6.5:** The RMS obtained when to a visual input is associated the most probable value of the relative distribution.

Motor Code	$Tree_7$	$Tree_{29}$	$PCA_7$	$PCA_{29}$
First Layer	$1.1 \pm 0.1$	$1.2 \pm 0.2$	$1.1 \pm 0.1$	$1.0 \pm 0.2$
Second Layer	$1.20 \pm 0.09$	$1.2 \pm 0.1$	$1.20 \pm 0.09$	$1.19 \pm 0.08$
Third Layer	$1.13 \pm 0.03$	$1.2 \pm 0.1$	$1.13 \pm 0.03$	$1.15 \pm 0.04$
Fourth Layer	-	$1.09 \pm 0.05$	-	$1.05 \pm 0.02$

#### 6.4.4 The ambiguity is described but not solved

As we described in the previous paragraph the MDN associates to a visual input a distribution over the motor space. As we said we used the negative log-likelihood to evaluate the different MDN performances. This makes us quite confident that, for a MDN with a low value of the negative log-likelihood, the output distributions will well describe the dispersion in the motor space of the motor data associated to the same or to very similar visual inputs. Although this we wish to stress that we are not suggesting that the network is solving the problem of associating to a visual input  $\mathbf{v}_i$  the relative motor representation  $\mathbf{m}_i$ . In fact, given  $\mathbf{v}_i$  as input to the network, we are not expecting that the relative motor data  $\mathbf{m}_i$  will always be the most probable according to the output distribution  $p(\mathbf{m}|\mathbf{v}_i)$ . We just expect that the value of the probability of  $\mathbf{m}_i$  will be in accordance with the distribution of the motor data that have the same or very similar visual input  $\mathbf{v}_i$ . So summarizing we are suggesting that the MDN:

- is not able to solve the visuo-motor ambiguity by itself;
- is well characterizing the non-functional mapping between the visual and motor representation.

To have some more convincing evidences of the previous hypothesis we realized two tests for the datasets:  $Tree_7$ ,  $Tree_{29}$  and  $PCA_7$ ,  $PCA_{29}$  partitioned by layer. For each of the dataset we chose the best architecture according to the results of the previous paragraph and develop the following test. For each visual input  $\mathbf{v}_i$  we sampled 100 elements from the distribution associated to the input. We then selected among the sampled values the once with the biggest probability. In this way we could associate to every visual input a motor data in the previously listed datasets, and evaluate the RMS for this kind of mapping. The values in the table 6.5 refers to the  $Tree_7$ ,  $Tree_{29}$  and  $PCA_7$ ,  $PCA_{29}$  partitioned according to layers. As we expected we obtained quite high values for the RMS error. This validate our hypothesis according

**Table 6.6:** The RMS obtained when to a visual input is associated the motor data that results to be the most near to the motor target.

Params/Motor Code	$Tree_7$	$Tree_{29}$	$PCA_7$	$PCA_{29}$
First Layer	$0.20 \pm 0.02$	$0.26 \pm 0.01$	$0.08 \pm 0.02$	$0.05 \pm 0.01$
Second Layer	$0.30 \pm 0.03$	$0.30 \pm 0.02$	$0.40 \pm 0.4$	$0.37 \pm 0.05$
Third Layer	$0.42 \pm 0.07$	$0.50 \pm 0.03$	$0.81 \pm 0.01$	$0.77 \pm 0.01$
Fourth Layer	-	$0.90 \pm 0.02$	-	$1.04 \pm 0.01$

to which the use of MDN is not enough to solve the visuo-motor ambiguity. The second test wished to verify that the MDNs were well describing the distribution of motor data relative to the same input. The visuo-motor mapping was realized according to the following procedure. We associated to each visual input the probability distribution returned as output of the MDN, then we drew very few samples from this distributions, 5 samples, and we associated to the visual input the sample more near to the relative motor target. We again evaluate the RMS for this kind of mapping. In the following tables are expressed the results we found: The values of the RMS in table 6.6 are much lower then the ones obtained in the previous table. Clearly we were expecting a decreasing of the RMS since we are choosing among the 5 motor samples the one that is more near to the target. Otherwise the decreasing of the RMS seem remarkable considering that it is obtained with very few samples. In the table is even worth noting that the best values of the RMS are obtained for the first layer of the PCA representations. Confirming the hypothesis that the first PCA component well partition the two classes of actions. Otherwise on the other levels the *Tree* representation performs always better then the *PCA*.

## 6.5 Motor involvement

Summarizing the previous results we can say that the mapping between the visual and motor data is actually a non-functional mapping. The MDN seem to be able to associate to a visual input a probability distribution over the space of the motor representation, that actually represent the motor data associated to the same visual input. We showed that this is true for both the motor representations *Tree* and *PCA*, even when the mapping is executed layer by layer. On the other hand these networks do not seem able by themselves to associate the visual input to the relative motor data. This was confirmed by the high values of the RMS when to a visual input  $\mathbf{v}$  the most probable motor data, according to the distribution  $p(\mathbf{m}|\mathbf{v})$ , was

associated.

In this paragraph we develop a test to show that the process of visuo-motor association can improve if some extra information on the motor codify are available. We have developed this test on two motor representations:  $Tree_7$  and  $PCA_7$ . In this test we exploited a probability distribution  $P_M(\mathbf{m})$  we defined over the space of motor  $Tree$  representations. The analytic form of this distribution is given in the previous chapter, where we even stressed how this distribution can be considered as representing the motor repertoire of the architecture. The test we developed is the following. Each visual data was given as input to the three MDNs, one per layer of the  $Tree_7$  and  $PCA_7$  representations, obtaining three probability distributions. From each of this distribution we drew 5 samples obtaining one set of data for each layer. As in the previous chapter we indicate these sets as:

$$\begin{aligned} L_1 &= \{c_1^1, \dots, c_1^5\}; & L_2 &= \{(c_2^1, c_3^1), \dots, (c_2^5, c_3^5)\}; \\ L_3 &= \{(c_4^1, \dots, c_7^1), \dots, (c_4^5, \dots, c_7^5)\}. \end{aligned}$$

We stored for each sampled data the relative probability. Possible motor representations associated with the visual input were obtained considering all the possible vectors that could be formed concatenating an element of the first set with one of the second and one of the third. So obtaining the set

$$O \equiv L_1 \otimes L_2 \otimes L_3 = \{(c_1^i, c_2^j, c_3^j, c_4^k, c_5^k, c_6^k, c_7^k)\}_{i,j,k=1}^5.$$

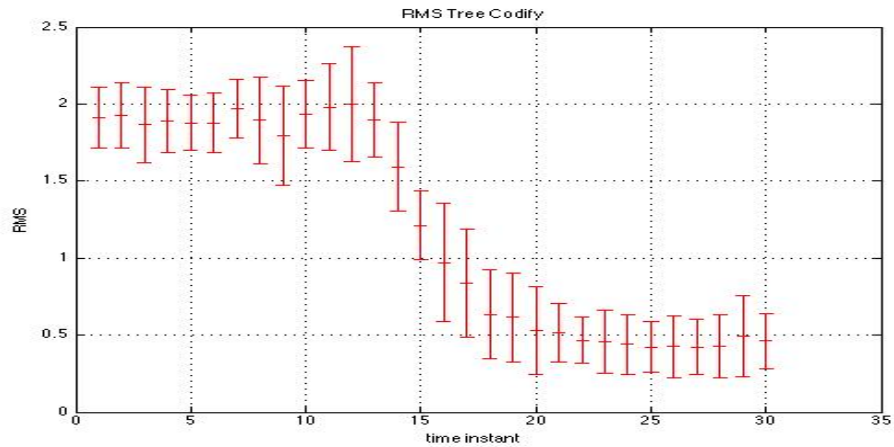
To each of the 125 vectors in the set  $O$  was associated the probability obtained as the product of the probability of the three element forming the vector. We will refer to this probability as  $p(\mathbf{m}|\mathbf{v})$  were  $\mathbf{m}$  is an element of the set  $O$ . The process until this step was totally equivalent for the  $Tree$  and  $PCA$  representation. At this point when using the  $PCA$  motor representation, we chose as motor representation,  $\mathbf{m}^*$ , to associate to the visual input  $\mathbf{v}^*$  the one with the highest value of the probability  $p(\mathbf{m}|\mathbf{v}^*)$ .

$$\mathbf{m}^* = \max_{\mathbf{m}} p(\mathbf{m}|\mathbf{v}^*). \quad (6.1)$$

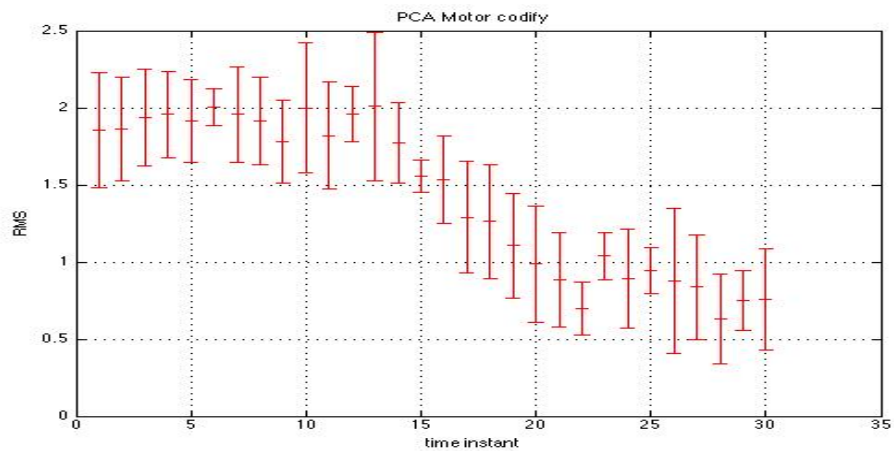
When the data were represented with the  $Tree$  representation a further step was developed. For each element in set  $O$  we evaluated its probability according to  $P_M$ . We finally associated to the visual element in input,  $\mathbf{v}^*$ , the element of the set  $O$  that result to have the biggest values of the product of the two probability  $P_M(\mathbf{m}) \cdot p(\mathbf{m}|\mathbf{v}^*)$ .

$$\mathbf{m}^* = \max_{\mathbf{m}} P_M(\mathbf{m}) \cdot p(\mathbf{m}|\mathbf{v}^*). \quad (6.2)$$

The plots in figure 6.13 show the values of the RMS for the two motor representations. The value of the RMS were evaluated for each of the 30 time frames describing a grasping action. We could evaluate in this way the visuo-motor mapping time by time during the grasp. Different considerations



(a)  $Tree_7$  representation.



(b)  $PCA_7$  representation.

**Figure 6.13:** The values of the RMS versus time. In subplot (a) the RMS relative to the  $Tree$  codify of the motor data. In subplot (b) the RMS for the  $PCA$  representation.

can be done on the two plots of figure 6.13. The first one is that for both the codify the values of the RMS remains high for the first 10-15 time instants. This is probably due to the fact that the two grasping actions are too similar at the beginning of the action and the involvement of the motor repertoire

as happen for the *Tree* representation cannot improve that much the visuo-motor association. The values of the RMS quickly decrease as the time takes values bigger then 15. In the case of *Tree* representation the descres is a bit more fast and the values reached by the RMS in the last time instant are smaller that the ones observed for the *PCA* representation.

## 6.6 Broadly and strictly neurons

The process described in the previous paragraph actually realizes a mapping among the visual inputs and the motor representations. In chapter 3 we defined the *usage* of a TPS (temporal postural synergy) as a way to measure how much a synergy is used to represent actions belonging to different classes. Now we are able to measure the *usage* even when an action is observed. In other words we can measure how much a TPS is used to represent an observed action. In chapter 3 we stressed that the TPSs evaluated for action representation could in fact be codified by some neural areas in the central neural system. In this way we could hypothesize that the use of a TPS in an action representation could consist in the activation of the motor area that codifies that synergy when the action is performed. According to this similarity among TPSs and motor areas, we could consider the different usage of a synergy when codifying an action observed or an action in the motor repertoire. We could consider if the different usage resembles the neurophysiological behaviour of strictly and broadly neurons.

The usage of the TPSs during action observation was evaluated in the following way. We collected the video frames relative to three different time instants into three sets.

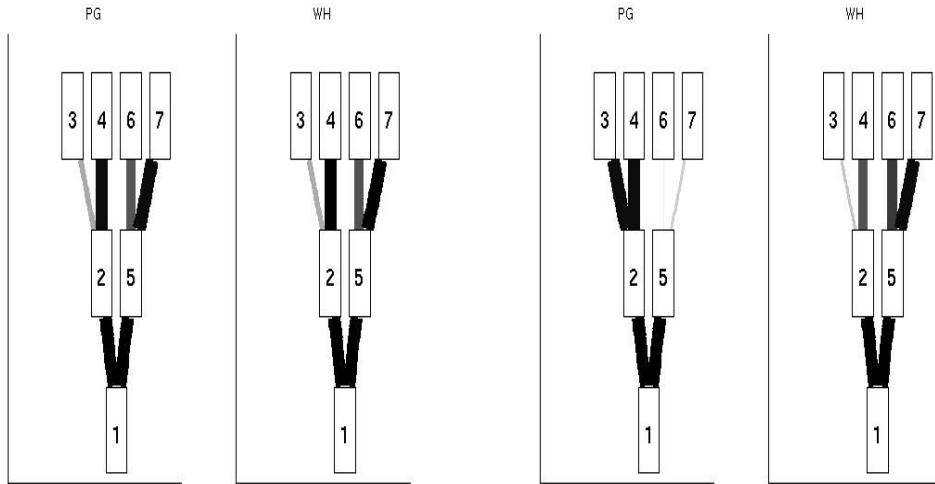
$$V_{10} = \{\mathbf{v}_i(10)\}_{i=1}^n \quad V_{20} = \{\mathbf{v}_i(20)\}_{i=1}^n \quad V_{30} = \{\mathbf{v}_i(30)\}_{i=1}^n$$

In the first set  $V_{10}$  are collected all the frames relative to the 10-th time instant, in the second set all the frames relative to the 20-th time instant and finally in the third set all the frame relative to the 30-th time instant. To each of these visual set a motor set was associated. To each visual representation a motor representation was associated according to the visuo-motor mapping described in the previous paragraph. In this way we obtained tree sets of motor representations at different time instants. These are the motor representation of observed actions.

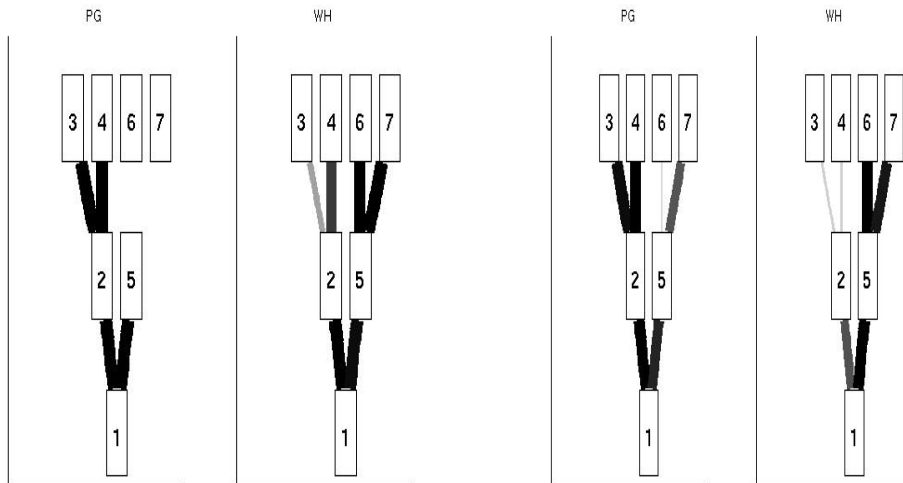
$$M_{10} = \{\mathbf{m}_i^{10}\}_{i=1}^n \quad M_{20} = \{\mathbf{m}_i^{20}\}_{i=1}^n \quad M_{30} = \{\mathbf{m}_i^{30}\}_{i=1}^n$$

Where the action representation indicated with  $\mathbf{m}_i^{10}$  is the one associated with the visual frame  $\mathbf{v}_i(10)$ .

We evaluated the *usage* of the TPSs over the 3 motor sets  $M_{10}$ ,  $M_{20}$ ,  $M_{30}$ , considering the motor representation [2, 2]. In figure 6.14 the usage is presented with the graphic description (see chapter 3 for details) for the 3 sets together with the usage of the TPSs when representing a data in the architecture motor repertoire. Considering the figure 6.14a, we can see that at



(a) *Usage*: observed action at time 10. (b) *Usage*: observed action at time 20.



(c) *Usage*: observed action at time 30. (d) *Usage* of motor representation.

**Figure 6.14:** *Usage* relative to motor representations of observed action at 3 different time instants (respectively in the subplots (a), (b), (c)), and of motor representation in the architecture motor repertoire (subplots (d)).

the beginning of the action (time instant 10) the system cannot distinguish the precision grip and whole hand action, and uses the TPSs more or less in the same way for representing both the actions. Let us consider instead the last time instant (time instant 30) of action observation. In figure 6.14c is evident that the system is using the synergies on the left part of the tree for representing precision grip actions and synergies on the right part of the tree for representing whole hand actions. This resemble the usage of the synergies when representing the motor data. Moreover comparing figure 6.14c and 6.14d we can see how for example the TPS number 3 has more or less the same usage when representing the observed actions or the actions in the motor repertoire, i.e. is used to represent PG action while is not so used to represent WH action in both cases. The usage of TPS number 3 resembles the behaviour of strictly neurons. These neurons in fact spike when the monkey is performing an action or when is looking at an action very similar to the one that elicit a spike when executed. On the other hand the TPS number 4 when coding actions in the motor repertoire is used to represent the PG actions while is not used to represent WH action. When instead used to represent observed actions the synergy number 4 is used for both kinds of action representations.



# Chapter 7

## Conclusions and future work

### 7.1 Contribution of this work

As we described in this thesis, a lot of research has been developed in order to evaluate the plausibility of a synergy representation of action in the brain. In particular different works provided indirect proofs in favor of this hypothesis. These works in fact showed that an efficient action representation could be obtained considering actions as an superposition of synergies. Using a similar paradigm to the ones developed in literature we realized an indirect proof for an action representation in terms of temporal postural synergies hierarchically organized in the brain. In particular we collected a big dataset of hand grasping actions. Following Gallese's studies (Gallese et al., 1996) we constituted a dataset made of 9 kinds of actions of three types: Precision Grip, Finger Prehention and Whole Hand Prehention. We realized, using an algorithm recently developed in the field of dictionary learning, a representation of these actions in terms of temporal postural synergies hierarchically organized. We showed that this kind of encoding actually represent action at multiple levels of detail. We moreover could verify that, when synergies were organized according to particular hierarchical structures, our action representation performed even better of the once proposed in literature, being more accurate and robust to noise. A detailed description of the dataset collection and the test developed on the representation obtained are described in **chapter 4**.

The other contribution of this thesis consisted in realizing the Hierarchical Visuo-Motor(HVM) architecture that, modeling some characteristics of the mirror neurons and more in general of action representation in the brain, try to depict a mechanism through which motor knowledge could be useful in the visual processing of actions. More in particular in this work we realized

the HVM architecture that:

1. uses the hierarchical action encoding previously described;
2. uses the same representation for codifying an action when observed or executed;
3. realizes a hierarchical mapping from visual to motor action representation;
4. identifies the principal difficulties of the visuo-motor mapping;
5. exploits some characteristics of the motor representation in order to improve the visuo-motor mapping.

The developments of points (1), (2) and (3) contributed to realize a model that is descriptively adequate of the biological system. In fact in the HVM architecture we used as motor representation the hierarchical synergy representation of action found in the first part of the work. This, as we stressed, has two main biologically plausible properties. The first is that represents actions in terms of synergies, the second is that represent actions with different degrees of detail. In order to model the behaviour of mirror neurons, that activate both during action execution and observation, in HVM architecture we decide to use the same action representation for observed and executed actions. The system, when observing an action, actually try to codify the action by projecting the visual input into the space of motor representations. Always in order to realize a biologically plausible architecture we build a system where the visuo-motor mapping is hierarchically organized. In **chapter 5** we explain in detail the characteristics of the architecture intended to model the biological findings just described.

Through the realization of points (4) and (5) HVM architecture gives a functional role to the motor representation in the visual processing of action. In particular we first characterized the principal computational difficulties of realizing a visuo-motor mapping, then we equipped our architecture with an its own motor repertoire and hypothesized a very simple and biologically plausible mechanisms to involve motor knowledge in visual elaboration. The description of this process is realized in **chapter 5**. Finally in **chapter 6** we could test that motor involvement actually improve the performance of HVM architecture in realizing the mapping, constituting a clue in favor of the Rizzolatti's direct matching hypothesis. In the same chapter we present a test that show the ability of our architecture to describe the different behaviour of broadly and strictly neurons.

## 7.2 Open questions and future work

Our action representation obtained through the use of temporal postural synergies is grounded on two main assumptions. The first one is that action can be represented as a linear superposition of synergies. The second one: areas in the motor cortex that encode the synergies in which the action is decomposed are all contemporary activated when an action is performed and remain active for the whole duration of the action. These assumptions were found to be plausible for a fast executed actions (Santello et al., 1998; Thakur et al., 2008), but must be even said that in literature there are works where hand actions are represented in terms of non-linear superposition of synergies (Vinjamuri et al., 2010b,a). These works realize representation where the synergies do not last for the whole action duration and can be multiple times recruited during action execution. Hence, an interesting development of this thesis work could be consider a synergy action representation that would be hierarchically organized represented by means of a non-linear superposition of synergies. These could be obtained through the use of already present algorithms in machine learning that enforce structure in the representation and sparsity in the atoms.

The results obtained by the HVM architecture showed that an architecture equipped with a motor repertoire, can actually use this motor knowledge in order to improve the visual processing of actions. Nevertheless we have to say that the improvement we obtained was not so evident as we expected. During our experiments, in order to facilitate the training of the artificial neural networks, we selected just a small part of the whole dataset of action at our disposition. In fact we used just two of the nine types of actions we recored. This was probably a too strong reduction of the dataset, entailing an excessive facilitation of the mapping task for the network and hence diminishing the improvement of the mapping when motor knowledge was used. So an important improvement of this work could results in repeating the tests on the HVM architecture for a bigger dataset. Another interesting development we are considering for our architecture would be extending the dataset including actions observed form different points of view. This could bring interesting results in the direction of modeling the different behaviour shown by mirror neurons according to the different perspective of action observation (Caggiano et al., 2011).



# Appendix A

## Principal Component Analysis

### A.1 Theoretical background

Principal component analysis (PCA) is a quite old algorithm (Hotelling, 1933; Pearson, 1901) used in machine learning to represent/code a bunch of data. This algorithm provides a representation of the data in terms of a linear superposition of orthonormal vectors named *principal components*. So for each data  $\mathbf{x}$ , the algorithm will find a set of vectors  $\mathbf{V}_i$  such that:

$$\mathbf{x} = \sum_i c_i \mathbf{V}_i. \quad (\text{A.1})$$

The sets of coefficient  $c_i$  relative to the data  $\mathbf{x}$ , will constitute the representation of the data in terms of principal components. Assuming a set of data  $D = \{\mathbf{x}_i\}$  the PCA algorithm proceeds in the following way:

- finds a direction of the greatest variance of the data.

$$\mathbf{V}_1 = \operatorname{argmax}_{\|\mathbf{V}\|=1} \sum_i (\mathbf{x}_i^T \mathbf{V})^2;$$

- finds a direction orthogonal to  $\mathbf{V}_1$  with the greatest variance:  $\mathbf{V}_2$ ;
- it repeats the last step, finding the vectors  $\{\mathbf{V}_1, \dots, \mathbf{V}_n\}$  until variance drops below a given threshold.

Here we will show that finding the directions of maximal variance is equivalent to finding the autovectors of the covariance matrix of the data. The covariance matrix  $C$  of our data is defined according to:

$$C = \sum_{i=1}^N \frac{1}{N} \mathbf{x}_i \mathbf{x}_i^T;$$

this is a symmetric positive semi-definite matrix. If the dimension of the data is  $p$ , then the covariance matrix will be a  $p \times p$  matrix. The covariance matrix can be diagonalized, resulting in a set of  $p$  couples of eigenvectors-eigenvalues  $(\mathbf{u}_i, \omega_i)$ . The eigenvectors would be orthogonal forming a basis of the  $p$ -dimensional space. Thus any vector  $\mathbf{x}$  in the dataset can be expressed as an overposition of the eigenvectors of the covariance matrix.

Here we will show that the variance of data in the direction of the eigenvector  $\mathbf{u}_i$  is equal to its eigenvalue  $\omega_i$  and that the direction of greatest variance correspond to the direction of the eigenvector associated to the maximum eigenvalue. The variance in the direction of  $\mathbf{u}_i$  can be calculated according to the following:

$$\langle (\mathbf{x}^T \mathbf{u}_i)^2 \rangle = \langle (\mathbf{u}_i^T \mathbf{x}^T \mathbf{x} \mathbf{u}_i) \rangle = \langle (\mathbf{u}_i^T \mathbf{C} \mathbf{u}_i) \rangle = \omega_i.$$

The angular brackets in the previous equation are intended as the mean over the data in our dataset. Let us evaluate the variance of the data in an arbitrary direction  $\mathbf{V}$ :

$$\langle (\mathbf{x}^T \mathbf{V}) \rangle = \left\langle \left( \mathbf{x}^T \left( \sum_i v_i \mathbf{u}_i \right) \right) \right\rangle = \sum_{ij} v_i \mathbf{u}_i^T \mathbf{C} \mathbf{u}_j v_j = \sum_i v_i^2 \omega_i$$

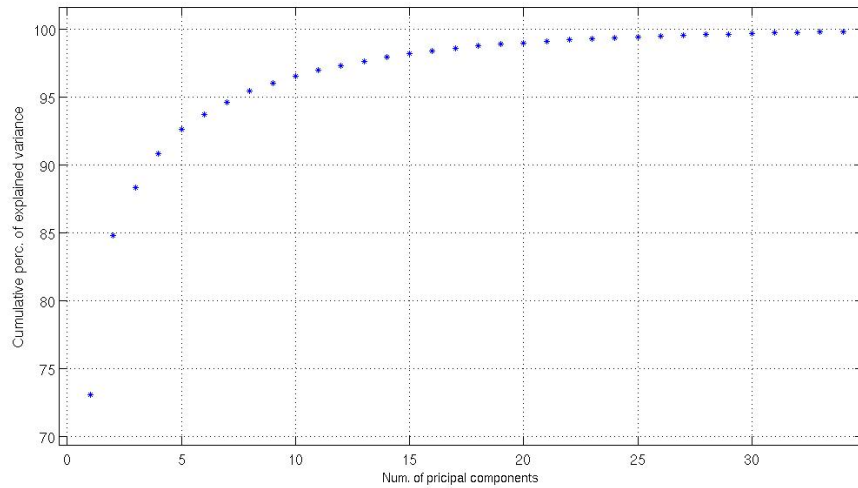
Since we are considering normalized vectors we will have  $\sum_i v_i^2 = 1$ , therefore the maximum values of the summation  $\sum_i v_i^2 \omega_i$  would be when all the  $v_i$  would be equal to zero, except the one relative to the eigenvector  $\mathbf{u}_{max}$  corresponding to the maximum eigenvalue  $\omega_{max}$ . In this way we even showed that the eigenvectors of the covariance matrix correspond to the direction of maximal variance of the dataset.

## A.2 PCA motor representaion

Multiple times in this thesis we referred to the PCA representation of motor data. To obtain this representation we proceeded in the following way. The principal components, according to the details of the algorithm just shown, were calculated organizing the vectors representing actions into a matrix, then calculating the covariance matrix for these vectors and finally diagonalizing this matrix. In the graph in figure A.1 the percentage of explained variance (*ExpVar*) is plotted versus the number of principal components. The percentage of explained variance of the first  $n$  principal components is defined as:

$$ExpVar(n) = \frac{\sum_{i=1}^n \omega_i}{\sum_{i=1}^N \omega_i}; \quad (\text{A.2})$$

where  $\omega_i$  are the eigenvalues of the covariance matrix in increasing order and  $N$  is the total number of eigenvectors. As can be observed in the figure



**Figure A.1:** Motor data PCA representation. The percentage of explained variance versus the number of principal component.

about 8 principal components are necessary to explain more than the 95% of the variance of the data. In this thesis we realized different PCA motor representations using a number of principal components ranging from 5 to 29.





# Appendix B

## Tree-Structured Synergy Method

In chapter 3 we showed how the Tree-Structure Synergy Methods (TSSM) allows for obtaining the hierarchical synergy representation of the motor data. We even mentioned that the algorithm we used in TSSM is due to Jenatton and colleagues (Jenatton et al., 2010). Here we will briefly describe how this algorithm actually works. We will use the following notations. Bold uppercase letters refer to matrices, e.g.,  $\mathbf{X}, \mathbf{V}$ , and bold lowercase letters designate vectors, e.g.,  $\mathbf{x}, \mathbf{v}$ . We denote by  $\mathbf{X}_i$  and  $\mathbf{X}^j$  the  $i$ -th row and the  $j$ -th column of a matrix  $\mathbf{X}$ , respectively. We use the notation  $x_i$  and  $v_{ij}$  to refer to the  $i$ -th element of the vector  $\mathbf{x}$  and the element in the  $i$ -th row and the  $j$ -th column of the matrix  $\mathbf{V}$ , respectively. Given  $\mathbf{x} \in \mathbb{R}^p$  we use the notation  $\|\mathbf{x}\|$  to refer to  $l_\infty$  norm. Given two vectors  $\mathbf{x}$  and  $\mathbf{y}$  in  $\mathbb{R}^p$ , we denote by  $\mathbf{x} \circ \mathbf{y} = (x_1y_1, x_2y_2, \dots, x_py_p) \in \mathbb{R}^p$  the element-wise product of  $\mathbf{x}$  and  $\mathbf{y}$ .

Let us start reconsidering the minimization problem of chapter 3:

$$\min_{\mathbf{U}, \mathbf{V}} \frac{1}{2np} \|\mathbf{X} - \mathbf{UV}^T\|_F^2 + \lambda \sum_{i=1}^n \sum_{j=1}^r \|\mathbf{D}_j \circ \mathbf{U}_i\|_\infty \quad (\text{B.1})$$

As we said this kind of minimization is executed in two stages. In the first stage, we named tree-structured stage, the coefficient  $\mathbf{U}$  of the representation are updated while the dictionary matrix  $\mathbf{V}$  is maintained fixed. In the second stage, we named synergy dictionary stage, the coefficients  $\mathbf{U}$  are maintained fixed while the  $\mathbf{V}$  are updated. The two stages just named are described in the following paragraphs.

## B.1 Tree-Structured Stage

The update of the  $\mathbf{U}$ 's values is performed in this stage, and, more importantly, following the approach suggested by (Jenatton et al., 2010), a tree-structured representation of the rows in  $\mathbf{X}$  is found. The main difficulty is that the optimization of the  $\mathbf{U}_i$ ,  $i \in 1, 2, \dots, n$ , for a fixed  $\mathbf{V}$  involves the nonsmooth regularization term  $\Omega(\mathbf{U}_i) = \sum_{j=1}^r w_j \|\mathbf{D}_j \circ \mathbf{U}_i\|$ , where  $w_j$  are positive weights<sup>1</sup>. In this case the update of the vectors  $\mathbf{U}_i$  can be performed using a *proximal* method. In general proximal approaches are used when one has to minimize a convex nonsmooth objective function which assumes the following general form:

$$f(\mathbf{u}) + \lambda\Omega(\mathbf{u})$$

where  $f(\mathbf{u})$  is the usual data-fitting term  $\frac{1}{2}\|\mathbf{x} - \mathbf{u}\mathbf{V}^T\|_2^2$  and  $\Omega(\mathbf{u})$  is a non-differentiable regularization term. In a nutshell, the proximal approach consists of two consecutive updating steps: first, the vector  $\mathbf{u}$  is updated using the standard gradient update rule w.r.t the first term of the objective function as follows:

$$\bar{\mathbf{u}} \leftarrow \mathbf{u} - \frac{1}{\sigma_{\mathbf{U}}} \nabla f(\mathbf{u}) = \mathbf{u} + \frac{1}{\sigma_{\mathbf{U}}} (\mathbf{x} - \mathbf{u}\mathbf{V}^T)\mathbf{V} \quad (\text{B.2})$$

then, starting from the value  $\bar{\mathbf{u}}$  the new value for  $\mathbf{u}$  is computed by applying a proximal operator  $\Pi_{\mathbf{U}}$  defined by the following minimization problem:

$$\Pi_{\mathbf{U}}(\mathbf{u}) = \underset{\mathbf{v}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{u} - \mathbf{v}\|_2^2 + \lambda\Omega(\mathbf{v}) \quad (\text{B.3})$$

Thus we obtain  $\mathbf{u}^{new} \leftarrow \Pi_{\mathbf{U}}(\bar{\mathbf{u}})$ . For a number of regularization terms the minimization problem expressed in (B.3) can lead to closed-form solutions. For example when  $\Omega(\mathbf{u})$  is the  $\ell_1$  norm of  $\mathbf{u}$  the corresponding proximal operator  $\Pi_{\mathbf{U}}$  is the well-known soft-thresholding operator. In the case of the regularization term used here this minimization problem can be solved by a *primal-dual* approach which enable us to implement the proximal operator defined in (B.3) by the procedure presented in Algorithm (1).

---

<sup>1</sup>Note that all  $w_j$  are fixed to 1 in the experiments

---

**Algorithm 1** Proximal operator.  $\Pi_{\lambda w_j}^*$  is the orthogonal projection on the ball of radius  $\lambda w_j$  of the dual norm  $\|\cdot\|_*$ .

---

**Input:**  $\mathbf{u} \in \mathbb{R}^r$  and  $\mathbf{D} \in \mathbb{R}^{r \times p}$

**Output:**  $\mathbf{v} \in \mathbb{R}^r$

**for**  $i \leftarrow 0$  **to** *MaxNumberOfIteration*

**for**  $j \leftarrow 1$  **to**  $r$

$$\mathbf{P}_j \leftarrow \mathbf{u} - \sum_{h \neq j} \mathbf{P}_h$$

$$\mathbf{P}_j \leftarrow \Pi_{\lambda w_j}^*(\mathbf{P}_j \circ \mathbf{D}_j)$$

**end for**

**end for**

$$\mathbf{v} \leftarrow \mathbf{u} - \sum_{j=1}^r \mathbf{P}_j$$


---

Summarizing, the optimization of the  $\mathbf{U}_i$  values is performed using the gradient descent rule expressed in (B.2) and, then, applying the proximal operator as defined previously.

## B.2 Synergy Dictionary Stage

This stage consists in updating the  $\mathbf{V}$ 's values while keeping fixed the values of  $\mathbf{U}$ . Note that the objective function in (B.1) is composed of two terms to be minimized, and the second term does not depend on  $\mathbf{V}$ . Therefore, the optimization problem posed in (B.1) can be, in this stage, reformulated as follows:

$$\min_{\mathbf{V}} \frac{1}{2np} \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2 \text{ s.t. } \forall i \|\mathbf{V}^i\|_2 \leq 1 \quad (\text{B.4})$$

Due the fact that the columns of  $\mathbf{V}$  are constrained to lie inside the unit ball, the update of  $\mathbf{V}$  is performed in two consecutive steps. First, we apply a standard gradient updating rule as follows

$$\bar{\mathbf{V}} \leftarrow \mathbf{V} + \frac{1}{\sigma_{\mathbf{V}} np} (\mathbf{X} - \mathbf{U}\mathbf{V}^T) \mathbf{U}^T \quad (\text{B.5})$$

where  $\eta$  is a parameter. Then, we use the projection operator  $\Pi(\mathbf{v}) = \frac{\mathbf{v}}{\max\{1, \|\mathbf{v}\|_2\}}$  in order to project the columns of  $\bar{\mathbf{V}}$  on the unit ball in  $\mathbb{R}^p$ . Consequently the update of  $\mathbf{V}$  is computed as follows:

$$\mathbf{V} \leftarrow \Pi(\mathbf{V} + \frac{1}{\sigma_{\mathbf{V}} np} (\mathbf{X} - \mathbf{U}\mathbf{V}^T) \mathbf{U}^T) \quad (\text{B.6})$$

The overall algorithm of TSSM is reported in algorithm 2. Note that a fixed step gradient descent procedure was adopted with the two learning rate  $\sigma_{\mathbf{U}}$  and  $\sigma_{\mathbf{V}}$  chosen equal to the Lipschitz constant of  $\nabla f(\mathbf{u})$  and  $\nabla f(\mathbf{v})$  respectively.

---

**Algorithm 2** Tree-structured synergies algorithm

---

**Input:**  $\mathbf{X} \in \mathbb{R}^{n \times p}$ ,  $\mathbf{U}^0 \in \mathbb{R}^{n \times r}$  and  $\mathbf{V} \in \mathbb{R}^{p \times r}$

$T_{max} \in \mathbb{Z}^+$ ,  $\lambda \geq 0$

**Output:**  $\mathbf{U} \in \mathbb{R}^{n \times r}$  and  $\mathbf{V} \in \mathbb{R}^{p \times r}$

**for**  $t \leftarrow 1$  **to**  $T_{max}$

**repeat until convergence**

$\mathbf{U}^t \leftarrow \mathbf{U}^{t-1} + \frac{1}{\sigma_{\mathbf{U}}}(\mathbf{X} - \mathbf{U}^{t-1}\mathbf{V}^T)\mathbf{V}$  gradient descent step

$\mathbf{U}^t \leftarrow \Pi_{\mathbf{U}}(\mathbf{U}^t, \lambda)$  proximal operator step

**end**

    possibly replace under-used atoms

**repeat until convergence**

$\mathbf{V}^t \leftarrow \mathbf{V}^{t-1} + \frac{1}{\sigma_{\mathbf{V}np}}(\mathbf{X} - \mathbf{U}\mathbf{V}^{(t-1)T})\mathbf{U}^T$  gradient descent step

$\mathbf{V}^t \leftarrow \Pi_{\mathbf{V}}(\mathbf{V}^t)$  proximal operator step

**end**

**end for**

**return**  $\mathbf{U}^t, \mathbf{V}^t$

---

# Appendix C

## Mixture Density Network

The Mixture Density Network (MDN) is an architecture intended to model non-functional relations among input and target. This architecture associates in fact to an input vector  $\mathbf{x}$  the parameters of a probability distribution over the targets set. Considering in fact a set of labelled data  $T = \{\mathbf{x}^n, \mathbf{t}^n\}_{n=1}^N$ , the MDN can be used to approximate the conditional distribution  $p(\mathbf{t}|\mathbf{x})$ . The MDN architecture will use to model the previous probability distribution an overposition of gaussians:

$$p(\mathbf{t}|\mathbf{x}) = \sum_{i=1}^M \alpha_i(\mathbf{x}) \phi_i(\mathbf{t}|\mathbf{x}); \quad (\text{C.1})$$

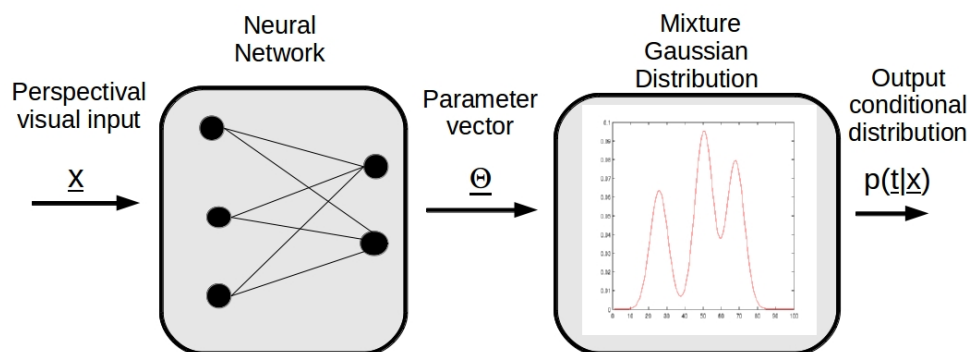
where the kernels  $\phi_i(\mathbf{t}|\mathbf{x})$  are given by:

$$\phi_i(\mathbf{t}|\mathbf{x}) = \frac{1}{(2\pi)^{c/2} \sigma_i^c(\mathbf{x})} \exp\left(-\frac{\|\mathbf{t} - \mu_i(\mathbf{x})\|^2}{2\sigma_i(\mathbf{x})}\right). \quad (\text{C.2})$$

As can be seen from the previous formula all the parameters of the mixture,  $\sigma_i$ ,  $\alpha_i$  and  $\mu_i$  are function of the input  $\mathbf{x}$ . The MDN architecture consists of a Feed Forward Neural Network (FFNN) that associates to the input values the parameters of the mixture of gaussians(see figure: C.1). The relation between  $\mathbf{x}$  and the mixture parameters can be learned in a supervised fashion using universal approximator as, for example, a two layer FFNN with non linear hidden units. The FFNN will be trained in order to minimize the negative log-likelihood of the data:

$$E = - \sum_{n=1}^N \ln \left\{ \sum_{i=1}^M \alpha_i(\mathbf{x}^n) \phi_i(\mathbf{t}^n|\mathbf{x}^n) \right\} \quad (\text{C.3})$$

By choosing a mixture model with a sufficient number of gaussian kernels and a FFNN with a sufficient number of hidden units, the MDN should be



**Figure C.1:** Mixture density network.

able to approximate with any accuracy any conditional distribution  $p(\mathbf{t}|\mathbf{x})$ . In the next paragraph we will give some insights on the way in which the FFNN associate to the input the mixture parameters, then we will show how the network can be trained using a backpropagation algorithm.

## C.1 Mixture Density Network

Let us consider a MDN with  $M$  kernels, i.e.  $M$  gaussians. The FFNN that would associate to the input the parameters of the mixture will have:

- $M$  outputs units, denoted  $z_j^\alpha$ , for the mixing coefficients  $\alpha_j(\mathbf{x})$ ;
- $M$  outputs units, denoted  $z_j^\sigma$ , for the standard deviation of the gaussians;
- $M \times c$  output units, denoted  $z_{jk}^\mu$ , for the gaussian means  $\mu_j$  with components  $\mu_{jk}$ .

In this way the network will have  $(c + 2) \times M$  outputs.

The mixing coefficients  $\alpha_i(\mathbf{x})$  of the distribution will be forced to sum to 1:

$$\sum_{i=1}^M \alpha_i(\mathbf{x}) = 1. \quad (\text{C.4})$$

This is imposed by obtaining the parameters  $\alpha_i(\mathbf{x})$  as the value of a "softmax" function on the network output:

$$\alpha_i = \frac{\exp(z_i^\alpha)}{\sum_{j=1}^M \exp(z_j^\alpha)}. \quad (\text{C.5})$$

The standard deviations of the gaussians are returned as the exponential of the output  $z_i^\sigma$ :

$$\sigma_i = \exp(z_i^\sigma). \quad (\text{C.6})$$

This is in order to avoid pathological configurations in which one or more of the standar deviations goes to zero. Finally the gaussian centers  $\mu_j$  are simple equal to the corresponding network outputs:

$$\mu_{jk} = z_{jk}^\mu. \quad (\text{C.7})$$

As we said previously a two layers FFNN can be trained to associate to the inputs the mixture parameters. The network is trained in order to reduce the error in equation C.3. The algorithm used in order to train the network is a gradient descent algorithm named Backpropagation (Rumelhart et al., 1985). This evaluates the gradient of the error as a function of the network parameters, the weights of the networks. Then it recudes the error changing the weights in the direction of the gradient of the function. The backpropagation algorithm can be applied any time the error function is a differentiable one respect to the weights of the network. It just needs as input the derivates of the error function respect to the output of the network. In this paragraph we will not give any detail on the backpropagation, we will just evaluate the derivates it needs as input.

Before starting our computation will be useful to define the following variables  $\pi_j(\mathbf{x}, \mathbf{t})$ :

$$\pi_j(\mathbf{x}, \mathbf{t}) = \frac{\alpha_j \phi_j}{\sum_l \alpha_l \phi_l}; \quad (\text{C.8})$$

this quantity can be viewed as the *responsability* that components  $j$  takes for explaining the observation  $\mathbf{t}$  as associated to the input  $\mathbf{x}$ . Let us start now evaluating the derivate of the error function respect to the  $z_j^\alpha$ . The error fuction depends on these variables through the equation C.5, moreover since

our error function is a summation of  $N$  terms  $E = \sum_n E_n$ , see equation C.3, we will thus evaluate the derivate of just one term of the summation.

Using the chain rule we can write:

$$\frac{\partial E^n}{\partial z_j^\alpha} = \sum_k \frac{\partial E^n}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial z_j^\alpha}. \quad (\text{C.9})$$

The first term of the sum in the right hand side of the previous equation is:

$$\frac{\partial E^n}{\partial \alpha_k} = -\frac{\phi_k}{\sum_{j=1}^M \alpha_j \phi_j} = -\frac{\pi_k}{\alpha_k}. \quad (\text{C.10})$$

The second term of the sum is:

$$\frac{\partial \alpha_k}{\partial z_j^\alpha} = \delta_{jk} \alpha_k - \alpha_j \alpha_k. \quad (\text{C.11})$$

Substituting the equations C.10 and C.11 into the equation C.9 we obtain:

$$\frac{\partial E^n}{\partial z_j^\alpha} = \sum_k -\frac{\pi_k}{\alpha_k} (\delta_{jk} \alpha_k - \alpha_j \alpha_k) = \alpha_j - \pi_j. \quad (\text{C.12})$$

For the derivatives corresponding to the  $\sigma_j$  parameters again remember that  $E^n$  depends on  $z_j$  only through the relation C.6.

$$\frac{\partial E^n}{\partial z_j^\sigma} = \frac{\partial E^n}{\partial \sigma_j} \frac{\partial \sigma_j}{\partial z_j^\sigma}, \quad (\text{C.13})$$

the first term on the right side of the equation can be calculated as follows:

$$\begin{aligned} \frac{\partial E^n}{\partial \sigma_j} &= -\frac{\alpha_j}{\sum_{j=1}^M \alpha_j \phi_j} \frac{1}{(2\pi)^{c/2}} \left[ -\frac{c}{\sigma^{c+1}} \exp\left\{ \frac{\|\mathbf{t} - \mu_j\|^2}{2\sigma_j^2} \right\} + \right. \\ &\quad \left. \frac{1}{\sigma_j^c} \exp\left\{ \frac{\|\mathbf{t} - \mu_j\|^2}{2\sigma_j^2} \right\} \frac{\|\mathbf{t} - \mu_j\|^2}{\sigma_j^3} \right] = \\ &= -\frac{\alpha_j \phi_j}{\sum_{j=1}^M \alpha_j \phi_j} \left[ -\frac{c}{\sigma_j} + \frac{\|\mathbf{t} - \mu_j\|^2}{\sigma_j^3} \right] = \\ &= \pi_j \left[ -\frac{c}{\sigma_j} + \frac{\|\mathbf{t} - \mu_j\|^2}{\sigma_j^3} \right] \end{aligned} \quad (\text{C.14})$$

Substituting in equation C.13 and considering that  $\frac{\partial \sigma_j}{\partial z_j^\sigma} = \sigma_j$  we obtain:

$$\frac{\partial E^n}{\partial z_j^\sigma} = \pi_j \left[ -c + \frac{\|\mathbf{t} - \mu_j\|^2}{\sigma_j^2} \right] \quad (\text{C.15})$$



Finally the derivate of the error function respect to the  $z_{jk}^\mu$  is:

$$\begin{aligned} \frac{\partial E^n}{\partial z_{jk}^\mu} &= - \frac{\alpha_j}{\sum_{j=1}^M \alpha_j \phi_j} \frac{\exp\left\{-\frac{\|t-\mu_j\|^2}{2\sigma_j^2}\right\}}{(2\pi)^{c/2} \sigma_j^c} \frac{(\mu_{jk} - t_k)}{\sigma_j^2} \\ &= \pi_j \left\{ \frac{(\mu_{jk} - t_k)}{\sigma_j^2} \right\} \end{aligned} \quad (\text{C.16})$$

$$\frac{\partial E^n}{\partial z_{jk}^\mu} = \pi_j \left\{ \frac{(\mu_{jk} - t_k)}{\sigma_j^2} \right\} \quad (\text{C.17})$$

The equations C.12, C.15 and C.17 will be used by the backpropagation algorithm in order to evaluate the derivates of the error function respect to the weights of the FFNN.



# Bibliography

- M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.
- E. Amico. *Rappresentazione multipla e gerarchica delle azioni di presa per lo studio dei neuroni specchio*. Università degli Studi di Napoli, "Federico II", 2011.
- A. Avenanti, D. Buetti, G. Galati, and S. M Aglioti. Transcranial magnetic stimulation highlights the sensorimotor side of empathy for pain. *Nature neuroscience*, 8(7):955–960, 2005.
- C. Basso, M. Santoro, A. Verri, and S. Villa. Paddle: proximal algorithm for dual dictionaries learning. In *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 379–386. Springer, 2011.
- C. M. Bishop. Mixture density networks. 1994.
- C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.
- C.M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
- M. Brass, H. Bekkering, and W. Prinz. Movement observation affects movement execution in a simple response task. *Acta Psychologica*, 106(1-2):3–22, 2001.
- V. Caggiano, L. Fogassi, G. Rizzolatti, P. Thier, and A. Casile. Mirror neurons differentially encode the peripersonal and extrapersonal space of monkeys. *Science*, 324(4):403–406, 2009.
- V. Caggiano, L. Fogassi, G. Rizzolatti, J. K. Pomper, P. Thier, M. A. Giese, and A. Casile. View-based encoding of actions in the mirror neurons of area f5 in the macaque premotor cortex. *Current Biology*, 21(1):144–148, 2011.

- C. Catmur, R. B. Mars, M. F. Rushworth, and C. Heyes. Making mirrors: Premotor cortex stimulation enhances mirror and counter-mirror motor facilitation. *Journal of Cognitive Neuroscience*, (23):2352–2362, 2011.
- F. Chersi, P. F. Ferrari, and L. Fogassi. Neuronal chains for actions in the parietal lobe: a computational model. *PloS one*, 6(11):e27652, 2011.
- T. T. J. Chong, R. Cunnington, M. A. Williams, N. Kanwisher, and J. B. Mattingley. fmri adaptation reveals mirror neurons in human inferior parietal cortex. *Current Biology*, 18(20):1576–1580, 2008.
- M. T. Ciocarlie and P. K. Allen. Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research*, 28(7):851–867, 2009.
- R. Cook, G. Brid, C. Catmur, C. Press, and C. Heyes. Mirror neurons: From origin to function. *Behavioral and Brain Sciences*, 2013.
- A. d’Avella, A. Portone, L. Fernandez, and F. Lacquaniti. Control of fast-reaching movements by muscle synergy combinations. *The Journal of neuroscience*, 26(30):7791–7810, 2006.
- A. P. Dempster, N. M. Laird, D. B. Rubin, et al. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977.
- G. Di Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti. Understanding motor events: a neurophysiological study. *Experimental brain research*, 91(1):176–180, 1992.
- J. Dushanova and J. Donoghue. Neurons in primary motor cortex engaged during action observation. *European Journal of Neuroscience*, 31(2):386–398, 1996.
- K. Engan, S. O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 5, pages 2443–2446. IEEE, 1999.
- L. Fadigà, L. Fogassi, G. Pavese, and G. Rizzolatti. Motor facilitation during action observation: a magnetic simulation study. *Journal of Neurophysiology*, 73:2608–2611, 1995.

- L. Fogassi, P.F. Ferrari, B. Gesierich, S. Rozzi, F. Chersi, and G. Rizzolatti. Parietal lobe: from action organization to intention understanding. *Science*, 308(5722):662–667, 2005.
- K. Friston. A theory of cortical responses. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1456):815–836, 2005.
- V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti. Action recognition in the premotor cortex. *Brain*, (119):593–609, 1996.
- S. T. Grafton and A. F. Hamilton. Evidence for a distributed hierarchy of action representation in the brain. *Human movement science*, 26(4):590–616, 2007.
- S. T. Grafton, M. A. Arbib, L. Fadiga, and G. Rizzolatti. Localization of grasp representations in humans by pet:2. observation compared with imagination. *Experimental Brain Research*, (112):103–111, 1996.
- A. F. de C. Hamilton and S. T. Grafton. Action outcomes are represented in human inferior frontoparietal cortex. *Cerebral Cortex*, 18(5):1160–1168, May 2008. doi: 10.1093/cercor/bhm150.
- M. Haruno, D. M. Wolpert, and M. Kawato. Mosaic model for sensorimotor learning and control. *Neural computation*, 13(10):2201–2220, 2001.
- T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*, volume 2. Springer, 2009.
- H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- M. Iacoboni, R.P. Woods, M. Brass, H. Bekkering, J.C. Mazziotta, and G. Rizzolatti. Cortical mechanisms of human imitation. *Science*, 286(5449):2526–2528, 1999.
- M. Iacoboni, I. Molnar-Szakacs, V. Gallese, G. Buccino, J. C. Mazziotta, and G. Rizzolatti. Grasping the intentions of others with one’s own mirror neuron system. *PLoS Biol*, 3(3):e79, 02 2005. doi: 10.1371/journal.pbio.0030079. URL <http://dx.doi.org/10.1371%2Fjournal.pbio.0030079>.
- T. Iberall, G. Bingham, and M.A. Arbib. Opposition space as a structuring concept for the analysis of skilled hand movements. *Experimental brain research series*, 15:158–173, 1986.

- M. Ito and J. Tani. Generalization in learning multiple temporal patterns using rnnpb. In *Neural Information Processing*, pages 592–598. Springer, 2004.
- R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.
- T. Jellema, C. I. Baker, B. Wicker, and D. I. Perrett. Neural representation for the perception of the intentionality of actions. *Brain Cognition*, 44: 280–302, 2000.
- R. Jenatton, J. Mairal, F. R. Bach, and G. R. Obozinski. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 487–494, 2010.
- C. Keysers and D. I. Perrett. Demystifying social cognition: a hebbian perspective. *Trends in cognitive sciences*, 8(11):501–507, 2004.
- J. M. Kilner, K. J. Friston, and C. D. Frith. Predictive coding: an account of the mirror neuron system. *Cognitive processing*, 8(3):159–166, 2007.
- J.M. Kilner, A. Neal, N. Weiskopf, K.J. Friston, and C.D. Frith. Evidence of mirror neurons in human inferior frontal gyrus. *Journal of Neuroscience*, 29(32):10153–10159, 2009.
- A. Kraskov, N. Dancause, M. M. Quallo, S. Shepherd, and R. N. Lemon. Corticospinal neurons in macaque ventral premotor cortex with mirror properties: a potential mechanism for action suppression? *Neuron*, 64(6): 922–930, 2009.
- K. R. Leslie, S. H. Johnson-Frey, and S. T. Grafton. Functional imaging of face and hand imitation: towards a motor theory of empathy. *Neuroimage*, 21(2):601–607, 2004.
- S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2):129–137, 1982.
- C.R. Mason, J.E. Gomez, and T.J. Ebner. Hand synergies during reach-to-grasp. *Journal of Neurophysiology*, 86(6):2896–2910, 2001.
- M. Matelli and G. Luppino. Parietofrontal circuits for action and space perception in the macaque monkey. *NeuroImage*, 14:27–32, 2001.

- G. McLachlan and T. Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.
- P. Molenberghs, R. Cunnington, and J. B. Mattingley. Brain regions with mirror properties: a meta-analysis of 125 human fmri studies. *Neuroscience & Biobehavioral Reviews*, 36(1):341–349, 2012.
- R. Mukamel, A. D. Ekstrom, J. Kaplan, M. Iacoboni, and I. Fried. Single-neuron responses in humans during execution and observation of actions. *Current Biology*, 20(8):750–756, 2010.
- A. Murata, V. Gallese, G. Luppino, M. Kaseda, and H. Sakata. Selectivity for the shape, size, and orientation of objects for grasping in neurons of monkey parietal area aip. *Journal of Neurophysiology*, 83(5):2580–2601, 2000.
- S. A. Overduin, A. d’Avella, J. M. Carmena, and E. Bizzi. Microstimulation activates a handful of muscle synergies. *Neuron*, 76(6):1071–1077, 2012.
- E. Oztop and M. A. Arbib. Schema design and implementation of the grasp-related mirror neuron system. *Biological cybernetics*, 87(2):116–140, 2002.
- E. Oztop, N. S. Bradley, and M. A. Arbib. Infant grasp learning: a computational model. *Experimental Brain Research*, 158(4):480–503, 2004.
- Erhan Oztop, Mitsuo Kawato, and Michael Arbib. Mirror neurons and imitation: A computationally guided review. *Neural Networks*, 19(3):254–271, 2006.
- K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.
- D. I. Perrett, A. J. Mistlin, M. H. Harries, and A. J. Chitty. Understanding visual appearance and consequences of actions. In *Vision and Action: The Control of Grasping*, pages 163–342. Goodale, 1990.
- M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Neural Networks, 1993., IEEE International Conference on*, pages 586–591. IEEE, 1993.
- G. Rizzolatti and M. A. Arbib. Language within our grasp. *Trends in neurosciences*, 21(5):188–194, 1998.

- G. Rizzolatti and M. Gentilucci. Motor and visual-motor functions of the premotor cortex. *Neurobiology of Neocortex*, pages 269–284, 1988.
- G. Rizzolatti and C. Sinigaglia. The functional role of the parieto-frontal mirror circuit: interpretations and misinterpretations. *Nat Rev Neurosci*, 11(4):264–74, 2010.
- G. Rizzolatti, R. Camarda, L. Fogassi, M. Gentilucci, G. Luppino, and M. Matelli. Functional organization of inferior area 6 in the macaque monkey. area f5 and the control of distal movements. *Experimental Brain Research*, 71(3):491–507, 1988.
- G. Rizzolatti, L. Fadiga, and V. Gallese. Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3(2):131–141, 1996.
- G. Rizzolatti, G. Luppino, and M. Matelli. The organization of the cortical motor system: new concepts. *Electroencephalography and Clinical Neurophysiology*, 106:283–296, 1998.
- G. Rizzolatti, L. Fogassi, and V. Gallese. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nat Rev Neurosci*, 2(9):661–670, 2001.
- G. Rizzolatti, Ferrari P.F., Rozzi S., and L. Fogassi. The inferior parietal lobule: where action becomes perception. In *Novartis Found Symposium*. US National Library of Medicine, 2006.
- D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.
- M. Santello and J. F. Soechting. Force synergies for multifingered grasping. *Experimental Brain Research*, 133(4):457–467, 2000.
- M. Santello, M. Flanders, and J. F. Soechting. Postural hand synergies for tool use. *The Journal of Neuroscience*, 18(23):10105–10115, 1998.
- M. Santello, M. Flanders, and J. F. Soechting. Patterns of hand motion during grasping and the influence of sensory guidance. *The Journal of Neuroscience*, 22(4):1426–1435, 2002.
- M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 111–147, 1974.



- G. Tessitore, C. Sinigaglia, and R. Prevede. Hierarchical and multiple hand action representation using temporal postural synergies. *Experimental brain research*, 225(1):11–36, 2013.
- P. H. Thakur, A. J. Bastian, and S. S. Hsiao. Multidigit movement synergies of the human hand in an unconstrained haptic exploration task. *The Journal of neuroscience*, 28(6):1271–1281, 2008.
- D. Tkach, J. Reimer, and N.G. Hatsopoulos. Congruent activity during action and action observation in motor cortex. *Journal of Neuroscience*, 27(48):13241–13250, 2007.
- E. Todorov and Z. Ghahramani. Analysis of the synergies underlying complex hand manipulation. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, volume 2, pages 4637–4640. IEEE, 2004.
- M.A. Umiltà, E. Kohler, V. Gallese, L. Fogassi, L. Fadigà, C. Keysers, and G. Rizzolatti. I know what you are doing: a neurophysiological study. *Neuron*, (32):91–101, 2001.
- R. Vinjamuri, M. Sun, C. Chang, H. Lee, J. Scabassi, Robert, and Z. Mao. Temporal postural synergies of the hand in rapid grasping tasks. *Information Technology in Biomedicine, IEEE Transactions on*, 14(4):986–994, 2010a.
- R. Vinjamuri, M. Sun, C. Chang, H. Lee, J. Scabassi, Robert, and Z. Mao. Dimensionality reduction in control and coordination of the human hand. *Biomedical Engineering, IEEE Transactions on*, 57(2):284–295, 2010b.
- S. Vogt, G. Buccino, A. M. Wohlschlagel, N. Canessa, N. J. Shah, K. Zilles, S.B. Eickhoff, H.J. Freund, G. Rizzolatti, and G.R. Fink. Prefrontal involvement in imitation learning of hand actions: effects of practice and expertise. *Neuroimage*, 37(4):1371–1383, 2007.
- G. Wahba and S. Wold. A completely automatic french curve: Fitting spline functions by cross validation. *Communications in Statistics-Theory and Methods*, 4(1):1–17, 1975.