



DOTTORATO DI RICERCA IN
SCIENZE COMPUTAZIONALI E INFORMATICHE
CICLO XXII

Consorzio tra Università di Catania, Università di Napoli Federico II,
Seconda Università di Napoli, Università di Palermo, Università di Salerno

SEDE AMMINISTRATIVA:
UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

GIOVANNI TESSITORE

FROM MOTOR TO SENSORY PROCESSING IN
MIRROR NEURON MODELS:
A NOVEL COMPUTATIONAL APPROACH

TESI DI DOTTORATO DI RICERCA

IL COORDINATORE
Prof. Luigi M. Ricciardi

Contents

1	Introduction	4
1.1	Background and motivations	4
1.2	The proposed approach	5
1.3	Overview	7
2	Mirror neurons	9
2.1	Biological Background	9
2.1.1	The VIP-F4 circuit	12
2.1.2	The AIP-F5ab circuit	13
2.1.3	The PF-F5c circuit	16
2.1.4	Relevant functional interpretation	19
3	Critical analysis of current computational models of mirror neurons	21
3.1	Same-activity hypothesis	22
3.1.1	The gap between same-activity hypothesis and experimental data	25
3.2	Same-input hypothesis	27
3.3	The confluence of same-activity and same-input hypotheses	32
3.4	A case study: the computation of grip-size	33
3.4.1	Model description	35
3.4.2	Testing the model	38
3.5	Same-input hypothesis and model of view-independent grip-size computation	49

4	A new computational approach for mirror neurons	50
4.1	Using motor information for internal input computation . . .	51
4.2	Mechanisms for computing internal inputs using motor in- formation	53
5	Model specification and modelling problems	58
5.1	Mapping visual input to hand configuration: an ill-posed problem	58
5.1.1	Inverse kinematics of a robot arm	59
5.2	A probabilistic framework for hand configuration estimation	63
5.2.1	Different sets of eigenpostures for different modality of object-directed actions	68
5.3	Eigen-postures selection mechanisms: the concept of affor- dances	69
5.3.1	Grasping Affordance (GA) model	69
5.3.1.1	Affordances for Grasping	70
5.3.1.2	General GA Model Description	71
5.3.1.3	GA Model specification and implementation	73
5.3.2	How GA model can be used to select the initial sets of eigenpostures	74
5.4	Generation of expected hand-configurations coefficients on the basis of selected sets of eigen-postures	75
6	Experiments and results	76
6.1	Different eigenposture sets for different classes of grasp ac- tions	76
6.1.1	Experimental set-up	78
6.1.2	Results	78
6.2	Testing the GA model for affordance extraction	84
6.2.1	Experimental set-up	84
6.2.2	Results	84
6.3	Soundness of the probabilistic approach for hand configu- ration estimation	86
6.3.1	Experimental set-up	87
6.3.2	Results	88
6.4	How motor information improves hand-configuration esti- mation	95
6.4.1	Experimental set-up	95
6.4.2	Results	96
6.4.3	Experimental set-up	100
6.4.4	Results	100

6.5	A test of the whole system	103
6.5.1	Experimental set-up	103
6.5.2	Results	103
7	Conclusion and future work	106
7.1	Contribution of this work	106
7.2	Open questions and future work	108
A	NeGOI model specification and implementation	109
A.1	View-based module	110
A.2	Prototypical view-invariant module	112
A.3	Grip-aperture module	114
A.3.1	Experimental setting	114
B	Mixture Density Networks	115
B.1	Feed-forward neural networks	117
B.2	Learning algorithm	119
B.3	Back-propagation in feed-forward networks	120
B.3.1	Back-propagation for sum-of-squares error function and sigmoid activation function	122
B.4	Mixture Density Networks	123
B.4.1	Back-propagation in Mixture Density Network	124
B.5	Implementation and Test	128
B.5.1	test 1: a simple uni-dimensional example	128
B.5.2	test 2: a two-dimensional example	135
C	Similarity measure between principal subspaces	141
C.1	Principal Component Analysis	141
C.2	Principal Subspace Comparison	144
C.3	A simple example	145
	Bibliography	148

1.1 Background and motivations

Mirror neurons exhibit the intriguing behavioural property of becoming active during both execution and observation of object-directed actions. The expression “object-directed action” is used to denote actions directed toward an object such as grasping, holding and tearing actions. Identified in the macaque’s F5 cortical motor area, these neurons were first described in seminal work by Giacomo Rizzolatti and co-workers (Rizzolatti et al., 1996; Gallese et al., 1996). According to a prominent interpretation, mirror neurons are involved in a circuit of cortical areas – usually referred to as mirror system (Cattaneo and Rizzolatti, 2009; Rizzolatti and Craighero, 2004) – subserving the control of one’s own actions and the recognition of observed actions. This interpretation posits significant functional commonalities between action control and action recognition processes, which take their origin in a set of shared neurobiological mechanisms.

Additional functional roles have been hypothesised for mirror systems in the framework of theories of language evolution (Arbib, 2005), mind-reading (Gallese and Goldman, 1998), and learning by imitation (Carr et al., 2003; Miall, 2003).

Various computational models have been advanced to account for the behaviour of mirror neurons in the broader context of mirror system functionalities (Haruno et al., 2001; Keysers and Perrett, 2004; Ito and Tani, 2004; Oztop et al., 2005; Oztop and Arbib, 2002).

In this PhD Thesis we point out that mirror neurons are usually modelled in accordance with the following hypotheses:

- ▷ **Same activity:** Let \mathcal{A} be an object-directed action. A mirror neuron exhibits the same activity irrespective of whether \mathcal{A} is carried out or

observed.

- ▷ **Same input:** Let \mathcal{A} be an object-directed action and let $m_{\mathcal{A}}$ be any mirror neuron which becomes active whenever \mathcal{A} is carried out or observed. Then, the same input signals are received, in both execution and observation conditions, by $m_{\mathcal{A}}$ and any other F5 neuron which directly affects $m_{\mathcal{A}}$'s behaviour. These input signals are the outcome of computational processes which do not involve the motor system.

The same-activity hypothesis turns out to be an idealization in the light of known experimental data about mirror neuron activation behaviours. The same-input hypothesis implies the existence of a complicated perceptual processing which is needed to give the same input to mirror neurons, irrespectively of whether one is in action observation or in action execution conditions. The main upshot of the analysis of computational models endorsing both same-activity and same-input hypotheses is that these models are descriptively inadequate and functionally uninformative: descriptively inadequate, insofar as these models fail to account for a wide variety of mirror neuron behavioural data; and functionally uninformative since mirror mechanisms do not play significant functional roles especially insofar as sensory processing is concerned.

The critical analysis of extant computational models endorsing both same-activity and same-input hypotheses prepares the ground for introducing a novel approach to the computational modelling of mirror neurons. In this new model, same-activity and same-input hypotheses are dispensed with; in particular the functional interaction between sensory input and mirror activation mechanisms is significantly modified according to a different interpretation of the direct-matching hypothesis.

1.2 The proposed approach

The direct matching hypothesis (Rizzolatti et al., 2001) states that the motor system plays a central role in action recognition. In this view, sensory inputs concerning an observed object-directed action \mathcal{A} are mapped onto motor representations of \mathcal{A} , which is recognized when its observation brings the observer's neural motor system to "resonate", that is, when the neural motor representation of \mathcal{A} becomes active in the observer's brain.

How is the resonating effect achieved? Procedurally, one can envisage different sorts of involvement for motor representations and processing

1.2. THE PROPOSED APPROACH

in action recognition. According to one view, pursued in most computational models of mirror neurons, sensory inputs are turned into motor information by means of a computational transformation unidirectionally flowing from sensory input to *direct internal input* (that is the brain signals that mirror neurons receive from directly afferent brain areas), and from the latter on to motor coding. According to this conception, mirror activity is a straightforward consequence of the view-independent character of mirror neurons inputs, thereby leading to impoverished functional roles for mirror mechanisms in action recognition processes. According to an alternative view, action recognition processes receive information from the brain motor system at earlier processing stages, and this information is deployed for the purpose of interpreting sensory inputs.

Major computational problems arising in connection with this approach concern the identification of available motor information, and its specific uses for the purpose of analyzing and interpreting sensory input during action observation.

In order to tackle these problems as first step we have attempted to isolate the motor information which is useful to both action control and visual input interpretation.

Santello and co-workers showed that hand configurations can be described by a restricted number of parameters. More precisely, a hand configuration, during grasping actions, can be described as a linear combination of a few number of vectors (called eigenpostures) in the space of hand joints.

In the context of grasping control, the opportunity of using simpler hand description models is argued for in (Iberall and Fagg, 1996; Iberall et al., 1986) and explored there by means of the notion of virtual finger, which enables one to reduce the degrees of freedom and thereby the complexity of the hand control problem.

In this work it is assumed that the motor information coded by the mirror system to improve visual input processing is related to eigenpostures.

In particular we have verified that some classes of grasping actions (such as precision grip and whole hand grasping actions) can be described by the coefficients of a linear combination of eigenpostures. And each class is associated to a different set of eigenpostures which is composed of few elements with respect to the total number of hand joints.

Note that the estimation of an actual hand configuration from its visual appearance only, is an inverse ill-posed problem. This is mainly due to hand self-occlusions which turns the mapping from hand visual description to hand configuration into a multi-valued mapping.

In this work, a probabilistic approach is proposed to model such map-

ping. In particular Mixture Density Networks have been used to model the distribution of hand configurations conditional on current visual input. However, the estimation of such distribution is a complicated task due to the large number of variables needed to describe hand configurations.

Hence, the benefit flowing from the use of eigenpostures is in the reduction of the parameters needed to describe hand configurations, and thus in the simplification of the problem of distribution estimation, because we now estimate the distribution of the coefficients of the linear combination of eigenpostures conditional on current visual input.

1.3 Overview

In *Chapter 2* we will review significant neurophysiological data on the brain areas involved in the generation and recognition of object-directed actions. In particular we will describe functional properties of the so called parieto-frontal circuits involved in visuo-motor transformations transforming visual information into actions. We will describe in depth the functional properties of mirror neurons, and in particular their classification into different classes on the basis of the congruence between activity during action execution and action observation.

In *Chapter 3* we will review current computational models of mirror neurons. We will give a critical description of each of them. It will turn out that current computational models are based on two main hypotheses: same-activity and same-input. Taken together, the two hypotheses, entail that the activity of mirror neurons is a trivial consequence of view-independence property of its inputs. However the same-input hypothesis posits severe computational challenges, insofar as a view independent scene description must be computed from visual input only. To underline the functional implications of the same-input hypothesis in perceptual processes the NeGOI model is reviewed at the end of the chapter. NeGOI is a computational model for extracting grip-aperture (which may be one of the input signals received by mirror neurons) in a view-independent fashion. Due to the view-independent property and the use of visual information only, NeGOI model is in accord with the same-input hypothesis.

In *Chapter 4* we will introduce a new computational model of mirror neurons in which the same-activity and same-input hypotheses are dispensed with. In particular, the functional interaction between sensory input and mirror activation mechanisms is significantly modified according to a special interpretation of the direct-matching hypothesis. The pro-

posed approach is investigated in a simplified scenario in which available motor knowledge is restricted to sets of hand configurations that a hand may assume in the course of specific kinds of object-directed actions; and sensory inputs are restricted to hand-related visual inputs collected from some fixed viewpoint. Accordingly, the simplified perceptual problem to solve is that of estimating actual hand configurations on the basis of both motor information and incoming visual inputs.

In *Chapter 5* we will deal with the specification of the model and the description of both the problems and the proposed solutions. In particular we will show in detail that the mapping from visual input to hand configuration is an ill-posed problem and the probabilistic framework may be used to model such mapping. Moreover we will show the existence of different sets of eigenpostures for different classes of grasping actions and we will describe the GA model as a way to select the initial sets of eigenpostures.

In *Chapter 6* we will show experimental results of the proposed approach.

In *Chapter 7* we will discuss the goals attained in this work and open questions to be addressed in the future.

Mirror neurons

Mirror neurons form a class of visuo-motor neurons identified in the macaque's F5 cortical motor area with the property of becoming active during both execution and observation of object-directed actions. These neurons were first described in seminal work by Giacomo Rizzolatti and co-workers (Rizzolatti et al., 1996; Gallese et al., 1996) and seem to be involved in a circuit of cortical areas – usually referred to as mirror system (Cattaneo and Rizzolatti, 2009; Rizzolatti and Craighero, 2004) – subserving the control of one's own actions and the recognition of observed actions.

Various computational models have been advanced to account for the behaviour of mirror neurons in the broader context of mirror system functionalities (for a recent review, see Oztop et al. (2006b)). We will give a detailed description and analysis of such models in Chapter 3.

In order to illustrate current computational models of mirror neurons, focussing on same-activity and same-input hypotheses and related drawbacks, a more extensive description of mirror neurons is needed.

Thus, we turn to present a review of significant neurophysiological information about other brain areas strictly related to mirror neurons behaviour.

2.1 Biological Background

In this section we will give a review of significant neurophysiological data on the brain areas involved in the generation and recognition of the so called “object-directed” actions. The expression “object-directed” action is used to denote a series of movements that relate body parts (effectors like hand or mouth) of a primate to a three-dimensional object, e.g. *grasping a*

2.1. BIOLOGICAL BACKGROUND

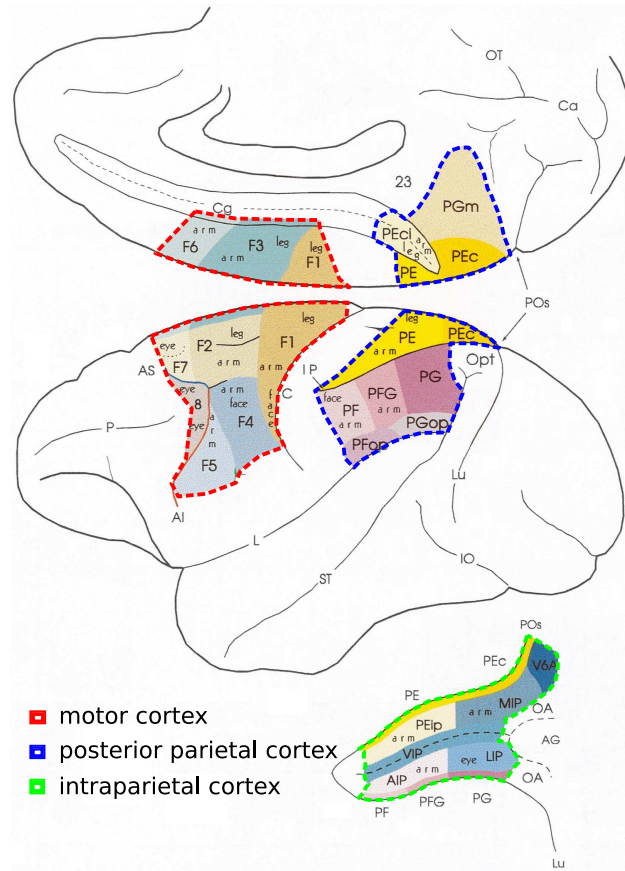


Figure 2.1: Schematic illustration of posterior parietal lobe, intra-parietal sulcus and motor cortex of monkey brain. Different colors represent different areas (picture taken from (Geyer et al., 2000)).

piece of food by a precision grip or tearing a sheet of paper are object-directed actions.

What follows is related to the brain of macaque if not differently specified.

This section will form the background for Chapter 3 and Chapter 4. In Chapter 3 we will make a critical analysis of extant computational models of mirror neurons. In Chapter 4 we will propose a new approach to the modelling of mirror neurons.

The brain regions of major interest for this work are the posterior parietal cortex, intra-parietal sulcus and motor cortex (see Figure 2.1). Recently, both motor and parietal cortex have been subdivided into different areas mainly on the basis of their functional properties (Geyer et al., 2000) (see Figure 2.1 different colors represent different areas).

2.1. BIOLOGICAL BACKGROUND

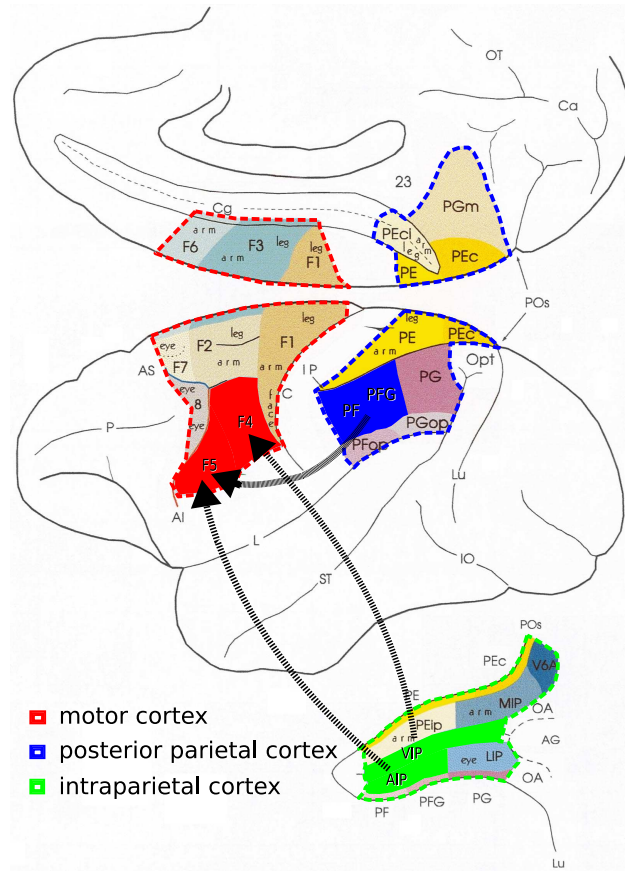


Figure 2.2: The three main parieto-frontal circuits involved in the execution and recognition of object-directed actions: VIP-F4, AIP-F5 and PF-F5.

In a standard interpretation, motor cortex is functionally involved only in the generation and control of actions, while the parietal and intra-parietal cortex is interpreted as an associative area integrating sensory information from various parts of the body. However, recent studies on motor areas, including those on mirror neurons, suggest that some motor areas are strongly involved in additional functional tasks such as the recognition of actions executed by others.

Some motor areas receive afferent connections from parietal areas and, in many cases, a motor area receives connection mainly from one parietal area leading to the so called *parieto-frontal circuits* (Matelli and Luppino, 2001). Broadly speaking, these circuits work in parallel and are responsible for visuo-motor transformations, transforming visual information into actions.

Areas mainly involved in object-directed actions are: F1, F4 and F5 mo-

2.1. BIOLOGICAL BACKGROUND

tor areas, PF and PFG posterior parietal areas, VIP and AIP intra-parietal sulcus areas. The motor area F5 consists of two main sectors, one of which is located on the dorsal convexity (F5c), and the other one on the posterior bank of the inferior arcuate sulcus (F5ab).

These areas establish three main circuits working in parallel: the VIP-F4 circuit, AIP-F5ab and PF-F5c circuits (see Figure 2.2). The VIP-F4 circuit is mainly involved in coding peripersonal space and arm movements, the AIP-F5ab circuit is mainly involved in coding object *affordances* and in hand movements, the PF-F5c circuit seems to be involved in the recognition of object-directed actions.

In the next subsections, we will describe these three circuits and in Section 2.1.4 we will summarize their main functional interpretations.

2.1.1 The VIP-F4 circuit

Area VIP is located in the intraparietal sulcus and receives both visual and somatosensory information. On the basis of functional properties, neurons of this area can be categorized into two main classes: *purely visual* (unimodal) neurons and *visual and tactile* (bimodal) neurons. Neurons belonging to the first class are selective to visual stimuli, in particular may become active for expanding or contracting visual stimuli or can be strongly selective for the direction and speed of stimuli moving along the sagittal plane. Bimodal neurons respond independently to visual and tactile stimuli. Interestingly, the visual receptive fields (RFs) are located in parts of the field of vision corresponding to the tactile RFs. Moreover many neurons respond only to visual stimuli located in the peripersonal space and in one third of visually-responsive neurons, the visual RF is encoded in egocentric and not in retinal coordinates. This means that the response of these neurons does not depend on the gaze direction but only on the location of the visual stimuli with respect to the body.

Area VIP is strongly connected to motor area F4. Microstimulation experiments have shown that in this area arm, neck, face and mouth movements are represented. Single neuron recordings have shown that many neurons fire during reaching movements directed toward the body or away from it while they do not respond to distal movements (that is movements of the hand). Neurons of area F4 have similar properties to neurons of area VIP but unimodal neurons are typically tactile while purely visual neurons are very rare.

Taken together these data indicate that this circuit plays a crucial role in encoding peripersonal space and in transforming object locations into

2.1. BIOLOGICAL BACKGROUND

appropriate movements toward them (Rizzolatti et al., 1998).

2.1.2 The AIP-F5ab circuit

Area AIP is located in the intraparietal sulcus (see Figure 2.1). Neurons of this area were studied in monkeys trained to reach and grasp objects of different sizes and shapes. The experiments were carried out both in darkness and in light. Most AIP neurons discharge during grasping of specific objects and their activity is mainly due to hand and finger movements and not to proximal arm movements nor to object position in space.

Three class of neurons have been identified: *motor-dominant*, *visual and motor*, and *visual-dominant* neurons. Neurons of the first class do not show any significant difference in activity when tested in darkness or light however they do not discharge during fixation of the object only. Visual and motor neurons are less active in darkness than in light while visual-dominant neurons fire vigorously only when the stimulus is visible. In contrast to motor-dominant neurons, many visually-responsive neurons discharge during fixation of the objects, even when fixation was not followed by a subsequent grasping movement. Finally, in most visual and motor neurons, the intrinsic characteristics of the object, effective in triggering a neuron and the type of grip encoded by that neuron, coincided. This means that, for example, a neuron which discharges during the fixation of a small object, discharges also during precision-grip actions which are usually used to grasp small objects.

In Figure 2.3 the behaviour of AIP neurons belonging to the three different classes is showed. Visual and motor and visual-dominant neurons were further subdivided on the basis of their activity during object fixation.

Area AIP is richly connected with motor area F5ab where neurons discharge during specific goal directed actions performed with the hand, the mouth or both. According to the action effective in triggering them, F5ab neurons were subdivided into various classes. Among them, the most represented are: grasping, holding, tearing and manipulating neurons. Most "grasping" neurons code specific types of hand prehension, such as for example, precision grip, whole-hand prehension, finger prehension. The temporal relation of neuron discharge with hand movements changes from neuron to neuron. Some neurons fire during the last part of grasping, others start to fire at finger aperture and continue during finger closure, others are activated in advance of the onset of finger movements (Rizzolatti et al., 1988). So area F5ab can be clustered into different subsets, each

2.1. BIOLOGICAL BACKGROUND

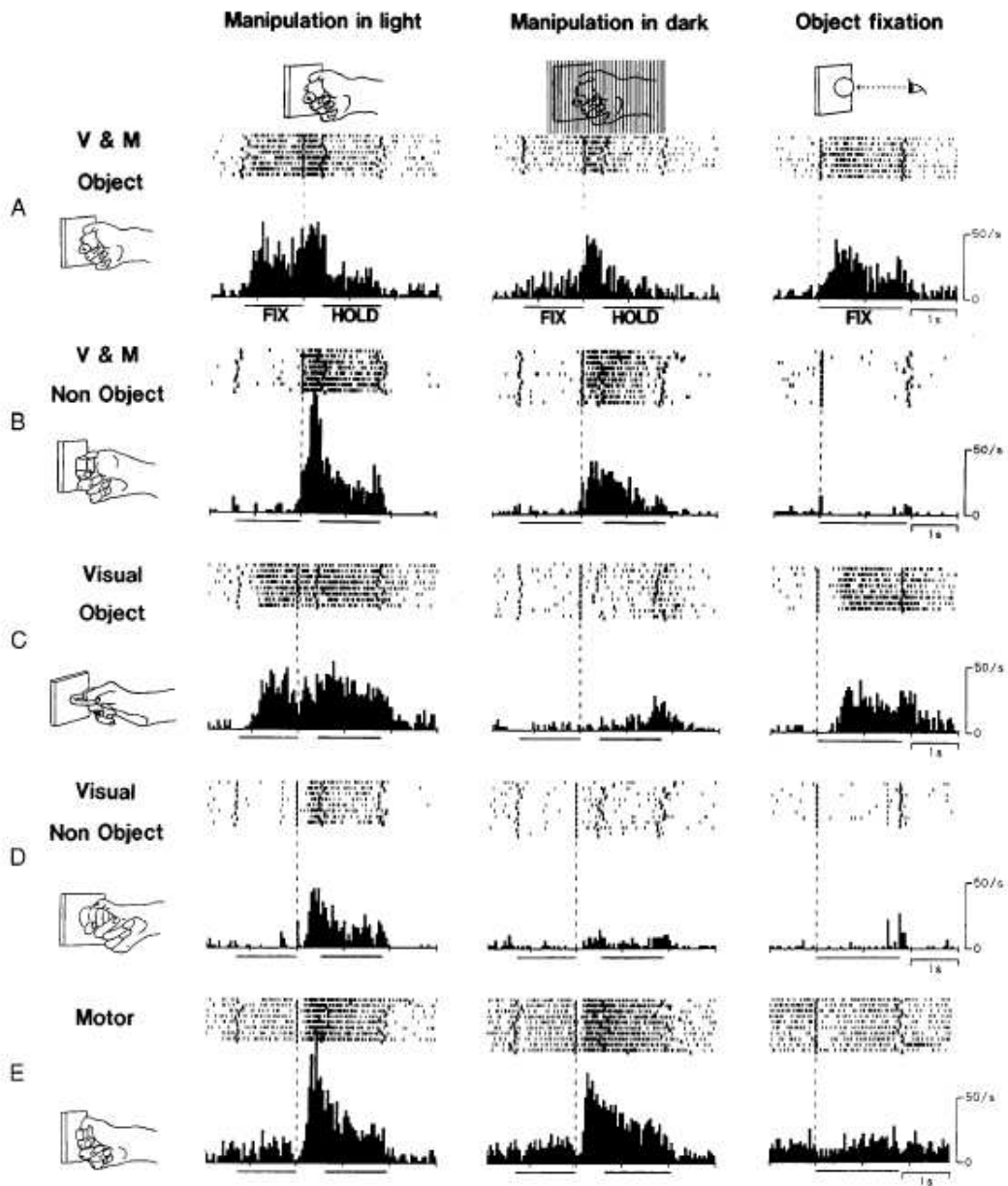


Figure 2.3: AIP neurons' behaviour (picture taken from (Murata et al., 2000)). *Motor-dominant* neurons do not show any significant difference in activity when tested in darkness or light and do not discharge during object fixation. *Visual and motor* neurons are less active in darkness than in light while *Visual-dominant* neurons fire vigorously only when the stimulus is visible. Some visual and motor and visual-dominant neurons fire just during object fixation (*Object*) while others do not fire if the action is not made (*Non Object*).

2.1. BIOLOGICAL BACKGROUND

one responsible for different aspects of the temporal segmentation of the movement (Rizzolatti et al., 1988, 1996). Furthermore, many grasping neurons discharge in association with a particular type of grip. Most of them are selective for one of the three most common grip types of the monkey: precision grip, finger prehension and whole hand prehension. Sometimes there is also specificity within the same general type of grip. For instance, considering the whole hand grasping, the prehension of a sphere, which requires the opposition of all fingers, is coded by neurons different from those coding the prehension of a cylinder, which requires the opposition of all fingers but the thumb.

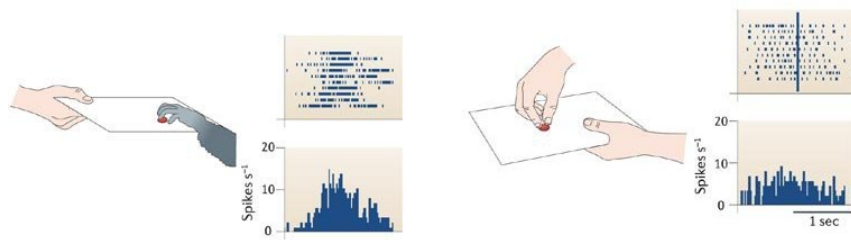
Taken together, the functional properties of F5ab neurons suggest that this area stores a set of motor schemata (Arbib, 1981), or, as it was previously suggested in (Rizzolatti and Gentilucci, 1988), a “vocabulary” of motor acts. Populations of neurons constitute the “words” composing this vocabulary. Some of them indicate the general category of an action (hold, grasp, tear, manipulate). Other populations specify the appropriate way to better adapt the hand to the grasped object (e.g. precision grip specific neurons vs. whole hand specific neurons).

A subset of F5 neurons has also visual properties. We refer to them as “visuo-motor neurons”. The visuo-motor neurons can be clearly partitioned into two classes on the basis of their visual properties: *canonical* neurons and *mirror* neurons (Rizzolatti et al., 1998; Rizzolatti and Craighero, 2004). Canonical neurons are mainly located in area F5ab while mirror neurons are mainly located in area F5c.

Canonical neurons discharge both during the execution of a object-directed action and at the visual presentation of an object at which the action will be directed. Often there is a congruence between the action coded by a given canonical neuron and the observed object that is able to evoke the visual discharge in that neuron. For instance, a canonical neuron that motorically codes a precision grip is also activated when the monkey looks at a small object (because a small object is usually grasped by a precision grip). The most common interpretation for visual discharge in canonical neurons is that there is a close link between the most common three-dimensional stimuli and the actions necessary to interact with them. Thus, every time a graspable object is visually presented, the related F5 canonical neurons “automatically” elicit the appropriate action (Fadiga et al., 2000).

In conclusion, the AIP-F5ab data suggest that this circuit plays a crucial role in transforming the intrinsic properties of the object into the appropriate hand movements (Jeannerod et al., 1995). The description of an object, possibly in terms of their affordances (Murata et al., 1997; Tessitore et al.,

2.1. BIOLOGICAL BACKGROUND



(a) Action execution. The monkey execute a grasping action with hand and the neuron fires vigorously.

(b) Action observation. The monkey observes the experimenter executing the same grasping action and the neuron fires again.

Figure 2.4: Behaviour of a grasping mirror neuron (pictures taken from (Iacoboni and Dapretto, 2006)).

2009), is carried out in AIP and then is transmitted to F5ab, where different types of actions are encoded (Murata et al., 1997). Strong support for a crucial role of AIP-F5ab circuit in visuomotor transformation was recently offered by studies in which the two areas were separately inactivated and the monkey has to perform grasping movements (Fogassi et al., 2001). The main effect observed following independent inactivation of AIP and F5ab was a disruption of the preshaping of the hand during grasping. The deficit consisted in a mismatch between the features of the object that had to be grasped and the posturing of finger movements. When the monkey was successful in grasping the objects, the grip was achieved only after a series of corrections that relied on tactile exploration of the object. These data clearly show that lesion of the AIP-F5ab circuit does not disrupt the ability to perform grasping movements, but only the capacity to transform the 3D properties of the object into appropriate hand movements.

2.1.3 The PF-F5c circuit

As previously said the other visuo-motor neurons of area F5 are the mirror neurons. These respond both when the monkey executes an object-directed action and when the monkey observes another individual (monkey or human experimenter) executing a similar action. In order to be triggered by visual stimuli, mirror neurons require an interaction between a biological effector (e.g. hand or mouth) and an object (see Figure 2.4).

The sight of an object alone, of an agent mimicking an action, or of an individual making intransitive (non object-directed) movements are all ineffective. The object significance for the monkey has no obvious influence

2.1. BIOLOGICAL BACKGROUND

on the mirror-neuron response. Actions directed toward a piece of food or a geometric solid produce responses of the same intensity (Rizzolatti and Craighero, 2004). Mirror neurons show a large degree of generalization. Presenting widely different visual stimuli, but which all represent the same action, is equally effective. For example, the same mirror neuron that responds to a human hand grasping an object responds also when the grasping hand is that of a monkey. Similarly, the response is typically not affected if the action is done near to or far from the monkey, in spite of the fact that the apparent size of the observed hand is obviously different in the two conditions (Rizzolatti and Craighero, 2004).

However, Mirror neurons behaviours show some congruence between the observed and executed action, in some cases this congruence is extremely strict, in other cases the congruence is broader. For example in (Gallese et al., 1996) are reported three different classes of mirror neurons on the basis of their visual properties: *strictly congruent*, *broadly congruent*, and *non-congruent*.

Strictly congruent mirror neurons are mirror neurons which exhibit activity during observed and executed actions corresponding both in terms of general action (e.g. grasping) and in terms of the way in which that action is executed (e.g. precision grip).

When there is a similarity, but not identity, between the observed and executed action the mirror neuron is classified as broadly congruent. Three different groups of broadly congruent neurons can be identified. Neurons of the first group are highly specific in terms of motor activity, discharging in association not only with the execution of a single type of action (grasping or holding), but also with a specific way to execute that type of action (e.g. grasping by a precision grip or finger prehension or whole hand prehension). However, unlike the strictly congruent neurons, they respond to the observation of various ways to execute a type of action (e.g. response for precision grip is different from that for whole hand prehension). A second group of broadly congruent mirror neurons is constituted of neurons less specific in terms of motor activity than the first group, that is they become active during one type of executed action regardless of the way the action is executed, but visually respond to two or more different type of actions (e.g. manipulation and grasping). The last group of broadly congruent neurons seem to be activated by the "goal" of the observed action regardless of how it was achieved. All these neurons are neurons that become active during the execution of grasping movements performed by the monkey itself, while they are activated by the observation of grasping movements performed by the experimenter with either hand or mouth (Gallese et al., 1996).

2.1. BIOLOGICAL BACKGROUND

Classification	Percentage
STRICTLY CONGRUENT	32% (29/92)
BROADLY CONGRUENT	61% (56/92)
GROUP ONE	8% (7/92)
GROUP TWO	50% (46/92)
GROUP THREE	3% (3/92)
NON-CONGRUENT	7% (7/92)

Table 2.1: Experimental data reveal complex patterns of mirror behaviours, in the way of activation congruence between execution and observation modes. The table summarizes experimental data wrt observed and executed action congruence as reported in Gallese et al. (1996).

Finally, mirror neurons which exhibit a response with a no clear-cut relationship between the observed action and the executed action movement of the monkey are called non-congruent.

Note that, the temporal relationships between action phases and mirror neuron firings have been less systematically and extensively investigated to the present date. However, different temporal patterns of mirror neuron activations were experimentally isolated: some mirror neurons become inactive as soon as an action is completed; some other mirror neurons keep on firing thereafter (Gallese et al., 1996); and mirror neurons were also recorded whose responses are peaked at different action phases (Umiltà et al., 2001).

Insofar as the connectivity of mirror neurons with other cortical areas is concerned, note that F5c area, where mirror neurons are mainly located, receives strong connections from parietal area PF. This area is in turn connected to temporal area STS. It has been shown in (Fogassi et al., 2005) that neurons with properties similar to mirror neurons exist in PF parietal area. In particular neurons of this area were studied when monkeys performed motor acts (e.g., grasping) embedded in different actions (e.g., grasping for eating wrt grasping for placing) and when they observed similar acts done by an experimenter. Mirror neurons of area PF respond to some grasp motor act (executed or observed) only when it is embedded in a specific action (e.g. grasping for eating but not during grasping for placing). Due to their behaviour it has been hypothesized that these neurons not only code the observed motor act but also allow the observer to understand the agent's intentions.

It was suggested by Keysers and Perrett (2004) and Oztop et al. (2005) that cortical area STS is involved in the process of extracting some high-

2.1. BIOLOGICAL BACKGROUND

level visual features. In particular, “shape-selective” cells have been found in area STS which respond very selectively to hand/object interactions exhibiting a mirror-like property (Keysers and Perrett, 2004; Perrett et al., 1989). Indeed, these neurons selectively respond to object-directed actions such as tearing, grasping, and manipulating in a view-invariant fashion. Unlike mirror neurons, however, shape-selective neurons fail to exhibit motor properties.

In conclusion PF-F5c circuit is strongly involved in the recognition of action made by others (Rizzolatti et al., 2001).

2.1.4 Relevant functional interpretation

We can summarize the above as follows:

- ▷ VIP-F4 circuit plays a crucial role in encoding peripersonal space and in transforming object locations into appropriate movements toward them.
- ▷ AIP-F5ab circuit plays a crucial role in transforming the intrinsic properties of the object into the appropriate hand movements.
- ▷ PF-F5c circuit is involved in the recognition of action made by others.

In particular as far as the circuits AIP-F5ab and PF-F5c are concerned, we can summarize experimental results as follows:

- ▷ F5 is a macaque’s cortical motor area strongly involved in *object-directed actions*. There is a strong congruence between the performed action and the discharge of F5 neurons. Most of them are selective for the general type of action (grasping, holding, tearing), that is, for *action type*. Furthermore some of them can be selective for both the general type of action and how that type of action is executed (precision-grip, finger prehension, whole hand prehension), that is, for *action modality*. Also, some neurons show a clear relation between their activity and the temporal phases of an action.
- ▷ In F5 area there is a subset of neurons which show visual properties: these are the visuomotor neurons. The set of visuomotor neurons can be partitioned into two classes: canonical neurons and mirror neurons.
- ▷ Canonical neurons discharge during both the execution of an object-directed action and at the sight of an object alone. If a canonical

2.1. BIOLOGICAL BACKGROUND

neuron discharges at the visual presentation of an object then it discharges also during the execution of actions which are usually performed on that object.

- ▷ Mirror neurons discharge during both the execution of an object-directed action and the observation of an object-directed action. If an F5 mirror neuron discharges during the execution of an object-directed action then it discharges also during the observation of an action that:
 - ▷ corresponds both in action type and in action modality (*strictly congruent mirror neurons*);
 - ▷ corresponds in action type only (*broadly congruent mirror neurons group 1*);
 - ▷ corresponds in action type but it responds also during the observation of other kinds of actions (*broadly congruent mirror neurons group 2*);
 - ▷ corresponds in action goal irrespectively of the effector used (*broadly congruent mirror neurons group 3*);
 - ▷ does not correspond in either action type or in action modality (*non-congruent mirror neurons*).
- ▷ Several findings suggest that the discharge of some mirror neurons is temporally linked to the phases of an observed action.
- ▷ Mirror neurons show a large degree of tolerance to scale, position and point of view.

Critical analysis of current computational models of mirror neurons

Various computational models have been advanced in literature in order to account for the behaviour of mirror neurons (see (Oztop et al., 2006b) for a relatively recent review). Mirror neurons are usually modelled there (Haruno et al., 2001; Keysers and Perrett, 2004; Ito and Tani, 2004; Oztop et al., 2005; Oztop and Arbib, 2002) in accordance with the following hypotheses:

- ▷ **Same activity:** Let \mathcal{A} be an object-directed action. A mirror neuron exhibits the same activity irrespective of whether \mathcal{A} is carried out or observed.
- ▷ **Same input:** Let \mathcal{A} be an object-directed action and let $m_{\mathcal{A}}$ be any mirror neuron which becomes active whenever \mathcal{A} is carried out or observed. Then, the same input signals are received, in both execution and observation conditions, by $m_{\mathcal{A}}$ and any other F5 neuron which directly affects $m_{\mathcal{A}}$'s behaviour. These input signals are the outcome of computational processes which do not involve the motor system.

The predictive and explanatory implications of these hypotheses are examined in Sections 3.1.1 and 3.3. The same-activity hypothesis turns out to be an idealization in the light of known experimental data about mirror neuron activation behaviours. And the sweeping functional implications of the same-input hypothesis are analyzed in the light of perceptual processes one has to posit to let mirror neurons receive the same input data, irrespectively of whether one is in action observation or in action execution conditions. The main upshot of this analysis is that computational

3.1. SAME-ACTIVITY HYPOTHESIS

models endorsing both same-activity and same-input hypotheses are descriptively inadequate and functionally uninformative: descriptively inadequate, insofar as these models fail to account for a wide variety of mirror neuron behavioural data; and functionally uninformative since mirror mechanisms do not play significant functional roles especially insofar as sensory processing is concerned. The critical analysis of extant computational models endorsing both same-activity and same-input hypotheses prepares the ground for introducing, in Chapter 4, a novel approach to the computational modelling of mirror neurons. There, same-activity and same-input hypotheses are dispensed with; and the functional interaction between sensory input and mirror activation mechanisms is significantly modified. The benefit flowing this approach is twofold: consistently with the direct matching hypothesis (Rizzolatti et al., 2001), a more central functional role is vindicated for mirror mechanisms in sensory processing; and the more substantive use which is thereby made of motor information enables one to simplify the computational processing of sensory inputs in action recognition processes.

3.1 Same-activity hypothesis

To begin with, let us examine the functional import of same-activity and same-input hypotheses in isolation, before turning to consider the predictive and explanatory consequences flowing from their confluence in computational models of mirror mechanisms. Computational models endorsing the same-activity hypothesis (Haruno et al., 2001; Keysers and Perrett, 2004; Ito and Tani, 2004; Oztop et al., 2005; Oztop and Arbib, 2002) predict that the activity of any mirror neuron during the execution of some object-directed action A does not differ from its activity during the observation of A . Finer-grained differences between these models emerge with respect to other features of mirror mechanisms. Let's see.

Distinguishing features of the model MNS1 (which stands for Mirror Neuron System 1 and whose block schema is shown in Figure 3.1) (Oztop and Arbib, 2002) include:

1. A hand-program controlling hand movements (both reach and grasp), computed on the basis of action-oriented object features (affordances) by a procedure which is supposed to model the activity of the AIP-F5 circuit and VIP-F4 circuit.
2. The hand-state hypothesis, according to which visual inputs concerning hand/target-object pairs are processed and coded into a vec-

3.1. SAME-ACTIVITY HYPOTHESIS

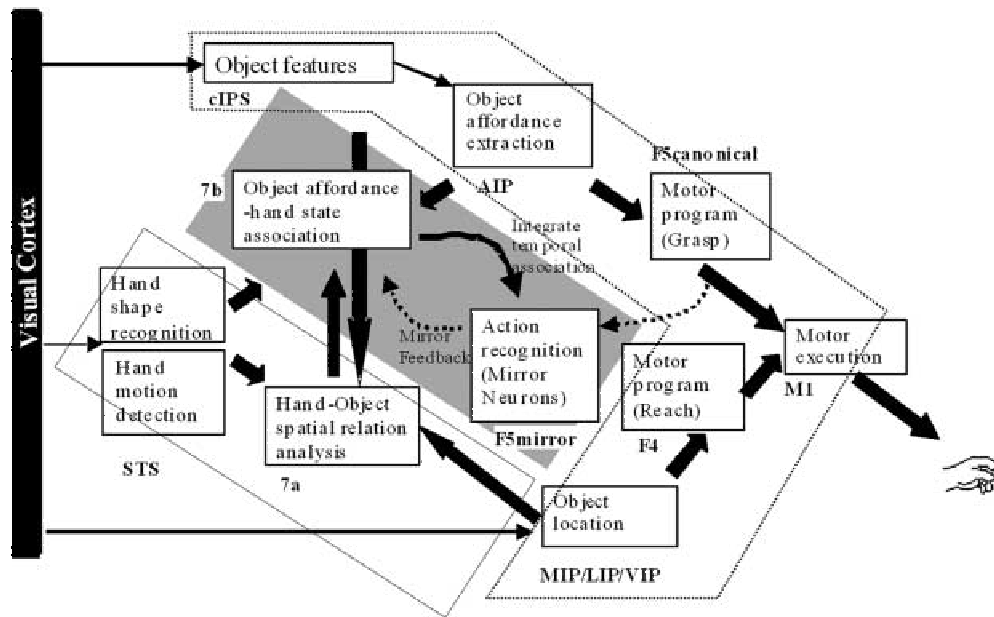


Figure 3.1: The MNS1 model is composed of several functional blocks which are related to the computation of different brain areas. Notably the *Object affordance extraction* and *Motor program(Grasp)* blocks model the AIP-F5 canonical circuit while *Object location* and *Motor program (Reach)* blocks model the VIP-F4 circuit. Mirror neurons are modelled by the *Action recognition* block.

tor, 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. The hand-state hypothesis will be examined in some detail in the next section.

- Mirror behaviours are modelled as the outcome of an action recognition module, which classifies ongoing actions on the basis of computed hand-state sequences. This functional module is implemented by means of a multi-layered feed-forward neural network. Notably, in self-observation conditions this network is trained to associate hand-state sequences to canonically encoded hand-programs. In an updated version of this model (Bonaiuto et al., 2007), a biologically more plausible neural network is substituted for the multi-layered feed-forward neural network.

On the whole, MNS1 construes mirror activity as the output of an action recognition module in both execution and observation modes. This

3.1. SAME-ACTIVITY HYPOTHESIS

action recognition module classifies the ongoing action on the basis of sequences of hand-state vectors. The same-activity hypothesis is embedded in the action recognition module of the MNS1 model, insofar as this module produces the same output irrespective of whether object-directed action \mathcal{A} is executed or observed.

The model presented in (Ito and Tani, 2004) takes the form of a distributed controller system based on a Continuous Recurrent Neural Network (CTRNN). The system includes both parametric and data input lines. The parametric input lines are fed with so-called *Parametric Bias* (PB) vectors, whereas both data input and data output lines are supplied with sensory-motor pairs. The system works in learning, generation, and recognition modes.

In learning mode, the system learns a mapping between PB vectors and behavioural patterns, identified with temporal sequences of sensory-motor pairs (s_t, m_t) . When learning is completed, the system operates in either generation or recognition mode. In generation mode, a specific PB vector fed into the parametric input line activates a closed-loop computation process: for each sensory-motor pair (s_t, m_t) given as data input at step t , the system predicts its updated value at step $t + 1$, producing on the output lines a sensory-motor pair (s_{t+1}, m_{t+1}) which is fed back into the system as data input. Accordingly, for each PB vector, the system generates sequences of sensory-motor pairs without using actual sensory inputs. In observation mode, the data input lines receive an actual sensory input s_t at step t . Using the current PB vector as supplementary information, the system generates an expected sensory value s_{t+1} on the output lines. This expected value is compared with actual sensory input s_{t+1} at step $t+1$, and the prediction error is used to update the PB vector by means of a “back-propagation through time” algorithm. Thus, if the system receives sensory sequences that are sufficiently similar to previously learned sequences, then the PB vectors tend to converge to values determined in the learning phase. This schematic account enables one to identify the same-activity hypothesis in this model too: mirror neuron behaviours are modelled by PB vectors, which assume the same values whenever some given action \mathcal{A} is either executed or observed.

Let us finally consider the forward model of mirror neuron behaviours proposed in (Oztop et al., 2005).

1. In action execution mode, the forward model predicts next sensory stimuli with respect to some desired bodily change. This information is used for action control purposes, in order to compensate for sensory delays involved in visual feedback loops.

3.1. SAME-ACTIVITY HYPOTHESIS

2. In action observation mode, predictions of next sensory stimuli by the forward model are used to infer the agent's intentions, on the basis of a comparison between computed "simulated perceptions" and actual movement perceptions.

Given some specific object-directed action, the forward model predicts the sensory consequences of action planning in action execution mode, and predicts the sensory correlates of agent's intentions in action observation mode. The same-activity hypothesis is at work in this model too, insofar as the output produced in action observation mode is the closest available match to the response provided in action execution mode.

The same-activity hypothesis is schematically represented in Figure 3.2. Mirror neurons are functionally represented there as black boxes, whose input may include proprioceptive information, perceptual information, and internal states, and whose output is (a code for) mirror neuron activity. Given some specific action \mathcal{A} , z and y represent mirror neuron input and output, respectively, during action execution; z' and y' represent mirror neuron input and output, respectively, during action observation. In this schema, the same-activity hypothesis is enforced by requiring that $y = y'$ for every given action \mathcal{A} .

3.1.1 The gap between same-activity hypothesis and experimental data

On the whole, experimental data reveal complex patterns of mirror behaviours, in the way of both temporal patterns and activation congruence between execution and observation modes. Table 2.1 summarizes experimental data wrt observed and executed action congruence. Only strictly congruent mirror neurons satisfy the same-activity hypothesis, insofar as these neurons are selectively active during the execution and the observation of actions which do not differ from each other in the way of action type and execution modality. Only about one-third of recorded mirror neurons are strictly congruent. Therefore, computational models endorsing the same-activity hypothesis do not fit about two-thirds of experimental data in (Gallese et al., 1996; Rizzolatti et al., 1996). And recent data showing that mirror neurons exhibit different behavioural responses in action observation modality, depending on spatial regions in which the action is being executed (Caggiano et al., 2009), add to the reasons for considering these computational models as descriptively inadequate. In the light of the above data, the same-activity hypothesis is aptly regarded

3.1. SAME-ACTIVITY HYPOTHESIS

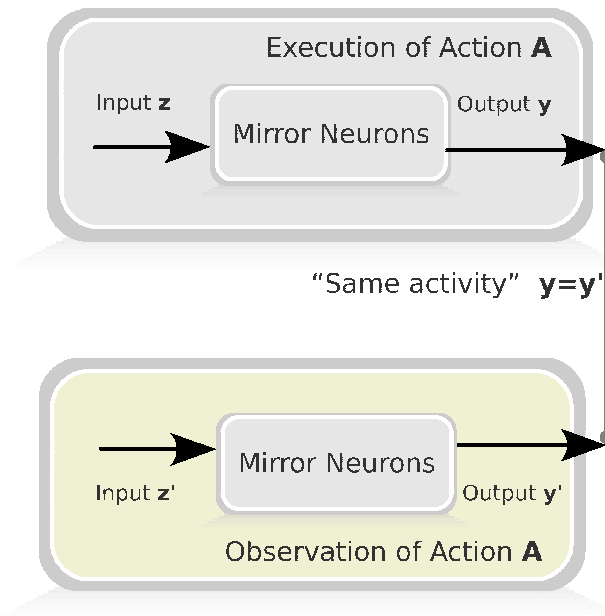


Figure 3.2: Same-activity hypothesis. Mirror neurons are functionally represented as black boxes. Given some specific action \mathcal{A} , z and y represent mirror neuron input and output, respectively, during action execution; z' and y' represent mirror neuron input and output, respectively, during action observation. The same-activity hypothesis is enforced by requiring that $y = y'$ for every given action \mathcal{A} .

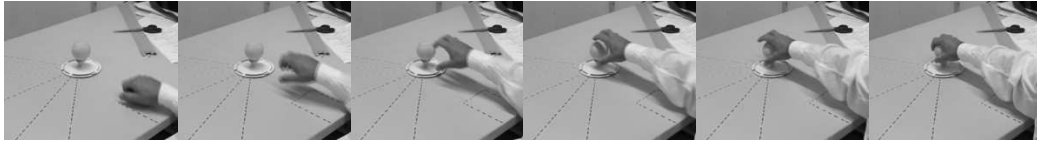


Figure 3.3: Visual sensory input to an executor of some object-directed action \mathcal{A} .

as an idealization, which enables one to abstract from some experimental data for the purpose of simplifying modelling tasks and isolating chief causal factors producing observed phenomena. Accordingly, a pertinent question to ask about computational models of mirror neurons is whether endorsing the same-activity hypothesis brings about significant benefits in the way of prediction or explanation. A negative answer to this question will be provided in Section 3.3, in connection with a special class of mirror neuron models, that is, computational models which jointly endorse the same-activity idealization and the same-input hypothesis. Preliminarily, let us turn to examine in the next section the functional implications which flow from the same-input hypothesis alone.

3.2 Same-input hypothesis

The F5 motor area receives inputs mainly from parietal areas PF and AIP. More generally, let us collectively call direct internal input to mirror neurons the complex of brain signals that mirror neurons and F5 neurons closely involved in mirror neuron activity receive from directly afferent brain areas. The same-input hypothesis significantly bears on the question of when and how direct internal input is computed. Indeed, this hypothesis entails that, for each object-directed action, mirror neurons (and possibly other strictly related F5 neurons) receive the same direct internal input irrespectively of whether some object-directed action \mathcal{A} is observed or executed. Accordingly, this assumption expresses a strong modelling commitment insofar as the actual sensory, proprioceptive, and internal state inputs to action executors do not, in general, coincide with sensory, proprioceptive, and internal state inputs to action observers. Consider, for example, the visual sensory input which is collected from the perspective of the executor of some object-directed action \mathcal{A} (Figure 3.3).

In general, this perspectival input differs (on account of observation angle, distance, illumination conditions, and so on) from the visual sensory input to an observer of the same action \mathcal{A} (Figure 3.4). Thus, sub-

3.2. SAME-INPUT HYPOTHESIS



Figure 3.4: Visual sensory input to an observer of some object-directed action \mathcal{A} .

stantive pre-processing is in many cases required in order to converge on the same direct internal input to mirror neurons (see, for example, (Prevete et al., 2008)), when starting from the different input collection conditions of action executors and action observers, respectively. To express schematically the same-input hypothesis, let us encapsulate once again mirror neuron computations as a functional module (see Figure 3.5).

In this schema, given some action \mathcal{A} , \mathbf{z} is the internal direct input to mirror neurons during action execution, and \mathbf{z}' is the internal direct input to mirror neurons during action observation. The same-input hypothesis amounts, in this schema, to assuming that for each given \mathcal{A} , $\mathbf{z} = \mathbf{z}'$ (see Figure 3.5). Since the total sensory, proprioceptive, and internal state input S_e to an action executor is usually quite different from the total sensory, proprioceptive, and internal state input S_o to an observer of the same action, then the same-input hypothesis entails that there are mechanisms transforming S_e into \mathbf{z} and S_o into \mathbf{z}' , which satisfy the additional restrictive condition that $\mathbf{z} = \mathbf{z}'$. To illustrate, consider again sensory inputs in the visual modality only. The same-input hypothesis entails the condition that viewpoint-independent information about object-directed actions can be extracted from different visual sensory inputs, and supplied to mirror mechanisms. Thus, in general, the pre-processing step presupposed by the same-input hypothesis involves extensive computational mappings into perspective-free perceptual information of perspectival visual inputs, that is, of visual inputs collected from the vantage points of action observers or executors, respectively (see Figure 3.6).

Current computational models of mirror neurons usually endorse the hypothesis that perspective-free perceptual information is fed into mirror mechanisms. Some of these models, however, merely presuppose that this perceptual processing problem admits a computational solution; some other models which outline a partial solution, that is, a solution which involves special restrictions on sensory input collection conditions. Let's see. Consider the hand-state hypothesis in (Oztop and Arbib, 2002), according to which one computes a hand-state vector making available perspective-free information concerning hand/target-object pairs. In this model, se-

3.2. SAME-INPUT HYPOTHESIS

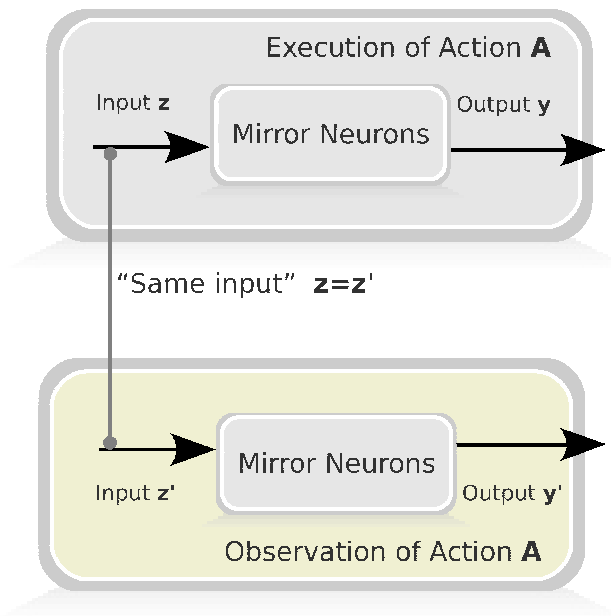


Figure 3.5: Same-input hypothesis. Mirror neurons are functionally represented as black boxes. Given some specific action \mathcal{A} , z and y represent mirror neuron input and output, respectively, during action execution; z' and y' represent mirror neuron input and output, respectively, during action observation. The same-input hypothesis amounts to assuming that for each given \mathcal{A} , $z = z'$.

3.2. SAME-INPUT HYPOTHESIS

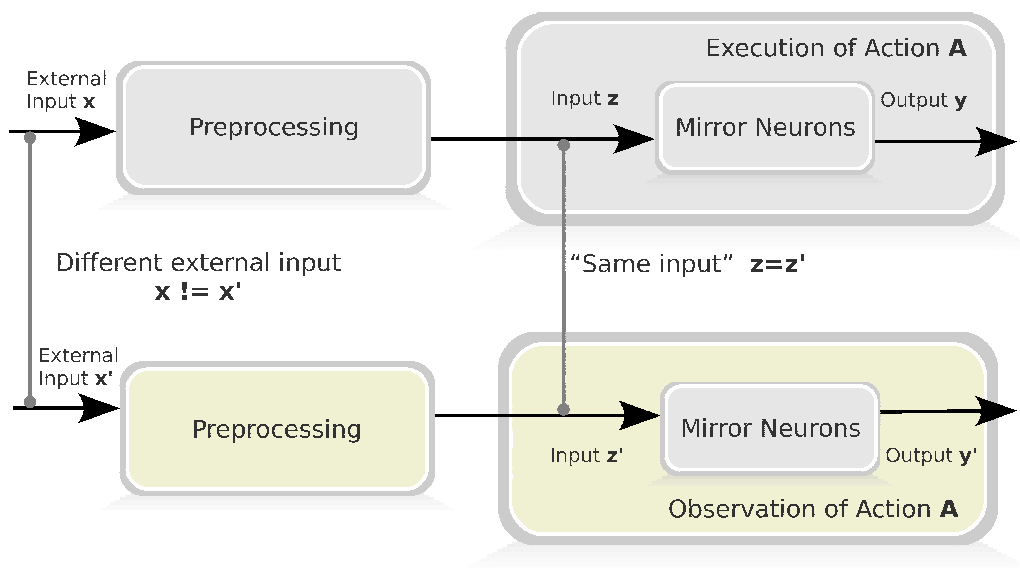


Figure 3.6: Pre-processing. Mirror neurons are functionally represented as black boxes. Given some specific action \mathcal{A} , x and x' represent perspectival sensory input from executor and observer vantage points, respectively. To obtain the same direct internal input ($z = z'$) starting from different perspectival inputs ($x \neq x'$) a substantive pre-processing step is required.

3.2. SAME-INPUT HYPOTHESIS

quences of hand-state vectors are the input data for an action classification module, whose output is identified with mirror neuron activity. Therefore, this model of mirror mechanisms presupposes substantive pre-processing steps, which enable one to extract invariant (perspective-free) information about object-directed actions from both executor and observer sensory inputs. Likewise, in (Haruno et al., 2001; Keysers and Perrett, 2004; Fritsch, 2007; Ito and Tani, 2004; Oztop et al., 2005; Oztop and Arbib, 2002), one presupposes that a view-independent description of the environment is computed and fed into mirror neuron mechanisms. A procedure for attaining view-independence is outlined in some other computational models in terms of frame of reference transformations (see Weber et al. (2008) for a review). In (Billard and Mataric, 2001), for example, view-independence is achieved by means of the following two-stage process: direction and orientation information about the executor's arms is extracted at first by reference to some selected point on the executor's body; this information is subsequently transferred onto the observer's frame of reference. This system affords a solution to the same input pre-processing problem only if a non-occluded and visually unambiguous initial position is perceptually available. More important, this solution presupposes that a kinematic model of a generic adult human is available for perceptual processing. In (Demiris and Hayes, 2002; Demiris and Johnson, 2003) a frame of reference transformation is carried out under the physical constraint that executor and observer be located in front of each other. To sum up. Some computational models of mirror neurons presuppose a solution to perceptual processing problems one has to posit in the light of the same-input hypothesis, insofar as one assumes there that the outcome of these perceptual processes is made available as an external input to the computational model. Some other computational models provide partial solutions to the same perceptual processing problems, insofar as special restrictions on sensory data collection and processing conditions are required for the proposed solution to work (such as the above mentioned physical constraint that executor and observer be located in front of each other). A general formulation of the same internal input computation problem raises several modelling challenges. A major mathematical problem arising in this context concerns the ill-posed character of many required transformations from perspectival sensory data to intrinsic features of object-directed actions. To illustrate, suppose that the description one is looking for is given by a sequence of configurations that a hand takes on during an object-directed action. However, the same visual input can be associated to various hand configurations, insofar as the hands of primates are highly complex structures including more than 20 degrees of

3.3. THE CONFLUENCE OF SAME-ACTIVITY AND SAME-INPUT HYPOTHESES

freedom, perceptually producing many different self-occlusions. Therefore, many visual sensory data x collected from any given vantage point are compatible with various distal hand configurations, and this fact suffices to conclude that the extraction of direct internal input z from any such x is an inverse ill-posed problem (Friston, 2005; Fritsch, 2007). If one allows for multiple vantage points in visual data collection processes, then self-occlusion affects increased numbers of visual inputs; by the same token, increased numbers of inverse ill-posed transformation problems will arise too. Additional information enabling one to turn these ill-posed, visual perception problems into well-posed functional mapping problems is provided in (Billard and Mataric, 2001), where a kinematic model of primate actions is made available. However, the underlying assumption that primates are able to compute kinematic models of this sort requires extensive empirical scrutiny.

3.3 The confluence of same-activity and same-input hypotheses

Let us now turn to consider the problematic character of the functional implications flowing from the conjunction of same-input and same-activity hypotheses. To preserve generality, let us suppose that mirror neuron activity is the outcome of two different computations, taking place during object-directed action execution and observation, respectively. If one encapsulates mirror neuron computations as functional modules (see Figure 3.7), this supposition entails that one has to posit two different functional modules F and G , encapsulating mirror activity in action execution and observation modes, respectively. Mirror neuron input and output during the execution of object-directed action \mathcal{A} are represented in F by means of z and $y = F(z)$, respectively. And mirror neuron input and output during the observation of \mathcal{A} are represented in G by z' and $y' = G(z')$, respectively. As discussed above, the same-input hypothesis implies that modules F and G receive the same direct internal input, irrespectively of whether \mathcal{A} is observed or executed. Thus, for every given \mathcal{A} , one obtains that $z = z'$. And, by the same-activity hypothesis, for every such \mathcal{A} one has that $y = y'$. Thus, the conjunction of same-input and same-activity hypotheses forces one to conclude that functional modules F and G coincide (see Figure 3.7). One can hardly belittle the consequences of this fact for modelling purposes. Indeed, to model mirror neuron activity under both hypotheses, it is sufficient to model mirror activity in ac-

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

tion execution mode only (or, alternatively, in action observation mode only). Accordingly, any function F (alternatively, any function G) will do, as long as F (or else G) is consistent with experimental data during action execution (during action observation, respectively). For this reason, computational models endorsing both same-activity and same-input hypotheses (Haruno et al., 2001; Keysers and Perrett, 2004; Oztop et al., 2005; Oztop and Arbib, 2002) are bound to assign an impoverished functional role to mirror neurons. In (Ito and Tani, 2004), mirror neuron activity is accounted for in terms of two genuinely different functional modules. In fact, one construes mirror neuron activity there as the outcome of a learning process during action observation, and as the output of some external process during action execution. In this case too, however, the model embodies a view-independence hypothesis about action description, so that the distinguishing features of mirror behaviours arise as a mere “side effect” of view-independence computational abilities. So far, descriptive and explanatory problems affecting extant computational models of mirror mechanisms have been isolated and analyzed. A novel approach to the computational modelling of mirror mechanisms is introduced in the next section, which dispenses with both same-activity and same-input hypotheses. Equally important, this approach makes room for more substantive functional roles of mirror mechanisms in action recognition processes.

3.4 A case study: the computation of grip-size

In the previous section we have argued that most computational models of mirror neurons usually endorse the hypothesis that perspective-free perceptual information is fed in input to mirror neurons. A general formulation of the same internal input computation problem raises several modelling challenges. A major mathematical problem concerns the ill-posed character of many required transformations from perspectival sensory data to intrinsic features of object-directed actions. This is particularly true for perspectival sensory data comprising only visual input and features of object-directed actions comprising a detailed hand description. A way to alleviate the ill-posed nature of this problem is to consider only a restricted number of features useful to object-directed actions description. For example if we consider the hand alone how many different features must be taken into account, and how detailed must the hand description be for the purpose of both recognition and grasping control tasks?

Grip aperture (Jeannerod, 1984), that is, the aperture between index finger and thumb, appears to be a key element of this set, insofar as it

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

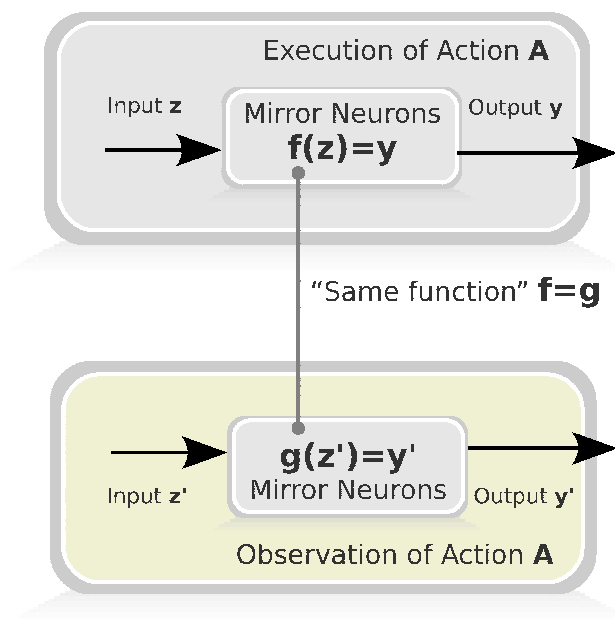


Figure 3.7: Same function. Mirror neurons are functionally represented as black boxes. Given some specific action \mathcal{A} , z and y represent mirror neuron input and output, respectively, during action execution; z' and y' represent mirror neuron input and output, respectively, during action observation. As discussed in the text, the same-input and same-activity hypotheses imply that, for every given \mathcal{A} , $z = z'$ and $y = y'$, respectively. Thus, the conjunction of same-input and same-activity hypotheses forces one to conclude that the functional modules F and G coincide.

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

is a crucial variable in the dynamical evolution of certain types of grasping actions. Some have even advanced and supported the hypothesis that grasping actions are basically coded in terms of changes in grip aperture ((Castiello, 2005) and (Jeannerod, 1984)). During a reach-to-grasp action, grip aperture initially increases until a maximum value is reached which exceeds object size; then grip aperture gradually decreases until it matches the actual object size; the grip-aperture largest value (maximum grip aperture) is reached within 60–70% of the grasp action duration and is linearly correlated with the size of the object. On the whole, grip aperture is a good candidate for being included into a parsimonious set of hand high-level features that are sufficient to describe overall hand movement during reach-to-grasp actions. Indeed, grip aperture is a relationship between fingers which plays a pivotal role in grasp actions, and the position of the other fingers correlates to grip movement because of the above-mentioned hand synergies. One should be careful to notice that in addition to grip aperture further object/hand features are certainly needed to describe ongoing actions in the general setting of object-directed action recognition. For example, at least one object property or relationship between hand and target, such as hand–object distance, is clearly needed.

In (Prevete et al., 2008) we have proposed a biologically plausible computational mechanism for extracting grip aperture in a view-independent fashion, whose architecture and experimental results are described in the next sections.

3.4.1 Model description

We have developed a neural network architecture and system for measuring grip aperture in an observer-independent way (NeGOI). This architecture, as it is pointed out in next section, is coherent with the computational model of the visual ventral stream for view-independent object recognition proposed by (Giese and Poggio, 2003; Riesenhuber and Poggio, 2002).

An assumption built into NeGOI is that grip aperture can be measured from the superposition of a small number of prototypical hand shapes corresponding to predefined grip-aperture sizes. In Figure 3.8 three prototypical hand shapes are shown, corresponding to fully opened grip aperture (PHS_1), middle size grip aperture (PHS_2), and fully closed grip aperture (PHS_3). The selection of these prototypical hand shapes, which have been used throughout the experimental work reported in the next section, is the outcome of a trade-off between model complexity and an effective algorithmic capacity to estimate grip aperture.

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE



Figure 3.8: Prototypical hand shapes. In NeGOI the grip aperture is measured from the superposition of three prototypical hand shapes: fully opened grip aperture (PHS_1), middle size grip aperture (PHS_2) and fully closed grip aperture (PHS_3). In this figure, from left to right, PHS_1 , PHS_2 and PHS_3 , respectively, are shown from a specific point of view.

The key idea underlying the NeGOI model is to introduce view-independent units (VIP units) that are selective for the prototypical hand shapes, and to integrate the output of VIP units in order to compute grip aperture. In particular, given a novel hand shape, the VIP units provide a similarity measure with respect to each prototypical hand shape. These three measures are integrated to estimate actual grip aperture. The integration is performed by a further unit, called here grip aperture (GA), which outputs a measure of the grip size in a view-independent fashion. View independence of VIP units is achieved by a “pooling operation” over a set of view-dependent units (VDP units) which are selective to both a prototypical hand shape and a specific viewpoint. Note that the prototypical hand shapes in question do not depend on the specific action.

The overall NeGOI approach can be briefly described as follows. There is a set of computing units hierarchically organized into three modules (see Figure 3.9). The output of the first module is computed by units which are selective to visual complex features and generalize across changes in scale and position. The second module is composed of both the view-dependent units VDP and the view-independent units VIP. The VDP units receive in input the output from the first module and send their output to the VIP units. The VDP units are selective to a specific prototypical hand shape, and generalize across transformation of the preferred stimulus to changes in scale and position, but fail to generalize across changes in viewpoints. The VIP units are selective to a specific prototypical hand shape and generalize across transformation of the preferred stimulus to changes in scale, position, and viewpoint. The output of the third module is computed by the GA unit. The third module receives in input similarity measures, computed by each VIP unit, of the current shape with respect to all the prototypical hand shapes. On the basis of these similarity measures, the GA unit computes the grip-aperture measure, normalized in the range from 0 (fully closed grip aperture) to 1 (fully opened grip aperture), by an “inte-

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

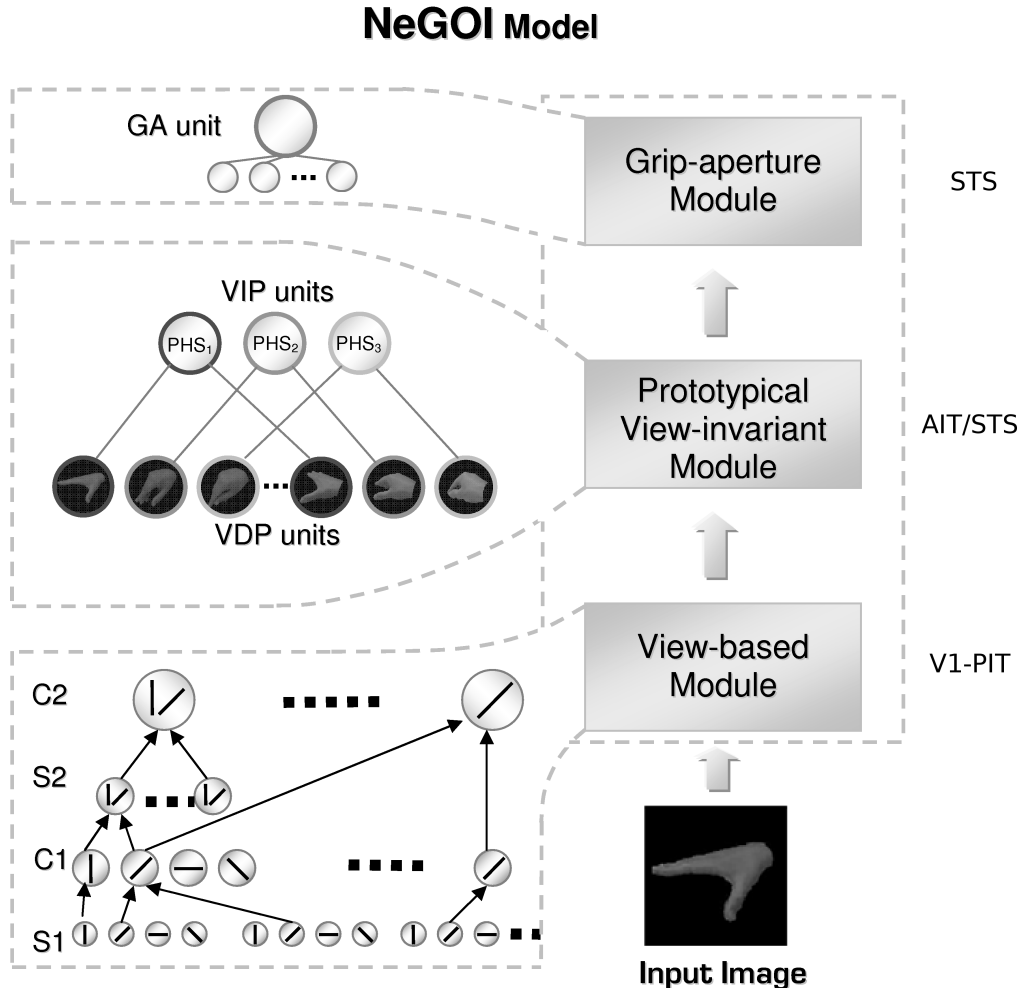


Figure 3.9: NeGOI model is composed of computing units hierarchically organized in three modules: view-based module, prototypical view-invariant module and grip-aperture module. The output of the first module is computed by units which are selective to visual complex features and generalize across changes in scale and position. The second module is composed of both view-dependent units, VDP, and view-independent units, VIP. The VDP units are selective to a specific prototypical hand shape, and generalize across transformation of the preferred stimulus to changes in scale and position, but fail to generalize across changes in view-points. The VIP units are selective to a specific prototypical hand shape and generalize across transformation of the preferred stimulus to changes in scale, position, and viewpoint. The output of the third module is computed by the GA unit. The GA unit computes the grip-aperture measure, normalized in the range from 0 (fully closed) to 1 (fully opened).

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE



Figure 3.10: Grip aperture. The figure shows four different grip apertures corresponding to roughly 2, 4, 6, and 8 cm.

gration" operation. The NeGOI architecture is explained in more details in appendix A.

In order to verify the capability of our system to measure actual grip aperture, and to test the view-independence property, two different sets of experiments were performed. These experiments were conducted complying with the experimental procedures described in appendix A.

3.4.2 Testing the model

Grip aperture

In order to test NeGOI's performance in measuring grip aperture, two different experiments were performed. In the first experiment, the correctness of NeGOI with respect to grip aperture measuring is tested. To this purpose five human subjects were asked to assume four hand shapes corresponding to four preassigned grip-aperture values equal to roughly 2, 4, 6, and 8 cm, respectively. In order to obtain the above-mentioned grip-aperture values each subject sequentially held four cubes whose sizes are equal to 2, 4, 6, and 8 cm, respectively. Each hand shape was recorded using a camera from a fixed viewpoint, thus obtaining a sequence of four images for each subject. Figure 3.10 shows one of the five hand-shape sequences obtained. Each image sequence is given in input to NeGOI, thereby obtaining a sequence of output values, i.e., a set of five NeGOI outputs for each given hand shape. The mean and the standard deviation were computed for each set of the five NeGOI outputs.

Since the four preassigned grip-aperture values increase linearly from 2 to 8 cm, the correctness of the NeGOI behaviour is validated if the computed mean values increase linearly as well.

The NeGOI outcomes showed a clear linear relationship with grip-aperture values (see Figure 3.11). In fact, by performing a linear regression between NeGOI output values and grip-aperture values, one obtains

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

an average determination index $r^2 = 0.98$.

The goal of the second experiment is to verify NeGOI ability “to follow” the grip aperture during a reach-to-grasp action. For this aim, notice that during a grasp action 1) the hand grip-aperture temporal profile has a typical bell shape, i.e., grip aperture initially increases until a maximum value is reached which exceeds object size; then grip aperture gradually decreases until it matches the actual object size, 2) the value of the maximum grip aperture occurs at roughly 70–80% of action duration and, finally, 3) the maximum grip aperture has a linear relationship with target size ((Castiello, 2005; Jeannerod, 1984)).

Thus we have recorded 8 human grasp actions using a camera from a fixed viewpoint (see Figure 3.12 for an example). The targets of these object-directed actions were 8 cubes of different sizes (cm 2, 3, . . . , 9 respectively). Accordingly, a significant test for NeGOI is to compare the properties of the grip aperture computed by the system during a reach-to-grasp action with the expected grip-aperture properties above-mentioned. For each reach-to-grasp action, the output values measured by NeGOI compared with the actual values are shown in Figure 3.13. It turns out that for every recorded grasp action the temporal profile of grip aperture as measured by NeGOI has a bell shape and it is consistent with the actual temporal profile. The maximum grip-aperture value shows a linear relationship with target size (see Figure 3.14). In fact, by performing a linear regression between maximum grip-aperture values and target sizes, one obtains an average determination index r^2 which is roughly equal to 0.85 (roughly 0.90 for actual values). The maximum grip-aperture values occur, on the average, at roughly 80% of action duration.

Hence it turns out that NeGOI is able to measure the grip aperture during a reach-to-grasp action.

Viewpoint independence

In order to test NeGOI’s viewpoint independence, two additional experiments were performed. In the first experiment one tests NeGOI capacity to measure, in a view-independent way, grip apertures corresponding to the three prototypical hand shapes. A subject was asked to assume three hand configurations as close as possible to the three prototypical hand shapes. For each hand configuration, the subject’s hand was recorded while rotating the camera in accordance with a viewpoint range, $view_1$ – $view_3$, roughly equal to 45° (from viewpoint $view_1$ to viewpoint $view_3$ and back). Thus, for each hand configuration, a sequence of about 120 images was obtained (a sample being shown in Figure 3.15). Each image was

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

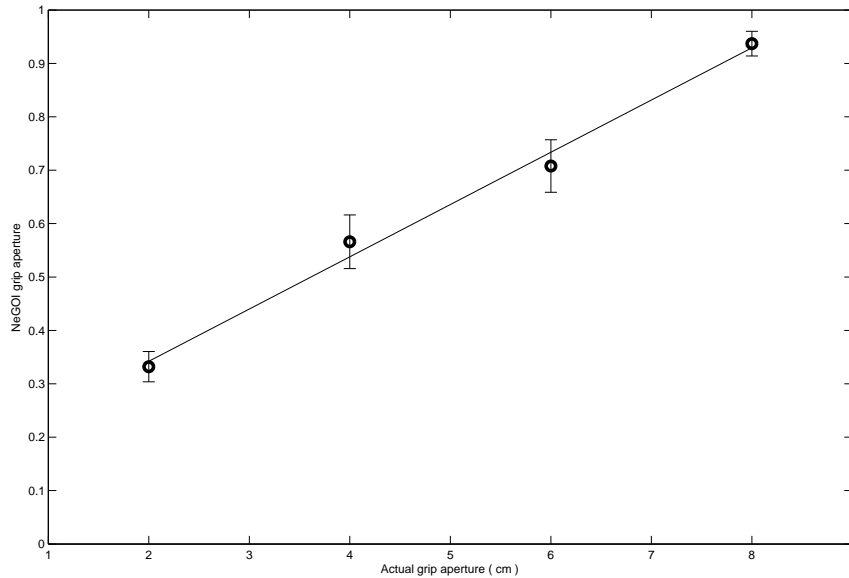


Figure 3.11: Linear regression. Five human subjects were asked to assume four hand shapes corresponding to four preassigned grip-aperture values equal to roughly 2, 4, 6, and 8 cm, respectively. Each hand shape was recorded by using a camera from a fixed viewpoint, thus obtaining a sequence of four images for each subject (Figure 3.10 shows one of the five hand-shape sequences obtained). Each image sequence is given in input to NeGOI, thereby obtaining a sequence of output values. The mean and the standard deviation were computed for each set of the five NeGOI outputs. Linear regression analysis between mean and actual grip-aperture values has been performed, obtaining an average determination index $r^2 = 0.98$. The y-axis represents the grip aperture as measured by NeGOI ($[0, 1]$). The x-axis represents the actual grip aperture.



Figure 3.12: Reach-to-grasp example. In this figure an example of reach-to-grasp action is shown.

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

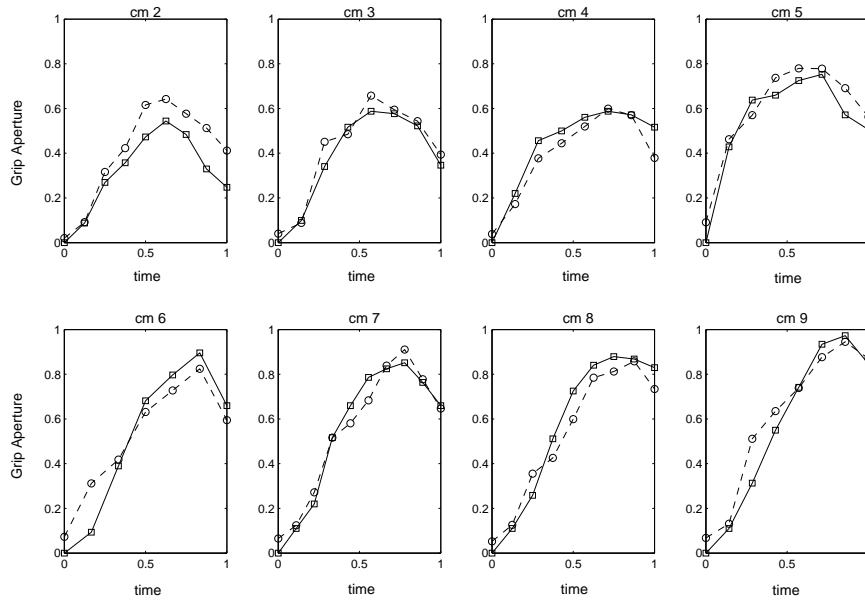


Figure 3.13: Grip aperture during a reach-to-grasp action. We have recorded 8 human grasp actions using a camera from a fixed viewpoint. The targets of these object-directed actions were 8 cubes of different sizes (cm 2, 3, . . . , 9 respectively). The graphic shows the grip aperture as measured by NeGOI compared with the actual values during eight reach-to-grasp actions performed on the eight objects. The dashed lines represent the grip aperture as measured by NeGOI, while the continuous lines represent the actual grip-aperture values. The grip aperture is normalized in the range $[0, 1]$. The x-axis represents the time normalized in $[0, 1]$. The time 0 corresponds at the beginning of the action, the time 1 corresponds with the moment when the hand touched the object.

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

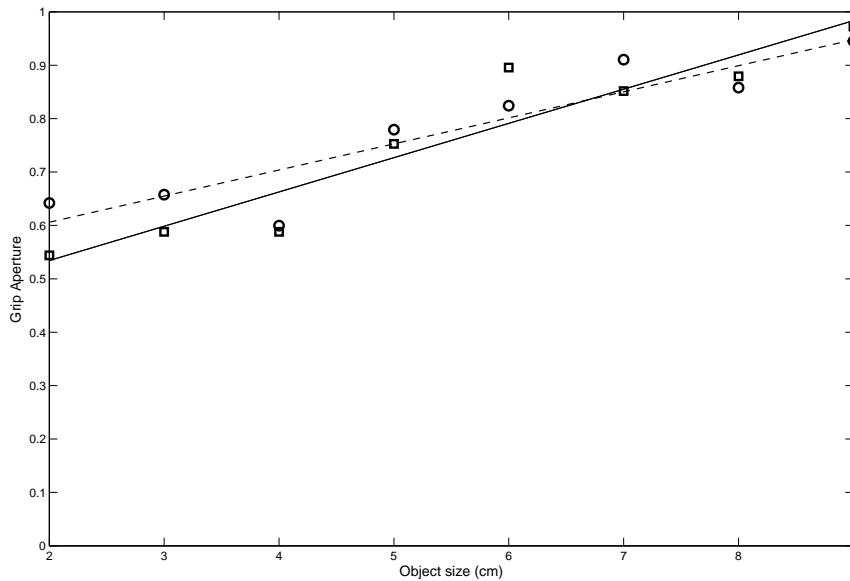


Figure 3.14: Maximum grip aperture. We have recorded 8 human grasp actions using a camera from a fixed viewpoint. The targets of these object-directed actions were 8 cubes of different sizes (cm 2, 3, . . . , 9 respectively). We obtained eight sequences of grip aperture values as computed by NeGOI. The graphic shows the maximum grip-aperture values of each sequence as computed by NeGOI (circles) compared with the actual maximum grip-aperture values (squares). The maximum grip-aperture values of each sequence shows a linear relationship with target size as reported in the literature. In fact, by performing a linear regression between maximum grip-aperture values and target sizes, one obtains an average determination index r^2 which is roughly equal to 0.85 (roughly 0.90 for actual values).

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE



Figure 3.15: Grip aperture for different viewpoints. Three hand configurations as close as possible to the three prototypical hand shapes (see Figure 3.8) were recorded while rotating the camera (see Figure A.1) around the Z -axis in accordance with a viewpoint range, $view_1$ - $view_3$, roughly equal to 45° (from viewpoint $view_1$ to viewpoint $view_3$ and back). For each hand configuration, a sequence of about 120 images was obtained. The figure shows a sample extracted from the sequence corresponding to the hand configuration fully opened grip aperture.

processed by NeGOI, resulting into a sequence of about 120 grip-aperture values for each hand configuration (see Figure 3.16 and Figure 3.17).

One verifies view independence by testing if each sequence of grip-aperture values is “almost stable”. This has been done by verifying the following two conditions:

$$max_i < min_{i+1} \text{ with } i = 1, 2$$

$$\sigma_i \ll \min_{h,k \in \{1,2,3\}} |\mu_h - \mu_k| = 0.49 \quad \forall i = 1, 2, 3$$

where max_i are the maximum values, min_i are the minimum values, σ_i are the standard deviations and μ_h are the mean values of each sequence of values computed by NeGOI (see Table 3.1). From Table 3.1 one can observe that the first condition is verified.

A subject was asked to assume both three hand configurations as close as possible to the three prototypical hand shapes (see Figure 3.8) and four specific hand configurations, different from the prototypical hand shapes, corresponding to grip-aperture values equal to roughly 2, 4, 6, 8 cm. For each hand configuration, the subject’s hand was recorded while rotating a camera in accordance with a viewpoint range, $view_1$ - $view_3$, roughly equal to 45° (from viewpoint $view_1$ to viewpoint $view_3$ and back). Thus, for each hand configuration, a sequence of about 120 images was obtained. Each image was processed by NeGOI, resulting into a sequence of about 120 grip-aperture values for each hand configuration. In the table we show mean, standard deviation, minimum and maximum value for each sequence.

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

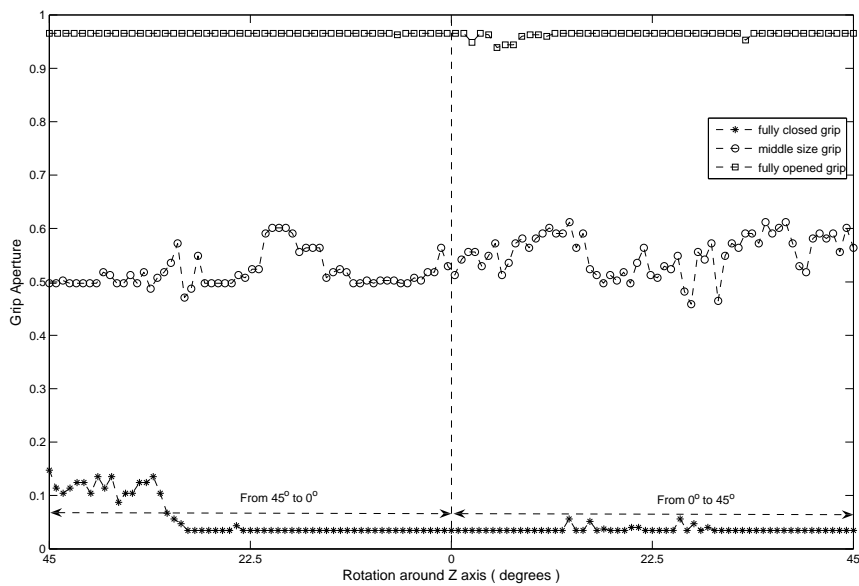


Figure 3.16: View independence for prototypical shapes. A subject was asked to assume three hand configurations as close as possible to the three prototypical hand shapes (see Figure 3.8). For each hand configuration, the subject's hand was recorded while rotating the camera (see Figure A.1) around the Z - *axis* in accordance with a viewpoint range roughly equal to 45° . For each hand configuration, a sequence of about 120 images was obtained. The graphic shows the three sequences of grip-aperture values as computed by NeGOI.

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

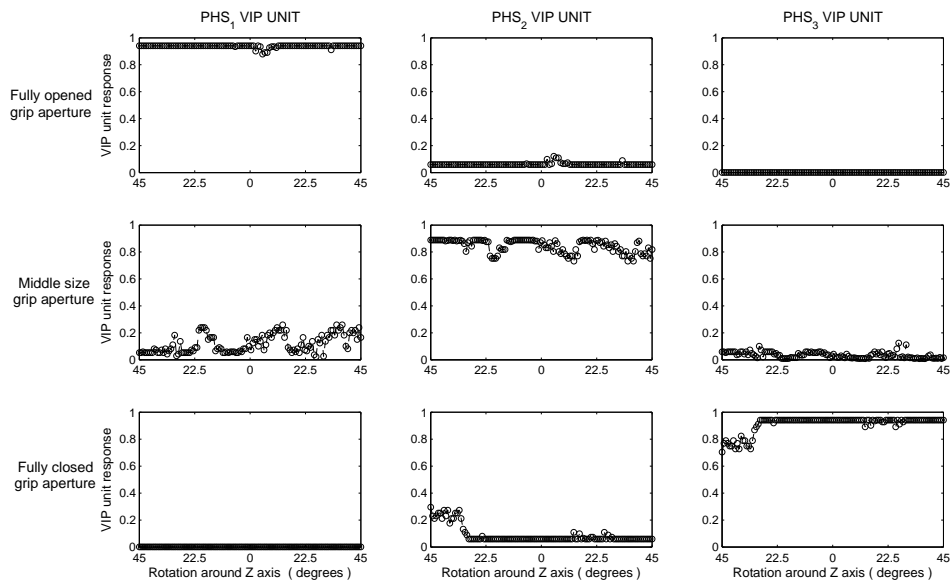


Figure 3.17: VIP unit response. Three specific hand shapes as close as possible to the three prototypical hand shapes (see Figure 3.8) were recorded while rotating the camera (see Figure A.1) in accordance with a viewpoint range roughly equal to 45° . For each hand configuration, a sequence of about 120 images was obtained. For each reach-to-grasp action, the graphic shows the responses of the three VIP units each one selective to one of the three prototypical hand shapes (PHS_1 , PHS_2 and PHS_3) and independent of the viewpoint. Notice that for each sequence (row) just one VIP unit assumes a high output value.

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

Actual grip aperture	Mean	Standard deviation	min	max
Fully opened grip (PHS_1)	0.96	0.01	0.94	0.96
Middle size grip (PHS_2)	0.54	0.04	0.46	0.61
Fully closed grip (PHS_3)	0.05	0.03	0.03	0.15
2 (cm)	0.28	0.05	0.20	0.41
4 (cm)	0.50	0.03	0.43	0.61
6 (cm)	0.72	0.04	0.64	0.81
8 (cm)	0.92	0.04	0.85	0.96

Table 3.1: A subject was asked to assume both three hand configurations as close as possible to the three prototypical hand shapes (see Figure 3.8) and four specific hand configurations, different from the prototypical hand shapes, corresponding to grip-aperture values equal to roughly 2, 4, 6, 8 cm. For each hand configuration, the subject's hand was recorded while rotating a camera in accordance with a viewpoint range, $view_1$ – $view_3$, roughly equal to 45° (from viewpoint $view_1$ to viewpoint $view_3$ and back). Thus, for each hand configuration, a sequence of about 120 images was obtained. Each image was processed by NeGOI, resulting into a sequence of about 120 grip-aperture values for each hand configuration. In the table we show mean, standard deviation, minimum and maximum value for each sequence.

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

To verify the second condition, for each sequence we performed a one-sided chi-square test at significance level $p < 0.01$, with the null hypothesis that the standard deviation is equal to 0.05 and the alternative hypothesis that the standard deviation is smaller than 0.05, obtaining on the basis of this test that one has to accept the alternative hypothesis.

The second experiment was devoted to test the NeGOI capacity to preserve the view invariance property for hand configurations different from the prototypical hand shapes. To this aim, we asked the subject to assume specific hand configurations different from the prototypical hand shapes, and corresponding to grip-aperture values equal to roughly 2, 4, 6, 8 cm. As in the previous experiment, for each hand configuration, the hand was recorded while rotating the camera in accordance with a viewpoint range, $view_1$ – $view_3$ (from viewpoint $view_1$ to viewpoint $view_3$ and back) obtaining again a sequence of about 120 grip-aperture values as computed by NeGOI. The capability of NeGOI to obtain a viewpoint independent measure is assessed by verifying whether each sequence of grip-aperture values is “almost stable”. The sequences of grip-aperture values shown in Figure 3.18 were obtained. Thus, the view-independence property is again verified if:

$$max_i < min_{i+1} \text{ with } i = 1, 2$$

$$\sigma_i \ll \min_{h,k \in \{1,2,3\}} |\mu_h - \mu_k| = 0.20 \quad \forall i = 1, 2, 3$$

where max_i are the maximum values, min_i are the minimum values, σ_i are the standard deviations and μ_h are the mean values of each sequence of values computed by NeGOI (see Table 3.1). From Table 3.1 one can observe that the first condition is verified.

To verify the second condition, for each sequence we performed a one-sided chi-square test at the significance level $p < 0.01$, with the null hypothesis that the standard deviation is equal to 0.06, and the alternative hypothesis that the standard deviation is smaller than 0.06, thereby suggesting that one has to accept the alternative hypothesis.

These tests confirm the hypothesis that NeGOI measures grip apertures in a view-independent fashion at least insofar as the range $view_1$ – $view_3$ is concerned.

3.4. A CASE STUDY: THE COMPUTATION OF GRIP-SIZE

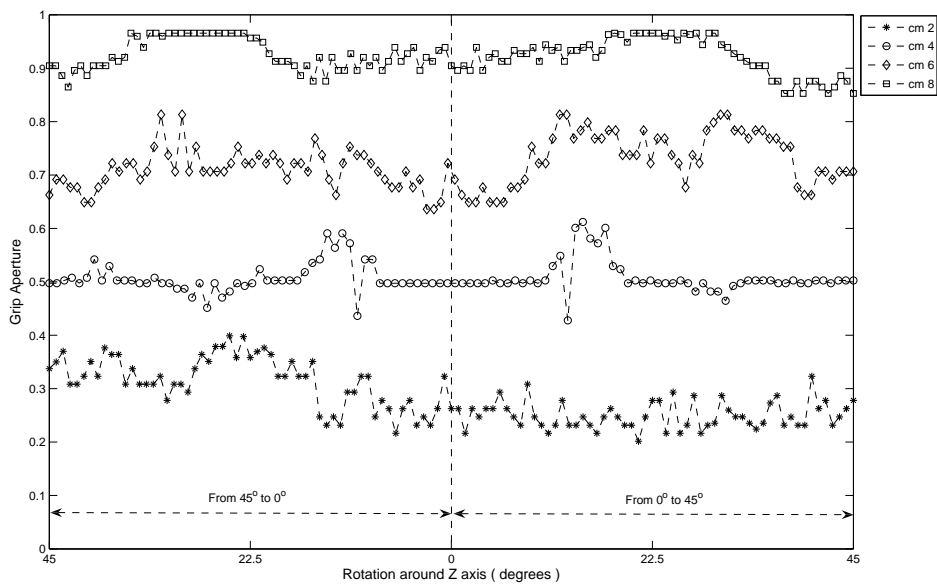


Figure 3.18: View independence. A subject was asked to assume four specific hand configurations corresponding to grip-aperture values roughly equal to cm 2, 4, 6, and 8. For each hand configuration, the subject's hand was recorded while rotating the camera around the Z - axis (see Figure A.1) in accordance with a viewpoint range, $view_1$ – $view_3$, roughly equal to 45° (from viewpoint $view_1$ to viewpoint $view_3$ and back). For each hand configuration, a sequence of about 120 images was obtained. The graphic shows the four sequences of grip-aperture values as computed by NeGOI.

3.5 Same-input hypothesis and model of view-independent grip-size computation

In order to develop a significant computational model of mirror neurons one must deal with same-activity and same-input hypotheses in the light of the considerations made in this chapter. In particular the same-input hypothesis raises a series of conceptual and computational problems. The latter kind of problems mainly concern how a view-invariant scene description (direct internal input) can be computed on the basis on visual input only. The NeGOI model provides evidence that at least one feature of the such description can be computed with some tolerance with respect to changes in point of view. But presumably other features, should be taken into account to provide the overall description. Accordings, two main questions arise:

- ▷ how many features should be included in scene description ?
- ▷ how tolerant may be a system which computes scene descriptions with respect to changes in points of view ?

The limitations imposed by the same-input hypothesis give rise to serious doubts about the feasibility that a similar description can be computed only by visual input. We will show that a comprehensive description (at least of the hand) can be computed if we use both visual input and motor information provided by mirror neurons.

A new computational approach for mirror neurons

According to the direct matching hypothesis (Rizzolatti et al., 2001), the motor system plays a central role in action recognition. Due to their behaviour mirror neurons provide empirical evidence in favour of this hypothesis. In fact it can be argued that an action \mathcal{A} is recognized, by an observer \mathcal{O} when its observation brings (a part of) the observer's neural motor system to become active in the same way as when \mathcal{A} is performed by \mathcal{O} . Thus mirror neurons properly represent the part of the motor system which activates during both action observation and execution. Even if this may be regarded as a convincing functional interpretation of experimental data, it is unsatisfactory from a computational point of view insofar as it does not specify how can be achieved this sort of "resonating effect" of the motor system.

In particular one can envisage different sorts of involvement for motor representations and processing in action recognition. According to one view, pursued in most computational models of mirror neurons, and spelled out in Section 3.2, perspectival sensory inputs are turned into motor information by means of a computational transformation unidirectionally flowing from perspectival sensory input to direct internal input, and from the latter on to motor coding. This conception leads to an impoverished functional roles for mirror mechanisms in action recognition processes as far as mirror activity is a straightforward consequence of the view-independent character of direct internal inputs. According to an alternative view, action recognition processes receive information from the brain motor system at earlier processing stages, and this information is deployed for the purpose of interpreting perspectival sensory inputs. Consistently with this latter view, the computational modelling approach pro-

4.1. USING MOTOR INFORMATION FOR INTERNAL INPUT COMPUTATION

posed here relies on some distinctive assumptions.

1. According to neurophysiological data (Rizzolatti et al., 1996; Umiltà et al., 2001), a necessary condition for mirror neuron activity to arise in observation modality is the evidence (usually perceptual evidence) that there is an object towards which the action is directed.
2. Given that the observer knows how to manipulate the object, then the observer knows the set of the more probable configurations a hand (more generally, an effector) can sequentially assume in actions that are directed towards that object. This assumption is akin to the affordance hypothesis in the psychology of perception (Gibson, 1979), according to which the observer selectively identifies properties enabling interactions with objects in the environment. In (Tessitore et al., 2009) a biologically plausible architecture is proposed for affordance extraction in the context of grasping actions.
3. Knowledge of hand (effector) configurations, which is codified in motor areas, can be used to form a priori hypotheses which constrain the computation of a mapping from perspectival sensory input to hand (effector) configuration coding.
4. Using the outcomes of this mapping jointly with perspectival sensory inputs, one can either corroborate or replace previously formed a priori hypotheses, and repeat, if necessary, the overall computational process.

Major computational problems arising in connection with this approach concern the identification of available motor information, and its specific uses for the purpose of analyzing and interpreting perspectival sensory input during action observation.

4.1 Using motor information for internal input computation

To begin with, it must be underlined that we are looking for motor information useful to both action control and recognition. In (Iberall and Fagg, 1996; Iberall et al., 1986), in the context of grasping control, the opportunity of using simple hand description models is explored. In particular, in this work the notion of virtual finger is introduced and explored in order to reduce the degrees of freedom, and thereby the complexity of the

4.1. USING MOTOR INFORMATION FOR INTERNAL INPUT COMPUTATION

hand control problem. A simplified control strategy during reach-to-grasp actions is also suggested, at the output stage, in the reduced number of hand shapes one can effectively assume and, hence, in the reduced number of features that are needed to achieve a meaningful hand description. Empirical evidence for the use of a reduced set of variables for representing hand shapes in the context of reach-to-grasp actions is provided by behavioural findings (see (Santello et al., 2002; Mason et al., 2001)). Notably, in Santello's work a principal component analysis (see Appendix C) is performed over a series of hand features that are monitored while a subject performs (or mimics) a reach-to-grasp action. The outcome of this analysis shows that the first *eigenposture* (principal components are called eigenpostures in this context) suffices to account for most hand feature variability, and the first two eigenpostures account for almost every aspect of whole hand feature variability. Apparently, coordinated movements of hand fingers result, during reach-to-grasp action, into a reduced number of physically possible hand shapes.

Thus it can be argued that motor information in the form of eigenpostures may be useful as far as the problem of hand grasping control is concerned.

In this work we will show that the same motor information (in the form of eigenpostures) may be useful to interpret perspectival sensory inputs.

More specifically the problem of analyzing and interpreting perspectival sensory input during actions will be explored within a probabilistic theoretical framework, and in connection with a simplified action recognition scenario. In the envisaged scenario perspectival sensory inputs are restricted to hand-related visual inputs collected from some fixed viewpoint. Accordingly, the simplified perceptual problem to solve is that of estimating actual hand configuration on the basis of both motor information and incoming visual inputs.

In this simplified setting, it is assumed that different classes of object-directed actions can be identified. Moreover, in accordance with experimental data analyzed in (Mason et al., 2001; Santello et al., 2002), it is assumed that a stereotyped/expected temporal sequence of hand configurations, described by a restricted number of parameters, can be associated to each of these classes. More precisely, each class C of object-directed actions is associated to a specific set of vectors in the space of hand-joints configurations. The eigenpostures associated to C span a low-dimensional sub-space of hand-joints configurations that a hand can assume during the execution of an object-directed action in C . Each set of eigenpostures must be selected so as to obtain a "sufficiently detailed" description of hand configurations during the execution of an action in C , in terms of

4.2. MECHANISMS FOR COMPUTING INTERNAL INPUTS USING MOTOR INFORMATION

an appropriate linear combination of eigenpostures. Thus, to each class of object-directed actions one associates a stereotyped temporal sequence of hand configurations, which is described in terms of the temporal evolution of coefficients in the selected linear combination of eigenpostures.

In this setting, an interpretive hypothesis about an observed action is given by some set of eigenpostures associated to a class of object-directed actions. This interpretive hypothesis is identified on the basis of motor knowledge, which is broadly construed here as comprising both eigenposture representation and selection processes. Accordingly, the incoming visual input is used to estimate the coefficients of the linear combination of the selected eigenpostures, rather than the computationally more demanding estimate of the whole set of hand joints parameters.

4.2 Mechanisms for computing internal inputs using motor information

Let us now proceed with the description of a formal framework which appears to accommodate the requirements outlined in the previous section towards a computational account of the interaction between motor and perceptual processes in action observation.

Let N be the number of distinct classes of object-directed actions, and let $S_k = \mathbf{e}_j^k$ be the set of eigenpostures (or principal subspace) associated to the k -th class, with $k = 1, 2, \dots, N$, $j = 1, 2, \dots, M_k$, and $\mathbf{e}_j^k \in \mathbb{R}^c$ where c is the number of degrees of freedom. Then a hand configuration \mathbf{t} , represented in terms of hand-joints parameters, during the execution of an object-oriented action belonging to the k -th class, is given by:

$$\mathbf{t} = \sum_j \beta_j \mathbf{e}_j^k \text{ with } \beta_j \in \mathbb{R} \quad (4.1)$$

Furthermore, it is assumed a selection mechanism of a subset of principal subspaces S_1, \dots, S_r on the basis of (usually perceptual) information concerning the object towards which the action is directed. To each principal subspace S_k one associates a probability $P(S_k)$, and a stereotyped/expected hand configuration in terms of coefficients β_j^k . Note that each principal subspace corresponds to a specific class of object-directed actions, and that the values of the associated probabilities are initially set to prior probabilities P_i , that is $P(S_i) = P_i$ with $i = 1, 2, \dots, r$. These P_i 's give an initial estimate of the probability of observing each class of object-directed actions. These probabilities will be updated on the basis of the in-

4.2. MECHANISMS FOR COMPUTING INTERNAL INPUTS USING MOTOR INFORMATION

coming visual input sequence $\mathbf{x}(t_1), \dots, \mathbf{x}(t_m)$. More specifically, for each principal subspace S_k and each time step t_h , one computes the probability that the expected/stereotyped hand-configuration corresponds to the actual hand-configuration on the basis of both incoming visual input $\mathbf{x}(t_h)$ and principal subspace S_k . These probabilities, let us call them π_h^k , are computed for the purpose of updating probabilities $P(S_1), \dots, P(S_r)$. In this way, throughout an action recognition process, each $P(S_k)$ value supplies a regularly updated estimate that the observed action corresponds to an object-directed action in the k -th class.

If one considers the incoming sensory input only, the problem of estimating probabilities π_h^k can be coped with by estimating at time t_h the probability distribution of $\mathbf{t}(t_h)$, given the incoming sensory input $\mathbf{x}(t_h)$, that is $p(\mathbf{t}(t_h)|\mathbf{x}(t_h))$. However, this task is, in general, computationally expensive because of the large number of components in \mathbf{t} . In contrast with this, the approach proposed here has the advantage of expressing \mathbf{t} in terms of a small number of coefficients, insofar as one assumes that \mathbf{t} lies in one of the principal subspaces S_k . Thus, given sensory input $\mathbf{x}(t_h)$, one estimates for each selected principal subspace S_k the conditional probability distribution $p_k(\boldsymbol{\beta}(t_h)|\mathbf{x}(t_h))$, and computes the π_h^k as $\pi_h^k = p_k(\boldsymbol{\beta}(t_h)|\mathbf{x}(t_h))$.

How can one estimate these conditional probability distributions? According to (Bishop, 1995), one may estimate the $p_k(\boldsymbol{\beta}(t_h)|\mathbf{x}(t_h))$ by means of mixture model:

$$p_k(\boldsymbol{\beta}(t_h)|\mathbf{x}(t_h)) = \sum_{i=1}^M \alpha_i(\mathbf{x}(t_h)) \phi_i(\boldsymbol{\beta}(t_h)|\mathbf{x}(t_h)) \quad (4.2)$$

where the $\phi_i(\boldsymbol{\beta}|\mathbf{x})$ are kernel functions, usually identified with Gaussian functions of the form $\phi_i(\boldsymbol{\beta}|\mathbf{x}) = \exp\left(-\frac{\|\boldsymbol{\beta}-\boldsymbol{\mu}_i(\mathbf{x})\|^2}{2\sigma_i(\mathbf{x})}\right)$. The parameters $\alpha_i(\mathbf{x})$ can be regarded as prior probabilities of $\boldsymbol{\beta}$ generated from the i -th component of the mixture $\phi_i(\boldsymbol{\beta}|\mathbf{x})$.

The coefficients of the mixture, $\alpha_i(\mathbf{x}(t_h))$, and the parameters of the kernel functions, $\phi_i(\boldsymbol{\beta}|\mathbf{x})$, ($\boldsymbol{\mu}_i(\mathbf{x}(t_h))$ and $\sigma_i(\mathbf{x}(t_h))$ for a Gaussian kernel), depend on sensory input $\mathbf{x}(t_h)$. A two-layer, feed-forward neural network can be used to model the relationship between visual inputs $\mathbf{x}(t_h)$ and corresponding mixture parameters. Accordingly, the problem of estimating the conditional probability distribution $p_k(\boldsymbol{\beta}(t_h)|\mathbf{x}(t_h))$ can be approached by combining a density model and a neural network structure (see Appendix B for more details).

Summarizing, the overall process in observation mode is expressible in terms of the processing steps showed in Algorithm 4.1.

4.2. MECHANISMS FOR COMPUTING INTERNAL INPUTS USING MOTOR INFORMATION

Algorithm 4.1 Action observation algorithm.

1. On the basis of (perceptual) information concerning the object towards which the action is directed, a set of principal subspaces S_1, \dots, S_r is selected. Each selected principal subspace S_k is associated to a probability value $P(S_k) = P_k$
 2. For $h \leftarrow 1$ to m DO
 - 2.1 Let $\mathbf{x}(t_h)$ be the current visual input;
 - 2.2 On the basis of the selected sets of eigenpostures, generate expected hand-configurations coefficients $\beta_1(t_h), \dots, \beta_r(t_h)$
 - 2.3 From the input $\mathbf{x}(t_h)$, compute probabilities $\pi_h^k = p_k(\beta(t_h)|\mathbf{x}(t_h))$ with $k = 1, \dots, r$;
 - 2.4 On the basis of the computed probabilities π_h^k , update the probability values associated to each selected principal subspace:

$$P(S_k) \leftarrow (P(S_k) * \pi_h^k) / \sum_{i=1, \dots, r} P(S_i) * \pi_h^i$$
 3. The final computed values $P(S_k)$ identify the probability that the observed action corresponds to an action belonging to the k -th class. Presumably, only one of these probabilities will reach a sufficiently high value.
-

Algorithm 4.2 Action execution algorithm.

1. On the basis of task selection and perceptual information concerning the object towards which the action is directed, a set of principal subspaces S_1, \dots, S_r is selected. An high prior probability is assigned to just one principal subspace $P(S_k) \gg P(S_j)$ with $j = 1, \dots, r$ and $j \neq k$
 2. For $h \leftarrow 1$ to m DO
 - 2.1 Let $\mathbf{x}(t_h)$ be the current visual input;
 - 2.2 On the basis of the selected sets of eigenpostures, generate expected hand-configurations coefficients $\beta_1(t_h), \dots, \beta_r(t_h)$
 - 2.3 From the input $\mathbf{x}(t_h)$, compute probabilities $\pi_h^k = p_k(\beta(t_h)|\mathbf{x}(t_h))$ with $k = 1, \dots, r$;
 - 2.4 On the basis of the computed probabilities π_h^k , update the probability values associated to each selected principal subspace:

$$P(S_k) \leftarrow (P(S_k) * \pi_h^k) / \sum_{i=1, \dots, r} P(S_i) * \pi_h^i$$
 3. If $P(S_k)$ falls below some fixed threshold, then action execution is interrupted.
-

4.2. MECHANISMS FOR COMPUTING INTERNAL INPUTS USING MOTOR INFORMATION

In action execution mode, the model works in a similar way as in the observation mode. However, in this case, on the basis of perceptual information about the object, together with the performing subject action intent, an high prior probability is assigned to just one principal subspaces $P(S_k) \gg P(S_j)$ with $j = 1, \dots, r$ and $j \neq k$. Subsequently, the probabilities $P(S_1), \dots, P(S_r)$ are updated in action execution mode on the basis of incoming visual input sequence $\mathbf{x}(t_1), \dots, \mathbf{x}(t_m)$. Moreover, one checks whether probability $P(S_k)$ falls below a certain threshold. If it does, this is an indication that something is not working properly in action execution. This overall process in action execution mode is expressible in terms of the processing steps showed in Algorithm 4.2.

The same process, in action execution and observation mode, is schematically illustrated by means of the functional diagram in Figure 4.1.

4.2. MECHANISMS FOR COMPUTING INTERNAL INPUTS USING MOTOR INFORMATION

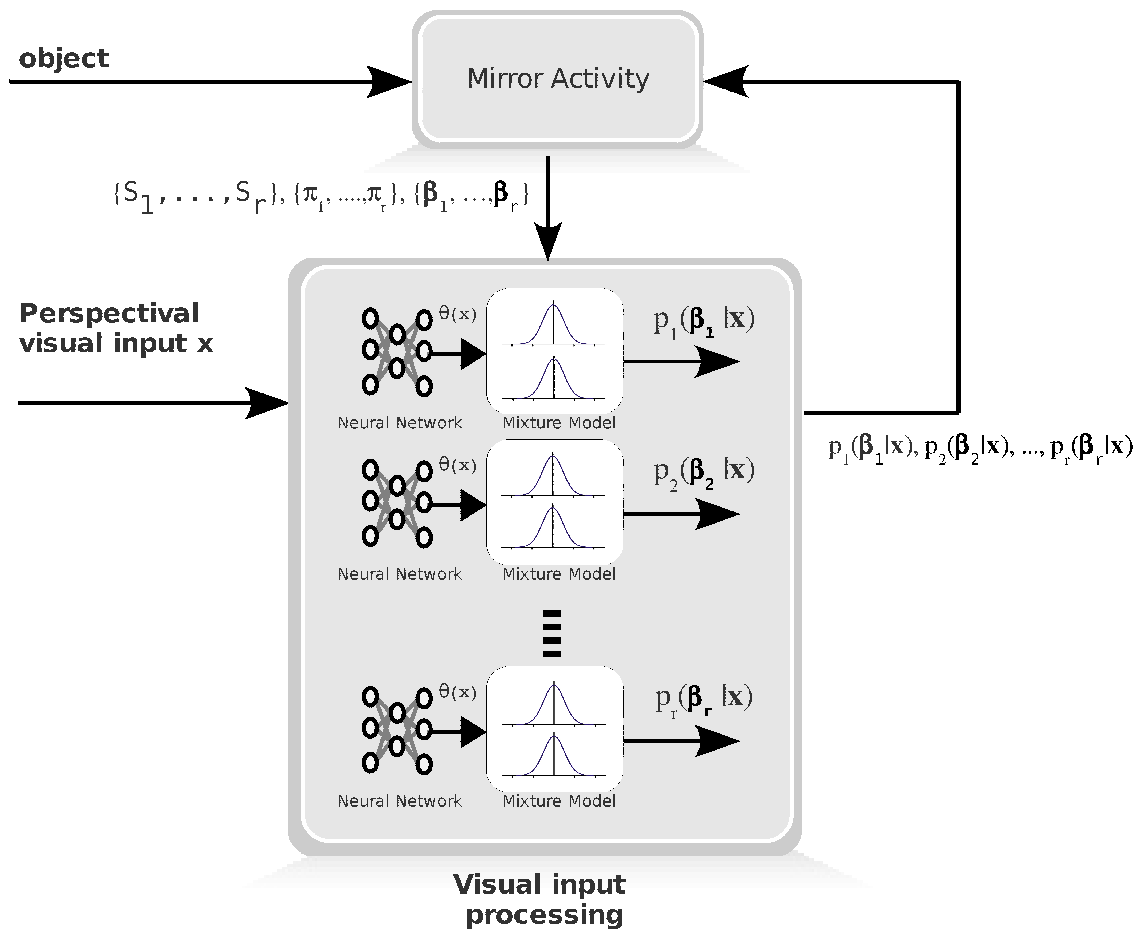


Figure 4.1: Functional diagram of interactions between motor and perceptual processes in action execution and observation. See text for details.

Model specification and modelling problems

In this chapter we will specify in more detail the computational model of mirror neurons introduced in previous chapter. We will extensively discuss the main modelling problems which were addressed during the construction of model. More specifically, a first problem concerns the mapping between visual input and hand configuration which is an ill-posed problem and cannot be faced by means of standard regression techniques. Therefore, this mapping is approached here from within a probabilistic framework. In this framework motor information, in terms of eigenpostures, is expected improve the computation of the mapping due to the reduction of variables needed to describe hand configurations. Moreover, the use of motor information presupposes the existence of both different sets of eigenpostures for different object-directed actions and a selection mechanism between them.

5.1 Mapping visual input to hand configuration: an ill-posed problem

As discussed in Chapter 3, a general formulation of the same internal input computation raises several modelling challenges. A major mathematical problem concerns the ill-posed character of many required transformations from perspectival sensory data to intrinsic features of object-directed actions. An example of such transformation is the determination of hand configuration from its visual appearance because the relation between visual input and hand configuration is not a functional mapping. Approaching such problem in a supervised learning fashion may lead to very

5.1. MAPPING VISUAL INPUT TO HAND CONFIGURATION: AN ILL-POSED PROBLEM

poor performance in the prediction of unseen hand configurations. This predicament affect standard regression techniques which try to minimize a sum-of-squares error function to training data pairs.

In Chapter 4 we have introduced a probabilistic framework and suggested the use of motor information as a way to overcome the ill-posed problem we are facing and model the relational (but no functional) mapping. In the next section we will discuss in more depth i) how the probabilistic framework may be used to model the non-functional mapping; ii) how motor information, in the form of eigenpostures, may significantly improve the computation of hand configurations.

As will be come clearer in the next section, the determination of hand configuration from its visual appearance, can be seen as the problem of constructing an inverse model. Thus, we start addressing the point i) by illustrating the use of the probabilistic framework in the simplified scenario of the determination of the inverse model of a robot arm.

5.1.1 Inverse kinematics of a robot arm

Consider the robot arm showed in Figure 5.1a, where we have indicated by $\mathbf{x} \equiv (x_1, x_2)$ the Cartesian coordinates of the end effector and by the vector $\mathbf{t} \equiv (\theta_1, \theta_2)$ the two joint angles of the arm. For every given values of θ_1 and θ_2 there exists a unique position of the end effector in the space, that is $g(\mathbf{t}) = \mathbf{x}$. This is known as the *forward kinematics* of the arm. However, given an arm position $\mathbf{x} = (\bar{x}_1, \bar{x}_2)$ we will have several joints configurations that give rise to the same arm position (see the example in Figure 5.1b). The problem of determining the joints angles given the end effector position is an *inverse kinematics* problem which, in general, cannot be modelled as a function.

Suppose we do not have an analytic formulation for both forward and inverse kinematics. We can collect a set of pairs $TS = \{\mathbf{x}^n, \mathbf{t}^n\}_{n=1, \dots, N}$ by giving to the system different values of \mathbf{t}^n and obtaining the corresponding \mathbf{x}^n . Now we can approach the determination of the forward and inverse kinematics in the framework of supervised learning.

Supervised learning can be described as follows (see Hastie et al. (2003) for a more comprehensive treatment of this topic): suppose we have a set of couple called *training set* $TS = \{\mathbf{x}^n, \mathbf{t}^n\}_{n=1, \dots, N}$ extracted from a deterministic but unknown function $f(\mathbf{x}^n) = \mathbf{t}^n + \epsilon$ where ϵ is an error which affects data and can be due, for example, to an error in the measure procedure. The goal of supervised learning is to construct an \hat{f} that for new $\mathbf{x} \notin TS$ is able to predict the right \mathbf{t} (*generalization property*). The problem

5.1. MAPPING VISUAL INPUT TO HAND CONFIGURATION: AN ILL-POSED PROBLEM

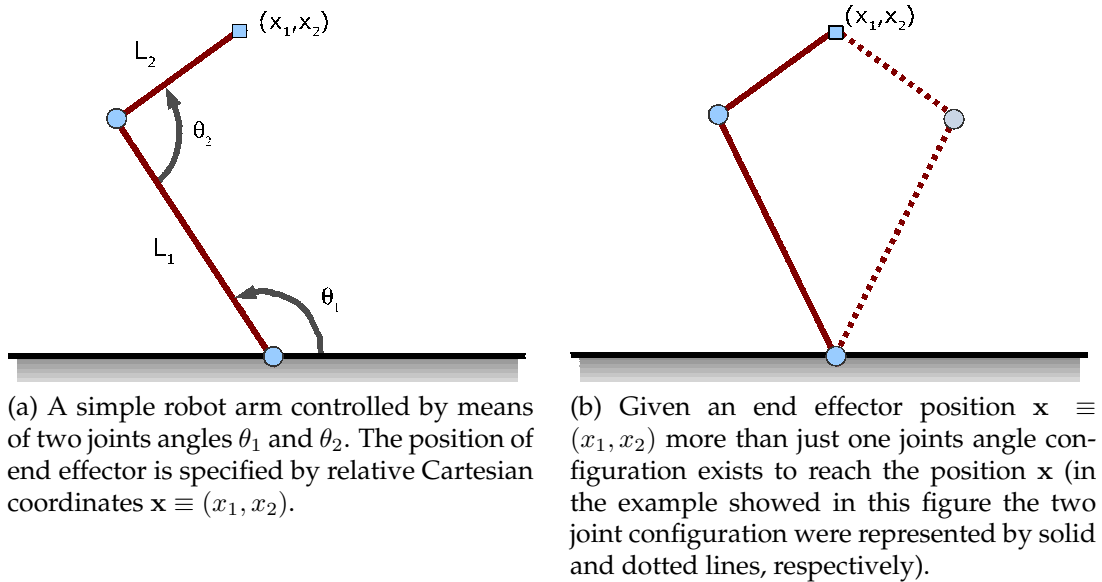


Figure 5.1: A simple robot arm example.

of inferring \hat{f} from TS can be formalized in the framework of statistical decision theory.

Let $\mathbf{x} \in \mathbb{R}^d$ denote a real valued random input vector and $\mathbf{t} \in \mathbb{R}^c$ a real valued random output variable with joint distribution $p(\mathbf{x}, \mathbf{t})$. We seek a function $f(\mathbf{x})$ for predicting \mathbf{t} given the value of the input \mathbf{x} . We now need a way to choose between different functions f . To do so, we introduce a *loss function* $L(\mathbf{t}, f(\mathbf{x}))$ which tells us how good is f in predicting \mathbf{t} given \mathbf{x} . The more common and convenient choice for L is *squared error loss*:

$$L(\mathbf{t}, f(\mathbf{x})) = (\mathbf{t} - f(\mathbf{x}))^2 \quad (5.1)$$

It can be shown that the function f which minimizes 5.1 is given by:

$$f(\mathbf{x}) = \langle \mathbf{t} | \mathbf{x} \rangle \quad (5.2)$$

where $\langle \mathbf{t} | \mathbf{x} \rangle$ denotes the conditional averages of the target data given \mathbf{x} that is:

$$\langle \mathbf{t} | \mathbf{x} \rangle = \int \mathbf{t} p(\mathbf{t} | \mathbf{x}) dt$$

the function f which satisfies 5.2 is called *regression function* (see Figure 5.2).

The result in 5.2 has important consequences if we want to apply this setting in the solution of both forward and inverse problems. In fact if

5.1. MAPPING VISUAL INPUT TO HAND CONFIGURATION: AN ILL-POSED PROBLEM

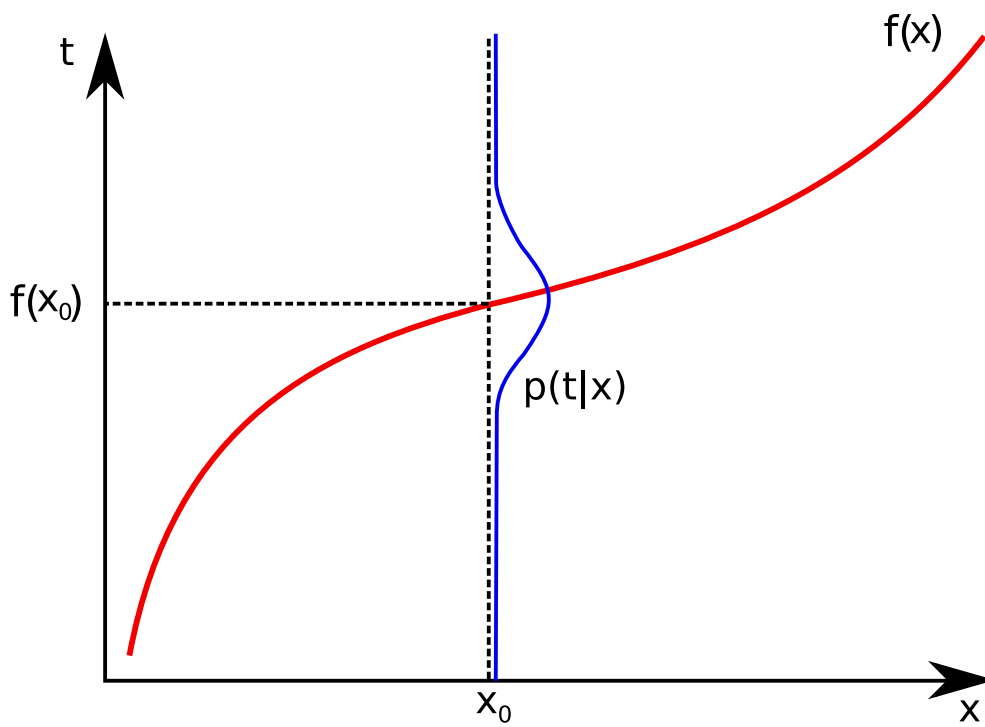


Figure 5.2: Regression function.

5.1. MAPPING VISUAL INPUT TO HAND CONFIGURATION: AN ILL-POSED PROBLEM

the relation between \mathbf{x} and \mathbf{t} is not functional, as is in general in inverse problems, then the resulting mapping is quite poor because in general the average of several solutions is not itself a solution.

To illustrate with a concrete example suppose that robot arm position and arm control are scalar values and that the unknown forward relation is given by:

$$x = t + 0.3\sin(2\pi t) + \epsilon \quad (5.3)$$

with $t \in [0, 1]$ and ϵ a random variable drawn from a uniform distribution in the range $[0, 1]$. A data set obtained by the previous relation is shown in Figure 5.3 together with the estimated (blue solid line) $\langle p(t|x) \rangle$. In this case the conditional average of the target data $\langle p(t|x) \rangle$ gives a good representation of the function from which the data was generated, and thus can be successfully used to estimate forward relation.

Consider now the inverse relation between x and t obtained by interchanging the roles of input and output of data as showed in Figure 5.4. In this case the relation between x and t is not a function and the conditional average of the target data $\langle p(t|x) \rangle$ gives a very poor representation of the data (see Figure 5.4 blue solid line) since the average of several correct target values (that is arm configurations) is not necessarily itself a correct value (that is an arm configuration).

This can be easily viewed in Figure 5.1b where at the same \mathbf{x} can be associated to two correct solutions $\mathbf{t}^1 \equiv (\theta_1^1, \theta_2^1)$ (showed by solid lines) and $\mathbf{t}^2 \equiv (\theta_1^2, \theta_2^2)$ (showed by dotted lines). However the mean solution between \mathbf{t}^1 and \mathbf{t}^2 given by $\mathbf{t}^3 = \left(\frac{\theta_1^1 + \theta_1^2}{2}, \frac{\theta_2^1 + \theta_2^2}{2} \right)$ is not itself a solution because the arm does not reach the position \mathbf{x} .

To sum up. The estimation of conditional average $\langle p(\mathbf{t}|\mathbf{x}) \rangle$ does not suffice to model inverse kinematics if the inverse relation between \mathbf{x} and \mathbf{t} is not functional.

An alternative approach is to estimate the whole distribution $p(\mathbf{t}|\mathbf{x})$ instead of the central value $\langle p(\mathbf{t}|\mathbf{x}) \rangle$ (Bishop, 1995). A powerful approach to estimate $p(\mathbf{t}|\mathbf{x})$ makes use of mixture models of the form:

$$p(\mathbf{t}|\mathbf{x}) = \sum_{i=1}^M \alpha_i(\mathbf{x}) \phi_i(\mathbf{t}|\mathbf{x}) \quad (5.4)$$

where M is the number of components in the mixture, $\alpha_i(\mathbf{x})$ are the mixing coefficients and $\phi_i(\mathbf{t}|\mathbf{x})$ are the mixture components functions usually chosen Gaussians of the form:

5.2. A PROBABILISTIC FRAMEWORK FOR HAND CONFIGURATION ESTIMATION

$$\phi_i(\mathbf{t}|\mathbf{x}) = \frac{1}{(2\pi)^{c/2}\sigma_i^c(\mathbf{x})} \exp\left\{-\frac{\|\mathbf{t} - \boldsymbol{\mu}_i(\mathbf{x})\|^2}{2\sigma_i^2(\mathbf{x})}\right\} \quad (5.5)$$

The mixture density network (Bishop, 1994, 1995), described in depth in Appendix B, can be used to estimate model's parameters. Here we want to give a broad idea of how the determination of the distribution $p(\mathbf{t}|\mathbf{x})$ can be used to give a more powerful description of the inverse relation between \mathbf{x} and \mathbf{t} .

Consider again the data showed in Figure 5.4 and supposed we have estimated the parameters of the distribution $p(t|x)$ expressed in the form 5.4. If the components ϕ_i of the mixture are well separated and have negligible overlap we can easily find the most probable branch associated to every given x by:

$$\arg \max_i \{\alpha_i(x)\}$$

Knowing the most probable branch, the most probable value t of the distribution is given by the center μ_i of the related component ϕ_i .

In Figure 5.5 it is shown the plot of the central value of the most probable branch as a function of x . The resulting map gives a good representation of the data with respect to the mapping obtained in Figure 5.4. In particular, note that in the region where the mapping is multi-valued this approach gives in output one of the possible solutions instead of the mean of all solutions.

5.2 A probabilistic framework for hand configuration estimation

In the previous section we have illustrated how a probabilistic framework can be used to model a relational but not functional mapping. In particular we have given an idea of how the estimation of $p(\mathbf{t}|\mathbf{x})$ using a mixture model may be used to give a more powerful representation of the inverse relation between \mathbf{x} and \mathbf{t} .

The problem of hand configuration estimation from its visual appearance can be approached as the construction of an inverse model, where the vector \mathbf{x} contains hand visual features while the vector \mathbf{t} contains hand joints configuration.

We can repeat the same probabilistic approach used for the inverse kinematic of the arm trying to estimate the distribution $p(\mathbf{t}|\mathbf{x})$. However, the problem here is that the vector \mathbf{t} has many components (in general

5.2. A PROBABILISTIC FRAMEWORK FOR HAND CONFIGURATION ESTIMATION

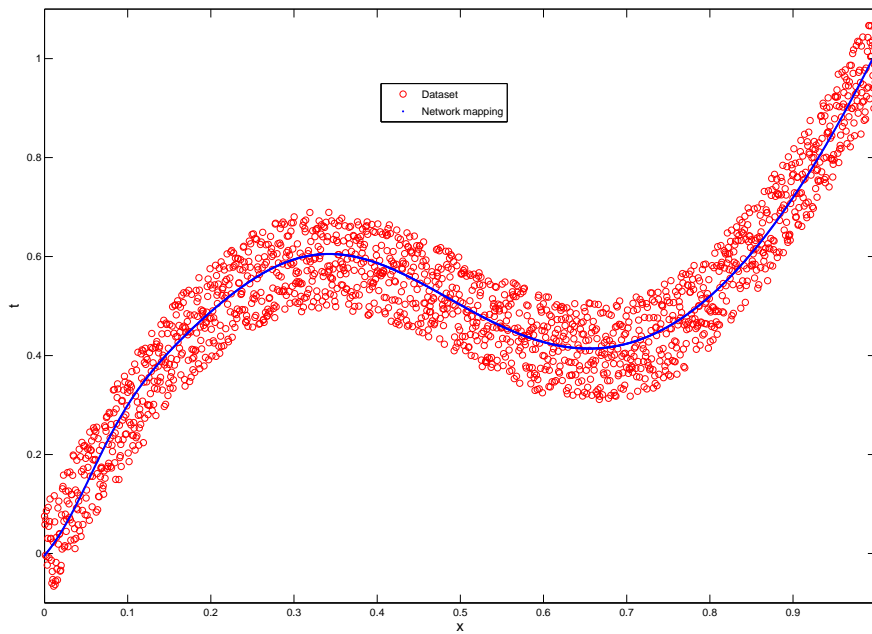


Figure 5.3: A dataset extracted by the forward relation expressed by equation 5.3. Note that in the case in which the underlying relation is functional the conditional mean give a good representation of the data.

5.2. A PROBABILISTIC FRAMEWORK FOR HAND CONFIGURATION ESTIMATION

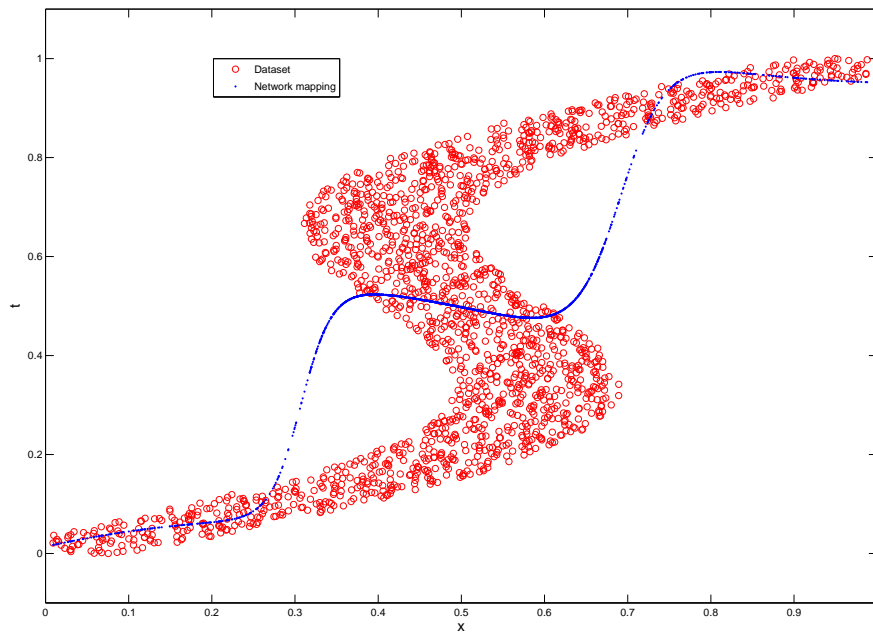


Figure 5.4: A dataset extracted by the forward relation by interchanging the role between x and t . Note that in this case the conditional mean give a poor representation of the data where the mapping is not functional.

5.2. A PROBABILISTIC FRAMEWORK FOR HAND CONFIGURATION ESTIMATION

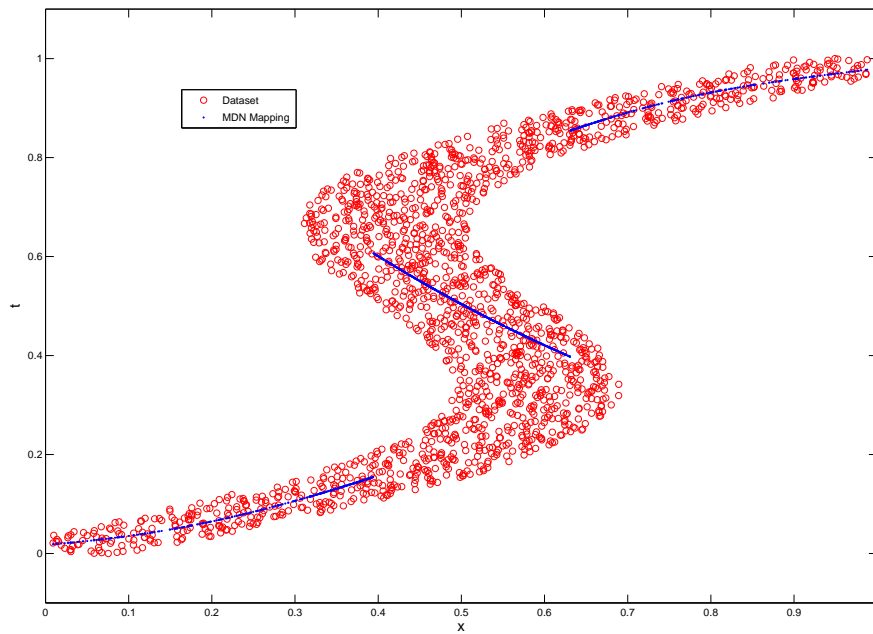


Figure 5.5: Progress of central value of the most probable branch as function of x . The resulting mapping is discontinuous giving however a good description of the data.

5.2. A PROBABILISTIC FRAMEWORK FOR HAND CONFIGURATION ESTIMATION

more than 20) and the determination of such distribution is a quite difficult task.

However it must be noted that we want to approximate $p(\mathbf{t}|\mathbf{x})$ in the particular case in which \mathbf{x} and \mathbf{t} are related to the execution or observation of object-directed actions.

It has been shown in the work of Santello et al. (2002) and successively in Mason et al. (2001) that during grasping actions the hand assumes a stereotyped sequence of hand configurations that can be described by a restricted number of parameters. More precisely, both works show that the hand configuration \mathbf{t} during grasping actions can be described as a linear combination of a small set of vectors (called eigenpostures) in the space of hand-joints configurations, that is:

$$\mathbf{t} = \sum_j^K \beta_j \mathbf{e}_j \text{ with } \beta_j \in R \quad (5.6)$$

where $\mathbf{e}_j \in \mathbb{R}^c$ are the eigenpostures' vectors and are computed by applying Principal Component Analysis (see Appendix C) over a dataset of hand joints configuration recorded by means of a dataglove during the execution of grasping actions.

As already discussed in Chapter 4, the idea is to express each vector \mathbf{t} in terms of a linear combination of eigenpostures and then to estimate the distribution $p(\beta|\mathbf{x})$ instead of the distribution of $p(\mathbf{t}|\mathbf{x})$.

However, two main questions arise in this case:

1. since mirror neurons' activity is supposed to be related to different sets of eigenpostures and since some mirror neurons show specificity for the modality of action execution (e.g. grasping with precision grip wrt grasping with whole hand) are there different sets of eigenpostures for different modalities of action execution?
2. the decomposition of \mathbf{t} in terms of a linear combination of eigenpostures holds if one knows in advance which action (or which actions) can be made. So a selection mechanism is needed to select the initial sets of eigenpostures on the basis of the object or contextual information.

In the next subsection we address 1 by describing how to make sure that different sets of eigenpostures exist for different modality of object-directed actions (e.g. whole hand prehension wrt precision grip prehension). Point 2 is instead faced in Section 5.3 by introducing the GA model. This model has been inspired by the sensory-motor computation occurring in the circuit AIP-F5 introduced in Chapter 2.

5.2. A PROBABILISTIC FRAMEWORK FOR HAND CONFIGURATION ESTIMATION

5.2.1 Different sets of eigenpostures for different modality of object-directed actions

In this section we will discuss how we can investigate if there are different sets of eigenpostures for different modality of object-directed actions. In Section 6.1, the same methodological approach will be applied to grasping actions to show the existence of different sets of eigenpostures for different types of grasping actions.

Consider a specific object-directed action and modality of action execution (i.e. grasping with precision grip). We can construct the relative set of eigenpostures by applying Principal Component Analysis (PCA) on a sufficient large dataset of hand joints configurations recorded during the execution of such type of action. We will usually take only a restricted number of eigenpostures (eigenvectors obtained from the PCA suffice to describe original data with enough accuracy) to form the set of eigenpostures.

This process can be repeated on data obtained during the execution of the same action made by a different subject or made by the same subject but directed toward different objects, thus leading to the creation of many sets of eigenpostures for the same type of modality of action execution. Moreover, the same process will be repeated for another modality of action execution, every time leading to the creation of different sets of eigenpostures.

We now need a way to measure the similarity between different sets of eigenpostures. In (Krzanowski, 1979) a similarity measure is proposed between principal subspaces¹ which is described in detail in appendix C.

The proposed similarity measure can be computed as follows.

If T^1 and T^2 are two sets of data (i.e. related to two modalities of grasping action) and L and M are the sets of corresponding selected k eigenpostures disposed column-wise, then the similarity measure between L and M is given by:

$$sim(L, M, k) = trace(L^T M M^T L)$$

For principal subspaces, that is subspaces spanned by the selected set of eigenvectors, of dimension k this similarity measure gives values belonging to $[0, k]$ where we have k for identical principal subspaces and 0 for orthogonal principal subspaces.

Suppose now that we have collected N different sets of eigenpostures. We can construct a similarity matrix of dimension $N \times N$ where each el-

¹A principal subspace is the space spanned by a set of eigenvectors

5.3. EIGEN-POSTURES SELECTION MECHANISMS: THE CONCEPT OF AFFORDANCES

ement (i, j) is the similarity measure between the i -th and the j -th set of eigenpostures.

The initial aim of showing that different sets of eigenpostures exists for different modalities of action execution can be approached as the problem of showing that the set of eigenpostures related to the same kind of action modality group together. This can be obtained by applying a clustering algorithm.

5.3 Eigen-postures selection mechanisms: the concept of affordances

A set of eigen-postures must be selected on the basis of perceptual information concerning the object toward which the action is directed and contextual information. In this section we will address this problem in the simplified case in which perceptual information is restricted to the visual information about the object toward which the action is directed.

In Tessitore et al. (2009) we have proposed a model for the extraction of hand-configurations useful to grasp a given object whose 2D visual representation is fed in input to the model. The model architecture is constructed on the basis of the concept of affordance, in particular on its interpretation for the special case of grasping affordance, and has some analogies with the computation of some neurons of the ventral visual stream up to parietal area AIP and F5 whose functional properties have been discussed in Section 2.1.2.

The model can be readily adapted to associate sets of eigenpostures instead of hand-configurations but we will discuss this point at the end of this section. Now we will first give a description of the model together with the underlying assumptions and interpretations in particular as the concept of affordance is concerned.

5.3.1 Grasping Affordance (GA) model

The notion of affordance was originally introduced by J. J. Gibson Gibson (1979) to single out perceived properties that enable one to interact with objects in the environment. Procedurally, the notion of affordance is framed in the context of *direct* perception theories, insofar as “higher-level” cognitive processes, such as access to semantic memory, logical inference, and object recognition processes are allegedly unnecessary to identify an affordance.

5.3. EIGEN-POSTURES SELECTION MECHANISMS: THE CONCEPT OF AFFORDANCES

A more precise understanding of the processes involved in identifying an affordance is crucial for the modelling of specific sensory-motor control mechanisms in biological systems. The existence of a particularly versatile sensory-motor control mechanism is witnessed by the wide range of sensory-motor associations that monkeys are able to perform. Notably, this behavioural ability persists upon presentation of many unknown/novel objects, thereby suggesting that a robust generalization process, based on perceived object properties, is at work there (Borghi, 2005).

In the context of grasping actions, neurophysiological data on the macaque's brain cortex are consistent with direct perception views of affordances. In particular, these data suggest that the anterior intraparietal area (AIP) is involved in the coding of object affordances (Rizzolatti and Sinigaglia, 2008), in the light of functional hypotheses concerning more extended brain circuits. The functional models of brain areas which have been found to deliver afferent signals to AIP include neither perceptual object recognition nor higher-level cognitive processes, such as planning and decision-making (Milner, 1998). Moreover, strong efferent pathways have been identified which connect AIP to pre-motor area F5 (Rizzolatti and Sinigaglia, 2008). Since F5 is prominently involved in the coding of object-oriented actions (such as grasping, holding, and manipulating), the AIP to F5 connections suggest the existence of some sort of *direct* functional link between perceptual feature detection and object-directed actions.

The computational model Grasping Affordances (GA) model (Tessitore et al., 2009), provides a precise explication of the notion of affordance in the context of grasping actions carried out by monkeys. This explication is consistent with both direct perception theories and neuroscientific models of the macaque's brain. It is consistent with direct perception theories, insofar as the identification of grasping affordances requires, according to the proposed computational model, neither object recognition processes nor access to semantic memory. It is consistent with neuroscientific models of the macaque's brain, insofar as (i) visual processes furnishing AIP inputs are modelled in accordance with the biological "Standard Model" proposed in (Riesenhuber and Poggio, 2000), and (ii) the overall system output does not conflict with neuroscientific data and modelling constraints insofar as inputs supplied by AIP to brain motor areas are concerned.

5.3.1.1 Affordances for Grasping

Affordances are not intrinsic properties of an object, but rather depend on the relationship between object and agent (Chemero, 2003). For example, differences in primate and feline effectors account to a large extent for the

5.3. EIGEN-POSTURES SELECTION MECHANISMS: THE CONCEPT OF AFFORDANCES

different affordances that objects convey to humans and cats, respectively. As one moves to consider more specifically grasping affordances for monkeys and humans, one should still be careful to note that graspable objects do not merely 'afford' our grasping them. Indeed, multiple opportunities for grasping arise in connection with many graspable objects. For example, a mug can be grasped by handle, lateral side, and top. These grasps can be distinguished from each other in terms of hand shape and wrist rotation obtaining just before grasping the object (Tucker and Ellis, 2000). Accordingly, the grasping affordances associated to a graspable object will be identified in the GA model with a collection of (codes for) appropriate hand configurations assumed by a hand just prior to grasping the object (Oztop et al., 2006a; Tsiotas et al., 2005). Since a graspable object may be grasped in several ways, this means that multiple hand configurations can be associated to any given object in the GA model.

5.3.1.2 General GA Model Description

From the above discussion, three main requirements have emerged for a computational model of grasping affordances to be empirically adequate and to move beyond previous computational models which include affordance extraction functionalities: (a) the model must provide computational solutions for significant processing steps along the path from V1 to AIP; (b) the model must enable one to extract multiple hand-configurations from the same graspable object; (c) the model must possess generalization capabilities with respect to novel/unknown objects.

To accomplish (a), the visual pathway was modelled starting from primary visual cortex V1 and reaching, through areas V2 and V4, into the posterior infero-temporal area (PIT), which is identified as the cortical region supplying visual monocular information to AIP (?). A biologically plausible model of the ventral visual stream, named *Standard Model*, was proposed in (Riesenhuber and Poggio, 2000). A component of the Standard Model, the view-based Module, accounts for computations along the path from V1 to PIT which makes inputs available to AIP. Accordingly, the Monocular Perception (MP) Module (see Figure 5.6) which is an implementation of the view-based module was developed and included in the GA model.

To accomplish (b), that is, to provide a computational solution to the multiple affordance extraction problem, we must compute a multi-valued function which relates any visual input to a collection of hand-configurations. This resemble the problem described in Section 5.1 where at the same hand visual description more one hand joints configuration can be associated.

5.3. EIGEN-POSTURES SELECTION MECHANISMS: THE CONCEPT OF AFFORDANCES

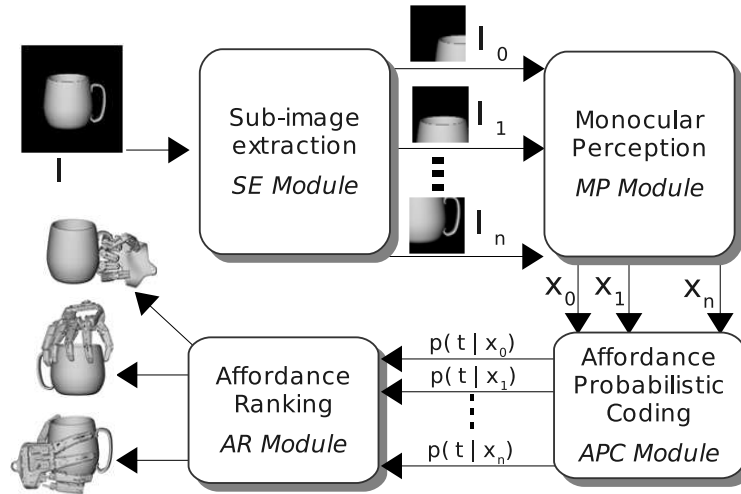


Figure 5.6: The GA model is formed by four modules: the SE Module, the MP Module, The APC Module, and the AR Module. This computational model receives an image depicting an object as input, and produces a list of affordances (appropriate grasps for the given object) as output.

Thus the same probabilistic approach was pursued.

More precisely, let $X \subseteq \mathcal{R}^d$ be the d -dimensional space of visual inputs, and let $T \subseteq \mathcal{R}^c$ be the c -dimensional space of hand configurations. Then the *Affordance Probabilistic Coding* (APC) (see Figure 5.7) will compute the distribution $p(t|x)$ with the same mixture model expressed in Equation 5.4.

To accomplish (c), that is, generalization capabilities enabling one to extract affordances from novel objects, a starting point was provided by the observation that the agent usually focuses its attention on the part of the object at which the grasping action is directed (Schiegg et al., 2003). This behaviour suggests the possibility of associating parts of a graspable object to affordances, and to store this “mereological” information for use when novel graspable objects are presented. For example, one may learn to associate appropriate affordances to handles and cylinders, respectively, and to use this information when a cup (resulting from the “composition” of handle and cylinder) is presented. This process was actually implemented by sliding an “attention window” on the image of an object, and by extracting a collection of grasping affordances at each displacement step. This function is achieved by the Subimage Extraction (SE) Module (see Figure 5.6). Finally, a post-processing step was required as well, in order to select the more plausible affordances. The post-processing step is accomplished

5.3. EIGEN-POSTURES SELECTION MECHANISMS: THE CONCEPT OF AFFORDANCES

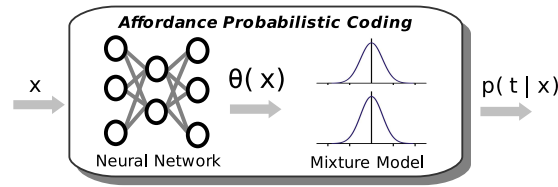


Figure 5.7: The APC Module is formed by a neural network and a Gaussian mixture model. Given an x vector, the neural network computes the required Gaussian parameters $\theta(x)$ to approximate $p(t|x)$ (see (Bishop, 1995) for more details).

by Affordance Ranking (AF) Module (see Figure 5.6). APC and AR modules account for the AIP affordance computation. The online learning of sensorimotor associations might be grounded onto a basic grasping ability such as described in (Oztop et al., 2004). Learning of sensorimotor associations may occur by collecting pairs of visually presented "object part" and related "hand-configuration" every time a successful grasp is made. Since the focus of this work is not on the acquisition of sensorimotor associations, however, we suppose here that a series of such pairs is already available.

5.3.1.3 GA Model specification and implementation

The GA model takes the image of an object as input and supplies the object's grasping affordances as output. It is composed by four modules, as shown in Figure 5.6. The input image I , represented in gray scale, is processed by the SE Module, which extracts n subimages I_j , $j = 1, \dots, n$. The number of subimages depends on the dimensions of the window W sliding on the image I , the image size, and the window displacement step DS .

Each subimage is then sent as input to the MP Module. The MP Module takes a sub-image I_j as input, and gives a 256 feature vector as output x^j . The latter is presented as input to the APC Module, which computes the corresponding $p(t|x^j)$.

To estimate $p(t|x^j)$, one uses a mixture model of the form expressed in eq. 5.4, whose parameters $\alpha_k(x)$, $\mu_k(x)$ and $\sigma_k(x)$ (for Gaussian kernel as in eq. 5.5) depend on the visual input x . The relationship between visual inputs x and corresponding mixture parameters is modelled by means of a two-layer, feed-forward neural network with H hidden nodes. Therefore, the ACP Module has a combined density model and neural network structure, as shown in Figure 5.7.

5.3. EIGEN-POSTURES SELECTION MECHANISMS: THE CONCEPT OF AFFORDANCES

Since the APC Module receives n feature vectors \mathbf{x}^j in input, its overall output is formed by n density functions $p(\mathbf{t}|\mathbf{x}^j)$. Note, however, that the desired output is a set $T = \{\mathbf{t}^1, \mathbf{t}^2, \dots, \mathbf{t}^L\}$ corresponding to the L distinct hand-configurations that enable one to grasp the viewed object. Therefore, a non-trivial output selection problem remains to be solved at this stage: one has to isolate hand-configurations which differ from each other as much as possible, and whose probability value is sufficiently high.

This requirement corresponds, for each single feature vector \mathbf{x} and related $p(\mathbf{t}|\mathbf{x})$, to choose as member of the set T the gaussians' centers $\mu_k(\mathbf{x})$ of the mixture associated to the higher values of $\alpha_k(\mathbf{x})$. In the case of n probability distributions $p(\mathbf{t}|\mathbf{x}^1), \dots, p(\mathbf{t}|\mathbf{x}^n)$, in order to obtain a behaviour similar to the single distribution case, one may proceed as follow:

1. generate s sample points from each distribution, obtaining $n \times s$ points, each of which defines a hand configuration. Not every hand configuration thus obtained corresponds to grasps for the input object; only those gathering around the kernel's means do, while the other points are distributed in a sparse manner;
2. a clustering over the $n \times s$ points is performed;
3. the clusters are ranked according to the order of their variance values, and the first L clusters with lower variances are selected because a lower variance implies less uncertainty about the hand configurations;
4. finally, the set T will be formed by the centers of the selected clusters.

5.3.2 How GA model can be used to select the initial sets of eigenpostures

As discussed in previous section, given a new object \mathbf{x} , the GA model can be used to predict appropriate hand configurations $\mathbf{t}^1, \dots, \mathbf{t}^L$ that can be used to interact with that object. The GA model can be also used to select the initial sets of eigenpostures. Let N be the number of distinct classes of object-directed actions, and let $S_k = \mathbf{e}_j^k$ be the set of eigenpostures associated to the k -th class. We will suppose that all the eigenpostures \mathbf{e}_j^k are stored in some brain area and our problem is just to select them in an appropriate manner.

To this aim we will further suppose that an unique index can be assigned to each eigenpostures. So, in order to select the set of eigenposture $\mathbf{e}_1^k, \dots, \mathbf{e}_c^k$ with associated indexes ind_1, \dots, ind_c , the vector $\mathbf{t} \equiv (t_1, \dots, t_c)$

5.4. GENERATION OF EXPECTED HAND-CONFIGURATIONS COEFFICIENTS ON THE BASIS OF SELECTED SETS OF EIGEN-POSTURES

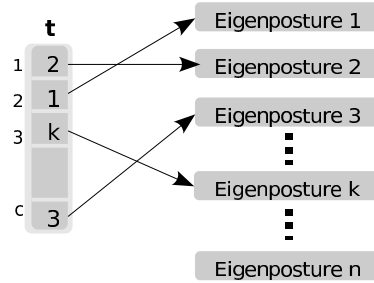


Figure 5.8: The selection mechanism used by the GA model to identify a set of eigenpostures.

will has components $t_i = ind_i$ (see Figure 5.8). Accordingly to this approach, given a new object \mathbf{x} , the GA model will give in output a set of vectors $\mathbf{t}^1, \dots, \mathbf{t}^L$ each of which identify a specific set of eigenpostures.

5.4 Generation of expected hand-configurations coefficients on the basis of selected sets of eigen-postures

The algorithms showed in Algorithms 4.1 and 4.2 presuppose that a sequence of hand joints configurations coefficients $\beta_k(t_h)$ is generated for each selected set of eigenpostures. Each of these sequences can be seen as a sort of *prototype action* associated to every sets of eigenpostures.

Each prototype action can be constructed as follows. For each class of object-directed action, we collect a sufficient number of actions each of which have an associated sequence of hand joints configurations coefficients $\beta_k^i(t_1), \dots, \beta_k^i(t_m)$ where i identify the i -th action instance, k the object-directed action class and m the length of the actions. The prototype action is formed by the sequence $\bar{\beta}_k(t_1), \dots, \bar{\beta}_k(t_m)$ where each element $\bar{\beta}_k(t_j)$ is obtained as:

$$\bar{\beta}_k(t_j) = \frac{1}{N} \sum_{i=1}^N \beta_k^i(t_j)$$

that is the element at time t_j of the sequence is obtained as the mean of all coefficients of the collected actions at time t_j .

Experiments and results

6.1 Different eigenposture sets for different classes of grasp actions

A basic hypothesis of the computational model of mirror neurons introduced in Chapter 4 concerns the existence of different sets of eigenpostures for different classes of object-directed actions. In this work we have focussed on one type of object-directed actions, that is grasping actions, and our objective is to show that different sets of eigenpostures exist for different modalities of grasping actions execution.

Although mirror neurons have been studied on a restricted number of grasping actions performed by monkeys¹, here we have chosen to work with a more comprehensive set of grasping actions performed by human beings (see Table 6.1). The dataset used has been provided by *Lira Lab University of Genova, Italy* and consists of five different grasping actions (*cylindrical, spherical, tripod, flat and pinch*) some of which are executed toward different objects and all of them repeated by twenty subjects. The pinch grasp is the same as the precision grip grasping while all others types of grasping are variants of the whole hand grasping.

Table 6.1 summarizes the set of grasping actions together with the objects toward which the actions have been directed.

¹For instance in Gallese et al. (1996) the activity of mirror neurons has been recorded during the execution/observation of grasping with precision grip and grasping with a whole-hand prehension.

6.1. DIFFERENT EIGENPOSTURE SETS FOR DIFFERENT CLASSES OF GRASP ACTIONS




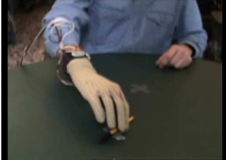

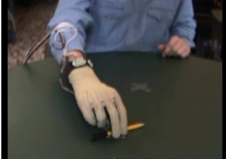





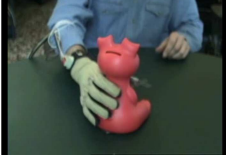

Grasp type	object			Grasp type	object
SPHERICAL	BALL				
TRIPOD	BALL			FLAT	HAMMER
PINCH	PEN			PINCH	SCOTCH-TAPE
TRIPOD	PEN			SPHERICAL	SCOTCH-TAPE
PINCH	DUCK			TRIPOD	SCOTCH-TAPE
TRIPOD	DUCK			FLAT	LEGO
CYLINDRICAL	PIGLET			PINCH	LEGO

Table 6.1: The Grasp dataset used in this work were provided by *Lira Lab University of Genova, Italy*. It is composed of five different grasping actions: cylindrical, spherical, tripod, flat and pinch. The objects used are: lego, scotch-tape, duck, piglet, hammer, pen and ball. Some actions were directed toward more than one object.

6.1. DIFFERENT EIGENPOSTURE SETS FOR DIFFERENT CLASSES OF GRASP ACTIONS

6.1.1 Experimental set-up

For each grasping action the dataset contains the sequence of hand configurations obtained with the dataglove *CyberGlove* (*CyberGlove; Virtual Technologies, Palo Alto, CA, USA*) endowed with 22 sensors.

The experimental setting used to record the dataset is illustrated in Figure 6.1. The subject was seated at a table with a clearly visible surface mark (X) placed at a comfortable distance for grasping execution. For each target object and grasping action modality a subject was asked to position the right hand approximately on a starting position, and to reach and grasp the target object placed on mark X and to place the object on another surface mark (X) near the previous mark. Each grasping action was repeated twenty times. An example of grasping action execution is shown in Figure 6.2.

6.1.2 Results

For our tests we have used only the data related to the first subject.

We have thus constructed 13 (resulting from the combination of different grasping modality and objects) different matrices indicated with $\{T^i\}_{i=1,\dots,13}$ each containing all vectors t disposed row-wise related to the execution of a given grasping action directed toward a specific object. Each matrix T^i has thus $c = 22$ columns equal to the number of sensors and a number of rows equal to the sum of the lengths of all actions for that kind of grasping and object.

In order to obtain eigenpostures we have performed Principal Components Analysis on each matrix T^i . PCA technique is described in appendix C.

Table 6.2 summarizes the cumulative variance obtained by selecting the first four eigenpostures (eigenvectors) ordered by decreasing value of associated eigenvalues λ_i . For k selected eigenpostures the cumulative variance is computed as follows:

$$variance_k = \frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^c \lambda_i} \cdot 100$$

where we have multiplied by 100 to obtain a result expressed as percentage of total variability.

The results show that four components suffice for capturing $\sim 90\%$ of data variability for all modalities of grasping actions.

6.1. DIFFERENT EIGENPOSTURE SETS FOR DIFFERENT CLASSES OF GRASP ACTIONS

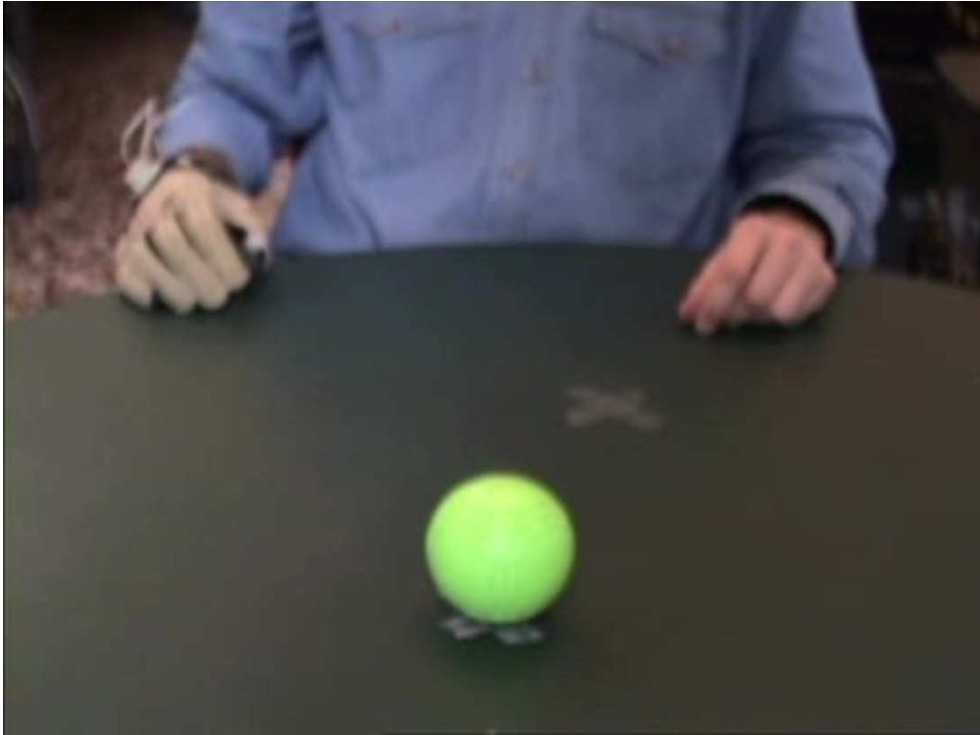


Figure 6.1: The picture illustrates the experimental setting used to record the dataset. The subject was seated at a table with a clearly visible surface mark (X) placed at a comfortable distance for grasping execution. For each target object and grasping action modality a subject was asked to position the right hand approximately on a starting position, and to reach and grasp the target object placed on mark X and to place the object on another surface mark (X) near the previous mark.

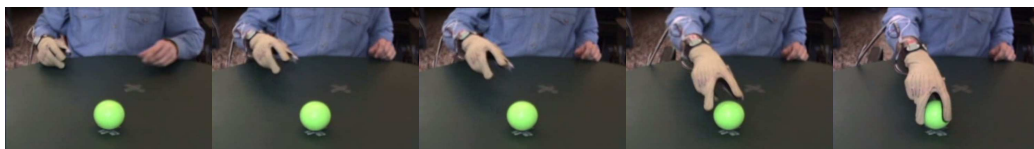


Figure 6.2: An example of grasping action contained in the used dataset. In particular the example was related to a tripod grasping action directed toward a ball.

6.1. DIFFERENT EIGENPOSTURE SETS FOR DIFFERENT CLASSES OF GRASP ACTIONS

	BALL SPHERICAL	BALL TRIPOD	PEN PINCH	PEN TRIPOD	DUCK PINCH	DUCK TRIPOD	PIGLET CYLINDRICAL
1	81	66	48	83	84	73	73
2	87	81	78	92	93	92	84
3	92	86	86	94	95	94	89
4	95	89	89	96	97	96	92

	HAMMER FLAT	SCOTCH-TAPE PINCH	SCOTCH-TAPE SPHERICAL	SCOTCH-TAPE TRIPOD	LEGO FLAT	LEGO PINCH	
1	77	81	74	41	77	57	
2	86	88	84	69	85	74	
3	89	92	90	78	91	81	
4	92	94	93	83	93	87	

Table 6.2: The table summarize the cumulative variance for the first four principal components resulting from the PCA analysis on each grasping action. As can be seen four principal components suffice for capturing ~ 90 of data variability for all actions.

Thus, we have obtained a set of eigenpostures for each modality of grasping action and object; we have computed the similarity measure between each pairs of principal subspace as proposed in (Krzanowski, 1979).

The obtained similarity matrix is shown in Figure 6.3.

In order to show the existence of different sets of eigenpostures we have performed a divisive clustering (Hastie et al., 2003) between principal subspaces. This kind of clustering algorithm starts with all objects grouped into a single cluster and recursively divides each cluster if the maximum distance² between elements exceeds a certain threshold $toll$.

The result of clustering is shown in Table 6.3 and in Table 6.4. The former is related to the results obtained with a threshold $toll = 0.75$.

In this case only three clusters were obtained. As can be seen, all the pinch grasps have been grouped together in *cluster 2*. In the same cluster one other kind of grasping action is contained: a flat one directed toward a lego object. It can be argued, however, that for this action there is a strong similarity in hand shape with the other kind of pinch actions, in particular with the *pinch lego* grasping action. Moreover, note that the cylindrical grasping action, for which we have instances only directed towards the piglet toy object, is placed into a different cluster. Finally, the other three classes of grasping actions, that is tripod, spherical and flat are grouped

²The distance between two principal subspaces is computed as the inverse of the similarity, that is $dist(L, M, k) = 1/sim(L, M, k)$ (see also Section 5.2.1).

6.1. DIFFERENT EIGENPOSTURE SETS FOR DIFFERENT CLASSES OF GRASP ACTIONS

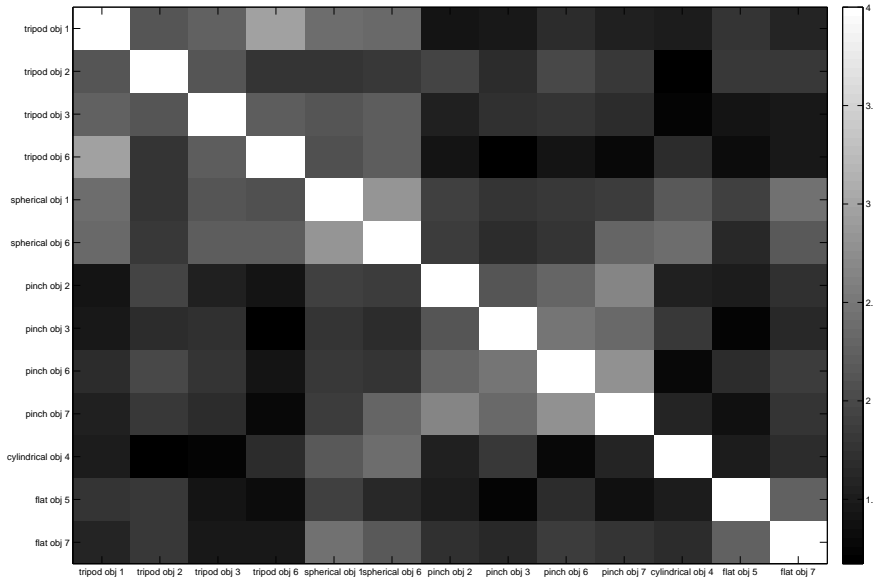


Figure 6.3: Similarity matrix between all computed principal subspaces.

together in *cluster 1*.

We can achieve some insight into the nature of the clustering result by changing the *toll* value. For $toll = 0.7$ we obtain four clusters whose elements are summarized in Table 6.4. As a result, *cluster 1* of the previous experiment has been split into two different clusters: *cluster 1* and *cluster 4*. If we compare the elements of each clusters we note that, even when the grasping actions are different, they share a common hand shape due to the object shape features.

This leads us to the following concluding considerations:

- ▷ different sets of eigenpostures exist for the two main modalities of grasping actions. In fact all the pinch grasps, which are the same as precision grip grasping actions, group together. All other grasping actions, which are very similar to whole hand grasping action are mainly grouped (except for the cylindrical grasp) in one other cluster;
- ▷ distinguishing between sets of eigenpostures may become more or less difficult on the basis of the object at which the action is directed.

6.1. DIFFERENT EIGENPOSTURE SETS FOR DIFFERENT CLASSES OF GRASP ACTIONS














<i>cluster 1</i>	<i>cluster 2</i>	<i>cluster 3</i>
spherical scotch-tape 	pinch pen 	cylindrical piglet 
flat hammer 	pinch duck 	
tripod ball 	pinch scotch-tape 	
spherical ball 	pinch lego 	
tripod duck 	flat lego 	
tripod scotch-tape 		
tripod pen 		

Table 6.3: Results of the divisive clustering with $toll = 0.75$ on the different sets of eigenpostures.

6.1. DIFFERENT EIGENPOSTURE SETS FOR DIFFERENT CLASSES OF GRASP ACTIONS

<i>cluster 1</i>	<i>cluster 2</i>	<i>cluster 3</i>	<i>cluster 4</i>
spherical scotch-tape 	pinch pen 	cylindrical piglet 	flat hammer 
tripod ball 	pinch duck 		tripod pen 
spherical ball 	pinch scotch-tape 		
tripod duck 	pinch lego 		
tripod scotch-tape 	flat lego 		

Table 6.4: Results of the divisive clustering with $toll = 0.7$ on the different sets of eigenpostures.

6.2. TESTING THE GA MODEL FOR AFFORDANCE EXTRACTION

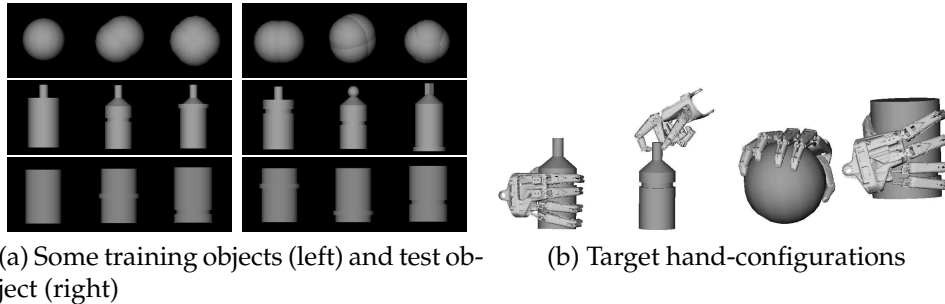


Figure 6.4: Examples of spherical, cylindrical and bottle objects used to train and test the system, and target hand-configurations.

6.2 Testing the GA model for affordance extraction

The GA model was designed so as to extract multiple hand-configurations, and to generalize its affordance-extraction capability with respect to novel objects. Two experiments were performed to test the extraction and generalization abilities, respectively. The results of these tests corroborate the possession of the extraction ability, in addition to the required generalization ability as far as novel objects obtained from the composition of known object parts are concerned

6.2.1 Experimental set-up

6.2.2 Results

The first test, which is concerned with the extraction of multiple hand-configurations, makes use of three different prototypical object images: a sphere, a cylinder and a bottle. It is assumed that the first two objects can be grasped using a power grasp only, whereas the bottle can be grasped in two different ways, by precision and power grasps. For each of these prototypical object images, similar images were generated by means of small contour variations. For each prototype, the resulting training and test sets were composed by 20 and 10 images, respectively (Figure 6.4)

In order to generate target hand configurations, GraspIt! (Miller and Allen, 2004), a robotic grasping simulator, was used. In particular, the robotic hand called Robonaut, endowed with 14 degrees of freedom, was chosen. Consequently, in the GA model hand configurations are identified by a vector of 14 components, where each component represents just

6.2. TESTING THE GA MODEL FOR AFFORDANCE EXTRACTION





Bottle Grasp	Bottle Grasp 2	Spherical	Cylindrical
1.2% \pm 0.4	1.9% \pm 0.6	3.9% \pm 1.4	1.3% \pm 0.4
			

Table 6.5: For each object class, the mean and standard deviation of the average error over all objects in the test set is reported here. Moreover, for each class mean hand-configuration over all objects in the class is exhibited.

	H	M	Image size	W	DS	Cluster
Test 1	5	2	160 \times 160	160 \times 160	0	None
Test 2	5	2	500 \times 500	160 \times 160	30	5

Table 6.6: Model parameters for each test. Image size, W and DS are expressed in pixels.

one hand joint’s angle. Spherical and cylindrical objects are associated to a single hand configuration, generated manually by changing the Robonaut’s degrees of freedom. Bottle objects are associated with two distinct hand configurations: a precision grasp, applied on the object’s top part, and a power one applied on the lateral part (see Figure 6.4). Training set targets are generated adding some Gaussian noise to these hand configurations. In this test, the attention window encompasses the whole object. Thus, for each object there is a single feature vector \mathbf{x} with an associated $p(\mathbf{t}|\mathbf{x})$. Hand configurations are obtained by selecting $\mu_k(\mathbf{x})$ associated with the higher values of $\alpha_k(\mathbf{x})$. The model parameters are summarized in Table 6.6. For the i -th degree of freedom, percentage error is defined as $\frac{|t_i - y_i|}{\max_i - \min_i} \times 100$, where y_i is the model output, and \max_i and \min_i are the max and the min value, respectively, for the i -th degree of freedom. *Average error* between model output hand configuration and target hand configuration is defined as the mean of percentage error over all degrees of freedom. For all test objects in each class, mean and standard deviation of average error is computed and shown in Table 6.5.

The second experiment is meant to test generalization capabilities with respect to novel objects. To test this ability, the system was trained to associate *parts* of an object to hand-configurations. Subsequently, the system was given in input a novel object resulting from the "composition" of previously known parts. In this test, a cup is used, which is obtained from the composition of a cylinder and a handle. Examples of both training

6.3. SOUNDNESS OF THE PROBABILISTIC APPROACH FOR HAND CONFIGURATION ESTIMATION

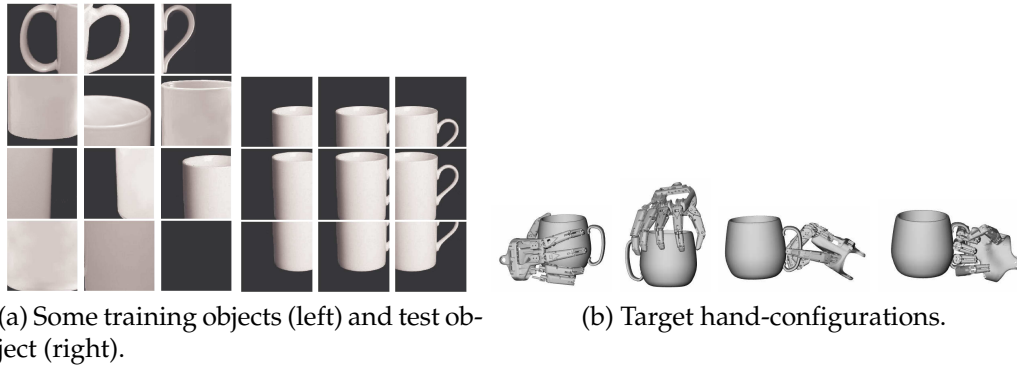


Figure 6.5: (a) Examples of training and test images (see text). (b) Examples of target hand-configurations.

images and the cup used as test image are shown in Figure 6.5. There are four kinds of training images: (a) cup handles; (b) upper and lower cup parts; (c) lateral cup parts; (d) non-graspable cup parts. Two target hand-configurations are associated with images (a); only one hand-configuration is associated to images (b) to (d). The training set targets are generated adding some Gaussian noise to hand configurations. Targets for non-graspable cup parts images are drawn from a Gaussian distribution with a large variance, so as to reflect the fact that in this case no plausible hand-configuration candidate exists. The K-Mean clustering algorithm is implemented by the AR Module, setting to 5 the number of clusters. In Table 6.7, cluster centroids are shown together with cluster variance. The fifth cluster was discarded in view of its large variance. Note that the first four cluster centroids are very similar to target hand configurations (Figure 6.5) with respect to which mean percentage error was computed.

6.3 Soundness of the probabilistic approach for hand configuration estimation

In this section the soundness of the probabilistic approach is tested with respect to a standard regression technique in the task of hand configuration estimation from its visual appearance. The selected standard regression technique is a two layer feedforward neural networks (FFW) (see Section B.1 for a brief introduction to feedforward neural networks).

The performances of the two systems have been tested on a reach to grasp action observed from different points of view. We have chosen to use multiple observation points of view to stress the problem of fingers

6.3. SOUNDNESS OF THE PROBABILISTIC APPROACH FOR HAND CONFIGURATION ESTIMATION






Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
				
$\sigma = 0.12$	$\sigma = 0.12$	$\sigma = 0.09$	$\sigma = 0.09$	$\sigma = 0.34$
Mean and standard deviation of percentage error				
$1.9\% \pm 2$	$2.5\% \pm 2$	$2\% \pm 1.2$	$1.8\% \pm 1.5$	(discarded)

Table 6.7: The graph visualizes the obtained cluster centroids. Compare these images with target hand configurations of Figure 6.5. The fifth cluster was discarded in view of its large variance. The percentage error with respect to target was mediated over all degrees of freedom.

self-occlusions. The results show that:

- ▷ when the problem of fingers self-occlusions is immaterial the two systems have comparable performance;
- ▷ when the number of fingers self-occlusions increases then the probabilistic approach outperforms a standard regression technique.

6.3.1 Experimental set-up

In order to compare the two systems a dataset of pairs $\{\mathbf{x}^n, \mathbf{t}^n\}_{n=1, \dots, N}$, with \mathbf{x}^n vectors of hand visual features and \mathbf{t}^n vectors of hand joints configurations, must be collected to train and test the two systems.

To achieve this goal a dataglove together with a 3D rendering software have been used.

The dataglove is a HumanGlove (*HumanGlove, Humanware S.r.l., Pontedera, Pisa, Italy*) (see Figure 6.6) endowed with 16 sensors. The dataglove is connected to a 3D rendering software which read the values of the sensors and constantly updates a 3D human hand model (see Figure 6.7). Thus, we are able to collect pairs *sensors values - hand image*.

In order to extract the visual features vectors \mathbf{x}^n , each image of size 670×490 pixels is converted to grayscale, subsampled at size 151×112 pixels and linearized into a single vector of size 1×16912 . A PCA algorithm is applied over a large dataset of collected input image and only the first principal component is computed. Each image is projected in the space of the first principal component and is therefore coded by a scalar value x^n .

One may doubt that a single principal component suffices to give a good representation of data variability. However this choice is motivated

6.3. SOUNDNESS OF THE PROBABILISTIC APPROACH FOR HAND CONFIGURATION ESTIMATION



Figure 6.6: The Human-glove is used to obtain hand joints configurations. This glove is endowed with 16 sensors.

by the following consideration: the input images are constructed with a 3D rendering software, and thus have little or negligible noise with respect to real world images. This fact simplifies drastically the problem of hand configuration estimation. By contrast we want to stress the problem of computing the joints configurations from visual inputs when the mapping from visual inputs to hand configurations does not assume functional form. For this reason, by taking only the first principal component, we have discarded a lot of input information and made the problem “more difficult” with respect to problems in which the visual input is constructed by many principal components.

6.3.2 Results

We have tested the two systems on grasping actions made with a precision-grip. A precision grip action has been repeated by a human being twenty times and the 3D simulator have synthesized the related hand configurations with respect to five different points of view as showed in Figure 6.8. All these points of view are related to an observer who should recognize the correct hand configurations.

We have collected a dataset of $N = 2059$ elements for each point of view. This dataset was split into three different subsets for training, validation and testing respectively.

6.3. SOUNDNESS OF THE PROBABILISTIC APPROACH FOR HAND CONFIGURATION ESTIMATION

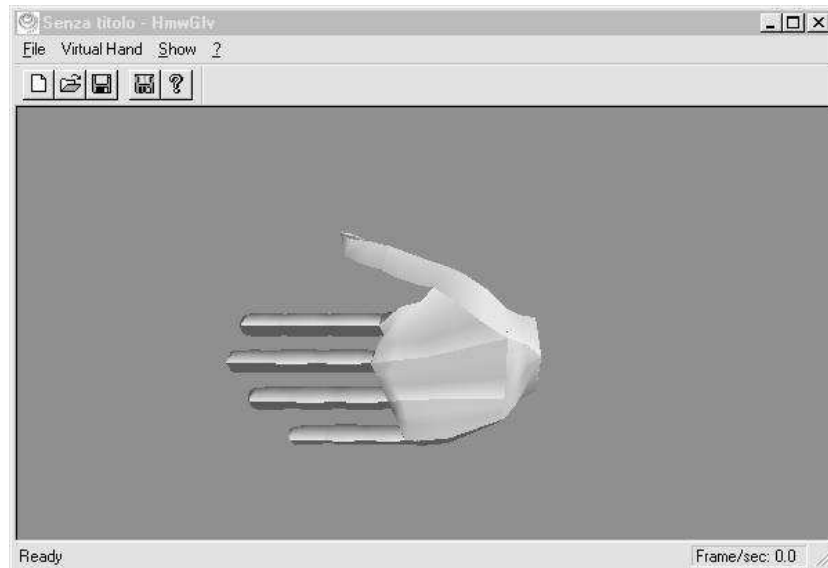


Figure 6.7: A 3D rendering software is used to construct a 3D hand model starting from hand joints configuration. The 3D hand model can be used to obtain 2D hand images from arbitrary points of view.

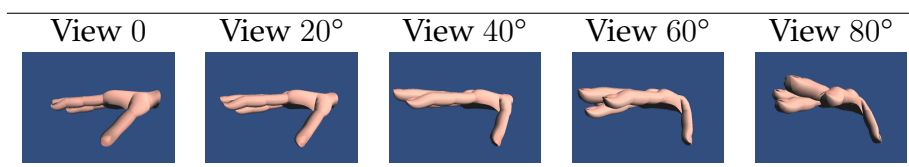


Figure 6.8: Five different points of view have been used for this test. All of them are related to an observer which has to recognize the correct hand configuration.

6.3. SOUNDNESS OF THE PROBABILISTIC APPROACH FOR HAND CONFIGURATION ESTIMATION

	FFW	MDN
Dim. x	1	1
Dim. t	16	16
H	5 – 10 – 15	10
K	None	from 5 to 40 at step 5
N^{train}	686	686
N^{valid}	686	686
N^{test}	687	687
T	10	10

Table 6.8: The table summarizes the parameters used to test both FFW and MDN.

The FFW was trained using different numbers, H , of hidden units. The training was repeated T times for each hidden nodes configuration.

The MDN was trained with $H = 10$ hidden units for the neural network component and different numbers, K , of kernels for the mixture component. Again for each kernel configuration the training was repeated T times.

The error for the two systems have been computed by means of a forward model as described in Section B.5 and schematically illustrated in Figure 6.9 in which the two systems implement the inverse model block. The forward model is just another FFW which has to learn the relation between t and x and must be able to predict the visual input x associated to a given new hand-configuration t . The network which implements the forward model has 10 hidden units and has been trained on the same training set as the FFW system and MDN system by interchanging the role of input and target. The forward model has a Root Mean Square Error (RMS) on the test set equal to 0.025.

For each configuration (of nodes for FFW and of kernels for MDN) and each trial the RMS error was computed and mediated over all trials T . Thus, an RMS error value was obtained for each configuration and only the best configuration was selected. The same training and test process was repeated for each point of view. The testing parameters are summarized in Table 6.8.

Figure 6.10 shows the RMS error for different points of view.

As can be seen the error increases for both systems as the angle at which the hand is observed increases. However, MDN error is almost always less than FFW error (only for view point 40° the error is almost the

6.3. SOUNDNESS OF THE PROBABILISTIC APPROACH FOR HAND CONFIGURATION ESTIMATION

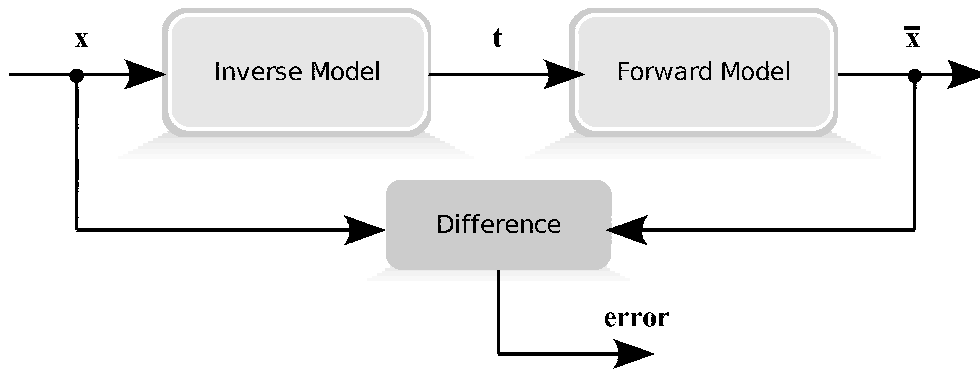


Figure 6.9: The schema used to compute the error of the two systems FFW and MDN. See Section B.5 for more details.

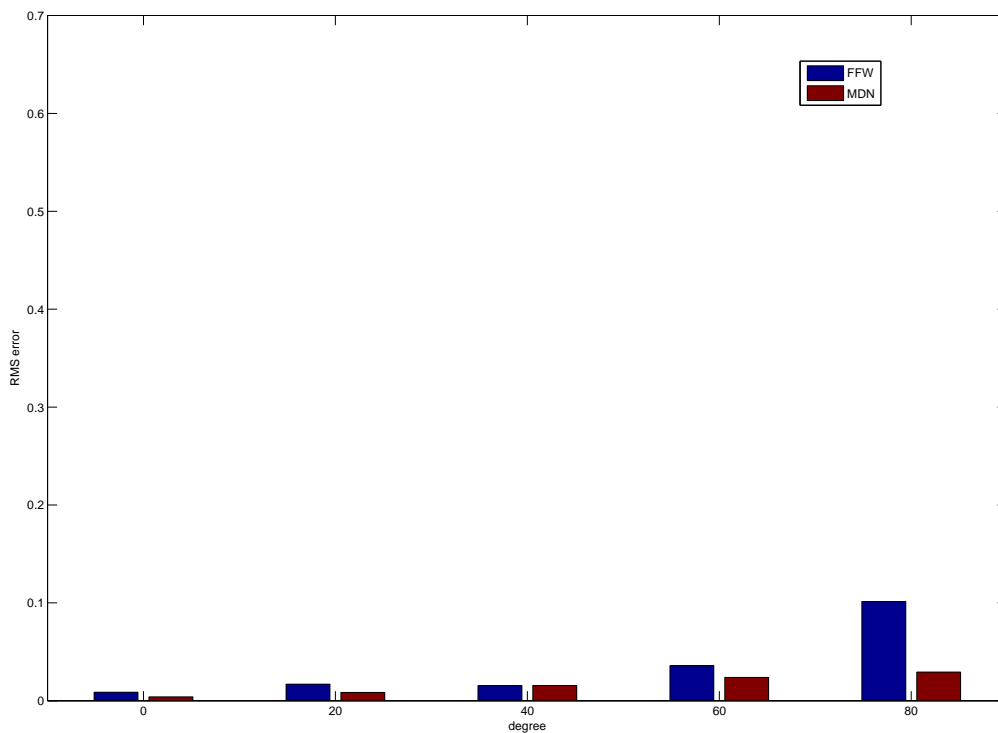


Figure 6.10: RMS error for both FFW and MDN system. As can be seen the error increases as the angle at which the hand is observed increases. In almost all case but one the MDN system have a minor RMS error with respect to the FFW system.

6.3. SOUNDNESS OF THE PROBABILISTIC APPROACH FOR HAND CONFIGURATION ESTIMATION

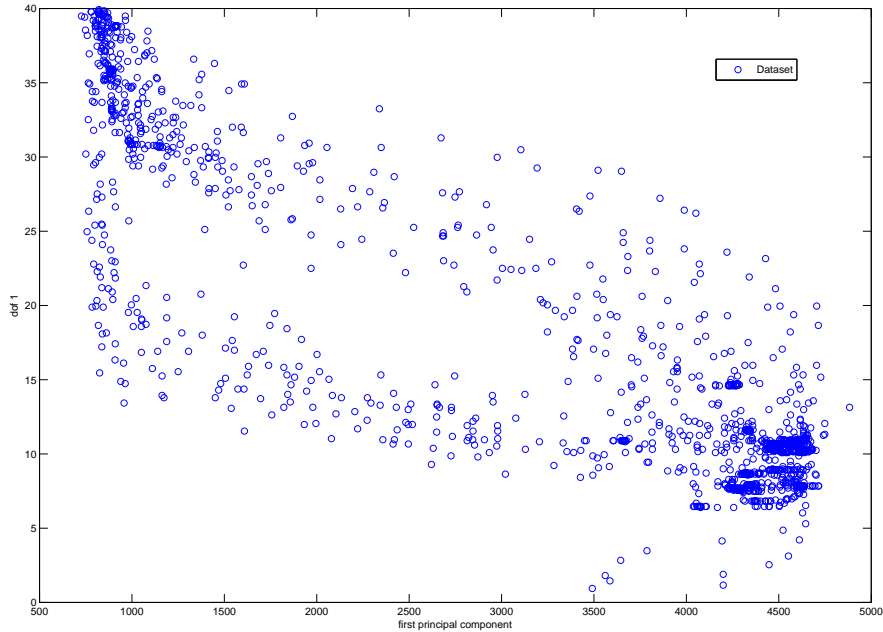


Figure 6.11: Datapoints for the point of view 40° and for the joint 1. As can be seen for many points the mapping between x and t is single valued. This is the reason of the similarity between the performance of FFW and MDN.

same for the two systems).

We can achieve more insight into the nature of error by looking at (Figure 6.11) where the plot of the datapoints x is shown, related to the view point 40° with respect to dof 1. As can be seen the mapping between x and t is in some zone multi-valued while in other zones it is single valued. Moreover, there are many points for which the mapping is single-valued and this behaviour is common to many others joints. This is the reason for the similarity between the performance of FFW and MDN.

If we compare the performances of the two systems only in the zone in which the mapping is multi-valued we obtain the results showed in Figure 6.12. In this case the difference between the performance of MDN and FFW is significant and MDN outperforms the results of MDN everywhere.

It is important to note that the points of the multi-valued zone correspond to positions of the hand in which there are more self-occlusions between fingers (e.g. when the hand is closed on the object).

6.3. SOUNDNESS OF THE PROBABILISTIC APPROACH FOR HAND CONFIGURATION ESTIMATION

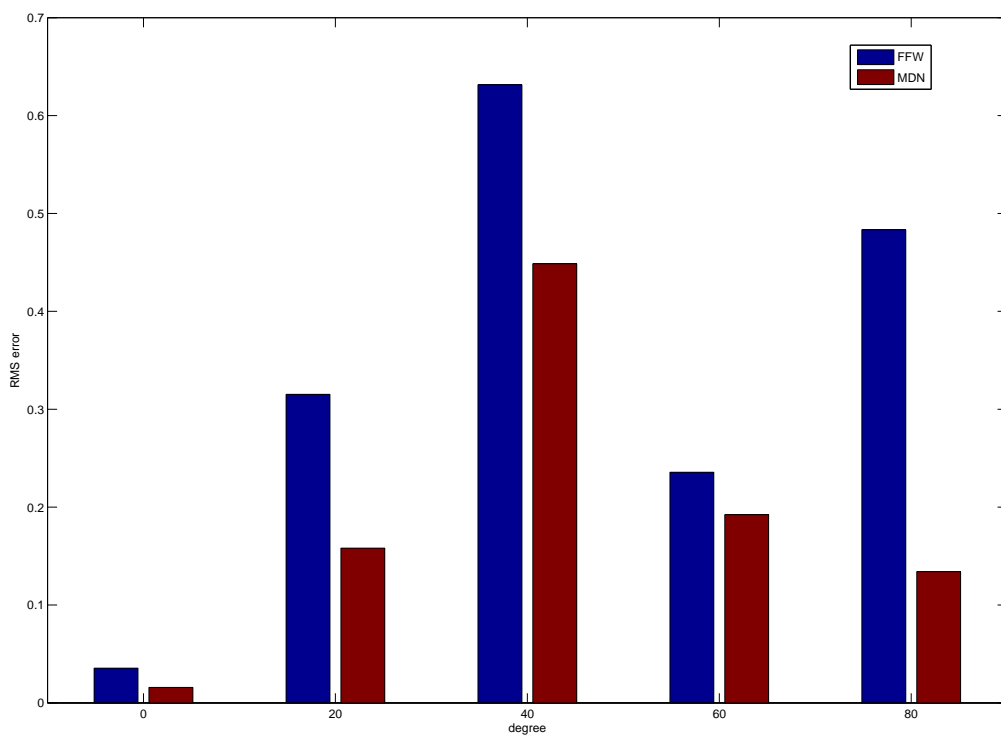


Figure 6.12: Performance of the two systems in the zone in which the mapping is multi-valued. The difference between the performance of MDN and FFW is significant and MDN outperform the results of MDN for all points of view.

6.3. SOUNDNESS OF THE PROBABILISTIC APPROACH FOR HAND CONFIGURATION ESTIMATION

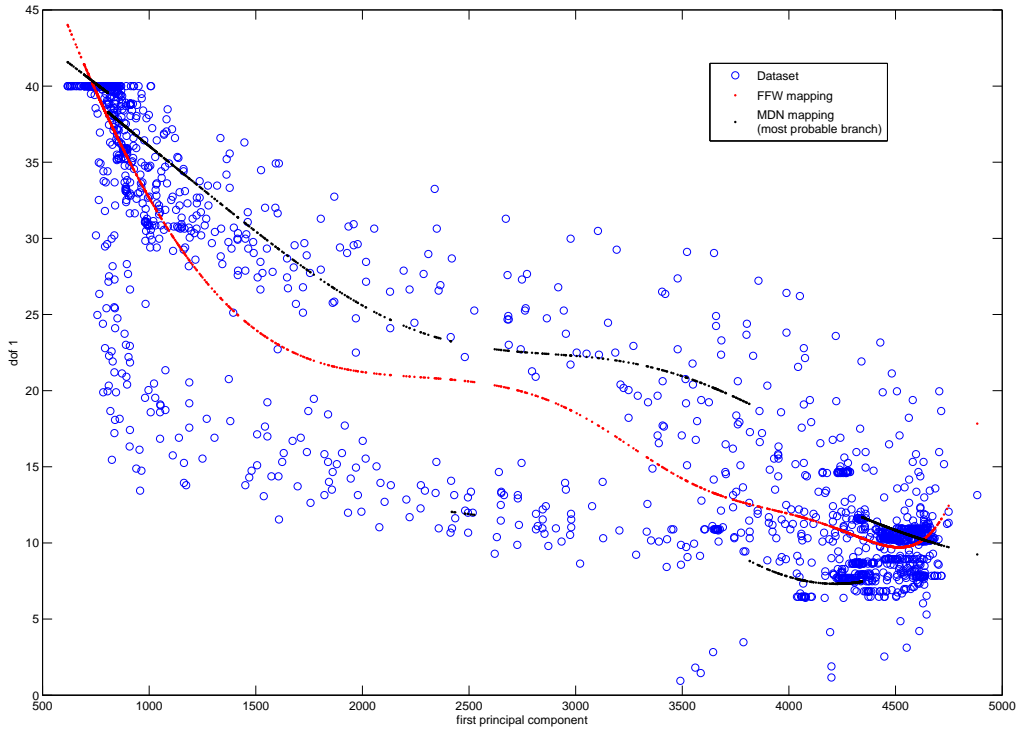


Figure 6.13: Comparison between the MDN and the FFW mapping. As can be seen where the mapping is multi-valued the MDN gives in output one of the possible solutions while FFW mediate over all possible solutions.

In order to give an idea of why the MDN system works better we can look at the mapping obtained for FFW and MDN, and showed in Figure 6.13. As can be seen, where the mapping is multi-valued the MDN gives in output one of the possible solutions while FFW mediate over all possible solutions which are not a hand configuration.

To sum up. We have shown that the probabilistic approach outperforms the standard regression technique in the task of hand configuration estimation especially when the number of self-occlusions between fingers increases.

6.4 How motor information improves hand-configuration estimation

In the previous section, we have shown that the probabilistic approach can be used to outperform a standard regression technique in hand configuration estimation. In this section we want to show that even better results can be obtained if we use motor information.

As explained in previous chapters motor information is a prior information furnished by motor system to visual system in order to interpret incoming visual input. For example, if we are looking an object we are already able to infer possible actions that can be directed toward that object. Moreover, we can benefit from knowing how such action can be performed by providing to the visual system additional useful information to improve hand configuration estimation.

More specifically, given the probabilistic framework introduced previously, motor information can be used in two ways:

1. knowing which object-directed action can be performed given the current object, we can create “specialized” modules (where each module implements the probabilistic framework), one for each object-directed action. Each module will be responsible for the estimation of hand configuration during a particular action;
2. for each object-directed action, we know how such kind of action can be performed, in particular we know the associated set of eigenpostures useful for action control. We can use such information to decompose t in terms of the linear combination of the set of eigenpostures associated to that action, and estimate the distribution of the parameters β of the linear combination $p(\beta|x)$ instead of the distribution $p(t|x)$.

In the next section we will show that we improve performance wrt both 1 and 2.

6.4.1 Experimental set-up

The first test will show that specialized modules, one for each object-directed action, work better, in terms of reduced error, with respect to a single module in hand configuration estimation (see Figure 6.14). In the following we will call *single system* the system formed by a single MDN and *multiple systems* the system formed by many MDN, one for each action.

6.4. HOW MOTOR INFORMATION IMPROVES HAND-CONFIGURATION ESTIMATION

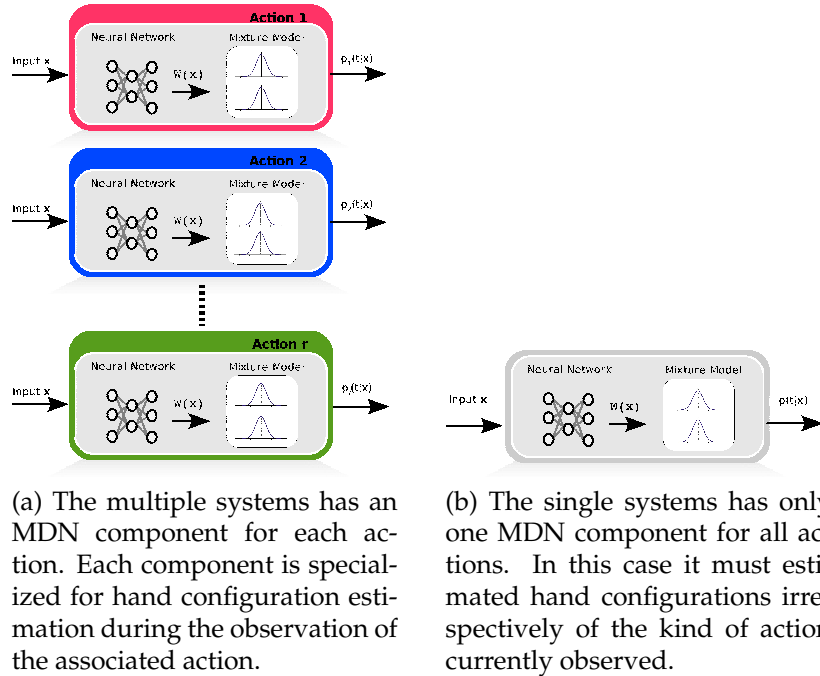


Figure 6.14: Comparison between the multiple system and the single system architecture.

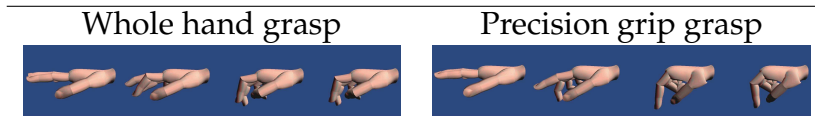


Table 6.9: Two types of grasping action have been used for this test: a whole hand grasping action and a precision grip grasping action.

In order to compare the two systems we have recorded two different grasping actions: a whole hand grasping action and a precision grip grasping action. Some representative frames for each of the two actions, as reconstructed by the 3D rendering software, are shown in Figure 6.9.

Both actions have been repeated twenty times and the obtained input images have been processed as in the previous experiment in order to extract feature vectors.

6.4.2 Results

We have collected a dataset of $N = 2200$ elements for each action. Both datasets were split into three different sets for training, validation and

6.4. HOW MOTOR INFORMATION IMPROVES HAND-CONFIGURATION ESTIMATION

	SINGLE SYSTEM	MULTIPLE SYSTEMS MDN_1	MULTIPLE SYSTEMS MDN_2
Dim. x	1	1	1
Dim. of t	16	16	16
H	10	10	10
K	from 5 to 40 at step 5	from 5 to 40 at step 5	from 5 to 40 at step 5
Data source	whole hand grasp precision grip grasp	whole hand grasp	precision grip grasp
N^{train}	686	686	686
N^{valid}	686	686	686
N^{test}	687	687	687
T	10	10	10

Table 6.10: The table summarizes the parameters used to test both *Single system* and *Multiple Systems*. Note that for the multiple system we have two different MDN each of which is trained on only one grasping action.

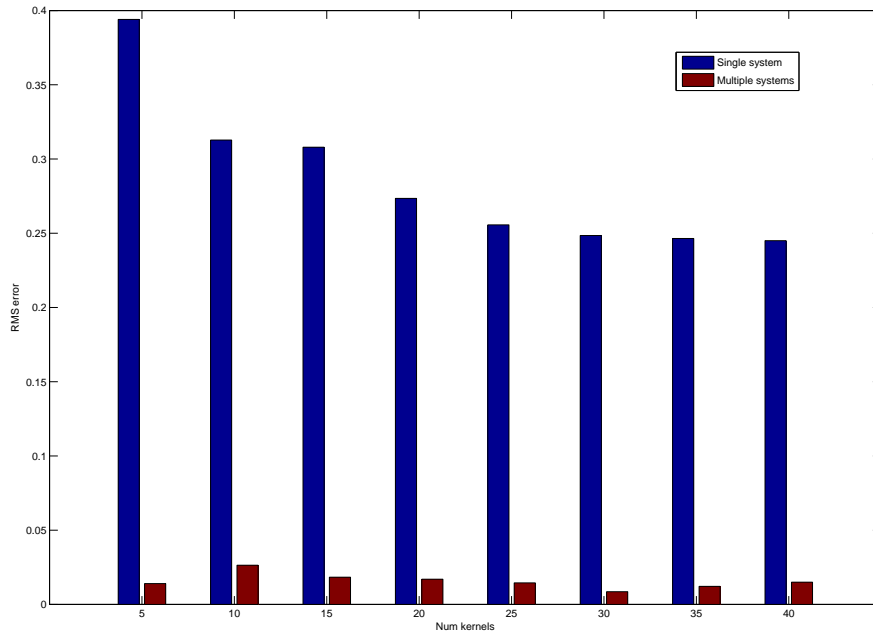
testing respectively. For the multiple system we have two different MDNs each of which has been trained on data related to one of the two actions. The MDN related to the single system has, instead, been trained on data related to both actions. The training parameters have been summarized in Table 6.10.

The error for the two systems have been computed by means of a forward model as described in Section B.5. The error was computed on the test set of both actions. However, for multiple system we have selected the output of the MDN related to the action currently given in input as test. In Table 6.10 the testing parameters are summarized.

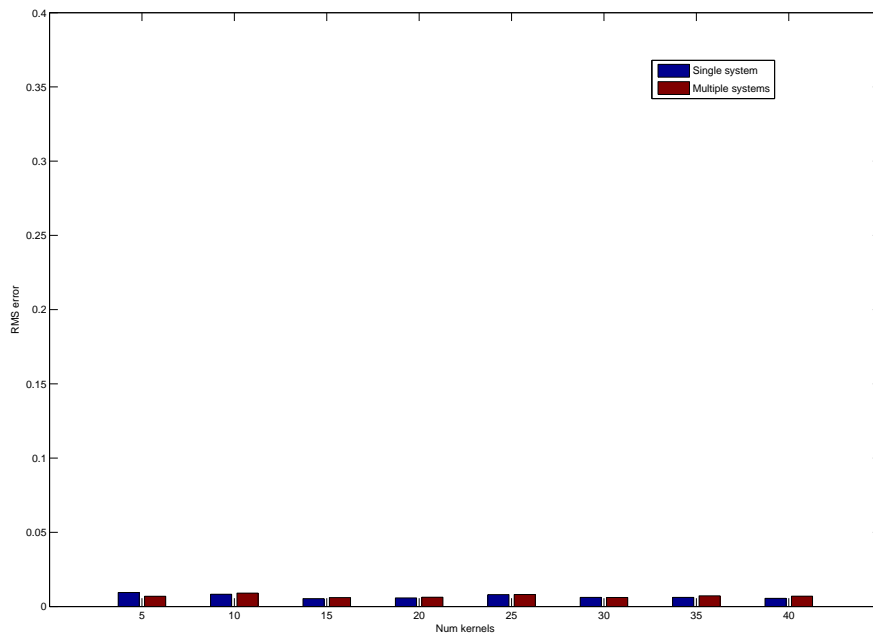
In Figure 6.15 it is shown the Root Mean Square Error (RMS) for different kernel configurations for both single system and multiple system. The two systems have almost similar error on whole hand grasping action, while on precision grip action the multiple system has considerable less error for every kernel configuration.

This behaviour is brought out in Figure 6.16 where we have plotted the error related to the best kernel configuration for both single system and multiple system.

6.4. HOW MOTOR INFORMATION IMPROVES HAND-CONFIGURATION ESTIMATION



(a) RMS error for precision grip grasping action related to each kernel configuration. The multiple systems outperform the single system for every kernel configurations.



(b) RMS error for whole hand grasping action related to each kernel configuration. The multiple systems and the single system have comparable performance.

6.4. HOW MOTOR INFORMATION IMPROVES HAND-CONFIGURATION ESTIMATION

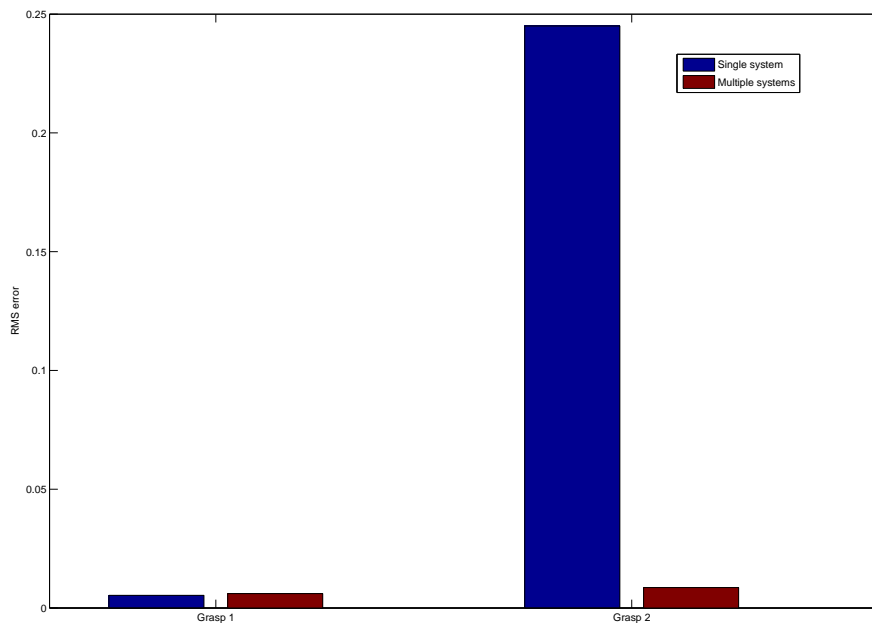


Figure 6.16: RMS error for precision grip and whole hand grasping actions related related to the best kernel configuration for the two systems. The multiple systems outperform the single system for latter grasping action while have comparable performance with respect to the single system for the former grasping action.

6.4. HOW MOTOR INFORMATION IMPROVES HAND-CONFIGURATION ESTIMATION

	FFW	MDN WITHOUT EIGENPOSTURES	MDN WITH EIGENPOSTURES
Dim. \mathbf{x}	1	1	1
Dim. \mathbf{t} (β)	16	16	3
H	5 – 10 – 15	10	10
K	None	from 5 to 40 at step 5	from 5 to 40 at step 5
N^{train}	686	686	686
N^{valid}	686	686	686
N^{test}	687	687	687
T	10	10	10

Table 6.11: The table summarizes the parameters used to test a classical regression system, in this case a feedforward neural network, and two different MDN systems. The former is the same as the test described in section 6.3.1 while the latter benefits from the decomposition of \mathbf{t} in terms of eigenpostures.

6.4.3 Experimental set-up

In this experiment we will show the benefit flowing by the decomposition of \mathbf{t} in terms of a linear combination of eigenpostures. More specifically we will show that we can outperform the result showed in Section 6.3.2 if we estimate the distribution $p(\beta|x)$ instead of the distribution $p(\mathbf{t}|x)$.

Eigenpostures have been computed in the same way as described in Section 6.1.2. Three components suffice to describe more than 90% of the total variability of hand-joints configurations \mathbf{t} .

The test was performed on the same data of experiment reported in section 6.3.1 on five different point of view. The test parameters are summarized in Table 6.11.

6.4.4 Results

Figure 6.17 shows the RMS error for the best configuration, over all trials and kernels or nodes configurations, for the five points of view. The results of FFW and MDN without eigenpostures are the same as Figure 6.10. The system with eigenpostures gives almost always better results with respect to MDN without eigenpostures.

Again if we compute the error where the mapping is not functional the differences between the three systems change significantly.

6.4. HOW MOTOR INFORMATION IMPROVES HAND-CONFIGURATION ESTIMATION

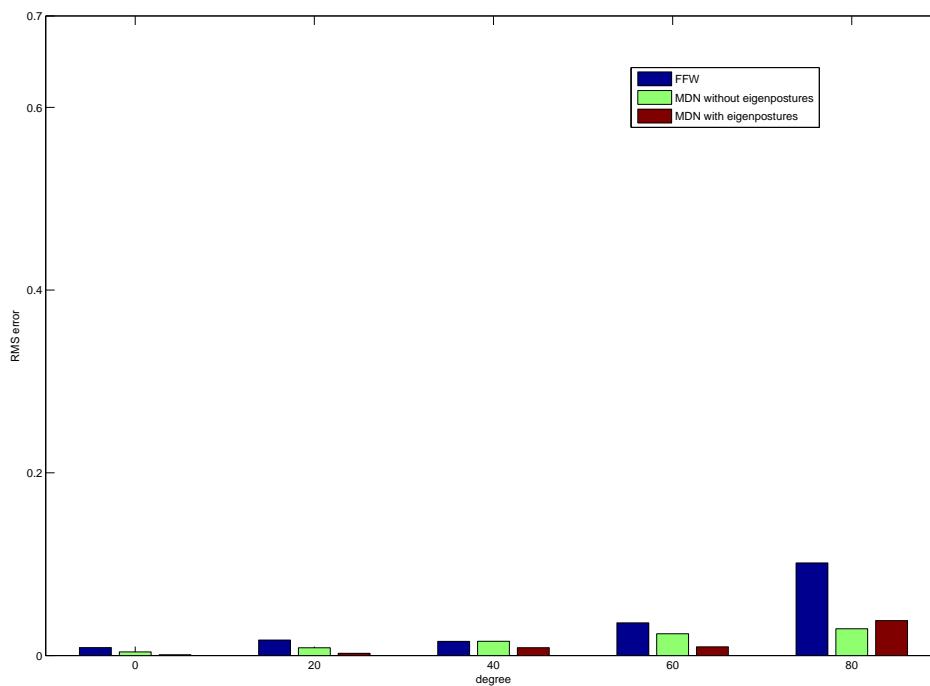


Figure 6.17: RMS error for both FFW and MDN systems. As can be seen the error increases as the angle at which the hand is observed increases. In almost all case but one (view point 80°) the MDN system with eigenpostures have a minor RMS error with respect to the MDN system without eigenpostures. However the MDN with eigenposture always have less RMS error with respect to FFW.

6.4. HOW MOTOR INFORMATION IMPROVES HAND-CONFIGURATION ESTIMATION

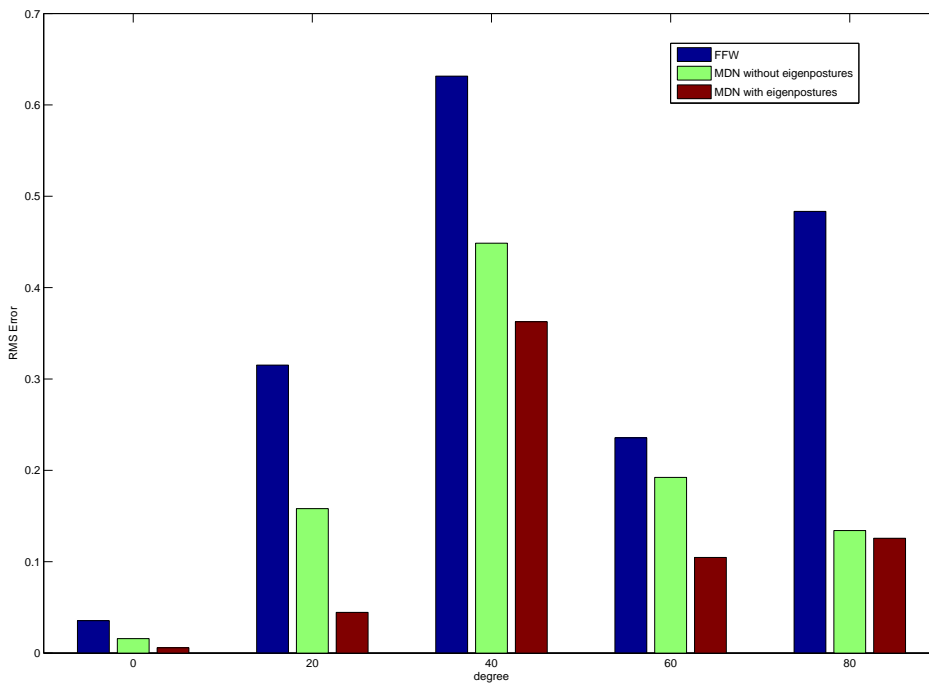


Figure 6.18: Performance for both FFW and MDN systems in the zone in which the mapping is multi-valued. As can be seen MDN system with eigenpostures have always a minor RMS error with respect to both FFW system and MDN system without eigenpostures.

6.5. A TEST OF THE WHOLE SYSTEM

N_{WH}	N_{WH}^p	N_{PG}	N_{PG}^p	m
52	25	51	25	41

Table 6.12: Parameters used to test the whole system in action observation mode.

6.5 A test of the whole system

The objective of this test is to show the soundness of the whole system in action observation mode as described in Algorithm 4.1.

In this preliminary test there are the same two classes of actions taken into consideration for the test described in Section 6.4, that is a whole hand grasping action (WH) and a precision grip grasping action (PG). We assume that the GA model can be used, as discussed in Section 5.3.2, to select the correct two sets of eigenpostures associated to the two classes of grasping actions. We further assume that two prototype actions, in terms of sequences of hand joints configurations coefficients, can be constructed as discussed in Section 5.4.

6.5.1 Experimental set-up

We have recorded N_{WH} actions of type WH and N_{PG} actions of type PG. We have used the HumanGlove together with the 3D rendering system to obtain both hand joints configurations and related visual input. The visual input x^n for each frame were constructed as explained above. Moreover a PCA algorithm have been applied to each of the two set of hand joints configurations related to actions of class WH and PG. The first three components suffice to obtain a whole variance of more than 90%. So each hand joints configurations have been expressed in terms of the coefficients β . All the recorded actions have been aligned to a fixed length of m frames. N_{WH}^p actions of class WH and N_{PG}^p of class PG have been used to construct the two prototype sequences $\beta_1^{WH}, \dots, \beta_m^{WH}$ and $\beta_1^{PG}, \dots, \beta_m^{PG}$, while the remaining actions have been used to test the system. In Table 6.12 the used parameters for this test are summarized.

6.5.2 Results

Each test action is classified accordingly to the Algorithm 4.1. We have assumed that GA model is able to select two different sets of eigenpostures $S_1 \equiv WH \text{ grasp}$ and $S_2 \equiv PG \text{ grasp}$. The initial probability of each set of

6.5. A TEST OF THE WHOLE SYSTEM

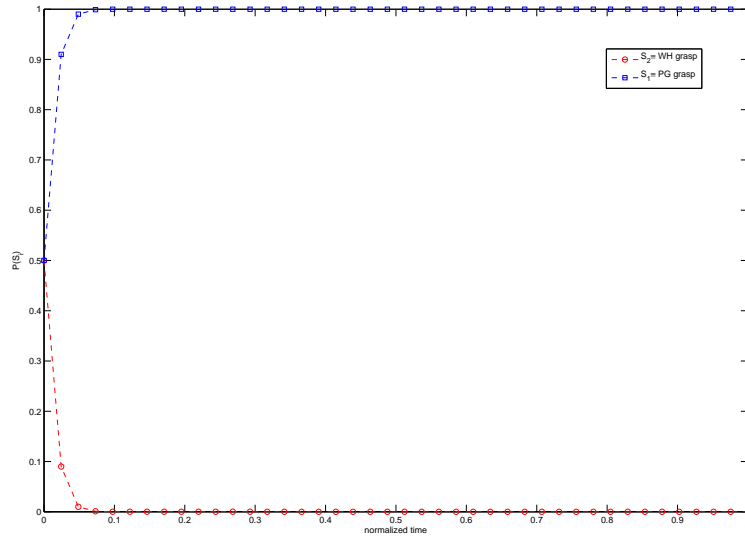


Figure 6.19: System response example 1

eigenpostures, $P(S_1)$ and $P(S_2)$, have been fixed to 0.5. At each time step t_h , such probabilities are updated on the basis on incoming visual input $x(t_h)$ as follows:

$$P(S_k) \leftarrow (P(S_k) * \pi_h^k) / \sum_{i=1,2} P(S_i) * \pi_h^i$$

where the $\pi_h^k = p_k(\beta(t_h)|x(t_h))$ with $k = 1, 2$, are computed with the same probabilistic framework presented in the above tests. In particular two MDNs have been used, one for each class of action, both with 10 hidden units and 10 kernels.

Each test action is assigned to the class for which the probability $P(S_k)$ is maximum at the end of the action.

All the 27 actions of class *WH* were correctly classified while 23 on a total of 26 test actions were correctly classified for the class *PG*.

In Figure 6.19 it is shown a typical system response for a *PG* action. As can be seen, in this simplified settings, the system is able to predict the correct action soon. This is what it happens for most cases.

In Figure 6.20 it is reported the response of the system to another *PG* action. In this case the system predict the wrong action in the first phase of the action and then the correct action in the last part of the action.

6.5. A TEST OF THE WHOLE SYSTEM

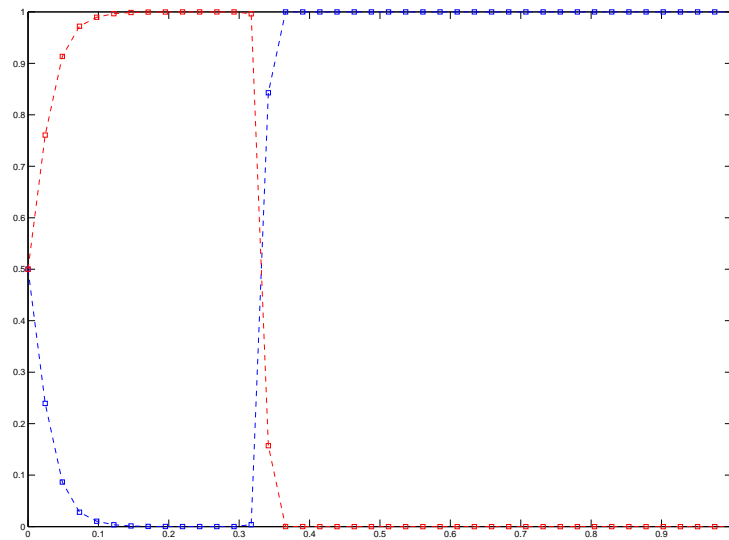


Figure 6.20: System response example 2

Conclusion and future work

7.1 Contribution of this work

We have argued that descriptively and explanatorily more adequate computational models of mirror neurons are needed. One of the main aspects that future computational modelling efforts should carefully take into account is, in our view, the functional interaction between sensory input and mirror activity.

As discussed in Chapter 3, in most current computational models of mirror neurons, motor information is the target of a computational process unidirectionally flowing from visual information to motor information. According to the same-input hypothesis, such computational process provides to mirror neurons a view-invariant scene description computed by the visual input only. If the same-activity hypothesis is also assumed in the model, the mirror behaviour arises simply as a side effect of the view-invariant property of the scene description computation.

This view of the underlying computation of mirror neurons behaviour has two major drawbacks: (a) it posits significant computational challenges insofar as the computation of a detailed scene description from visual input only is a difficult and often ill-posed task, (b) it makes the role of mirror neurons shallow.

The NeGOI model, in accord with the same-input hypothesis and presented in Chapter 3, provides evidence that at least one feature of the scene description, the grip-size, can be computed with some tolerance with respect to changes in point of view. Nevertheless, this model posits a cascade of complex computational processes to compute just one feature, and presumably other features should be taken into account to provide the overall description, giving rise to serious doubts about the feasibility that a similar

7.1. CONTRIBUTION OF THIS WORK

description can be computed by visual input only.

Accordingly, in this work, it was argued that motor involvement in sensory input interpretation may “facilitate” this interpretive process, in addition to assigning causally more significant roles to mirror mechanisms.

The following steps were taken in order to corroborate the claim that motor information can be used to improve sensory processing:

1. motor information useful to both action control and sensory processing was identified;
2. how the identified motor information can be used to improve the performance of the visual processes was described;
3. a model of the visual-motor interaction during object-directed actions was proposed;

In connection with (1), in Chapter 4, eigenpostures were identified as motor information useful to both action control and perspectival sensory inputs processing. Eigenpostures allow hand configurations to be identified with a very restricted number of variables. In Santello’s works it is shown that such eigenpostures exist for a class of object directed actions (grasping actions). However we also require that different sets of eigenpostures exist for different modalities of action execution. The experimental results showed in Section 5.2.1 endorse this requirement. Moreover the GA model examined in Section 5.3.1 provides a selection mechanism to choose the appropriate sets of eigenpostures during both action execution and action observation.

In connection with (2), in Section 6.4 the motor information in terms of sets of eigenpostures, selected by the GA model, has been proposed to be used in order to improve visual processing insofar as: i) more specialized mappings between perspectival sensory input and direct internal input can be achieved; ii) each mapping can benefit from the hand representation in terms of a very restricted number of eigenpostures. The results presented in Section 6.4 support both points i) and ii).

In connection with (3), the computational model outlined in Chapter 4 and specified in Chapter 5 provides a computational account of how motor information interacts with perspectival sensory input. In this approach, conjoined motor knowledge and visual inputs are the data enabling one to estimate the conditional probability distribution of hand configurations by combining a density model and a feed-forward neural network.

7.2 Open questions and future work

In order to work out a more detailed computational treatment for mirror neurons behaviour, additional issues have to be identified and properly addressed. Although, in our computational approach we have proposed a novel and specific functional interpretation of mirror neurons activity in relation to sensory input processing, a more detailed relation between mirror neurons spikes activity and model's variables must be provided. In particular the spike frequencies occurring within some subsets of mirror neurons might be regarded as encoding the probabilities of each sets of eigenpostures. In this way, accordingly to the Algorithm 4.1, during action execution, just one set of eigenposture S_k is selected with assigned high probability, while all others sets S_j with $j \neq k$ have very low probabilities. If the action is properly executed, the same neurons will continue to have an high spike frequency in order to assign high probability to S_k . During the observation of the same action, different sets of eigenpostures are selected with the same initial probabilities. Presumably S_k is included in this set and will have, at the end of the action, an high probability with respect to all other sets of eigenpostures. Mirror neurons encoding S_k probability will fire again giving rise to the mirror property.

The results showed in Section 6.5 are a first step towards a more systematic analysis of the model in relation to mirror activity. The preliminary results show a fast-growing of the probability associated to the set of eigenpostures of the action currently given in input to the model. Moreover this behaviour is common to almost all action instances taken into consideration. This is an encouraging result despite of the simplified scenario used for the test with a restricted number of actions and a simplified external input x .

As far as the selection mechanism is concerned, we have seen the ability of the GA model in associating the correct hand joints configurations in response to input objects. In Section 5.3.2 we have proposed a way for using GA model for the selection of sets of eigenpostures even if no tests have been performed for the modified system. Moreover the GA selection mechanisms keep into consideration only object's features while others features related to the task could be taken into consideration.

To sum up. Various open questions must be addressed in the future mostly related to the correspondence between model and mirror neurons activity and to the selection mechanism of the sets of eigenpostures.



NeGOI model specification and implementation

As discussed Section 3.4.1 the NeGOI model is composed of a sequence of three main modules: view-based module, prototypical view-invariant module and grip-aperture module. The basic idea underlying the view-based module is that extraction of position and scale invariant complex features is obtained by a hierarchical and interleaved organization of a series of processing layers. There, layers of computing units which combine simple filters into more complex ones, in order to increase pattern selectivity, are interleaved with layers based on a max operation, in order to build invariance to position and scale. The prototypical view-invariant module is composed of a layer of units combining the output of a previous layer of units that are selective to specific views of prototypical hand shapes. The key idea here is that selectivity to a specific prototypical hand shape in a view-independent way, i.e, the model selectivity and generalization properties, can be obtained by means of a Gaussian Radial Basis Function (GRBF) network (Bishop, 1995). The GRBF network is endowed with a hidden unit for each prototypical hand shape and point of view. The third module, grip-aperture module, is based on the hypothesis that a view-independent measure of grip aperture can be obtained by integrating the activity of the view-independent units in the previous module which is interpreted as providing a similarity measure between a generic hand shape and the prototypical hand shapes. Thus, given an image representing a generic hand shape during an object-directed action, the behaviours of the overall model are as follow: (1) the view-based module extracts a set of scale and position invariant complex features; (2) from these complex features the prototypical view-invariant module recognizes the three pro-

prototypical hand shapes extracting a similarity measure between input hand shape and prototypical hand shapes; finally, (3) from these similarity measures the grip-aperture module generates a measure of grip aperture. Figure 3.9 shows a schematic representation of the NeGOI model. The following subsections provide a more detailed description of how the three NeGOI modules have been implemented.

A.1 View-based module

The view-based module is organized in a hierarchical fashion and comprises a finite number of ordered levels: S1, C1, S2 and C2. Each level is composed of various computing units which receive as input the outputs of the units belonging to the previous level. The view-based module computation starts from the units belonging to the lowest level, S1, which implement simple local filters on an input graylevel image, selective to “bars” located in a specific position, and having both a specific scale and a specific orientation. The computing units of the level immediately above, C1, implement a max operation increasing the receptive field of the local filters in order to build invariance to position and scale. The S2 units combine the responses of the C1 units implementing more complex filters in order to increase pattern selectivity. Finally, the output of the C2 units, the output of the view-based module, is again a max operation over the S2 output. The output of the view-based module represents a position and scale invariant feature vector. More specifically, the S1 layer is composed of a series of Gabor filter (Daugman, 1993) with 16 different sizes (from 7×7 to 37×37 pixels with a step equal to 2 pixels) and 4 different orientations (0° , 45° , 90° and 145°). Thus, for each pixel located at position (x, y) there are 64 S1 units. The S1 layer is organized into 8 bands and the k -th band contains all the units which, independent of the orientations, correspond to Gabor filters with sizes equal to $d_k = (7 + 4k) \times (7 + 4k)$ and $d_{k+1} = (7 + 4k + 2) \times (7 + 4k + 2)$, with $k = 0, 1, \dots, 7$. The C1 units pool information coming from different S1 units having the same orientation but different position and scale. The C1 units are organized into 8 bands too. For each orientation t , the C1 units belonging to the k -th band are organized in a grid of size $M_k \times N_k$, with $M_k = \frac{M}{r_k}$ and $N_k = \frac{N}{r_k}$ where $r_k = (8 + 2k)$ is called pooling range, and $M \times N$ is the size of the input image. Let us call $s_{t,d}(x, y)$ the output of a S1 unit corresponding to a Gabor filter located at position (x, y) on the image, with size $d \times d$ and orientation equal to t° ; then the output of a C1 unit belonging to the band k and located at position (i, j) of the grid corresponding to the orientation

t is computed as follows:

$$c_{k,t}^1 = \max \{s_{d,t}^1(x, y) : (x, y) \in I_{i,j}^k, d \in \{d_k, d_{d+1}\}\}$$

where $I_{i,j}^k$ is the set of S1 units providing input to the unit $c_{k,t}^1(i, j)$ defined as follows

$$I_{i,j}^k = \{(x, y) : (i - 1) * r_k < x \leq i * r_k, (j - 1) * r_k < y \leq j * r_k\}$$

In this manner, a first step towards a position and scale invariant property is achieved by a max operation over a number of S1 units that are selective to the same simple feature and with receptive fields close to each other, and over S1 units that are selective to different sizes of the same simple feature.

The units of the S2 level are selective to more complex features than the C1 units. This is achieved by combining the simple filters implemented at level C1. In particular, each complex feature is obtained as a combination, without repetitions, of the four simple features corresponding to orientations of 0° , 45° , 90° and 135° respectively, these taken four at a time, thus obtaining 256 complex features. Let us represent each complex feature as a 4-tuple $CF^h = (t_0^h, t_1^h, t_2^h, t_3^h)$ with $h \in 1, 2, \dots, 256$ in and $t_m^h \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$.

Again, level S2 is organized in 8 bands. The k -th S2 band is organized in 256 grids of sizes $(M_k - 1) \times (N_k - 1)$. The units belonging to a specific grid h are selective to the specific complex feature CF^h at different scales and positions on the input image. More formally, the output of the S2 units belonging to the k -th band and the h -th grid at the location (i, j) is computed by a Gaussian function as follows:

$$\begin{aligned} s_{k,h}^2(i, j) = & \exp \left(- \left(\left(c_{k,t_0^h}^1(i, j) - 1 \right)^2 + \left(c_{k,t_1^h}^1(i, j + 1) - 1 \right)^2 \right) + \right. \\ & \left. + \left(c_{k,t_2^h}^1(i + 1, j) - 1 \right)^2 + \left(c_{k,t_3^h}^1(i + 1, j + 1) - 1 \right)^2 / 2\sigma^2 \right) \end{aligned}$$

where $(t_0^h, t_1^h, t_2^h, t_3^h) = CF^h$.

Level C2, just as level C1, increases scale and position invariance while preserving selectivity to the complex features extracted in the S2 level. This level is composed of 256 units. Let us call c_h^2 the output of the C2 unit selective to the complex feature CF^h , with $h = 1, 2, \dots, 256$. This output is computed as follows:

A.2. PROTOTYPICAL VIEW-INVARIANT MODULE

$$c_h^2 = \max_{i,j,k} \{ (s_{k,h}^2(i,j)) \}$$

Hence, the output of the view-based module is a position and scale invariant feature vector of size equal to 256.

A.2 Prototypical view-invariant module

The prototypical view-invariant module is made up of two layers. The first layer is composed of three ordered groups of units receiving as input from the view-based module the scale and position independent feature vector. Each group is composed of N ordered units. Let VDP_{ij} be the j -th unit belonging to i -th group, with $i = 1, 2, 3$ and $j = 1, 2, \dots, N$. Each VDP_{ij} unit is scale and position independent; it is, however, selective to both prototypical hand shape PHS_i and viewpoint j . The second layer is composed of three viewpoint-independent units selective to the three prototypical hand shapes. Let VIP_i be the i -th unit of the second layer, with $i = 1, 2, 3$. The unit VIP_i receives connections from units belonging to i -th group of the first layer only. Each unit VIP_i is selective to the prototypical hand shape PHS_i but is viewpoint independent.

The units of the first layer are the hidden neurons of a Gaussian Radial Basis Function neural network.

There are nine hidden neurons ($N = 9$) only, i.e., three neurons for each selected prototypical hand shape: VDP_{1j} selective to both fully opened grip aperture and viewpoint $view_j$, GV_{2j} selective to both middle size grip aperture and viewpoint $view_j$, GV_{3j} selective to both fully closed grip aperture and viewpoint $view_j$, with $j = 1, 2, 3$. The $view_1$, $view_2$ and $view_3$ viewpoints are sample viewpoints differing from each other by about 22° (see Figure A.1). These viewpoints correspond to rotations of about 22° of a camera around an axis Z (perpendicular to the surface of a table and centered on the target object).

In the second layer, the VIP_i neurons ($i = 1, 2, 3$) compute a linear combination of the outputs of VDP_{ij} , with $j = 1, 2, 3$. Therefore, the VIP_1 , VIP_2 and VIP_3 neurons, once trained, are selective to fully opened, middle size and fully closed grip aperture, respectively (see Figure 3.8).

Both VDP_{ij} and VIP_i neurons were trained using three different training sets. Each set, compose of 450 frames of the three prototypical hand shapes recorded from the $view_j$ viewpoint, comprises 150 frames representing fully opened grip aperture, 150 frames representing middle size grip aperture, and 150 frames representing fully closed grip aperture.

A.2. PROTOTYPICAL VIEW-INVARIANT MODULE

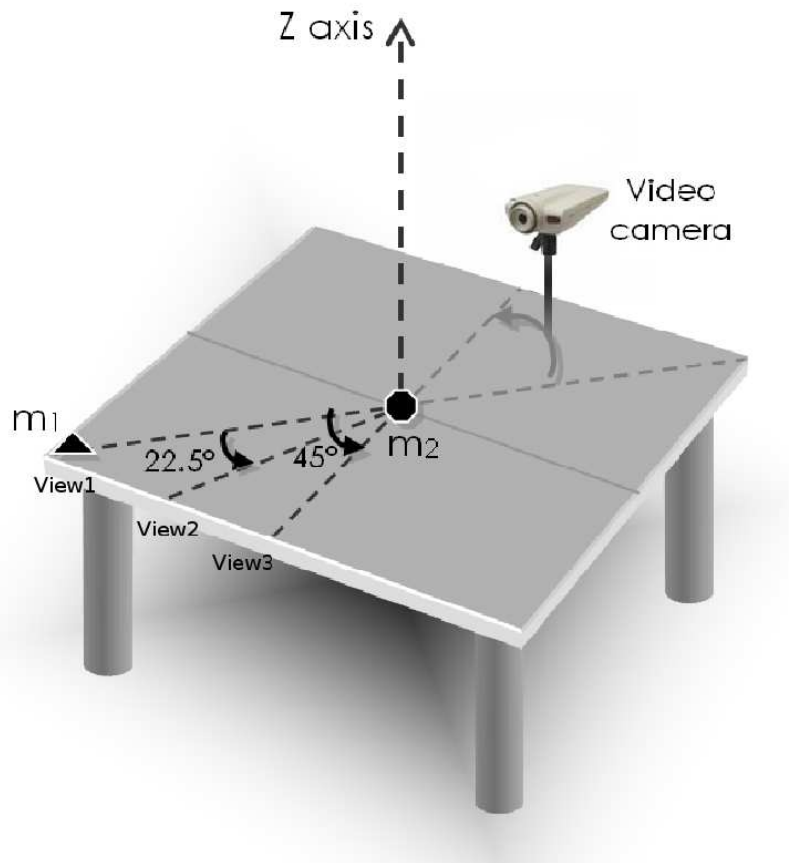


Figure A.1: Experimental setting. A subject was seated at a table with two clearly visible surface marks (m_1 and m_2) placed at a distance of roughly 40cm from each other. For each target object, a subject was asked to position the right hand on starting position m_1 , and to reach and grasp the target object placed on mark m_2 . Each action was recorded using a camera placed at a fixed distance (roughly 70cm) from the target and at a fixed height (roughly 50cm) from table surface. The camera is able to rotate around the Z axis.

In order to increase model tolerance to viewpoint variations it is sufficient to augment the model by adding into each group of the first layer further view-invariant units that are selective for different viewpoints which are separated from each to other by about 22° .

A.3 Grip-aperture module

The grip-aperture module consists of a RBF network composed of a number of hidden nodes (in our implementation the number of hidden nodes is equal to two) and one output node. The output node is the unit GA . This module receives inputs from every VIP_i in the prototypical view-invariant module. The output of GA is a scale, position, and viewpoint independent value belonging to the interval $[0, 1]$. The GA unit was trained under the hypothesis that the output of the VIP_1 , VIP_2 and VIP_3 units are Gaussian centered on fully opened grip aperture, middle size grip aperture, and fully closed grip aperture, respectively.

A.3.1 Experimental setting

A subject was asked to perform a variety of reach-to-grasp actions. These actions were carried out while the subject was seated at a table with two clearly visible surface marks (m_1 and m_2) placed at a distance of roughly $40cm$ from each other: each reach-to-grasp action starts at m_1 and ends at m_2 (Figure A.1). For each target object, the subject was asked to position the right hand on starting position m_1 , and to reach and grasp the target object placed on mark m_2 . Each action was recorded using a camera placed at a fixed distance (roughly $70cm$) from the target and at a fixed height (roughly $50cm$) from table surface. As mentioned above, this experimental setting enables one to rotate the camera around the axis Z perpendicular to table surface and centered on target object.

Accordingly, each action is represented as a sequence of frames (160×160 pixels). Each frame is a gray-level image. This gray-level image is relayed as input to the NeGOI system.

Mixture Density Networks

Given a set of unlabeled data $T = \{\mathbf{t}^i\}_{i=1,\dots,N}$ with $\mathbf{t}^i \in \mathbb{R}^c$, a powerful, general framework for modelling unconditional distributions $p(\mathbf{t})$, makes use of mixture models of the form:

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

where M is the number of components in the mixture. The parameters α_i are called *mixing coefficients*, and can be regarded as prior probabilities of the data vector \mathbf{t} having been generated from the i -th component of the mixture. The kernel functions are usually chosen Gaussian of the form:

$$\phi_i(\mathbf{t}) = \frac{1}{(2\pi)^{c/2} \sigma_i^c} \exp \left\{ -\frac{\|\mathbf{t} - \boldsymbol{\mu}_i\|^2}{2\sigma_i^2} \right\} \quad (\text{B.2})$$

where c is the dimension of the data vector \mathbf{t} , the vector $\boldsymbol{\mu}_i$ represents the center of the i -th kernel, with components μ_{ik} , while σ_i^2 represents the variance. A maximum likelihood approach can be pursued for the determination of model's parameters (Redner and Walker, 1984).

Consider now a set of labeled data $T = \{\mathbf{x}^i, \mathbf{t}^i\}_{i=1,\dots,N}$, mixture models can be used to approximate conditional distributions $p(\mathbf{t}|\mathbf{x})$ if we allow the mixture's parameters to be function of \mathbf{x} , that is:

$$p(\mathbf{t}|\mathbf{x}) = \sum_{i=1}^M \alpha_i(\mathbf{x}) \phi_i(\mathbf{t}|\mathbf{x}) \quad (\text{B.3})$$

again m is the number of components in the mixture, the parameters $\alpha_i(\mathbf{x})$ are the *mixing coefficients*, and can be regarded as prior probabilities,

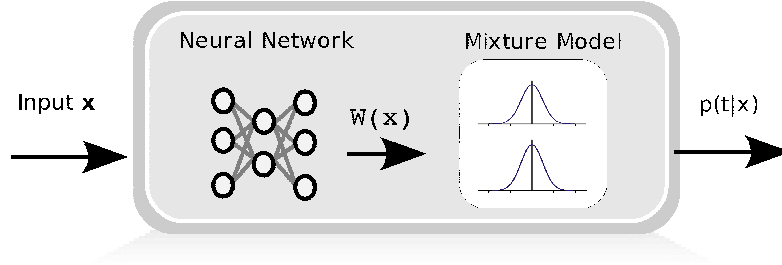


Figure B.1: A Mixture Density Network (MDN) is a general framework to estimate conditional distributions $p(\mathbf{t}|\mathbf{x})$. It is composed of a universal approximator, in particular a two layer feed-forward neural network, and a mixture model. The neural network is used to learn the relation between input vector x and mixture model parameters $W(\mathbf{x}) \equiv (\alpha_1(\mathbf{x}), \dots, \alpha_M(\mathbf{x}), \boldsymbol{\mu}_1(\mathbf{x}), \dots, \boldsymbol{\mu}_M(\mathbf{x}), \sigma_1(\mathbf{x}), \dots, \sigma_M(\mathbf{x}))$. These parameters are feed in input to the mixture model in order to estimate $p(\mathbf{t}|\mathbf{x})$.

conditional on \mathbf{x} , of the target vector \mathbf{t} having been generated from the i -th component of the mixture. Note that in this case the mixing coefficients α_i as well as the kernel functions parameters $\boldsymbol{\mu}_i$ and σ_i are function of the input vector \mathbf{x} . Also in this case the kernel functions are usually chosen Gaussian of the form:

$$\phi_i(\mathbf{t}|\mathbf{x}) = \frac{1}{(2\pi)^{c/2}\sigma_i^c(\mathbf{x})} \exp \left\{ -\frac{\|\mathbf{t} - \boldsymbol{\mu}_i(\mathbf{x})\|^2}{2\sigma_i^2(\mathbf{x})} \right\} \quad (\text{B.4})$$

The relation between \mathbf{x} and mixture parameters can be learned in a supervised fashion by universal approximator, for example a two layer feedforward neural networks with non-linear hidden units, thus leading to a combined structure such that showed in Figure B.1 called **Mixture Density Networks (MDN)** (Bishop, 1995, 2006).

By choosing a mixture model with a sufficient number of kernel functions, and a neural network with a sufficient number of hidden units, the MDN can approximate as closely as desired any conditional density $p(\mathbf{t}|\mathbf{x})$.

The neural network component of the MDN framework can be any standard feed-forward structure with universal approximation capabilities.

Before proceeding to the MDN model's description in a more depth, we will give a brief introduction to feed-forward neural networks.

B.1 Feed-forward neural networks

A graphical representation of Feed-Forward multi-layered neural networks (from now on FFW) is shown in Figure B.1. It is composed of successive layers of elementary units. Each units compute its input and its activation value by means of an *activation function*. A unit can receive and send connection to other units.

In the feed-forward multi-layered neural networks the units are organized in layers and a unit of layer can send connection only to units of the next layer.

The units of the first layer (represented bottom in Figure B.1) and called input units do not compute its inputs but simply have an activation, indicated with x_i with $i = 1, \dots, d$, equal to input currently given to the network.

The units of the last layer (represented top in Figure B.1) are called outputs units and related activation is indicated with y_i with $i = 1, \dots, c$. The units of the other layers are called hidden units. In the next we will always consider networks with only one hidden layer whose activation will be indicated with z_i with $i = 1, \dots, s$. It has been shown that such networks can approximate arbitrarily well any functional continuous mapping from one finite-dimensional space to another, provide the number of hidden units is sufficiently large (Funahashi, 1989). At each connection between unit i and unit j it is associated a weight $w_{ji} \in \mathbb{R}$. We will indicate with $w_{ji}^{(1)}$ the weight connecting the input unit i with the hidden unit j while with $w_{kj}^{(2)}$ the weight connecting the hidden unit j with the output unit k .

We can now write the analytic function computed by the network.

Each hidden units compute its inputs a_j as follows:

$$a_j = \sum_{i=1}^d w_{ji}^{(1)} x_i + w_{j0}^{(1)} \quad (\text{B.5})$$

where $w_{j0}^{(1)}$ is the *bias* for the hidden unit j . We can include these parameters in the sum of B.5 by the inclusion of an extra input variable x_0 , whose activation is fixed to one as shown in Figure B.1, thus leading to the following expression for a_j :

$$a_j = \sum_{i=0}^d w_{ji}^{(1)} x_i \quad (\text{B.6})$$

B.1. FEED-FORWARD NEURAL NETWORKS

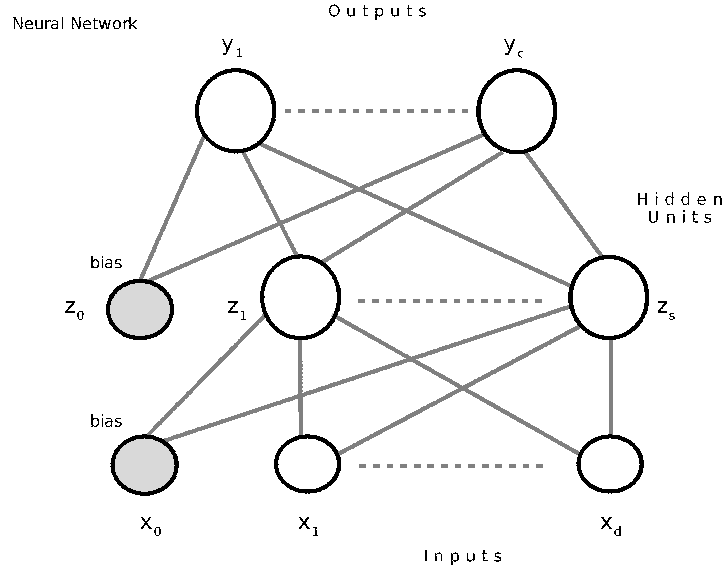


Figure B.2: Feed-forward neural network with two layers of adaptive weights. The bias parameters of the first and second layer are shown as weights from an extra input or hidden unit with activation fixed to value 1.

The activation of hidden unit j is obtained by applying an activation function $g(\cdot)$ to a_j :

$$z_j = g(a_j) \quad (\text{B.7})$$

where $g(\cdot)$ is usually chosen as logistic sigmoid of the form:

$$g(a) \equiv \frac{1}{1 + \exp(-a)} \quad (\text{B.8})$$

The outputs units again construct a linear combination of its inputs (that is the activations of the hidden units):

$$a_k = \sum_{j=1}^s w_{kj}^{(2)} z_j + w_{k0}^{(2)} \quad (\text{B.9})$$

Again, we can absorb the bias into the weights, by including an extra hidden unit with activation $z_0 = 1$, to give:

$$a_k = \sum_{j=0}^s w_{kj}^{(2)} z_j \quad (\text{B.10})$$

The activation of the k -th output unit is the obtained as follows:

$$y_k = \tilde{g}(a_k) \quad (\text{B.11})$$

where $\tilde{g}(\cdot)$ is the activation function of the output layer usually chosen as linear function.

So the computation of the network can be expressed as follows:

$$y_k = \tilde{g} \left(\sum_{j=0}^s w_{kj}^{(2)} g \left(\sum_{i=0}^d w_{ji}^{(1)} x_i \right) \right) \quad (\text{B.12})$$

B.2 Learning algorithm

The estimation of MDN or FFW model's parameters take place by minimizing a convenient error function usually derived from the maximum likelihood principle.

In order to minimize the error function we can use standard gradient descent algorithm with related optimization techniques to improve convergence performance.

Broadly speaking gradient descent algorithm works as follows: it starts with an initial guess for the model parameters (e.g. random), we can group all of them in the vector $\Theta^{(0)}$, then it iteratively updates model parameters such that, at each step, say τ , it moves a short distance in the direction of the greatest rate of decrease of a pre-fixed error function $E(\Theta)$, i.e. in the direction of the negative gradient, evaluated at $\Theta^{(\tau)}$:

$$\Delta\Theta^{(\tau)} = -\eta \nabla E |_{\Theta^{(\tau)}}$$

In order to actually implement a gradient descent algorithm it is needed a differentiable error function and a way to efficiently calculate the gradient of E .

In the case of standard feed-forward neural network the back-propagation algorithm exists as a way to efficiently compute the gradient of differentiable error functions. We will see that such algorithm apply to both MDN model.

First of all we briefly describe general back-propagation algorithm for a general network having feed-forward topology and differentiable non-linear activation functions.

B.3. BACK-PROPAGATION IN FEED-FORWARD NETWORKS

B.3 Back-propagation in feed-forward networks

Many error functions, such as that we will use with MDN model, comprise a sum of terms one for each data point in the training set so that:

$$E(W) = \sum_{n=1}^N E^n(W) \quad (\text{B.13})$$

Thus our objective is to evaluate $\nabla E^n(W)$.

In a general feed-forward network, each unit computes a weighted sum of its inputs of the form:

$$a_j = \sum_i w_{ji} z_i \quad (\text{B.14})$$

where z_i is the activation of a unit, or input, which sends a connection to unit j , while w_{ji} is the weight associated to the connection between unit i and unit j .

The activation of a generic unit z_j is obtained by applying a non-linear activation function $g(\cdot)$ to a_j , that is:

$$z_j = g(a_j) \quad (\text{B.15})$$

It is important to note that variables z_i in B.14 could be inputs while unit z_j could be an output.

Consider now the evaluation of the derivative of E^n with respect to a generic weight w_{ji} .

E^n depends on w_{ji} only through the summed input a_j to unit j . So by applying the chain rule for partial derivative we obtain:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E^n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} \quad (\text{B.16})$$

We now define δ_j as:

$$\delta_j \equiv \frac{\partial E^n}{\partial a_j} \quad (\text{B.17})$$

The second term of the right hand side of B.16 is simply:

$$\frac{\partial a_j}{\partial w_{ji}} = z_i \quad (\text{B.18})$$

Substituting B.18 and B.17 in B.16 we obtain:

B.3. BACK-PROPAGATION IN FEED-FORWARD NETWORKS

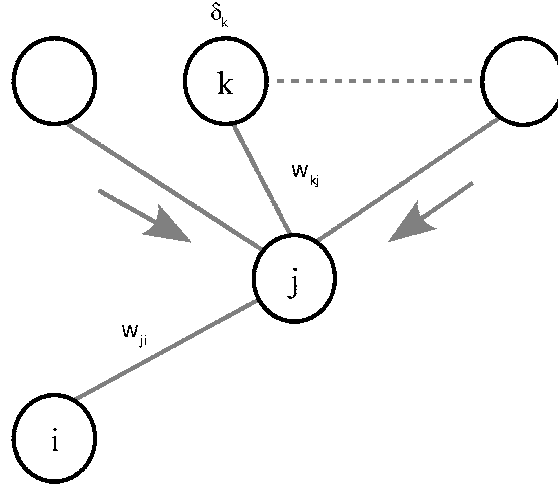


Figure B.3: Illustration of the calculation of δ_j for hidden unit j by back-propagation of the δ 's from those units k to which unit j sends connections.

$$\frac{\partial E}{\partial w_{ji}} = \delta_j z_i \quad (\text{B.19})$$

This means that the evaluation of the derivative of E^n with respect to the weight w_{ji} is given by multiplying the value of δ of the unit j for the output z_i of the unit i .

For the output units the evaluation of δ_k is straightforward:

$$\delta_k \equiv \frac{\partial E}{\partial a_k} = g'(a_k) \frac{\partial E^n}{\partial y_k} \quad (\text{B.20})$$

where we have used equation B.14 and we have called y_k the z_k because in this case z_k are output units.

In order to obtain the evaluation of B.20 we must substitute the appropriate expression of $g'(a)$ and $\frac{\partial E^n}{\partial y}$. This will be the main discussion for the MDN model.

For the hidden units the expression for δ_j takes the form:

$$\delta_j \equiv \frac{\partial E^n}{\partial a_j} = \sum_k \frac{\partial E^n}{\partial a_k} \frac{\partial a_k}{\partial a_j} \quad (\text{B.21})$$

where the sum runs over all units k to which units j sends connections. This is schematically illustrated in Figure B.3.

Now note that in the expression B.21 $\frac{\partial E^n}{\partial a_k}$ is simply the definition of δ_k . The second term of the sum in the right hand side of B.21 is for every k :

B.3. BACK-PROPAGATION IN FEED-FORWARD NETWORKS

$$\frac{\partial a_k}{\partial a_j} = g'(a_j)w_{kj} \quad (\text{B.22})$$

substituting in B.21 we finally obtain the expression for δ_j :

$$\delta_j \equiv \frac{\partial E^n}{\partial a_j} = g'(a_j) \sum_k \delta_k w_{kj} \quad (\text{B.23})$$

It is important to observe the presence of δ_k terms in the equation B.23. This means that the computation of the gradient of E proceeds in the following way: starting by the output layer we compute the δ_k terms and so the expression of the derivative with respect to the corresponding weights parameters; we then *back-propagate* the δ_k terms to the higher layer in order to evaluate the derivative for the corresponding weights.

B.3.1 Back-propagation for sum-of-squares error function and sigmoid activation function

The back-propagation algorithm illustrated in the previous section is related to general forms of differentiable error functions with respect to the network outputs, differentiable activation functions and feed-forward network topology. In this section we will briefly show such algorithm in the particular case of sum-of-squares error function, logistic sigmoid activation function for hidden units and identity activation function for output units.

The sum-of-square error function for the input n -th takes the form:

$$E^n = \frac{1}{2} \sum_{k=1}^c (y_k - t_k)^2 \quad (\text{B.24})$$

so the B.20 becomes:

$$\delta_k = y_k - t_k \quad (\text{B.25})$$

Now note that the derivative of the sigmoid activation function of hidden units expressed in B.8 is simply:

$$g'(a) = g(a)(1 - g(a)) \quad (\text{B.26})$$

so the B.23 becomes:

$$\delta_j = g(a_j)(1 - g(a_j)) \sum_k w_{kj} \delta_k \quad (\text{B.27})$$

B.4. MIXTURE DENSITY NETWORKS

but $g(a_j)$ is simply z_j and so:

$$\delta_j = z_j(1 - z_j) \sum_k w_{kj} \delta_k \quad (\text{B.28})$$

The back-propagation algorithm is summarized in Table B.1.

B.4 Mixture Density Networks

In an MDN with M kernel function, the network component will have M output units, denoted with z_j^α , for the mixing coefficients $\alpha_j(\mathbf{x})$, M output units, denoted with z_j^σ , for the kernel width $\sigma_j(\mathbf{x})$, and $M \times c$ output units, denoted with z_{jk}^μ , for the kernel centers $\boldsymbol{\mu}_j$ with components μ_{jk} . So the network will have $(c + 2) \times M$ output units.

In order to ensure that the mixing coefficients $\alpha_i(\mathbf{x})$ can be interpreted as probabilities and to ensure that the distribution is correctly normalized ($\int p(\mathbf{t}|\mathbf{x})d\mathbf{t} = 1$), they must satisfy the constraints:

$$\sum_{i=1}^M \alpha_i(\mathbf{x}) = 1 \quad (\text{B.29})$$

$$0 \leq \alpha_j(\mathbf{x}) \leq 1 \quad (\text{B.30})$$

This is obtained by relating the coefficients $\alpha_i(\mathbf{x})$ with the networks outputs by a “softmax” function:

$$\alpha_i = \frac{\exp(z_i^\alpha)}{\sum_{j=1}^M \exp(z_j^\alpha)} \quad (\text{B.31})$$

Moreover it is convenient to represent the variances σ_j in terms of the exponentials of the corresponding network outputs:

$$\sigma_j = \exp(z_j^\sigma) \quad (\text{B.32})$$

This help to avoid pathological configurations in which one or more of the variances goes to zero, since this would require the corresponding $z_j^\sigma \rightarrow \infty$.

Finally the centers $\boldsymbol{\mu}_j$ are simple related to the corresponding network outputs:

$$\mu_{jk} = z_{jk}^\mu \quad (\text{B.33})$$

B.4. MIXTURE DENSITY NETWORKS

B.4.1 Back-propagation in Mixture Density Network

In this section we derive the back-propagation algorithm for the Mixture Density Network in the case of a specific error function. As said above, in order to have an effective expression for the back-propagation algorithm we need to define an error function and the expressions for the derivatives of the error with respect to the output of the network, that is we must specify the terms of the equation B.20.

As error function we will use the following, derived from the likelihood principle:

$$E = - \sum_n \ln \left\{ \sum_{j=1}^M \alpha_j(\mathbf{x}^n) \phi_j(\mathbf{t}^n | \mathbf{x}^n) \right\} \quad (\text{B.34})$$

Since such error function is composed of a sum of terms $E = \sum_n E^n$, one for each input, we can consider the derivatives $\delta_k^n = \frac{\partial E^n}{\partial z_k}$ for a particular input n and then find the derivatives of E by summing over all inputs.

Note that the output units of the MDN have linear activation functions, $g(a) = a$, so the quantities δ_k^n can also be written as $\frac{\partial E^n}{\partial a_k}$.

The ϕ_j can be regarded as conditional density functions, with prior probabilities α_j . It is convenient to introduce the corresponding posterior probabilities in order to have some simplification of the subsequent analysis:

$$\pi_j(\mathbf{x}, \mathbf{t}) = \frac{\alpha_j \phi_j}{\sum_{l=1}^M \alpha_l \phi_l} \quad (\text{B.35})$$

such probabilities sum to unity, that is:

$$\sum_{j=1}^M \pi_j = 1 \quad (\text{B.36})$$

We show now the form of the derivative of E with respect to each type of network outputs z_j^α , z_j^σ and z_j^μ . We start by considering the derivatives of E^n with respect to those network outputs which correspond to the mixing coefficients α_j .

First of all note that the error E^n depends on the outputs z_j^α by means of the relation B.31. Moreover as a result of the softmax transformation, the value of α_k depends on all the network outputs which contribute to the mixing coefficients.

From the chain rule and taking into account the contribution of all z_j^α we can write:

B.4. MIXTURE DENSITY NETWORKS

$$\frac{\partial E^n}{\partial z_j^\alpha} = \sum_k \frac{\partial E^n}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial z_j^\alpha} \quad (\text{B.37})$$

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

$$\frac{\partial E^n}{\partial \alpha_k} = - \frac{\phi_k(\mathbf{t}|\mathbf{x})}{\sum_{j=1}^M \alpha_j(\mathbf{x}) \phi_j(\mathbf{t}|\mathbf{x})} = - \frac{\pi_k}{\alpha_k} \quad (\text{B.38})$$

Where we have used the relation B.35.

The second term of the sum in the right hand side is:

$$\begin{aligned} \frac{\partial \alpha_k}{\partial z_j} &= \frac{\delta_{jk} \exp(z_k^\alpha) \sum_{l=1}^M \exp(z_l^\alpha) - \exp(z_j^\alpha) \exp(z_k^\alpha)}{\left[\sum_{l=1}^M \exp(z_l^\alpha) \right]^2} \\ &= \delta_{jk} \alpha_k - \frac{\exp(z_j^\alpha)}{\sum_{l=1}^M \exp(z_l^\alpha)} \cdot \frac{\exp(z_k^\alpha)}{\sum_{l=1}^M \exp(z_l^\alpha)} \\ &= \delta_{jk} \alpha_k - \alpha_j \alpha_k \end{aligned}$$

So we have obtained:

$$\frac{\partial \alpha_k}{\partial z_j} = \delta_{jk} \alpha_k - \alpha_j \alpha_k \quad (\text{B.39})$$

Where δ_{jk} is Kronecker delta symbol defined as $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ otherwise.

Substituting B.38 and B.39 into B.37 and making use of B.36 we obtain:

$$\begin{aligned} \frac{\partial E^n}{\partial z_j^\alpha} &= \sum_k \frac{\partial E^n}{\partial \alpha_k} \frac{\partial \alpha_k}{\partial z_j^\alpha} \\ &= \sum_k - \frac{\pi_k}{\alpha_k} (\delta_{jk} \alpha_k - \alpha_j \alpha_k) \\ &= - \frac{\pi_j}{\alpha_j} \alpha_j + \alpha_j \sum_k \alpha_k \end{aligned}$$

that is:

$$\frac{\partial E^n}{\partial z_j^\alpha} = \alpha_j - \pi_j \quad (\text{B.40})$$

For the derivatives corresponding to the σ_j parameters again remember that E^n depends on z_j only through the relation B.32 and so from the chain rule:

B.4. MIXTURE DENSITY NETWORKS

$$\frac{\partial E^n}{\partial z_j^\sigma} = \frac{\partial E^n}{\partial \sigma_j} \frac{\partial \sigma_j}{\partial z_j^\sigma} \quad (\text{B.41})$$

in order to compute the term $\frac{\partial E^n}{\partial \sigma_j}$ we make use of B.34, B.35 and B.4:

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

So we have obtained:

$$\frac{\partial E^n}{\partial \sigma_j} = -\pi_j \left\{ \frac{\|\mathbf{t} - \boldsymbol{\mu}_j\|^2}{\sigma_j^3} - \frac{c}{\sigma_j} \right\} \quad (\text{B.42})$$

The term $\frac{\partial \sigma_j}{\partial z_j^\sigma}$ is simple:

$$\frac{\partial \sigma_j}{\partial z_j^\sigma} = \sigma_j \quad (\text{B.43})$$

Substituting B.42 and B.43 into B.41 we obtain:

$$\frac{\partial E^n}{\partial z_j^\sigma} = -\pi_j \left\{ \frac{\|\mathbf{t} - \boldsymbol{\mu}_j\|^2}{\sigma_j^2} - c \right\} \quad (\text{B.44})$$

Finally, using B.34, B.35 and B.4, and taking into account that the parameters μ_{jk} are given directly by the z_{jk}^μ network outputs, we have:

$$\begin{aligned} \frac{\partial E^n}{\partial z_{jk}^\mu} &= \frac{\alpha_j}{\sum_{j=1}^M \alpha_j \phi_j} \cdot \frac{\exp\left\{-\frac{\|\mathbf{t} - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right\}}{(2\pi)^{\frac{c}{2}} \sigma_j^c} \cdot \frac{(\mu_{jk} - t_k)}{\sigma_j^2} \\ &= \pi_j \left\{ \frac{\mu_{jk} - t_k}{\sigma_j^2} \right\} \\ &\quad \frac{\partial E^n}{\partial z_{jk}^\mu} = \pi_j \left\{ \frac{\mu_{jk} - t_k}{\sigma_j^2} \right\} \end{aligned} \quad (\text{B.45})$$

B.4. MIXTURE DENSITY NETWORKS

Algorithm B.1 Back-propagation procedure

INPUT An input-target couple $\{\mathbf{x}^n, \mathbf{t}^n\}$
input-hidden weights matrix $W1$, hidden-output weights matrix $W2$

OUTPUT derivative of E^n with respect to input-hidden weights $derivative_1$
derivative of E^n with respect to hidden-output weights $derivative_2$

```
1: function BACK-PROPAGATION( $\mathbf{x}^n, \mathbf{t}^n, W1, W2$ )
     $\triangleright$  Computing hidden units input and activation
2:   for  $j \leftarrow 1$  to  $s$  do
3:      $a[j] \leftarrow \text{computeHiddenUnitInput}(j, x, W1)$ 
4:      $z[j] \leftarrow \text{sigmoid}(a[j])$ 
5:   end for
     $\triangleright$  Computing network output and related delta
6:   for  $k \leftarrow 1$  to  $c$  do
7:      $y[k] \leftarrow \text{computeNetworkOutput}(k, z, W2)$ 
8:      $delta_{out}[k] \leftarrow t[k] - y[k]$ 
9:   end for
     $\triangleright$  Computing delta for hidden units
10:  for  $j \leftarrow 1$  to  $s$  do
11:     $delta_{hidden}[j] \leftarrow 0$ 
12:    for  $k \leftarrow 1$  to  $c$  do
13:       $delta_{hidden}[j] \leftarrow delta_{hidden}[j] + delta_{out}[k] * W2[k][j]$ 
14:    end for
15:     $delta_{hidden}[j] \leftarrow z[j] * (1 - z[j]) * delta_{hidden}[j]$ 
16:  end for
     $\triangleright$  Computing derivative with respect to input-hidden weights
17:  for  $j \leftarrow 1$  to  $s$  do
18:    for  $i \leftarrow 1$  to  $d$  do
19:       $derivative_1[j][i] \leftarrow delta_{hidden}[j] * x[i]$ 
20:    end for
21:  end for
     $\triangleright$  Computing derivative with respect to hidden-output weights
22:  for  $k \leftarrow 1$  to  $c$  do
23:    for  $j \leftarrow 1$  to  $s$  do
24:       $derivative_2[k][j] \leftarrow delta_{out}[k] * z[j]$ 
25:    end for
26:  end for
27:  return  $derivative_1, derivative_2$ 
28: end function
```

B.5 Implementation and Test

Both FFW network and MDN have been implemented in Matlab. The FFW network weights and the weights of the MDN's network component have been initialized on the basis of Nguyen-Widrow method (Nguyen and Widrow, 1990). In order to speed up the learning process the Rprop (Riedmiller, 1994) variant of the gradient descent algorithm have been implemented.

B.5.1 test 1: a simple uni-dimensional example

As first example of MDN capability we consider the problem of approximate a one-dimensional multi-valued function as described in Bishop 1996.

Multi-valued functions are typical in inverse problems such as robot kinematics. For such problems there exists a well-defined forward problem which can be described by a functional, single-valued, mapping. However for the same problem the inverse mapping can be often multi-valued.

Suppose that the forward relation between x and t is given by:

$$t = x + 0.3\sin(2\pi x) + \epsilon \tag{B.46}$$

with $x \in [0, 1]$ and ϵ is a random variable drawn from a uniform distribution in the range $[-0.1, 0.1]$.

In Figure B.4 it is shown a data set obtained from B.46 together with a FFW mapping. It can be noted how the network approximate the conditional average of target data $\langle p(t|x) \rangle$ giving rise to a good representation of the function from which the data was generated.

Consider now the inverse problem obtained by interchanging the roles of input and output of data in Figure B.4. In this case, as Figure B.5 shows, the network mapping gives a very poor fit to the data, as it again tries to represent the conditional average of the target values.

The MDN model is also applied in this case with five hidden units for the network component and three kernels for the mixture.

In Figure B.6 is shown contour plot of $p(t|x)$ as estimate by MDN while in Figure B.7 it is shown the plot of the priors $\alpha_j(x)$ as function of x for the

B.5. IMPLEMENTATION AND TEST

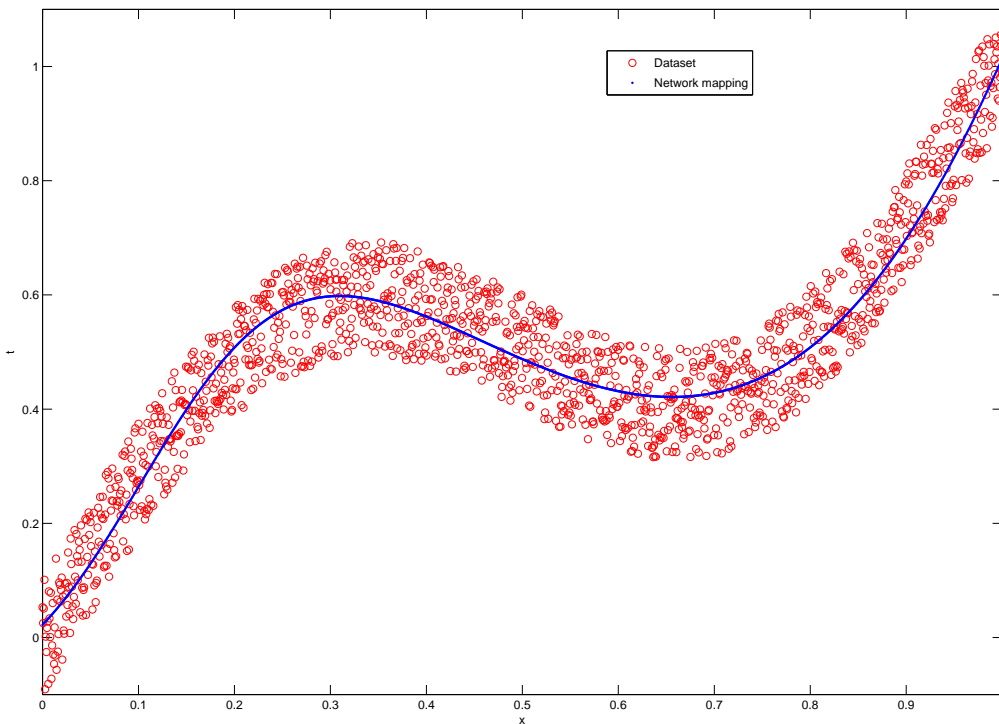


Figure B.4: Red bulled are points obtained from equation B.46. The underlying relation between x and t is a single-valued function and can be easily approximate by standard neural networks. Note how network mapping for a any given x , in blue, approximate the conditional average of target data $\langle p(t|x) \rangle$.

B.5. IMPLEMENTATION AND TEST

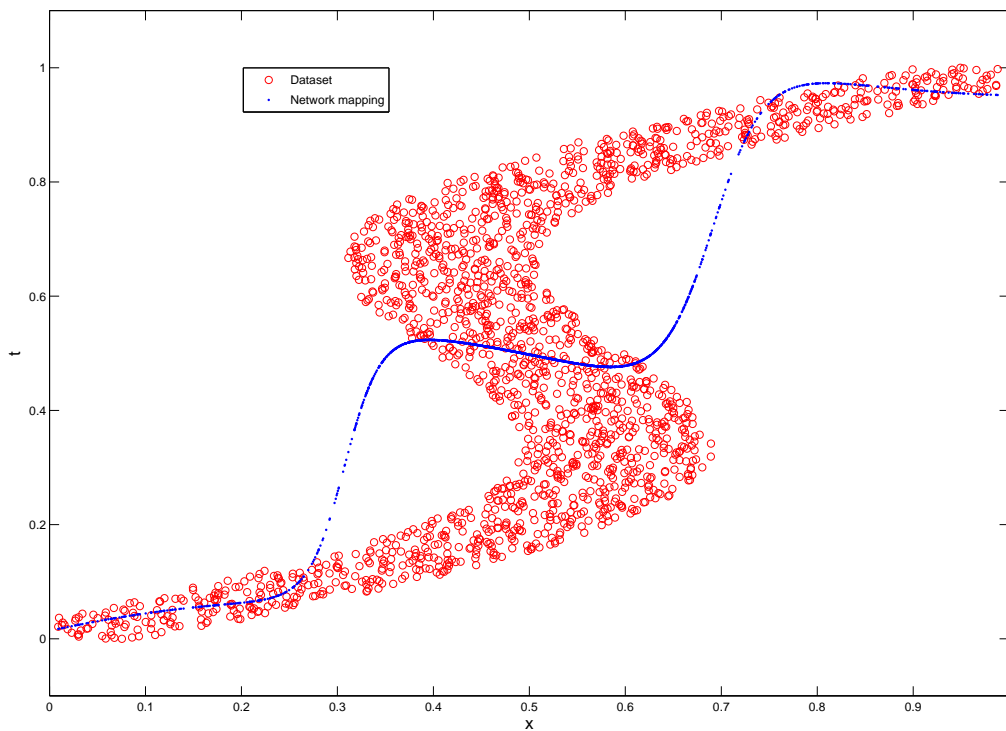


Figure B.5: The underlying relation between x and t is a multi-valued function. In this case standard neural networks give a very poor description of the data since it again try to approximate conditional average of t 's given x .

B.5. IMPLEMENTATION AND TEST

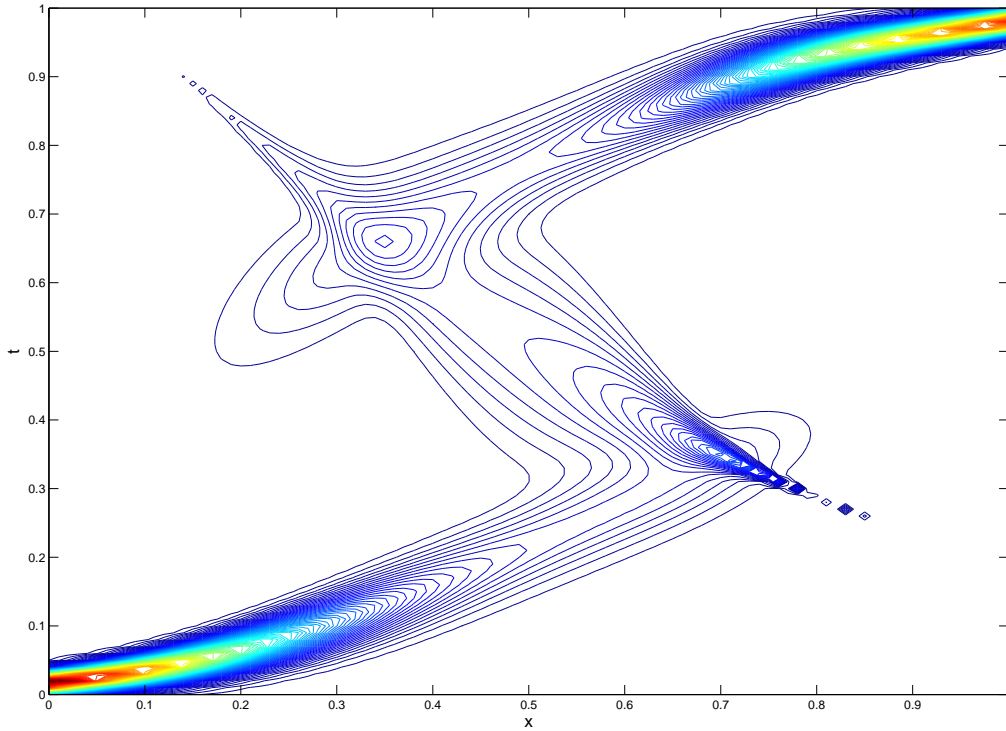


Figure B.6: Contour plot of conditional distribution $p(t|x)$ as estimated by Mixture Density Network.

three kernel functions. As can be seen the MDN uses only one kernel when the mapping is single-valued and more kernels otherwise. Moreover in Figure B.8 it is shown how each kernel contributes to overall estimation of $p(t|x)$.

In some cases we are interested in finding an output value for every given input value x (for example in the case of control problems). If the components of the distribution are well separated and have negligible overlap we can easily find the most probable branch and associate with x the related central value. In fact since each component of the mixture model is normalized $\int \phi_j(\mathbf{t}|\mathbf{x})d\mathbf{t} = 1$ then the most probable branch of the solution is given by:

$$\arg \max_j \{\alpha_j(\mathbf{x})\} \quad (\text{B.47})$$

In Figure B.9 it is shown the plot of the central value of the most prob-

B.5. IMPLEMENTATION AND TEST

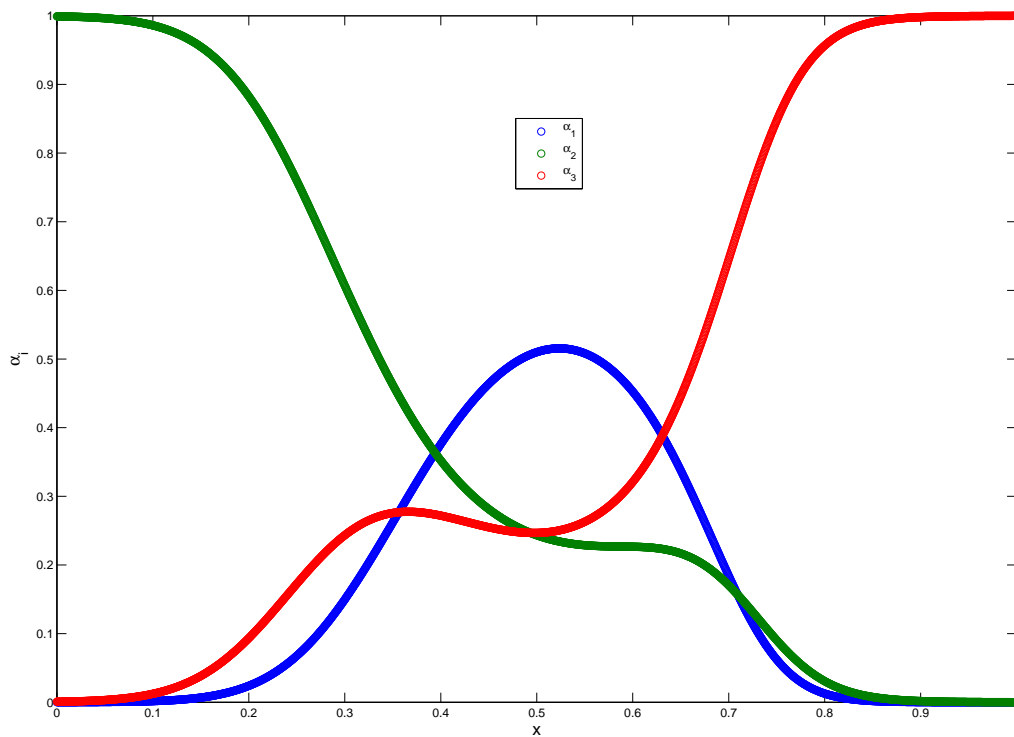


Figure B.7: Progress of prior probabilities α_1, α_2 and α_3 as function of x . Note that where the function is single-valued only one kernel have prior probability different from zero.

B.5. IMPLEMENTATION AND TEST

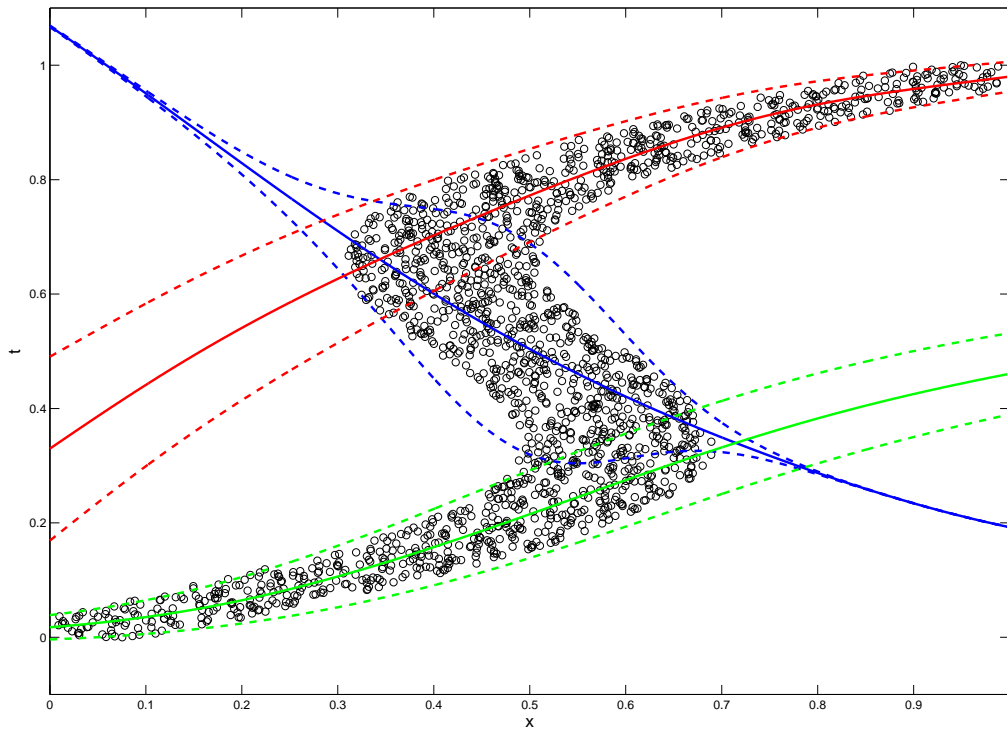


Figure B.8: Plot of central values μ_1, μ_2 and μ_3 as function of x together with relative standard deviation $\mu_1 \pm \sigma_1, \mu_2 \pm \sigma_2$ and $\mu_3 \pm \sigma_3$.

B.5. IMPLEMENTATION AND TEST

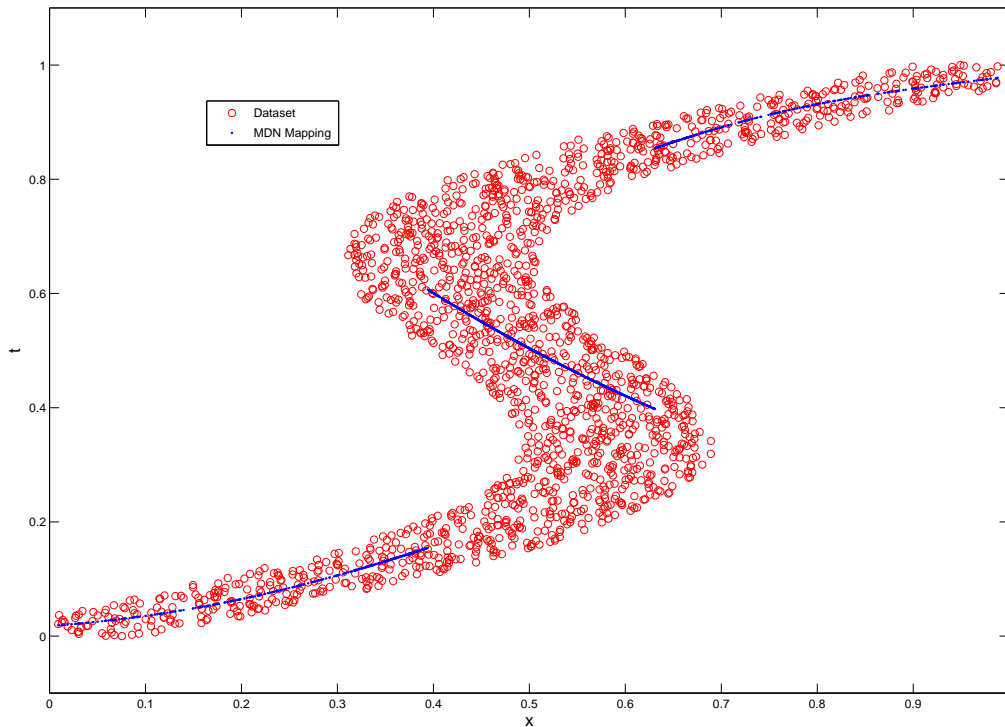


Figure B.9: Progress of central value of the most probable branch as function of x . The resulting mapping is discontinuous giving however a good description of the data.

able branch as a function of x . As can be seen we have obtained a discontinuous mapping which however give a good representation of the data with respect to FFW mapping (see Figure B.5). Differently from FFW mapping with the MDN approach we may obtain one of the possible solutions instead of the mean of all solutions. This give rise to a significant performance improvement especially for problems (such as control problems) in which the mean of more solutions is not itself a solution.

In order to quantify the performance of the two systems (MDN and FFW) on the basis of how the predicted outputs are solutions and not the mean of different solutions we can use the schema reported in Figure B.10 as done in (Bishop, 1994).

The underlining idea is simple: given a particular x the inverse model (in our case modeled with the MDN or FFW) predict the corresponding

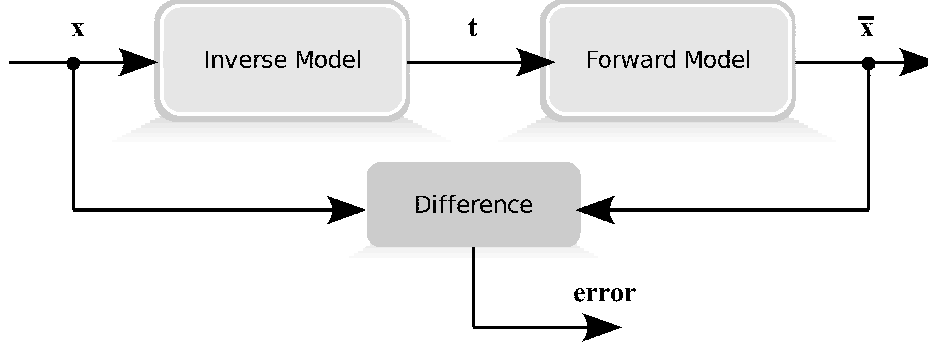


Figure B.10: Schema used to quantify how the response of the MDN (FFW) is one of the solutions of the inverse problem given a certain input x .

t ; if t is a solution then giving t in input to the forward model we must re-obtain x . So we can quantify how the predicted t is a solution by computing the distance between x and \tilde{x} which is the output obtained from the forward model feed with the predicted t (see Figure B.10).

The error between x and \tilde{x} is computed using the root-mean-square (RMS) error¹.

As expected there is a significant difference in term of RMS error between MDN and FFW. For the former $RMS = 0.011$ while for the last $RMS = 1.2216$.

B.5.2 test 2: a two-dimensional example

In this test we follow the same step of test 1 but now $\mathbf{t} \equiv (t_1, t_2)$ is a two-dimensional vector. Our objective is to give a broad idea of why the estimation of $p(\mathbf{t}|x)$ become more and more difficult when t increases its dimensionality.

The forward model which describes the relation between x and \mathbf{t} is the following:

$$x = t_1 + 0.3\sin(2\pi t_1) + t_2 + 0.3\sin(2\pi t_2) + \epsilon \quad (\text{B.48})$$

where $\mathbf{t} \in [0, 1]^2$ while ϵ is a random variable drawn from a uniform distribution in the range $[-0.1, 0.1]$.

¹Root-mean-square error (RMS) between target vectors \mathbf{t}^n and model outputs \mathbf{y}^n is computed as: $E^{RMS} = \frac{\sum_{n=1}^N \|\mathbf{y}^n - \mathbf{t}^n\|^2}{\sum_{i=1}^N \|\mathbf{t}^n - \bar{\mathbf{t}}\|^2}$. Here $\bar{\mathbf{t}}$ is defined to be the average target vector, that is: $\bar{\mathbf{t}} = \frac{1}{N} \sum_{n=1}^N \mathbf{t}^n$. The RMS has a value of unity when the model predicts the test data “in the mean” while a value of zero when the model’s prediction is perfect.

B.5. IMPLEMENTATION AND TEST

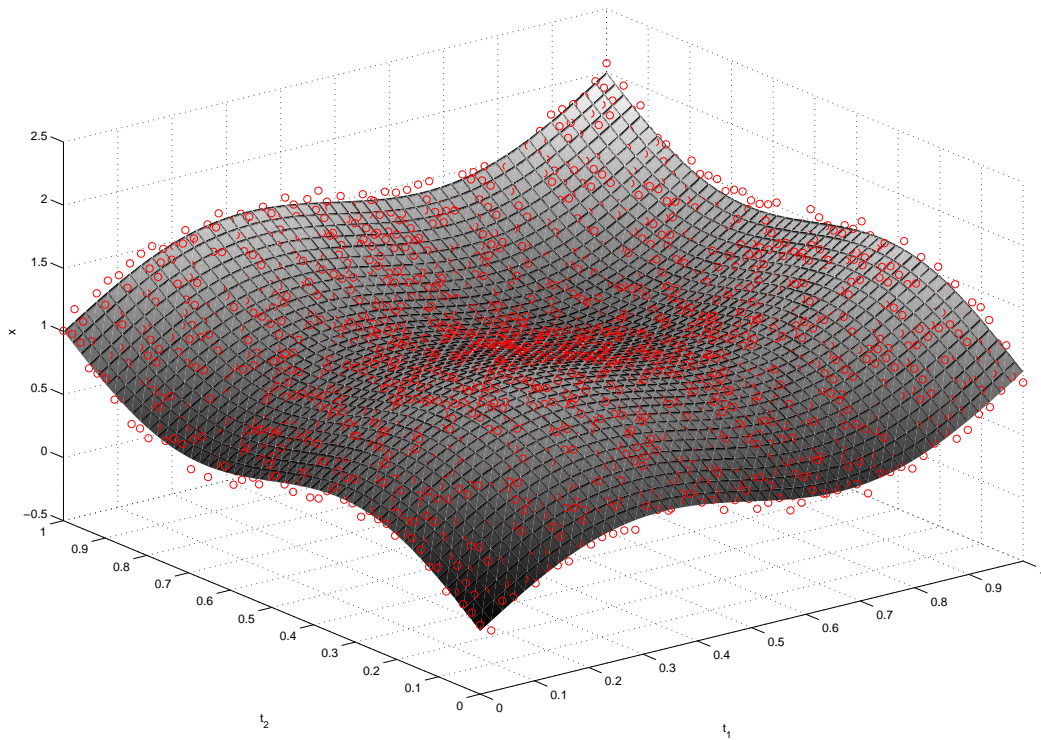


Figure B.11: Red bulled are points obtained from equation B.48. The underlying relation between x and $\mathbf{t} \equiv (t_1, t_2)$ is a single-valued function and can be easily approximate by standard neural networks.

B.5. IMPLEMENTATION AND TEST

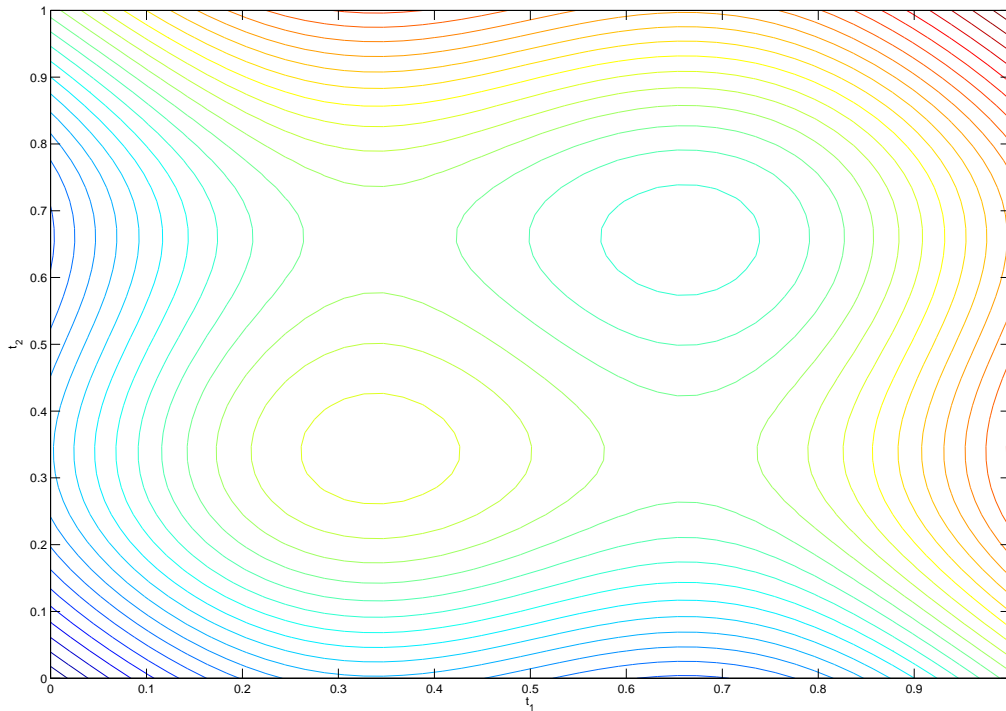


Figure B.12: Contour plot of the function expressed by relation B.48. As can be seen the inverse relation between x and t is multi-valued since for a certain value of x (denoted by a different color) there will be several different values of t .

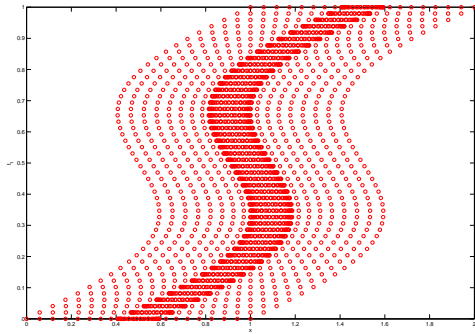
In Figure B.11 it is shown a data set of 2500 points created by using B.48 together with the mapping obtained using FFW. As can be seen the network give a good representation of the underling function which describe the forward model.

Again we try to estimate the inverse relation between x and t with an MDN networks. The mapping from x to t is multivalued as can be seen from the contour plot of Figure B.12.

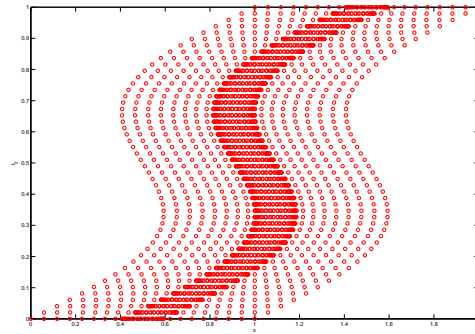
In Figure B.13 it is shown the relation between x and the components of t . These graphics may be misleading because one may thinks that few kernels suffice to have a good representation of $p(t|x)$. However this is not the case as can be seen from Figure B.14 where we have plotted the points t for $x = 1$.

In Figure B.15 it is shown the contour plot of $p(t|x = 1)$ estimated by

B.5. IMPLEMENTATION AND TEST



(a) Plot of x versus t_1



(b) Plot of x versus t_2

Figure B.13: Relation between x and the components of the vector $\mathbf{t} \equiv (t_1, t_2)$

two different MDN with three and ten kernels respectively and both with ten hidden units for the network component. As can be seen MDN model with ten kernels give a more comprehensive description of the distribution of \mathbf{t} . This is confirmed by computing the error as for test 1. The RMS error for the MDN with three kernels is 0.2607 while for MDN with ten kernels is 0.0683.

B.5. IMPLEMENTATION AND TEST

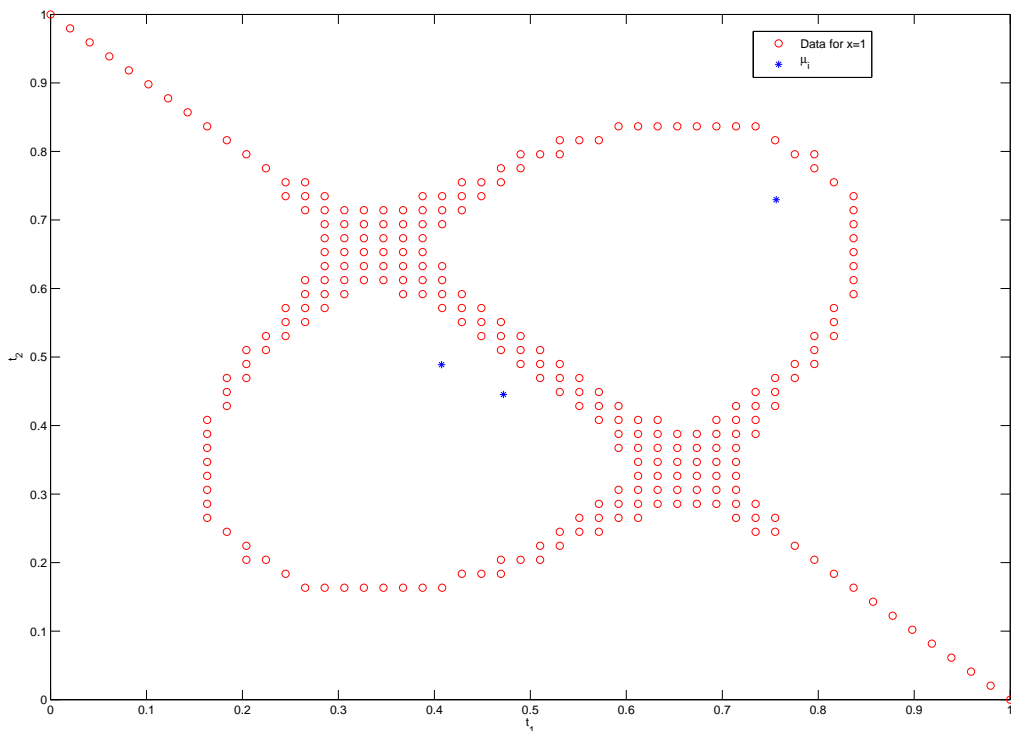
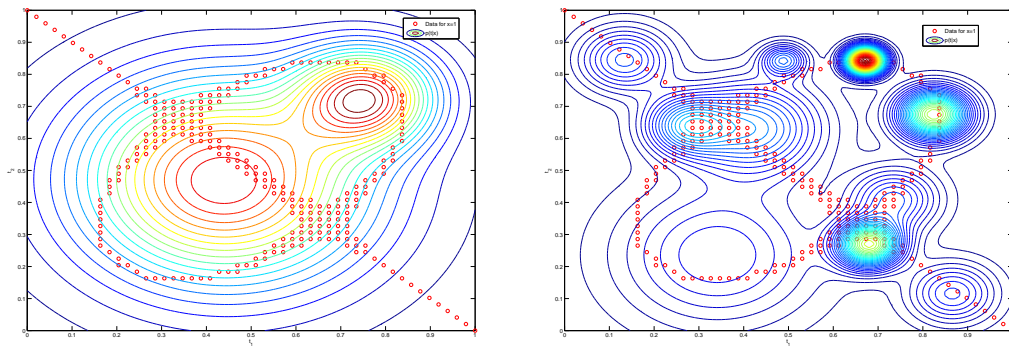


Figure B.14: The red bullets are points of the dataset corresponding to $x = 1$. In blue are shown the location of the three kernel centers used to approximate $p(t|x)$. As can be seen from the next figure, three kernels do not suffice to give a good representation of the distribution of the data points.

B.5. IMPLEMENTATION AND TEST



(a) Contour plot of the distribution $p(\mathbf{t}|x)$ as estimated with an MDN with 3 kernels

(b) Contour plot of the distribution $p(\mathbf{t}|x)$ as estimated with an MDN with 10 kernels

Figure B.15: Contour plot of the distribution $p(\mathbf{t}|x)$ with 3 and 10 kernels. As can be seen increasing the number of kernels leads to a better description of the distribution of the data points.



Similarity measure between principal subspaces

In this appendix the principal component analysis is introduced together with a similarity measure between principal subspaces as reported in (Krzanowski, 1979).

The principal component analysis (from now on PCA) is a widely used technique for dimensionality reduction, lossy data compression, feature extraction and data visualization (Jolliffe, 2002).

There are two different but equivalent formulations for PCA. The former defines the PCA as the orthogonal projection of the data onto a lower dimensional *linear* space (called *principal subspace*) such that the variance of the projected data is maximized. The latter defines the PCA as the linear projection that minimizes the mean squared distance between the data points and their projections.

In Figure C.1 is shown a two dimensional example of PCA where the red bullets represent original data points while green bullets represent projected data points onto a one dimensional principal subspace.

In the following sections we will first briefly describe the maximum variance formulation of PCA and then give a description of the similarity measure between principal subspaces.

C.1 Principal Component Analysis

Consider a data set X of N observations $\{\mathbf{x}^n\}_{n=1,\dots,N}$ where $\mathbf{x}^n \in \mathbb{R}^d$. Our objective is to project the data onto a space with dimensionality $k < d$ while maximizing the variance of the projected data.

C.1. PRINCIPAL COMPONENT ANALYSIS

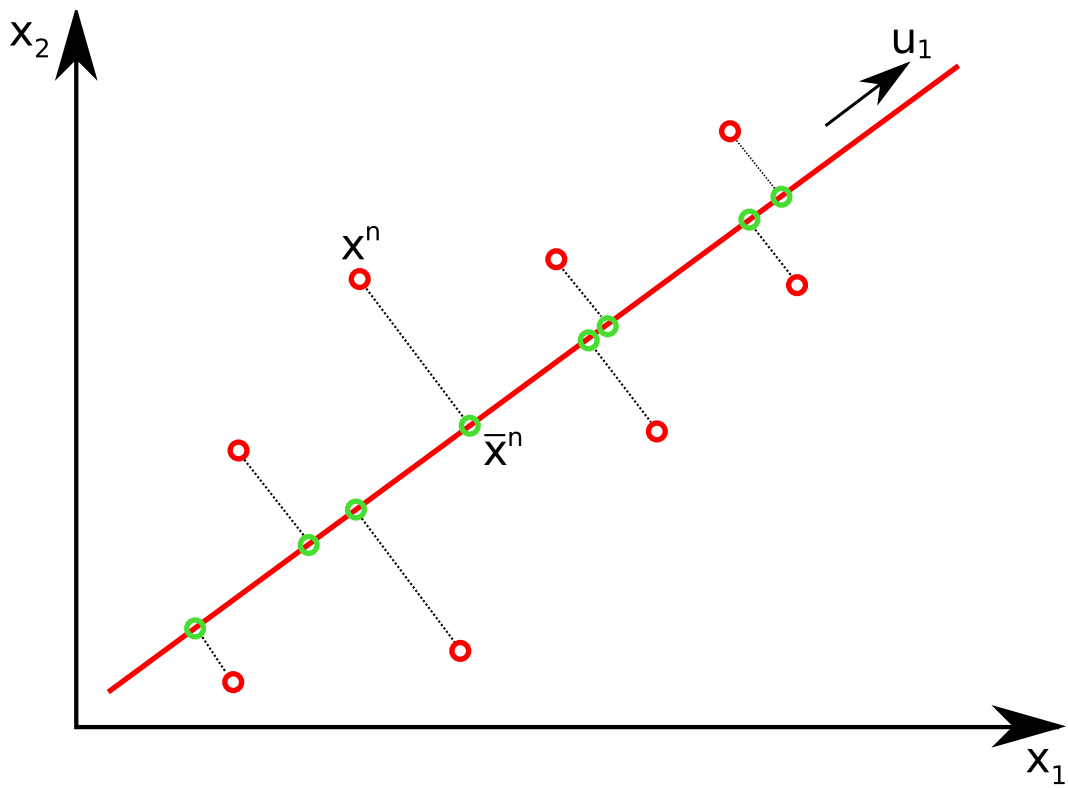


Figure C.1: Two dimensional example of PCA. The red bullets represent original data points while green bullets represent projected data points onto a one dimensional principal subspace.

C.1. PRINCIPAL COMPONENT ANALYSIS

We will start by considering a projection onto a one-dimensional space, that is $k = 1$. The extension to multi-dimensional space is straightforward. We can define the direction of this space using a vector \mathbf{u} of dimension $d \times 1$.

Each data points \mathbf{x} is projected onto a scalar value $\mathbf{u}^T \mathbf{x}$ where the superscribe T indicate the transpose of vector \mathbf{u} . If we indicate with $\bar{\mathbf{x}}$ the sample mean of the data set, then the projected mean is $\mathbf{u}^T \bar{\mathbf{x}}$. The variance of the projected data is given by:

$$\frac{1}{N} \sum_{n=1}^N (\mathbf{u}^T \mathbf{x}^n - \mathbf{u}^T \bar{\mathbf{x}})^2 = \mathbf{u}^T S \mathbf{u} \quad (\text{C.1})$$

where S is the covariance matrix defined as:

$$S = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}^n - \bar{\mathbf{x}})(\mathbf{x}^n - \bar{\mathbf{x}})^T \quad (\text{C.2})$$

We want to maximize the projected variance with respect to \mathbf{u} . However we must limit \mathbf{u} in order to prevent the solution $\|\mathbf{u}\| \rightarrow \infty$. To do this we impose the normalization condition $\mathbf{u}^T \mathbf{u} = 1$.

By means of a Lagrange multiplier we can convert such constrained maximization problem into an unconstrained one as follows:

$$\mathbf{u}^T S \mathbf{u} + \lambda(1 - \mathbf{u}^T \mathbf{u}) \quad (\text{C.3})$$

by setting the deriving with respect to \mathbf{u} equal to zero we obtain:

$$S \mathbf{u} = \lambda \mathbf{u} \quad (\text{C.4})$$

so \mathbf{u} must be an eigenvector of S . Multiplying both side by \mathbf{u}^T and taking into account that $\mathbf{u}^T \mathbf{u} = 1$ we have:

$$\mathbf{u}^T S \mathbf{u} = \lambda \quad (\text{C.5})$$

So the variance is maximized when we choose \mathbf{u} equal to the eigenvector having the largest eigenvalue λ of S .

if we consider the general case of an k -dimensional projection space, the optimal linear projection for which the variance of the projected data is maximized is defined by k eigenvectors $\mathbf{u}^1, \dots, \mathbf{u}^k$ of the data covariance matrix S corresponding to the k largest eigenvalues $\lambda_1, \dots, \lambda_k$.

C.2. PRINCIPAL SUBSPACE COMPARISON

Algorithm C.1 PCA algorithm

INPUT The data set matrix X of dimension $N \times d$
the number of components k

OUTPUT The projected data set matrix Y of dimension $N \times M$

1: **function** PCA(X, k) ▷ Centering data

2: $X_{mean} \leftarrow \text{computeMean}(X)$

3: $X_c \leftarrow X - X_{mean}$

4: $Cov_X \leftarrow X_c^T X_c$ ▷ Computing covariance matrix

5: $[U \text{ Lambda}] \leftarrow \text{diagonalize}(Cov_X)$ ▷ Computing eigenvectors,
eigenvalues
 ▷ Sorting eigenvectors by decreasing values of corresponding
 eigenvalues

6: $U \leftarrow \text{sortDescending}(U, \text{Lambda})$

7: $U_k \leftarrow \text{getFirstKcomponents}(U, k)$ ▷ Taking only the first k
eigenvectors

8: $Y \leftarrow X_c U_k$

9: **return** Y

10: **end function**

C.2 Principal Subspace Comparison

In the previous section we have briefly described the PCA algorithm as a way for describe a set of d -dimensional data onto a subspace of lower dimension.

Now, suppose we have several data set X^1, X^2, \dots, X^K each of which with same number of variable d . We apply PCA algorithm for each dataset and we search a criteria of congruence between the generated principal subspaces.

Such criteria has been proposed in Krzanowski (1979) and is exposed in the following.

Consider two data set X^1 of size $N_1 \times d$ and and X^2 of size $N_2 \times d$ each of which has undergone PCA. Suppose that k components are considered adequate for the purposes of representing each sample.

Let L and M be the matrices, both of size $d \times k$, of the first k eigenvectors of X^1 and X^2 respectively disposed column-wise.

It holds the following theorem:

Theorem 1. *The minimum angle between an arbitrary vector in the space of the first k principal components of X^1 and the one most nearly parallel to it in the*

C.3. A SIMPLE EXAMPLE

space of the first k components of X^2 is given by $\cos^{-1}(\lambda_1)^{\frac{1}{2}}$, where λ_1 is the largest eigenvalue of $G \equiv L^T M M^T L$.

Theorem 2. Let λ_i be the i th largest eigenvalue of G , \mathbf{u}^i its associated eigenvector of size $k \times 1$, and $\mathbf{v}^i = L\mathbf{u}^i$ ($i = 1, \dots, k$). Then $\mathbf{v}^1, \dots, \mathbf{v}^k$ form a set of mutually orthogonal vectors embedded in subspace X^1 and $M M^T \mathbf{v}^1, \dots, M M^T \mathbf{v}^k$, a corresponding set of mutually orthogonal vectors in subspace X^2 into which the differences between the subspaces can be partitioned. The angle between the i th pair $\mathbf{v}^i, M M^T \mathbf{v}^i$ is given by $\cos^{-1}(\lambda_i)^{\frac{1}{2}}$ ($i = 1, \dots, k$).

The first Theorem shows that the pair \mathbf{v}^1 and $M M^T \mathbf{v}^1$ gives the two closest vectors in the original space when one is constrained to be in subspace X^1 and the other in subspace X^2 . The second Theorem shows that continuing the decomposition of G , the pair \mathbf{v}^2 and $M M^T \mathbf{v}^2$ gives directions, orthogonal to the previous ones, along which the next smallest angle between the subspaces is represented.

These arguments allow us the definition of a measure of similarity between two subspaces. Let's see.

Let θ_{ij} be the angle between the i -th principal component of X^1 and the j -th principal component of X^2 then $\cos\theta_{ij}$ is the element (i, j) of the matrix $T = L^T M$. So it holds the following relation:

$$\sum_{i=1}^k \lambda_i = \text{trace } G = \text{trace } T T^T = \sum_{i=1}^k \sum_{j=1}^k \cos^2 \theta_{ij} \quad (\text{C.6})$$

The previous relation says that the sum of the eigenvalues of G equals the sum of squares of the cosines of the angles between each of the k eigenvectors defining the principal components of X^1 and X^2 . This value can be used as similarity measure between the two principal subspaces. The value of the sum is easily seen to lie between k , in this case the two principal subspaces are equal, and 0 in this case the two principal subspaces are orthogonal.

C.3 A simple example

To give an idea of how the similarity measure introduced above works let us consider a simple two dimensional example. We have two group of data X^1 and X^2 both composed of 1000 points extracted from a normal distribution with zero mean $\mathbf{x} \equiv (x, y) = 0$ and diagonal covariance matrix with $\sigma_{xx} = 0.1$ and $\sigma_{yy} = 1$. So both group of data have great variability

C.3. A SIMPLE EXAMPLE

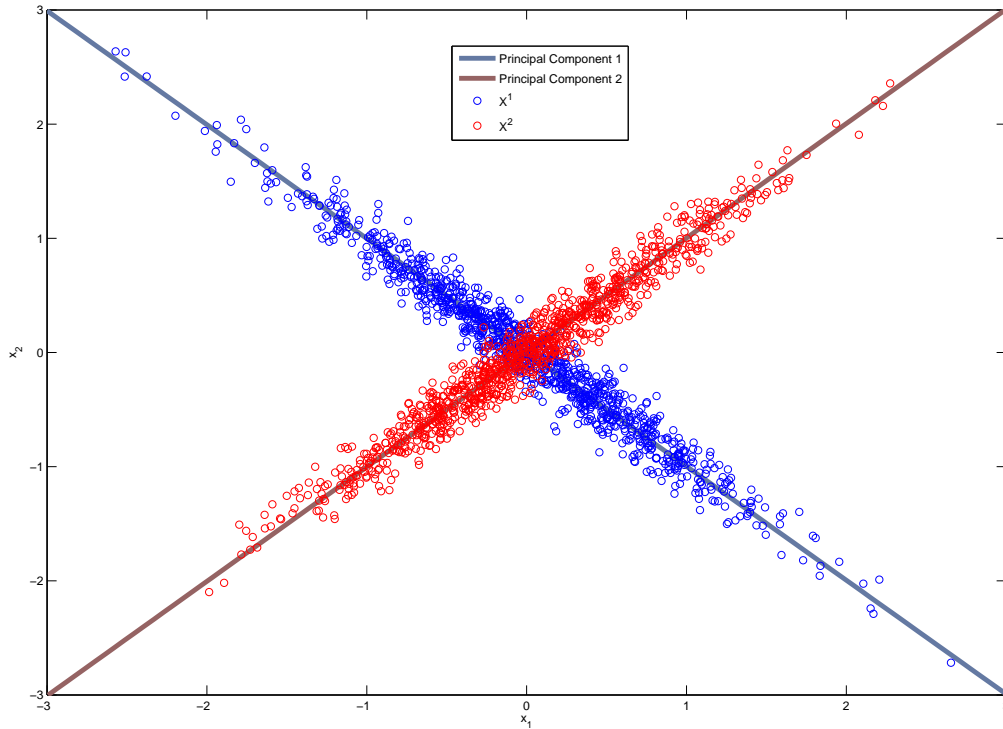


Figure C.2: Data points belonging to the two data set X^1 (blue) and X^2 (red). The corresponding principal components in the two sets have the same direction but different eigenvalues because direction of greatest spread of data is orthogonal between the two sets.

along only one direction. Point of the two group are rotated of 45° and 135° respectively by means of a rotation matrix of the form:

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

The points of the two groups so obtained as plotted in Figure C.2 together with principal components obtained by applying PCA algorithm. Principal components of the two group are very similar and in the graph only that of the first group are shown. However the associated eigenvalues are completely different in the two cases as one can see from Table C.1. In fact the direction of maximum variance for point of X^1 is orthogonal to the direction of maximum variance for point of X^2 .

C.3. A SIMPLE EXAMPLE

Eigenvalues	Eigenvectors X^1	Eigenvalues	Eigenvectors X^1
0.0105	(-0.7061, -0.7081)	0.9297	(-0.7080, -0.7063)
1.0627	(-0.7081, 0.7061)	0.0101	(-0.7063, 0.7080)

Table C.1: Eigenvectors and corresponding eigenvalues for the two group X^1 and X^2

Let us now consider two principal subspaces of dimension one generated by the first principal component of the two group X^1 and X^2 . Since they are orthogonal to each others the similarity measure 0 (recall that the in this case such measure lies between 0 and 1). However if we consider two principal subspace of dimension two the subspaces coincide and in fact the similarity measure is 2 (in this case such measure lies between 0 and 2).

Bibliography

- Arbib, M. A. (1981). Perceptual structures and distributed motor control. *Handbook of Physiology, section 1: The Nervous System, vol II: Motor Control*, pages 1449–1480.
- Arbib, M. A. (2005). From monkey-like action recognition to human language: An evolutionary framework for neurolinguistics. *Behavioral and Brain Sciences*, 28(2):105–124.
- Billard, A. and Mataric, M. (2001). Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture. *Robotics and Autonomous Systems*, 37:145–160.
- Bishop, C. M. (1994). Mixture density networks. Technical report, Aston University, Birmingham.
- Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- Bonaiuto, J., Rosta, E., and Arbib, M. A. (2007). Extending the mirror neuron system model, i: Audible actions and invisible grasps. *Biol. Cybern.*, 96(1):9–38.
- Borghini, A. M. (2005). *Grounding Cognition: The role of perception and action in memory, language, and thinking*, chapter Object concepts and action. Cambridge University Press.

C.3. A SIMPLE EXAMPLE

- Caggiano, V., Fogassi, L., Rizzolatti, G., Thier, P., and Casile, A. (2009). Mirror neurons differentially encode the peripersonal and extrapersonal space of monkeys. *Science*, 324(5925):403–406.
- Carr, L., Iacoboni, M., Dubeau, M. C., Mazziotta, J. C., and Lenzi, G. L. (2003). Neural mechanisms of empathy in humans: A relay from neural systems for imitation to limbic areas. *PNAS*, 100(9):5497–5502.
- Castiello, U. (2005). The neuroscience of grasping. *Nature Reviews Neuroscience*, 6(9):726–736.
- Cattaneo, L. and Rizzolatti, G. (2009). The Mirror Neuron System. *Arch Neurol*, 66(5):557–560.
- Chemero, A. (2003). An outline of a theory of affordances. *Ecological Psychology*.
- Daugman, J. G. (1993). *An information-theoretic view of analog representation in striate cortex*, pages 403–423. Computational neuroscience. MIT Press, Cambridge, MA, USA.
- Demiris, J. and Hayes, G. (2002). *Imitation as a dual-route process featuring predictive and learning components: a biologically plausible computational model*, pages 327–361. Imitation in animals and artifacts. MIT Press, Cambridge, MA, USA.
- Demiris, J. and Johnson, M. (2003). Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning. *Connection Science*, 15(4):231–243.
- Fadiga, L., Fogassi, L., Gallese, V., and Rizzolatti, G. (2000). Visuomotor neurons: ambiguity of the discharge or ‘motor’ perception? *International Journal of Psychophysiology*, 35:165–177.
- Fogassi, L., Ferrari, P. F., Gesierich, B., Rozzi, S., Chersi, F., and Rizzolatti, G. (2005). Parietal lobe: from action organization to intention understanding. *Science*, 308(5722):662–667.
- Fogassi, L., Gallese, V., Buccino, G., Craighero, L., Fadiga, L., and Rizzolatti, G. (2001). Cortical mechanism for the visual guidance of hand grasping movements in the monkey. a reversible inactivation study. *Brain*, 124:571–586.
- Friston, K. (2005). A theory of cortical responses. *Philos Trans R Soc Lond B Biol Sci*, 360(1456):815–836.

C.3. A SIMPLE EXAMPLE

- Fritsch, C. (2007). Predictive coding: an account of the mirror neuron system. *Cognitive Processing*, 8:159–166.
- Funahashi, K.-i. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Netw.*, 2(3):183–192.
- Gallese, V., Fadiga, L., Fogassi, L., and Rizzolatti, G. (1996). Action recognition in the premotor cortex. *Brain*, 119:593–609.
- Gallese, V. and Goldman, A. (1998). Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences*, 2(12):493–501.
- Geyer, S., Matelli, M., Luppino, G., and Zilles, K. (2000). Functional neuroanatomy of the primate isocortical motor system. *Anat Embryol*, 202:443–474.
- Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin.
- Giese, M. A. and Poggio, T. (2003). Neural mechanisms for the recognition of biological movements and action. *Nature Reviews Neuroscience*, 4:179–192.
- Haruno, M., Wolpert, D. M., and Kawato, M. (2001). Mosaic model for sensorimotor learning and control. *Neural Computation*, 13(10):2201–2220.
- Hastie, T., Tibshirani, R., and Friedman, J. H. (2003). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, corrected edition.
- Iacoboni, M. and Dapretto, M. (2006). The mirror neuron system and the consequences of its dysfunction. *Nature Reviews Neuroscience*, 7(12):942–951.
- Iberall, T., Bingham, G., and Arbib, M. A. (1986). Opposition space as a structuring concept for the analysis of skilled hand movements. *H. Heuer and C. Fromm (Eds.)*, 15:158–173.
- Iberall, T. and Fagg, A. H. (1996). *Neural Network models for selecting hand shapes*, pages 243–264. A.M. Wing, P. Haggard and J.R. Flanagan, Editors, *Hand and Brain: the Neurophysiology and Psychology of Hand Movements*, Academi.

C.3. A SIMPLE EXAMPLE

- Ito, M. and Tani, J. (2004). Generalization in learning multiple temporal patterns using RNNPB. In *ICONIP: International Conference on Neural Information Processing*, pages 592–598.
- Jeannerod, M. (1984). The timing of natural prehension movements. *Journal of Motor Behavior*, 16(3):235–54.
- Jeannerod, M., Arbib, M. A., Rizzolatti, G., and Sakata, H. (1995). Grasping objects: the cortical mechanisms of visuomotor transformation. *Trends in Neurosciences*, 18(7):314–320.
- Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, second edition.
- Keysers, C. and Perrett, D. I. (2004). Demystifying social cognition: a hebbian perspective. *Trends Cogn Sci*, 8(11):501–507.
- Krzanowski, W. J. (1979). Between-groups comparison of principal components. *Journal of the American Statistical Association*, 74(367):703–707.
- Mason, C. R., Gomez, J. E., and Ebner, T. J. (2001). Hand synergies during reach-to-grasp. *J Neurophysiol*, 86(6):2896–2910.
- Matelli, M. and Luppino, G. (2001). Parietofrontal circuits for action and space perception in the macaque monkey. *NeuroImage*, 14:S27–S32.
- Miall, R. C. (2003). Connecting mirror neurons and forward models. *Neuroreport*, 14(17):2135–2137.
- Miller, A. T. and Allen, P. K. (2004). Graspit! a versatile simulator for robotic grasping. *Robotics & Automation Magazine, IEEE*.
- Milner, A. D. (1998). Neuropsychological studies of perception and visuomotor control. *Philos Trans R Soc Lond B Biol Sci*.
- Murata, A., Fadiga, L., Fogassi, L., Gallese, V., Raos, V., and Rizzolatti, G. (1997). Object representation in the ventral premotor cortex (area f5) of the monkey. *Journal of Neurophysiology*, 78:2226–2230.
- Murata, A., Gallese, V., Luppino, G., Kaseda, M., and Sakata, H. (2000). 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.
- Nguyen, D. and Widrow, B. (1990). Improving the learning speed of 2-layer neural network by choosing initial values of the adaptive weights. In *IJCNN: International Joint Conference on Neural Networks*, pages 21–26.

C.3. A SIMPLE EXAMPLE

- Oztop, E. and Arbib, M. A. (2002). Schema design and implementation of the grasp-related mirror neuron system. *Biological Cybernetics*, 87:116–140.
- Oztop, E., Bradley, N. S., and Arbib, M. A. (2004). Infant grasp learning: a computational model. *Experimental Brain Research*, 158(4):480–503.
- Oztop, E., Imamizu, H., Cheng, G., and Kawato, M. (2006a). A computational model of anterior intraparietal (AIP) neurons. *Neurocomputing*, 69(10-12):1354–1361.
- Oztop, E., Kawato, M., and Arbib, M. A. (2006b). Mirror neurons and imitation: A computationally guided review. *Neural Networks*, 19:254–271.
- Oztop, E., Wolpert, D. M., and Kawato, M. (2005). Mental state inference using visual control parameters. *Cognitive Brain Research*, 22(2):129–151.
- Perrett, D. I., Harries, M. H., Bevan, R., Thomas, S., Benson, P. J., Mistlin, A. J., Chitty, A. J., Hietanen, J. K., and Ortega, J. E. (1989). Frameworks of analysis for the neural representation of animate objects and actions. *J. exp. Biol.*, 146:87–113.
- Prevete, R., Tessitore, G., Santoro, M., and Catanzariti, E. (2008). A connectionist architecture for view-independent grip-aperture computation. *Brain Research*, 1225:133–145.
- Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood and the em algorithm. *SIAM Review*, 26(2):195–239.
- Riedmiller, M. (1994). Rprop - description and implementation details. Technical report, Institut für Logik, Komplexität und Deduktionssysteme, University of Karlsruhe.
- Riesenhuber, M. and Poggio, T. (2000). Models of object recognition. *Nature Neuroscience*, 3:1199–1204.
- Riesenhuber, M. and Poggio, T. (2002). Neural mechanisms of object recognition. *Current Opinion in Neurobiology*, 12(2):162–168.
- Rizzolatti, G., Camarda, R., Fogassi, L., Gentilucci, M., Luppino, G., and Matelli, M. (1988). Functional organization of inferior area 6 in the macaque monkey: II. area f5 and the control of distal movements. *Experimental Brain Research*, 71(3):491–507.

C.3. A SIMPLE EXAMPLE

- Rizzolatti, G. and Craighero, L. (2004). The mirror-neuron system. *Annual Review of Neuroscience*, 27:169–192.
- Rizzolatti, G., Fadiga, L., Gallese, V., and Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. *Cognitive Brain Research*, 3(2):131–141.
- Rizzolatti, G., Fogassi, L., and Gallese, V. (2001). Neurophysiological mechanisms underlying the understanding and imitation of action. *Nat Rev Neurosci*, 2(9):661–670.
- Rizzolatti, G. and Gentilucci, M. (1988). Motor and visual-motor functions of the premotor cortex. *Neurobiology of Neocortex*, pages 269–284.
- Rizzolatti, G., Luppino, G., and Matelli, M. (1998). The organization of the cortical motor system: new concepts. *Electroencephalography and Clinical Neurophysiology*, 106:283–296.
- Rizzolatti, G. and Sinigaglia, C. (2008). *Mirrors in the Brain. How Our Minds Share Actions and Emotions*. Oxford University Press: Oxford.
- Santello, M., Flanders, M., and Soechting, J. F. (2002). Patterns of hand motion during grasping and the influence of sensory guidance. *Journal of Neuroscience*, 22(4):1426–1235.
- Schiegg, A., Deubel, H., and Schneider, W. (2003). Attentional selection during preparation of prehension movements. *Visual Cognition*.
- Tessitore, G., Borriello, M., Prevede, R., and Tamburrini, G. (2009). How direct is perception of affordances? a computational investigation of grasping affordances. In Howes, D. Peebles, R. C., editor, *9th International Conference on Cognitive Modeling - ICCM2009*, Manchester, UK.
- Tsiotas, G., Borghi, A., and Parisi, D. (2005). Objects and affordances: An artificial life simulation. In *Proceedings of the Cognitive Science Society*.
- Tucker, M. and Ellis, R. (2000). Micro-affordance: the potentiation of components of action by seen objects. *British Journal of Psychology*, 91(4):451–471.
- Umiltà, M., Kohler, E., Gallese, V., Fogassi, L., Keysers, C., and Rizzolatti, G. (2001). I know what you are doing: A neurophysiological study. *Neuron*, 31(19):155–165.

C.3. A SIMPLE EXAMPLE

Weber, C., Elshaw, M., Triesch, J., and Wermter, S. (2008). *Humanoid Robots: Human-like Machines*, chapter 30, pages 577–600. I-Tech Education and Publishing, Vienna, Austria.