# COMPUTATIONAL APPROACHES TO BRAIN MECHANISMS OF ACTION RECOGNITION AND EMOTION

A Thesis

by

Murat Kırtay

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Computer Science

Özyeğin University
August 2015

# COMPUTATIONAL APPROACHES TO BRAIN MECHANISMS OF ACTION RECOGNITION AND EMOTION

Approved by:

Professor Erhan Öztop, Advisor
Department of Computer Science
*Özyeğin University*

Professor Tankut Barış Aktemur
Department of Computer Science
*Özyeğin University*

Professor Sanem Sarıel
Department of Computer Engineering
*İstanbul Technical University*

Date Approved: 07-07-2015

*To Non-fraud Academicians, Singularitarians, Cosmists, Hackers and AGI Researchers*

# ABSTRACT

Through evolution living beings have gained unique features to deal with apparently easy but computationally expensive problems such as mate selection, learning sensorimotor skills and decision making. Thus, understanding how a biological system can process sensory information, interpret the probable results and find a solution in relatively short time to faced problems have become an attractive research area for computational neuroscience, artificial intelligence (AI) and robotics. In this thesis we focused on mirror neurons in the ventral premotor cortex (area F5) and the functional aspects of emotions from a computational but biologically plausible way. In the former part, the raw neural firing data from area F5 of macaque monkeys are analyzed to undercover neural representation using a decoding framework. For this, we propose two methods to detect mirror neurons by using machine learning and statistical analysis techniques. In the later part, we present that higher level emotions (those that have putatively evolved after the basic emotions of fear, anger etc.) are the behavioral manifestation of self-regulation mechanisms of computational (neuronal) energy expenditure for cognitive processing. To realize this proposal, we chose a tractable computational mechanism that may be considered as a model of neural computation mechanisms of the brain and deploy it on a robotic platform (Darwin-OP).

# ÖZETÇE

Evrim süreci boyunca canlılar çözümü kolay görünen fakat hesaplama gücü ve yükü bakımından yüksek problemleri çözebilmek için çok çeşitli özellikler kazanmışlardır. Bu problemlere örnek olarak eş seçimi, yeni beceriler edinme ve beklenmeyen durumlarda karar verme gösterilebilinir. Karmaşık biyolojik sistemlerin bilgi edinme, edinilen bilgiyi sonuçlandırma ve göreceli kısa zaman aralığında problem çözme becerileri hesaplamalı sinirbilim, yapay zeka ve robotik alanında çalışan araştırmacılar için ilgili literatürlerde var olan problemlerin çözümü için ilgi çekiçi araştırma alanı olarak öngörülmektedir. Bu tez çalışmasında makak maymunlarının beyinde yer alan F5 bölgesindeki ayna nöronlarının sinirsel çözümlemesini ve duygu tabanlı karar vermenin hesaplama yetisi bakımından işlevseliği ve biyolojik açıdan ikna ediciliği hakkında elde edilen sonuçlara yer verilmiştir. İlk bölümde, makak maymunundan farklı deney şartlarından elde edilen sinirsel veri kümelerine, makine öğrenmesi ve istatistiksel analiz yöntemleri uygulanmasıyla elde edilen sonuçların detaylarına yer verilmiştir. Bu yöntemler sonucunda F5 bölgesindeki ayna nöron adayları seçilmiş ve bu nöronların sinirsel gösterimleri hakkında literaturde önceden elde edilmemiş bulgulara yer verilmiştir. İkinci bölümde ise temel duygulardan hemen sonra gelişen üst seviye dugyuların vücut mekanizmasında var olan enerji seviyesinin değişiminin ve denetiminin işlevsellik ve biyolojik açıdan incelenmesine ve insansı robot platformu üzerinde gerçeklenmesine yer verilmiştir.

# ACKNOWLEDGMENTS

What I can not create, I do not understand.

*Richard Feynman*

I would like to thank my advisor Erhan Öztop and my undergraduate period supervisors Uluç Saranlı and Veysel Gazi for sharpening my academic career with their endless energy, invaluable academic morality and helping me to discover my enthusiasm towards biologically inspired robotics and neuroscience.

I also extend my thanks to Barış Aktemur and Güray Erkol for their help and understanding when I was a teaching assistant of their lectures. I am pretty sure that in a parallel universe they are co-advising my PhD dissertation which is titled as "Functional Programming Approaches on Quantum Mechanics". I had pleasure of learning Computer Vision from Furkan Kıraç who is an excellent role model for being an engineer and academician. I am grateful to Sanem Sarıel for accepting to be a member of my defense jury and her tireless efforts for giving talks and having conversation with robotics club members.

I am fortunate enough to meet two of most talented robotics guys: Utku Culha and İsmail Uyanık. I thank them for their guidance during my very first year in undergraduate period. Additionally, I am very thankful to Bilkent Robotics Club members who are Murat Aslan, Mustafa Uğur Daloğlu, Bilgesu Erdoğan, Gizem Tabak and Burak Yücesoy.

I would like to give special thanks, in no particular order, to my Italian friends: Linda Caira, Daniela Sassano, Daniele Evangelista, Matteo Pisani, Lorenzo Carnevale, Paola Manzo, Sara Morrone, Francesco de Siena, Stefania Esposito and Sidhant Hasija who are always there to mitigate pain in our stressful research period.

I want to thank Gönül Aycı because of her never ending understanding towards my late night and early morning discussions. Moreover, I always appreciate my "social hack" friend Bersu Uysalel who is always ready to listen my Star Wars, the singularity, transhumanism mumbo-jumbos and I thank to "black-eyed elvish lady" Ayşe Karagöz for her help and friendship during unbearable problems that we faced in the academia.

One of the most important aspect of my graduate year is to work with talented undergraduate students who are Can Göçmen, Taha Doğan Günes and Deniz Sökmen. I am in certain that they will "make the world a better place".

I had the pleasure to meet and work with Can Eren Sezener (aka the Golden Boy). We discussed numerous topics including but not limited to rationality, quantified self, black swans, deep learning, Shane Legg and Demis Hassabis.

Finally, I am forever thankful to Fadime Taşcı for her endless support, making unbearable times fun and always being up for showing me there is a life outside of hacking robots and neurons.

Most importantly, I owe my great thanks to my family for their endless love, support and encouragement.

**Statement of Funding**

The study in the second chapter of the thesis was supported by the Scientific and Technological Research Council of Turkey (TUBİTAK) with project number 113S391.

**Statement of Contribution**

In Chapter 2, recorded data provided by the project collaborators from University of Crete and utilized methods are discussed in the meetings with Erhan Öztop. In Chapter 3, the applied vision algorithm was tuned in collaboration with Alp Pehlivan. The rest of this thesis is my own original work.

# TABLE OF CONTENTS

**DEDICATION** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **iii**

**ABSTRACT** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **iv**

**ÖZETÇE** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **v**

**ACKNOWLEDGEMENTS** . . . . . . . . . . . . . . . . . . . . . . . . **vi**

**LIST OF TABLES** . . . . . . . . . . . . . . . . . . . . . . . . . . . **x**

**LIST OF FIGURES** . . . . . . . . . . . . . . . . . . . . . . . . . . . **xi**

**I INTRODUCTION** . . . . . . . . . . . . . . . . . . . . . . . . . . . **1**

1.1 Motivation and Background . . . . . . . . . . . . . . . . . . . . . 1

1.1.1 Existing Works on Mirror Neurons . . . . . . . . . . . . . . . 3

1.1.2 Existing Works on Emotions . . . . . . . . . . . . . . . . . . 4

1.2 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

1.3 Thesis Organization . . . . . . . . . . . . . . . . . . . . . . . . 6

**II ACTION RECOGNITION (MIRROR NEURONS)** . . . . . . . . **7**

2.1 Experiments and Neural Data Set Features . . . . . . . . . . . . . 8

2.1.1 Experiment conditions . . . . . . . . . . . . . . . . . . . . . 9

2.1.2 Neural data set specifications . . . . . . . . . . . . . . . . . 10

2.2 Evaluation Metrics . . . . . . . . . . . . . . . . . . . . . . . . . 11

2.3 Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

2.3.1 Linear regression based decoding . . . . . . . . . . . . . . . 13

2.3.2 Cross validation . . . . . . . . . . . . . . . . . . . . . . . . 15

2.4 Results and Discussions . . . . . . . . . . . . . . . . . . . . . . 16

2.4.1 Mirror neuron detection methods . . . . . . . . . . . . . . . 17

2.4.2 Cross decoding . . . . . . . . . . . . . . . . . . . . . . . . . 18

2.4.3 Population level cross decoding . . . . . . . . . . . . . . . . 21

**III EMOTIONS** . . . . . . . . . . . . . . . . . . . . . . . . . . . . **23**

3.1 Biological Background . . . . . . . . . . . . . . . . . . . . . . . . . 24

3.2 Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

3.2.1 Behaviour trough self regulation of the Hopfield Neural Network 26

3.2.2 Behaviour trough energy regulation . . . . . . . . . . . . . . 27

3.3 Experimental Setup . . . . . . . . . . . . . . . . . . . . . . . . . . 29

3.3.1 Hardware setup and data flow . . . . . . . . . . . . . . . . . 30

3.3.2 Darwin-OP and behavior execution process . . . . . . . . . . 32

3.4 Results and Discussions . . . . . . . . . . . . . . . . . . . . . . . . 33

3.4.1 Analysis of energy conservation . . . . . . . . . . . . . . . . 33

3.4.2 Emotion based network dynamics . . . . . . . . . . . . . . . 36

**IV CONCLUSIONS** . . . . . . . . . . . . . . . . . . . . . . . . . . **38**

4.1 Action Recognition (Mirror Neurons) . . . . . . . . . . . . . . . . . 38

4.1.1 Execution and observation decoding . . . . . . . . . . . . . . 38

4.1.2 Cross decoding . . . . . . . . . . . . . . . . . . . . . . . . . 39

4.2 Emotions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

**References** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **41**

# LIST OF TABLES

1 Condition and object matrix for unit 10 . . . . . . . . . . . . . . . . . 11

2 Condition and object matrix for unit 13 . . . . . . . . . . . . . . . . . 11

3 Unit10 execution condition prediction rates . . . . . . . . . . . . . . . 12

4 Single unit linear LOO regression prediction ranks for NOCUEOBS event . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

5 Single unit linear regression LOO prediction ranks for EXECUTION event . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

6 Single unit linear regression prediction ranks for EXECUTION ($W_{nocueobs}$ used) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

7 Single unit linear regression prediction ranks for EXECUTION ($W_{exec}$ used) event . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

8 Single unit linear regression prediction rates for nocueobs2execution and EXECUTION events . . . . . . . . . . . . . . . . . . . . . . . 20

9 Multi unit linear regression prediction ranks for EXECUTION ($W_{nocueobs}$ used) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

10 Multi unit linear regression prediction rates for Execution ($W_{nocueobs}$ used) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

11 Pattern and behavior matrix . . . . . . . . . . . . . . . . . . . . . . 36

# LIST OF FIGURES

1 Marr's three levels of analysis . . . . . . . . . . . . . . . . . . . . . . . . 2

2 Purposeful movements and object grasping types . . . . . . . . . . . 8

3 Observation with cue . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

4 Observation without cue . . . . . . . . . . . . . . . . . . . . . . . . . 9

5 Object fixation with cue . . . . . . . . . . . . . . . . . . . . . . . . . 10

6 Execution . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

7 Weight transfer among conditions . . . . . . . . . . . . . . . . . . . . 19

8 Retina image and trained patterns . . . . . . . . . . . . . . . . . . . 28

9 Retina images and converged patterns . . . . . . . . . . . . . . . . . 28

10 Experimental setup . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

11 Data Flow between Darwin-OP and PC . . . . . . . . . . . . . . . . . 31

12 Darwin-OP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

13 Converged pattern curves for Fig. 8(c) and average curves of all patterns . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

14 Experiment media snapshots . . . . . . . . . . . . . . . . . . . . . . 37

# CHAPTER I

# INTRODUCTION

This thesis concerns the development of computational approaches for modeling brain mechanisms, namely mirror neurons and emotions. Moreover, emotion related modeling is realized both in simulation and hardware (e.g. robot platform). In the first part, we automatize the detection of the mirror neurons among a set of neurons which was generally done in a heuristic and often manual way. In the second part, we propose a function for emotion, and implement it in an existing cognitive process of a humanoid robot to demonstrate its plausibility. With this we show that computational energy conservation might lead to emergent emotion. Furthermore, we test our implementation both in simulation and on hardware to exemplify that proposed emotion mechanism not only perform well in simulation, but it also can be realized on a real robot platform.

## *1.1 Motivation and Background*

Brain and its sensorimotor mechanisms have been considered as a potential source of solution to various existing problems in robotics and artificial intelligence. This is the reason why, brain inspired approaches substantially increased in many fields of engineering such as cognitive robotics, machine learning, computational neuroscience and artificial general intelligence. In this thesis we specifically focused on two brain mechanisms which are action recognition (e.g. mirror neurons) and emotions due to their unique roles in the brain and promising impacts on related fields.

In order to get insights from the brain mechanisms, we follow a model based approach to investigate mirror neurons and emotions. By doing so, we developed a decoding framework for mirror neurons and emotion based neural energy conservation mechanism to get preliminary findings and results for building computational models of mirror neuron and emotions. According to Marr's three levels of analysis, a complex biological system (e.g. human brain, vision system) can be understood by analyzing them into three levels of abstraction [1]. In the first level the computational model of the system should be derived to understand the goals of the system. In the second level the algorithmic level should be constructed to obtain representations and algorithms of the system that used to achieve the goals of the system. In the last level the physical realization of the system on a hardware platform should be implemented to finalize the abstraction levels. In this thesis we partially performed these three levels for mirror neurons and a simplified version was fully realized for emotion studies.

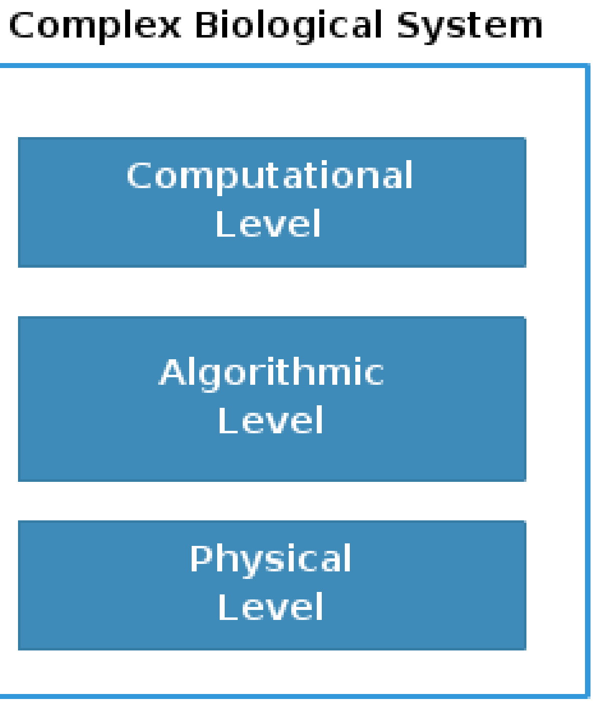


**Figure 1:** Marr's three levels of analysis

On the one hand, we analyze the neural data to propose a biologically realistic computational model for mirror neurons for the later studies. On the other hand, we realize a version of these three levels of analysis framework to investigate functional aspects of the emotions.

### 1.1.1 Existing Works on Mirror Neurons

Mirror neurons were discovered in the ventral premotor cortex (area F5) of a macaque monkey which become activated while the monkey executes an action or observes similar action executed by the others [2]. Up until now, the functions of mirror neurons have been associated with empathy, language, imitation learning, autism spectrum disorder, cultural evolution, action understanding and observation, to mention a few [3], [4], [6], [5]. In this thesis we specifically emphasize action understanding and action observation characteristics of the macaque monkeys' mirror neurons in F5. To find a convincing answer to what is decoded by mirror neurons attracts considerable number of researchers and decoding a single neuron or neuron population studies have been vividly increased in the recent years.

In the majority of decoding related studies can be listed as observed action related parameters, the schema level motor plan and the intention understanding [7]. In the study [8], a support vector machines (SVM) based classifier used to decode four or six grip types and propose implantable cortical neuroprosthesis with a high accuracy rate which was above 96%. Similarly in [9] presents real-time prosthetic hand control trough decoding dorsal premotor cortex' neural data with four different shaped objects. To do this, the authors performed fuzzy k-nearest neighbor algorithm and obtained accuracy of 97.1%. Carpaneto et al. applied various machine learning algorithms on the area F5 neural data such as linear classifier, soft-max network, k-nearest neighbor and support vector machines to classify set of objects which have different grasping types. The authors concluded that applying support vector machine algorithm provided high accuracy rate, 95%, over other techniques. As can be seen these aforementioned studies have high accuracy rate to decode area specific neurons via performing different machine learning techniques. The emphasized accuracy rates obtained by applying manually tuned window width, adding more neurons for constructing input matrices and combining neurons which were obtained from different areas of the brain.

It should be noted that these studies have no specific ways to detect mirror neurons and their action/observation characteristics. Thus, we took different approach both for labeling mirror neurons and object level decoding which will be explained in detail in the Chapter 2's subsections.

### 1.1.2 Existing Works on Emotions

There have been studies from theoretical and experimental -both behavioral and neuroscientific- perspectives that investigated the role of emotions (e.g., fear and joy) in social behavior [13]. In particular, there have been recent efforts to determine the brain regions which contribute or affect emotional processes [13]. In the AI and robotics literature, there exists a number of studies that utilize the concept of emotions -at least functionality of the emotions- as cognitive processes that can be emulated on robots to short-cut cognitive processing, shift behavioral hierarchies, facilitate storage and recall of memories [14], [15]. Despite the accumulated knowledge on the source and function of emotion in biological systems, the transfer of this knowledge to robotics and AI has not been realized in full, to the extent to trigger a significant improvement or a paradigm shift. This is in part due to the unsettled issues about emotion in psychology and biological sciences. It would not be wrong to say that there is no universally accepted definition of emotion. Yet, the emotion related studies in robotics have been on the rise. To mention a few, some target demonstration of emotions through facial expressions [16], some address emotion based activity selection [18] and control of robot behavior through emotional states [19]; and yet some others focus human-robot interaction experiments [20].

The existing emotion related studies in robotics are mostly based on specific emotions such as fear, happiness, anger and sadness with specific learning algorithms [21], [22]. The results of these studies are mostly obtained from simulation environments and lack quantitative comparisons, that ideally should be obtain from real robot experiments. For instance in [21] Salichs et al. proposed a decision making system through implementing specific emotions such as happiness, sadness and fear via Q learning algorithm while in study [22], the authors implement artificial emotions of happiness, anger, sadness and fear to control Khepera robot in a simulation environment via reinforcement learning and artificial neural networks (ANN). Moreover, in [19], the authors performed emotional circuit which located in simulated Khepera robot's neural network architecture that create a link between either input and internal layers or input and output layers. Integrating the emotional circuit to the robot provides to control robot behaviors in order to reach higher fitness in simulation experiments. The work in [19] also used predefined emotion units, hunger, thirst and pain, which enable robots to reach higher fitness value by taking appropriate decisions.

The current literature in robotics indicates that most studies are targeted at implementing predefined emotions and emotional functionalities. As will be explained in the Chapter 3, we do not manually define emotions but ask the question what kind of processes in the cognitive system of (a biological or artificial) agent may yield emotion. In the current realization of our approach, we propose a biologically realistic neural network to explain how some type of emotions may emerge in a cognitive agent, and how emotions can create short-cut in decision making to reach "good enough" outcomes quickly [23].

## *1.2 Contributions*

The main contributions of this thesis can be categorized into two independent parts which belong to Chapter 2 and 3, respectively. In the chapter two we developed a decoding framework that can be used to detect mirror neurons given a set of neural firing. It should be noted that these neural decoding methods resulted in plausible answers to detecting mirror neurons, single neuron and population level object decoding performance and determining object specific neuron types. Moreover, we proposed a new term which was coined as *temporal mirror neuron* to describe how a mirror neuron candidate can change its own behavior throughout conducted experiments.

In the chapter 3 we propose that the regulation of neuronal energy for cognitive processes may lead to emergent behaviors which may be associated with and explained by the emotional state of the agent. Unlike existing studies in the literature we realized emotion based self-regulatory mechanism for computational energy conserving on a humanoid robot with simple cognitive architecture.

## *1.3 Thesis Organization*

In the second chapter we briefly introduce the motivation of the mirror neuron and our object decoding studies. Then, the obtained neural data set features explained and derivation of the evaluation metrics were introduced in detail. This chapter also presents the performed methods, obtained results and discussion of the study.

In the third chapter we summarize our motivation on emotions and biological background of the proposed method. Then, the data flow, experimental and hardware setups explained in detail. In subsections, the results of emotion based network dynamics and analysis of neural energy conservation were provided with emphasizing biological plausibility of the performed method and obtained results. In last chapter we provide conclusions of the conducted research and future work of the studies.

# CHAPTER II

# ACTION RECOGNITION (MIRROR NEURONS)

This chapter outlines our study on action recognition mechanism which exists in a monkey's brain. More specifically, mirror neurons and their neural representation in the monkey's area F5. In this study we firstly focus on detecting mirror neurons trough performing object level decoding and event-wise cross decoding in the available data set. In that we perform object level decoding via cross validated regression algorithm to achieve generalization performance by eliminating over-fitting problem. To apply event-wise cross decoding for detecting mirror neurons, we transfer obtained weight matrix in action execution event to action observation, and vice versa. As a result we find *best decoder* neurons from the neuron population and combining these *best decoder* neurons give rise to a multi-unit classifier which can decode set of four objects with high accuracy rate. The obtained results lead to detection mirror neurons and elimination of canonical neurons, which are located in F5 but they have no mirror characteristic, in available data set. By doing so we implement same cross validated regression algorithm on each period's neural firing data and object specific decoding performance extracted for each neuron. As an important finding of the study, we propose a "temporal mirror neuron" concept that imply a labeled mirror neuron can change its own behavior to engage with different objects throughout experiment duration. Note that, all these procedures, including visualization, are automatically generated by a software framework which was developed by us for neural decoding studies.

## *2.1 Experiments and Neural Data Set Features*

In this section and following subsections, we will introduce experiment conditions and neural data set features. The experiments carried out by project collaborators with four objects which have different grasping types in four experiment conditions which are listed as observation with cue (CUEOBS), observation without cue (NOCUEOBS), object fixation with cue (CUE-OBJFIX) and execution (EXECUTION). All grasping movements of the monkey are illustrated in the Figure 2 and the experimenter also grasps the object in same manner with the monkey.

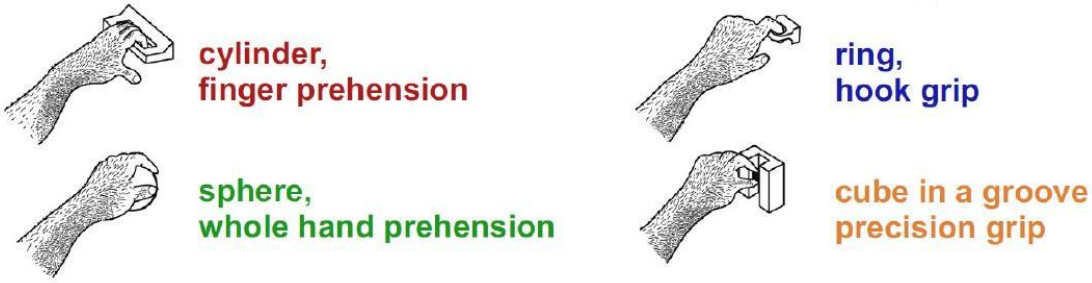
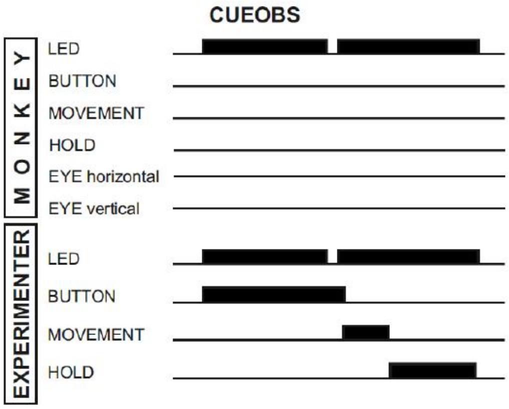


**Figure 2:** Purposeful movements and object grasping types

The conducted experiment procedures begin with when the monkey reaches a selected 3D object, fixate it and press a key in till LED diminish. Throughout an experiment, the monkey located on experimenter's right side and all experiment components are visible to the monkey. The monkey's unit spiking data and experimenter's kinematic information are recorded for preprocessing data analysis to answer what kind of information whether kinematic data or an object's features are possibly responsible for firing a unit while observing or executing an action. To do this various experiments with different conditions have been conducted with different objects and they are explained following subsections in detail.



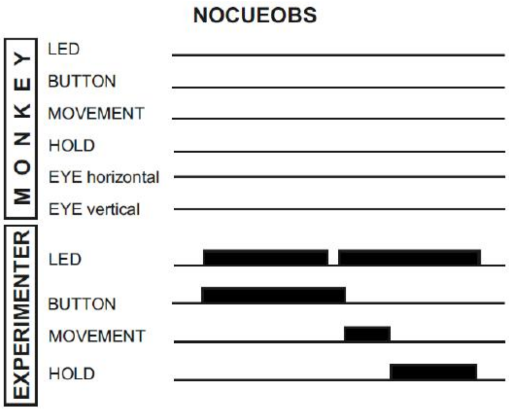


**Figure 3:** Observation with cue

**Figure 4:** Observation without cue

### 2.1.1 Experiment conditions

*2.1.1.1 Observation with cue (CUEOBS)*

In this experiment condition cueing LED is on and the monkey observes both experimenter's movements while reaching and grasping an object. The event time line was depicted in the Figure 3 and during event time span experimenter's kinematic data are recorded.

*2.1.1.2 Observation without cue (NOCUEOBS)*

In this condition, the cueing LED was off for monkey and the experimenter was performing instructions from a screen which was not visible to the monkey. It should be noted that observation with and without cueing LED conditions were randomly performed with no specific order. The event time line for NOCUEOBS can be seen in the Figure 4 and kinematic data are available for postprocessing purpose.

*2.1.1.3 Object fixation with cue (CUE-OBJFIX)*

Both the monkey and the experimenter take no reaching and grasping actions, and cueing LED was visible to the monkey. There are no movements that can be seen in the Figure 5. Therefore, comparing this event's neural data with other event will enable us to make reasonable conclusions about a unit set behaviors towards a specific

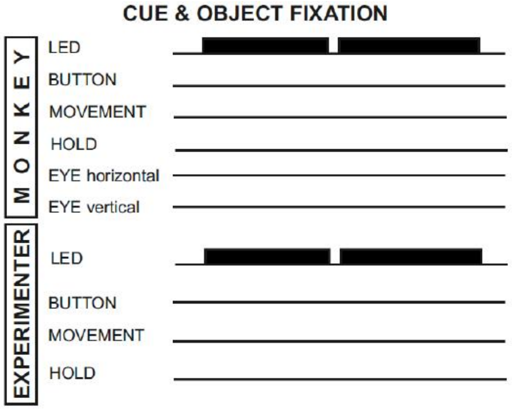


Figure 5: Object fixation with cue

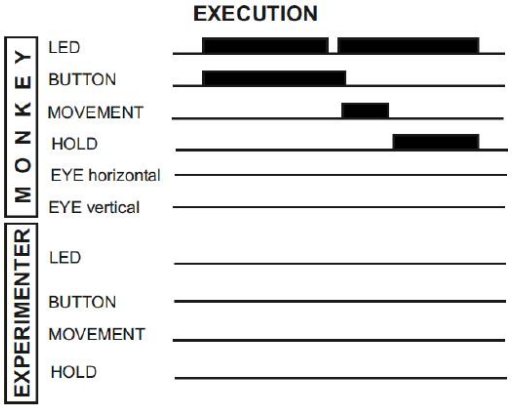


Figure 6: Execution

object and and experiment condition.

*2.1.1.4 Execution (EXEC)*

The monkey executes reaching and grasping task with visible cueing LED in this event (see Figure 6). The interpretation of this event neural data are important to label whether a unit has mirror neuron characteristics or not.

### 2.1.2 Neural data set specifications

In this part we will describe available neural data which are extracted in the area F5. The neural data consist of 192 enumerated neurons and the structure of firing vectors vary from each experiment condition. For instance, some neurons have no spike firing vector for sphere object in all experiment conditions and some neurons have fully populated firing vectors for each object and experiment condition. Therefore, the neural data set needs to be preprocessed to construct necessary input reading and output matrix. Hence, the unit information table was constructed for each neurons, as depicted Table 1 and Table 2, which contains number of spike reading vectors for each object and condition. A spike reading vector's size 1 × 7000 and contains 1 in each cell to mark firing in specific time, otherwise it contains 0 for non-activation. Note that the Table 1 and Table 2 data were constructed via customized function of the software framework and it represents firing information about unit 10 and unit

13. Constructing unit information table leads us to see that there are 68 units that have an $event \times objects$ matrix which is fully populated with spiking readings and we will analyze these units' data for decoding objects. Any missing cell information of $event \times objects$ matrix may lead to insufficient conclusions about specific unit behavior in an experiment conditions. For instance, we need both CUEOBS or NOCUEBS and EXECUTION data together to decide whether a unit can decode an objects and represents mirror neuron characteristics.

| **unit 10** | Cylinder | Sphere | Ring | Cube | Purposeless |
|---|---|---|---|---|---|
| CUEOBS | 31 × 7000 | 10 × 7000 | 10 × 7000 | 20 × 7000 | 10 × 7000 |
| NOCUEOBS | 32 × 7000 | 10 × 7000 | 10 × 7000 | 21 × 7000 | 10 × 7000 |
| CUE-OBJFIX | 10 × 7000 | 10 × 7000 | 10 × 7000 | 10 × 7000 | 13 × 7000 |
| EXEC | 11 × 7000 | 10 × 7000 | 10 × 7000 | 10 × 7000 | [ ] |

**Table 1:** Condition and object matrix for unit 10

As can be seen Table 1 and Table 2 $condition \times object$ matrices, some data are missing and this may vary from one unit to the other. That is why, we constructed an unit index vector for each experiment event to mark "best" units for each objects. For instance, unit 10 can be good candidate for applying decoding procedures since there are unit readings for all $condition \times object$ matrix cell, yet unit 13 seems not to be a good candidate.

| **unit 13** | Cylinder | Sphere | Ring | Cube | Purposeless |
|---|---|---|---|---|---|
| CUEOBS | 10 × 7000 | [ ] | 10 × 7000 | 9 × 7000 | 9 × 7000 |
| NOCUEOBS | 10 × 7000 | [ ] | 10 × 7000 | 10 × 7000 | 10 × 7000 |
| CUE-OBJFIX | 10 × 7000 | [ ] | 10 × 7000 | [ ] | 10 × 7000 |
| EXEC | [ ] | [ ] | [ ] | [ ] | [ ] |

**Table 2:** Condition and object matrix for unit 13

## *2.2 Evaluation Metrics*

In order to evaluate decoding performance of a unit we propose two evaluation metrics related to absolute specificity and relative specificity of a neuron for a given object set. By absolute specificity we underline that the prediction performance of a single neuron in only one class of an object which is denoted by in class percent (ICP). By relative specificity we highlighted that the prediction performance of a neuron while performing in all available classes which is shown with class percent (CP) in below table. The $object \times prediction$ matrices aer constructed for all units (see Table 3). In below table *class count(CC)* refers to how many times predicted error of an object is less than or equal to the threshold, *class percent (CP)*, relative specificity, is prediction rate over total number of successful prediction among all objects while *in class percent (ICP)*, absolute specificity, is prediction rate over total number of successful prediction in same class.

| **unit 10** | Class Count | Class Percent | In Class Percent |
|---|---|---|---|
| Cylinder | 1 | 7.6923 | 10 |
| Sphere | 5 | 38.4615 | 50 |
| Ring | 2 | 15.3846 | 20 |
| Cube | 5 | 38.4615 | 50 |

**Table 3:** Unit10 execution condition prediction rates

Note that total number of prediction for sphere object is 10 and among those predictions only 5 of them are above the threshold value. The CC, CP and ICP values for sphere object calculated based on following equations:

$$CC_{sphere} = 5,$$

$$CC_{total} = 13,$$

$$spherePrediction_{total} = 10,$$

$$CP_{sphere} = CC_{sphere} \times 100.0/CC_{total},$$

$$ICP_{shpere} = CC_{sphere} \times 100.0/spherePrediction_{total}$$

## *2.3 Methods*

### 2.3.1 Linear regression based decoding

Linear regression is a data analysis method that enables us to extract linear relationships among given variables. In that we apply linear regression on preprocessed neural data to predict a decoded object for a specific neuron in an experiment condition. The applied linear model is formulated in equation (1)

$$X * W = Y \tag{1}$$

where $Y$ is *response variable* and $X$ is *independent variable* which are referring to an object id vector and preprocessed neural data matrix, respectively. $W$ is a weight matrix that enable us to predict decoding parameters between neuron readings and objects. In equation (2), $Y$ vector elements consist of object ids, yet these values can be any kinematic parameters need to be predicted such as aperture, finger angles, arm velocity, etc.

#### *2.3.1.1 Single unit linear regression*

Equation (2) constructed with unit 10's execution condition data as $X$ with size of $40 \times 14$ since common number of spiking vector is 10 and we are predicting 4 objects which are enumerated from 1 to 4 as cylinder, sphere, ring and cube, respectively. The $X$ matrix can be constructed with random or consecutive order based on selection parameter for spiking vector. Note that window size and step size are performed as preprocessing step on raw neural data to reduce dimension from $1 \times 7000$ to $1 \times 14$. Furthermore, $X$ matrix readings constructed in sequential order based on their reading ids. The $Unit10 - Obj1 - r1$ stands for unit 10's first window sized spiking vector for object one.

$$\begin{pmatrix} \text{Unit10-Obj1-r1} \\ \text{Unit10-Obj2-r1} \\ \text{Unit10-Obj3-r1} \\ \text{Unit10-Obj4-r1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \text{Unit10-Obj1-r10} \\ \text{Unit10-Obj2-r10} \\ \text{Unit10-Obj3-r10} \\ \text{Unit10-Obj4-r10} \end{pmatrix} \times W = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \tag{2}$$

The predicted output values' error vector can be obtained after deriving $W$ vector from equation (2) by using equations in (3) and (4).

$$X * W = Y_{pred} \tag{3}$$

$$Y - Y_{pred} = Y_{error} \tag{4}$$

squared $Y_{error}$ vector elements with a threshold value (e.g. 0.5) paves the way for a rate that which is successfully predicted by this vector.

#### *2.3.1.2 Multi unit linear regression*

The regression procedures are same for single unit and multi units. In multi unit, equation (2)'s $X$ matrix was expanded the number of neuron population, accordingly. It should be noted that adding more units will blindly decrease $Y_{error}$ vector elements.

$$\begin{pmatrix} \text{Unit10-Obj1-r1} & \text{Unit19-Obj1-r1} \\ \text{Unit10-Obj2-r1} & \text{Unit19-Obj2-r1} \\ \text{Unit10-Obj3-r1} & \text{Unit19-Obj3-r1} \\ \text{Unit10-Obj4-r1} & \text{Unit19-Obj4-r1} \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \vdots & \vdots \\ \text{Unit10-Obj1-r10} & \text{Unit19-Obj1-r10} \\ \text{Unit10-Obj2-r10} & \text{Unit19-Obj2-r10} \\ \text{Unit10-Obj3-r10} & \text{Unit19-Obj3-r10} \\ \text{Unit10-Obj4-r10} & \text{Unit19-Obj4-r10} \end{pmatrix} \times W = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \tag{5}$$

In equation (5)'s $X$ matrix populated with neuron 10 and 19's spiking vectors in consecutive order.

### 2.3.2 Cross validation

To eliminate over-fitting in regression results, leave one out cross validation (LOO) was performed to $X$ input matrices of in equation (2) and (5). The LOO execution steps are introduced in Algorithm 1 and applied on single and multi unit $X$ matrices. In that the $X$ matrix which consists of neural activation data divided into two subsets which are training and test sets. One sample from $X$ matrix which is a row that contains spiking vector picked as test set, then remaining $X$ matrix's elements used as training set to derive $W$ vector. At the end, this $W$ vector performed with test set and prediction error noted. By this way, the neural data is not only employed on training set but also on the test set for mitigating over-fitting.

```
Data: Preprocessed $X$ matrix
Result: Mean error for leaving specific spiking reading vector
R = number rows in $X$ matrix;
for $k$= 1 to $R$ do
    Let $(X_k, Y_k)$ be the $k^{th}$ record;
    Temporarily remove $(X_k, Y_k)$ from the data set;
    Train on the remaining $R-1$ data point/samples;
    Note your error for $(X_k, Y_k)$;
end
When you've done all point/samples report mean error;
```

**Algorithm 1:** Leave One Out Cross-Validation Pseudo code

## 2.4 Results and Discussions

In this section two methods for detecting mirror neurons and object level decoding will be explained in detail. The goal of decoding was to find which neurons are effective in decoding the object type (which uniquely identified the motor action) in execution condition, to find which neurons are effective in decoding the object type (or the action directed towards it) in the observation condition, most critically, to assess whether transfer between execution and observation decoders (i.e. cross-decoding) can be observed. Note that leave one out cross validation applied on all regression tables and each neuron number ranked in decreasing order according to absolute (ICP) and relative (CP) specificity. It is observed that using absolute specificity give rise to more reliable decoding performance for a given neuron. Although we have four different experiment conditions, due to the considerable amount of the neural data we only provide the action observation without cue and action execution rank tables.

### 2.4.1 Mirror neuron detection methods

#### *2.4.1.1 Execution and observation decoding*

As stated in the introduction, it is known that mirror neurons in the area F5 of macaque monkeys have neural activation responses for specific type of grasp execution and observation. Based on this fact, all neurons were ranked according to absolute specificity metric for each individual object in order to observe whether the same neurons could be found both in action observation and action execution.

| **Rank** | Cylinder | | Sphere | | Ring | | Cube | |
|---|---|---|---|---|---|---|---|---|
| | CP | ICP | CP | ICP | CP | ICP | CP | ICP |
| 1 | 55 | 78 | 9 | 51 | 134 | 51 | 84 | 93 |
| 2 | 78 | 40 | 80 | 61 | 145 | 93 | 148 | 54 |
| 3 | 40 | 55 | 180 | 93 | 56 | 38 | 105 | 129 |
| 4 | 110 | 51 | 125 | 38 | 38 | 94 | 77 | 148 |
| 5 | 168 | 54 | 22 | 53 | 94 | 21 | 166 | 21 |
| 6 | 179 | 93 | 127 | 18 | 121 | 91 | 109 | 51 |
| 7 | 181 | 110 | 146 | 147 | 127 | 121 | 129 | 77 |
| 8 | 82 | 181 | 106 | 34 | 24 | 134 | 78 | 78 |
| 9 | 92 | 21 | 53 | 45 | 17 | 24 | 23 | 83 |
| 10 | 128 | 45 | 147 | 54 | 91 | 54 | 41 | 84 |
| 11 | 105 | 59 | 61 | 76 | 108 | 59 | 83 | 131 |
| 12 | 19 | 168 | 104 | 91 | 76 | 76 | 131 | 166 |
| 13 | 106 | 18 | 76 | 94 | 82 | 110 | 10 | 10 |
| 14 | 107 | 92 | 42 | 104 | 92 | 145 | 42 | 45 |

**Table 4:** Single unit linear LOO regression prediction ranks for NOCUEOBS event

In Table 4 and Table 5, observation and execution conditions' first 14th rankings are illustrated the cells with same neuron ids are automatically colored with the same RGB values. It can be stated that the neurons which are appeared in both observation and execution tables can be marked as a strong mirror neuron candidates among neuron population [10]. More interestingly, it is observed that some mirror neuron candidates behave like general decoder for all objects (e.g. neuron 54), some of them show selective behavior towards specific objects (e.g. 93), and some of them are single object decoder (e.g. 131).

| **Rank** | Cylinder | | Sphere | | Ring | | Cube | |
|---|---|---|---|---|---|---|---|---|
| | CP | ICP | CP | ICP | CP | ICP | CP | ICP |
| 1 | 78 | 59 | 133 | 15 | 110 | 56 | 10 | 93 |
| 2 | 19 | 80 | 38 | 56 | 92 | 34 | 134 | 18 |
| 3 | 59 | 84 | 23 | 42 | 55 | 92 | 109 | 40 |
| 4 | 168 | 168 | 142 | 53 | 76 | 131 | 127 | 56 |
| 5 | 17 | 15 | 43 | 95 | 9 | 54 | 129 | 131 |
| 6 | 121 | 34 | 95 | 18 | 95 | 93 | 41 | 109 |
| 7 | 105 | 77 | 125 | 20 | 167 | 94 | 148 | 41 |
| 8 | 80 | 93 | 24 | 23 | 23 | 95 | 43 | 53 |
| 9 | 84 | 105 | 20 | 51 | 34 | 166 | 93 | 54 |
| 10 | 142 | 131 | 148 | 54 | 94 | 167 | 91 | 59 |
| 11 | 77 | 17 | 42 | 77 | 141 | 150 | 18 | 82 |
| 12 | 34 | 54 | 146 | 84 | 166 | 9 | 24 | 91 |
| 13 | 128 | 56 | 76 | 94 | 41 | 15 | 38 | 107 |
| 14 | 22 | 78 | 79 | 132 | 83 | 21 | 40 | 148 |

**Table 5:** Single unit linear regression LOO prediction ranks for EXECUTION event

With the available data we highlight that for observation condition 12 distinct neurons that were marked as high performers ($\geq$ 40% successful prediction vs. chance level of 25%) were also found to be high performers in action execution condition.

### 2.4.2 Cross decoding

In order to apply cross decoding on observation and execution conditions we firstly extract W matrix for a specific condition. Then, transfer this obtained W to the other experiment condition in order to predict a set of object. As illustrated in Figure 7, by "transfer" we imply that the decoder parameters found with the neural discharge in the observation conditions are directly applied to neural firing recorded in the execution conditions, and vice versa.

The success of a decoder trained with data from one condition, on the data obtained from another condition is very important as it indicates that the representations in both conditions are the same. This $W_{conditon}$ transfer enable us to determine

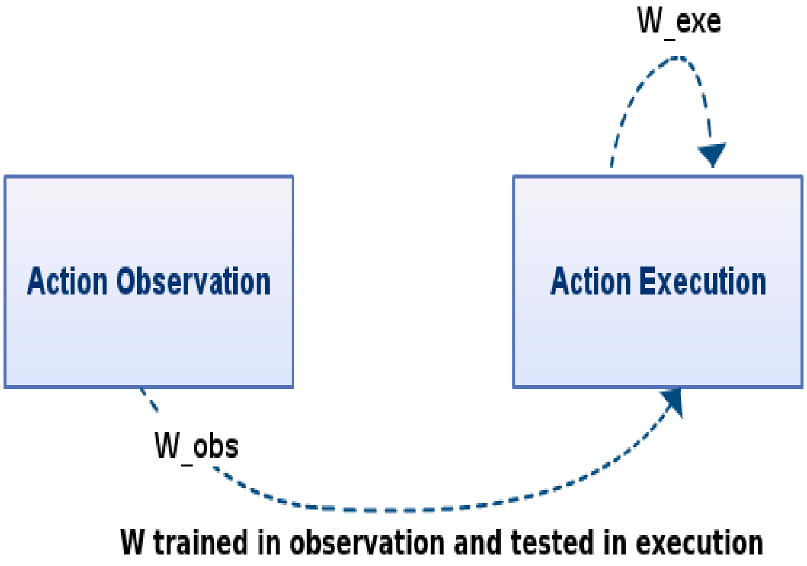


**Figure 7:** Weight transfer among conditions

both best decoder neurons for objects and best mirror neuron candidates among available neuron set. To observe decoding performance *in class percent* (ICP), or absolute specificity, used as a metric and sorted in decreasing order and first 7th rank are shown in Table 6 and Table 7.

| **Rank** | Cylinder | | Sphere | | Ring | | Cube | |
|---|---|---|---|---|---|---|---|---|
| | CP | ICP | CP | ICP | CP | ICP | CP | ICP |
| 1 | 91 | 133 | 109 | 18 | 147 | 41 | 21 | 38 |
| 2 | 95 | 91 | 104 | 53 | 82 | 61 | 38 | 93 |
| 3 | 133 | 92 | 18 | 10 | 41 | 53 | 55 | 106 |
| 4 | 180 | 105 | 83 | 83 | 45 | 51 | 121 | 131 |
| 5 | 181 | 131 | 43 | 105 | 61 | 79 | 132 | 51 |
| 6 | 76 | 167 | 56 | 168 | 110 | 82 | 141 | 146 |
| 7 | 134 | 17 | 105 | 9 | 57 | 106 | 146 | 150 |

**Table 6:** Single unit linear regression prediction ranks for EXECUTION ($W_{nocueobs}$ used)

In Table 6's cells are filled with neuron ids in which $W_{nocueobs}$ transferred to execution condition. Meanwhile, Table 7 was constructed with regression in execution

| **Rank** | Cylinder | | Sphere | | Ring | | Cube | |
|---|---|---|---|---|---|---|---|---|
| | CP | ICP | CP | ICP | CP | ICP | CP | ICP |
| 1 | 78 | 84 | 23 | 56 | 92 | 56 | 180 | 59 |
| 2 | 84 | 109 | 38 | 15 | 132 | 18 | 107 | 107 |
| 3 | 127 | 21 | 125 | 53 | 9 | 109 | 20 | 148 |
| 4 | 77 | 59 | 43 | 80 | 10 | 15 | 133 | 150 |
| 5 | 22 | 80 | 142 | 94 | 146 | 21 | 61 | 93 |
| 6 | 21 | 127 | 147 | 168 | 18 | 34 | 148 | 166 |
| 7 | 80 | 129 | 141 | 18 | 17 | 93 | 10 | 109 |

**Table 7:** Single unit linear regression prediction ranks for EXECUTION ($W_{exec}$ used) event

condition where $W_{exec}$ used for object decoding. It is observed that with cross decoding between observation to execution 5 distinct neurons perform well to decode in both conditions while decoding sphere and cube objects. Table 8 shows the decoding performance of listed neurons in pure execution and prediction with transferred $W_{nocueobs}$. Cross decoding performance rates are shown in Table 8. The high performer neurons for cross decoding are 53 and 18 with 70% and the minimum cross decoding rate is 33%.

| **Rank** | Cylinder | | Sphere | | Ring | | Cube | |
|---|---|---|---|---|---|---|---|---|
| | CP | ICP | CP | ICP | CP | ICP | CP | ICP |
| Cross decoding ($W_{obs}$) 53 | 12.50 | 20.00 | 43.75 | 70.00 | 43.75 | 70.00 | 0.00 | 0.00 |
| Execution ($W_{exe}$) 53 | 25.00 | 70.00 | 32.14 | 90.00 | 14.29 | 40.00 | 28.57 | 80.00 |
| Cross decoding ($W_{obs}$) 168 | 36.36 | 40.00 | 45.45 | 50.00 | 18.18 | 20.00 | 0.00 | 0.00 |
| Execution ($W_{exe}$) 168 | 23.08 | 60.00 | 34.62 | 90.00 | 11.54 | 30.00 | 30.77 | 80.00 |
| Cross decoding ($W_{obs}$) 18 | 22.22 | 20.00 | 77.78 | 70.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Execution ($W_{exe}$) 18 | 13.79 | 40.00 | 27.59 | 80.00 | 31.03 | 90.00 | 27.59 | 80.00 |
| Cross decoding ($W_{obs}$) 150 | 20.00 | 22.22 | 30.00 | 33.33 | 20.00 | 22.22 | 30.00 | 33.33 |
| Execution ($W_{exe}$) 150 | 18.52 | 55.56 | 22.22 | 66.67 | 25.93 | 77.78 | 33.33 | 100.00 |
| Cross decoding ($W_{obs}$) 93 | 0.00 | 0.00 | 27.27 | 30.00 | 18.18 | 20.00 | 54.55 | 60.00 |
| Execution ($W_{exe}$) 93 | 22.22 | 60.00 | 14.81 | 40.00 | 29.63 | 80.00 | 33.33 | 90.00 |

**Table 8:** Single unit linear regression prediction rates for nocueobs2execution and EXECUTION events

It should be noted that same tables are available for CUEOBS, NOCUEOBS and

EXECUTION conditions, yet due to excessive number of data, only *W_nocueobs* transferred results are shared in this subsection. The more information about conducted decoding results can be found in [11].

### 2.4.3 Population level cross decoding

After obtaining cross decoder neurons trough weight transfer from observation to execution condition, we analyzed population level cross decoding among available data set. To do this, we manually select four best “solo decoder” neurons and six randomly selected neurons to construct a neuron pool, then we get all possible two paired permutations to construct input matrices for applying cross decoding. The prediction ranks for these pairs can be seen in Table 9 and prediction rates for these pairs are illustrated in Table 10. The pairs with solo decoder in Table 9 are colored to understand how transferring a weight effects on neuron population decoding performance.

| **R** | Cylinder | | Sphere | | Ring | | Cube | |
|---|---|---|---|---|---|---|---|---|
| | CP | ICP | CP | ICP | CP | ICP | CP | ICP |
| 1 | 18 – 150 | 18 – 17 | 18 – 45 | 18 – 45 | 168 – 108 | 18 – 21 | 53 – 168 | 93 – 45 |
| 2 | 168 – 21 | 18 – 53 | 18 – 104 | 18 – 17 | 18 – 21 | 53 – 150 | 150 – 21 | 93 – 17 |
| 3 | 18 – 53 | 150 – 17 | 53 – 93 | 53 – 93 | 53 – 150 | 108 – 17 | 168 – 104 | 53 – 45 |
| 4 | 18 – 168 | 168 – 21 | 18 – 93 | 17 – 104 | 93 – 21 | 93 – 150 | 108 – 21 | 93 – 108 |
| 5 | 18 – 17 | 17 – 104 | 150 – 45 | 150 – 45 | 150 – 108 | 150 – 108 | 45 – 21 | 17 – 104 |
| 6 | 150 – 17 | 18 – 150 | 45 – 17 | 18 – 53 | 108 – 45 | 150 – 45 | 93 – 45 | 93 – 150 |
| 7 | 45 – 17 | 150 – 104 | 45 – 104 | 18 – 93 | 18 – 93 | 18 – 93 | 53 – 17 | 150 – 168 |

**Table 9:** Multi unit linear regression prediction ranks for EXECUTION ($W_{nocueobs}$ used)

According to obtained tables, it can be seen that some solo decoder neurons can perform well with different pairs while decoding different object. Recall that neuron number with 18, 53 and 168 are solo decoder for sphere object and the pairs with these neurons are emerged not only in sphere role but also in other object with significant performance rate. It should be noted that above tables are constructed via expanding

| **Rank** | Cylinder | | Sphere | | Ring | | Cube | |
|---|---|---|---|---|---|---|---|---|
| | CP | ICP | CP | ICP | CP | ICP | CP | ICP |
| 1 | 100.00 | 70.00 | 87.50 | 70.00 | 100.00 | 60.00 | 100.00 | 50.00 |
| 2 | 57.14 | 60.00 | 66.67 | 60.00 | 75.00 | 55.56 | 100.00 | 50.00 |
| 3 | 50.00 | 44.44 | 62.50 | 50.00 | 71.43 | 50.00 | 100.00 | 40.00 |
| 4 | 50.00 | 40.00 | 57.14 | 50.00 | 66.67 | 33.33 | 100.00 | 40.00 |
| 5 | 50.00 | 40.00 | 57.14 | 44.44 | 60.00 | 33.33 | 75.00 | 40.00 |
| 6 | 50.00 | 33.33 | 57.14 | 40.00 | 60.00 | 33.33 | 71.43 | 33.33 |
| 7 | 42.86 | 33.33 | 57.14 | 40.00 | 42.86 | 30.00 | 66.67 | 33.33 |

**Table 10:** Multi unit linear regression prediction rates for Execution ($W_{nocueobs}$ used)

rows of the input matrix in equation (5) because of adding two window sized firing vector in same row. By this way we change the complexity of the decoding system which was originally used in single neuron decoding. To control the complexity of the system we fixed the row length with the original complexity and shrink paired neurons data to match original window size. Fixing the complexity lead us to observe significant enhancement of the prediction rates. In that, for sphere object the first ranked unit's absolute specificity is 70%, yet fixing complexity gives rise to 100% and similar improvement observed in prediction rates observed for all objects.

# CHAPTER III

# EMOTIONS

This chapter presents our work on how emotion based behaviors may emerge through computational mechanisms. We hold that in addition to basic emotions such as anger and fear that serves bodily well being of the organism, high level emotions such as boredom and affection have evolved to facilitate low cost brain computations. Higher level of emotions can be considered as affective state of the organism or mood, rather than the reflex-like physiologically triggered emotional responses such as fear and anger. In large and complex brains (e.g. primate brains), the neuronal energy consumption for cognition is non-negligible. We propose that for such organisms computational regulatory mechanisms for decision making give rise to behaviors that can be explained by various emotional states. As a proof of concept for this idea, we envision a robotic cognitive system and a select function that we assign a neural cost for its operation. To be concrete, we use a small humanoid robot platform (Darwin-OP) and implement a neural network (Hopfield Network) that allows the robot to recall learned patterns that it sees through its camera. As a model of neural computational energy consumption, we postulate that a change in the state of a neural unit of the network consumes one unit of (neural) energy. Therefore, the total computational energy consumed is determined by the incoming stimuli. The robot is programmed to avoid high energy consumption by showing aversive behavior when the energy consumption is high. Otherwise, the robot demonstrates engaging behavior. For an external observer these responses may be perceived as robots having certain emotional (affectional) preference for input stimuli. In this study in addition to robot experiments, we also emphasize the biological support for our proposal and provide detailed exposition of biological background and its relevance for the hypothesis that (certain) emotions may emerge through computational mechanisms [17].

## *3.1 Biological Background*

It is known that biological systems have limited energy resources to survive and maximize their number of off-springs. Therefore, these systems, through evolution gained physical mechanisms -fur, sharp teeth, tails, etc.- and mental mechanisms-emotions, feelings, social bonding, etc.- to keep themselves alive and take decisions against stochastic and sometimes unpredictable events.

Emotions are one of the vital features which animals use to regulate their behaviors. Animals have emotions in terms of functionality [14], to control attention in order to solve faced immediate problems and make a decision under uncertain conditions [24]. Considering reproduction case in nature [23] mentioned, searching for "perfect mate" may took longer than one can afford. Meanwhile, the living beings have to satisfy other needs such as finding food, shelter and avoiding predators. Instead of searching and so consuming more energy to find "the perfect mate", shorter search and reasonable partner might be preferred as probability of reproducing might be higher in the latter. This choice cannot be evaluated as optimal or conscious behavioral choice [23]. Mate selection can be considered as a cognitive process which needs to be controlled according to cost-benefit trade-offs. Thus, emotions appear to provide organisms computational short-cut mechanisms on cognitive processes to facilitate energy economy via the adoption of "good enough" choices rather than searching for the best. This thesis proposes to reverse the statement "Emotions provide organisms short-cut mechanisms on cognitive processes to facilitate energy economy" as "The computational short-cut mechanisms on cognitive processes to facilitate energy economy give rise to what we call Emotions".

It is important to underline that by “energy” we imply both physical and “computational” energy. The idea behind energy consumption by neurons and emotions can be inferred from literature in neuroscience and neurobiology [25], [26]. These studies pointed out that the brain consumes high level of resting metabolic energy of body in different species. For instance, human brain alone consumes 20% of the total energy consumed by the body. These studies indicate that neuronal activities require considerable amount of energy in order to successfully execute cognitive activities such as reasoning, decision making and vision processing. In [25], Laughlin and his colleagues emphasize that in case of limitation in metabolic energy, the neurons, neuronal codes and neuronal circuits must have evolved to reduce the metabolic demands in order to keep brain’s batteries charged. In this vein, we capture these energy related metabolic features and establish a self-assessment mechanism enabling the robot to regulate its behavior according to the consumed (neural) energy.

## *3.2 Methods*

As outlined in the chapter three’ introduction, we propose that higher level emotions (those that have putatively evolved after the basic emotions of fear, anger etc.) are the behavioral manifestation of self-regulation mechanisms of computational (neuronal) energy expenditure for cognitive processing. To be concrete we implement a Hopfield Network on a small humanoid robot, namely Darwin OP to undertake a pattern recall task. We model the energy expenditure of a Hopfield unit which represents an artificial neuron or neuron population, as the number of times it changes its state during the processing of an incoming stimuli. Based on the total number of state changes, the robot decides the action to take and show either aversive (negative) or engaging (positive) behavior. We aim to show that the behavioral outcome of this energy regulation mechanism can be perceived (by human observer) as an affectional stance of the robot for the input stimuli it is seeing.

The methods which are performed in this chapter have been carried on a PC and humanoid robot platform. In the PC experiments we analyzed how the Hopfield Network dynamics behave while receiving different input patterns that have different levels of noise contamination. This analysis is used to determine a *threshold* value for neuro-computational energy consumption to be used in robot experiments. In the robot experiments, this threshold was used to guide the behavior modulation according to input stimuli. To be concrete, the energy self-monitoring logic used the obtained *threshold* value to generate aversive or engaging behavior.

### 3.2.1 Behaviour trough self regulation of the Hopfield Neural Network

Hopfield Neural Network consists of fully connected artificial neurons (or units) and behave as an auto-associative memory [28]. The Hopfield Neural Network is very suitable for parallel computation, yet its serial implementation can be inefficient for large number units (such as when having each unit represent the pixel of a high resolution retina). Nevertheless, it can be used for storing and recalling a small set of characters with a small retina, and serve as a simple model for neural computation taking place as part of the cognitive processing of a biological brain. In particular, the dynamic nature of the network and the dense connection create a complex system which can be used to define a neural energy measure on which agent behavior may be regulated.

The network dynamics is adopted as given in [28]. The network initialization and network read off required binary (0,1) to bipolar (-1,1) conversion, as the retinal input was represented as thresholded binary images. The output representation of a unit $i$ is shown by $S_i$,

$$S_i = sgn\left(\sum_{jk} W_{ijk} S_j S_k\right) \tag{6}$$

The $sgn()$ is defined as

$$sgn(x) = \begin{cases} -1, & x < 0 \\ \\ 1, & x \geq 0 \end{cases} \tag{7}$$

The weights are calculated as follows

$$W_{ijk} = \sum_{p} \xi_i^p \xi_j^p \xi_k^p \tag{8}$$

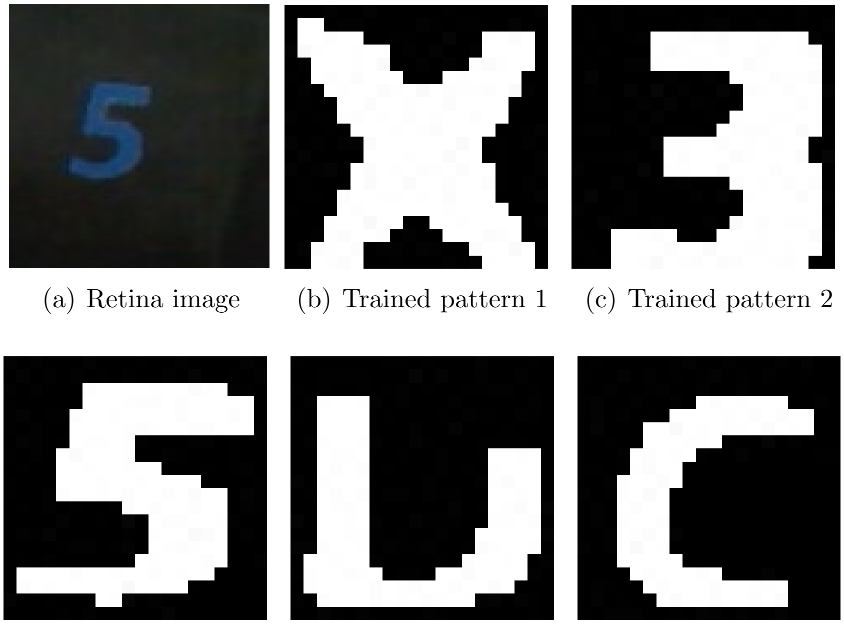
and $p$ indicates the patterns to be stored. In equation 8, $\xi_i^p$, $\xi_j^p$ and $\xi_k^p$ refer to the $j_{th}$, $j_{th}$ and $k_{th}$ bit of the pattern $p$, respectively. Asynchronous update rule is adopted which updates a randomly chosen unit until iteration conditions terminated or convergence is reached.

Implementation procedure of the Hopfield Neural Network begins with obtaining images from the camera of Darwin-OP are segmented based on color and scaled down to $20 \times 20$ pixels. This small image then converted into a binary image and the same binary image is used during learning, i.e. when computing the weight matrix, and also when querying the network for recall. Each pixel in the binary image is associated with a unit of the network.

After determining the weight matrix, to test recall performance, randomly contaminated patterns are generated. The network is initialized with an input image and updated asynchronously until convergence (or until an earlier stop criterion is reached). Hopfield Network creates spurious attractor states; the linear combinations of the stored patterns and their inverses become attractors. Therefore, the network may settle in states which do not directly correspond to a single stored pattern, or in the extreme it may converge to the complete inverse of a stored pattern (see Figure 14(d)).

### 3.2.2 Behaviour trough energy regulation

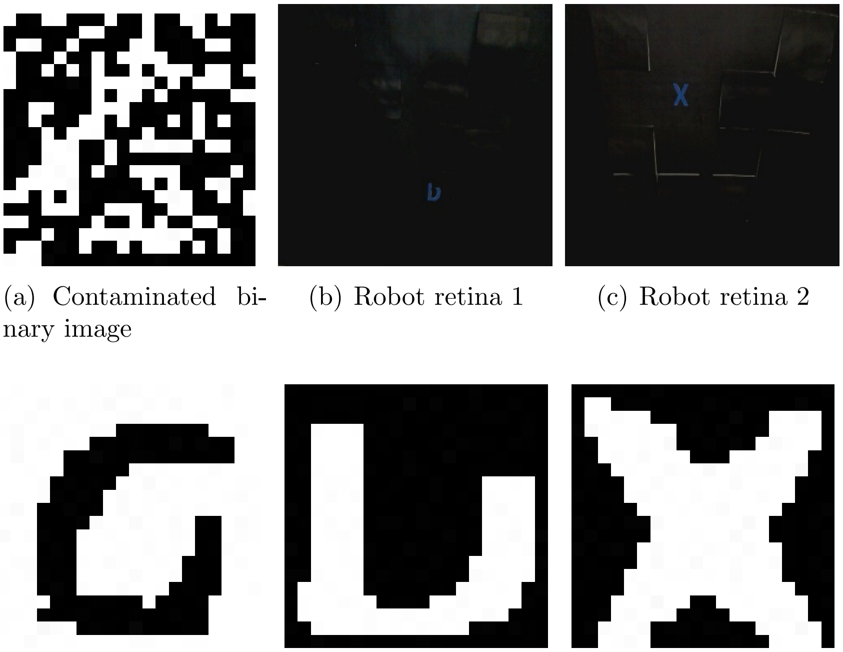
In the scenario we consider, the life cycle of the robot is very simple. It sees its environment (applies visual processing leading to a segmented, thresholded retinal



(a) Retina image (b) Trained pattern 1 (c) Trained pattern 2

(d) Trained pattern 3 (e) Trained pattern 4 (f) Trained pattern 5

**Figure 8:** Retina image and trained patterns



(a) Contaminated binary image (b) Robot retina 1 (c) Robot retina 2

(d) Output pattern 1 (e) Output pattern 2 (f) Output pattern 3

**Figure 9:** Retina images and converged patterns

image) and tries to recall what it has learned before. It does this by loading the input image to the Hopfield Network and running the dynamics until convergence state or the stop criterion is reached. If stop is reached with small energy consumption, the robot displays an engaging behavior; otherwise it displays a negative behavior. In the current implementation we used a simple squatting behavior to indicate that the robot would have evaded the current stimulus. This behavior is a substitute for the case where the robot would move away from the current input stimuli with the hope to find stimulus that may cost less brain energy.

With the stochastic update rule it is guaranteed that the Hopfield Network will converge a stable pattern. However, the time it takes to converge and the number of state changes that it would take is a complex function of the input stimulus. We postulated that each state change costs 1 unit of energy and defined a threshold energy level; after which the robot will disengage with the current stimuli. For those stimuli that convergence can be obtained with small amount of energy the robot shows an engaging behavior. Note that this may happen for a completely novel input as well as for those inputs that are similar to one of the learned patterns. For an outside observer it may be viable to say that robot likes this character rather than that character.

## *3.3 Experimental Setup*

Our framework includes the humanoid robot platform, a display panel that hosts the input stimuli that can be presented to the robot and supporting PC hardware. Hardware experiments were carried out using Darwin-OP as a continuous interactive loop. The evaluation experiments for the input patterns were run only by using a PC. These experiments were used to collect extensive data to assess the dynamics of energy consumption based on noise contaminated input patterns. The robot experiments were conducted to show how the proposed system can be embedded in a cognitive system of the robot. At the beginning of the experiment, the robot was placed so that it can see the panel that holds the input stimuli patterns. Input stimuli can be hidden or revealed using post-its. We assume that normally one pattern will be visible to the robot. This eliminates the need for an explicit attention system for the robot, and a simple color based segmentation can be used for vision processing.

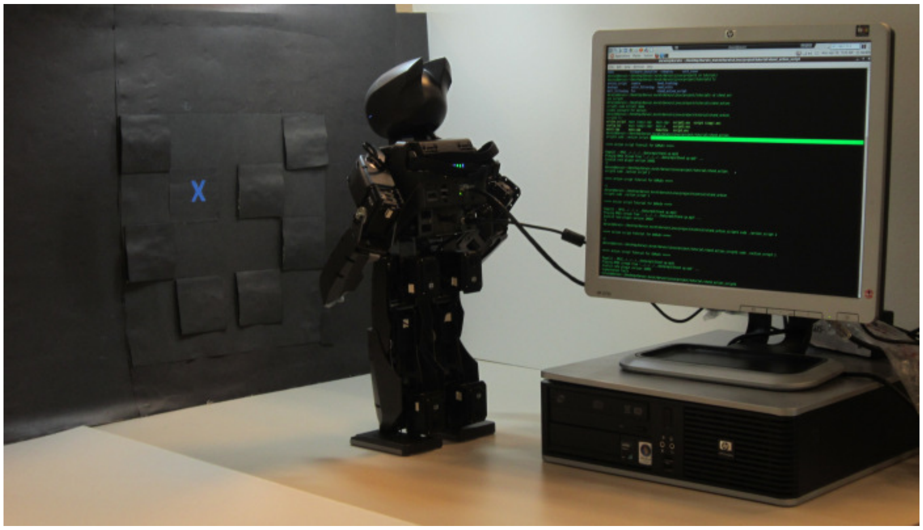


**Figure 10:** Experimental setup

### 3.3.1 Hardware setup and data flow

The hardware setup consists of a base-station (control PC) and the Darwin-OP robot which sets up a continuous behavior loop. The evaluation of PC experiments are run off-line. For these, stimuli are collected from the camera of the robot in order to train Hopfield network and same training stimuli are used in both PC and robot experiments. These obtained stimuli are contaminated with uniform noise systematically to probe the neural energy consumption and the convergence dynamics of the Hopfield network. The contaminated binary image is shown in Figure 14(a).

In the robot experiments, client/server programs both on Darwin-OP and the PC are executed, then experiment circle runs as itself without having external actor interaction. The conducted experiments' processes highlight that the robot can regulate its own states to minimize energy consumption while successfully executing recognition process and presenting a behavior.

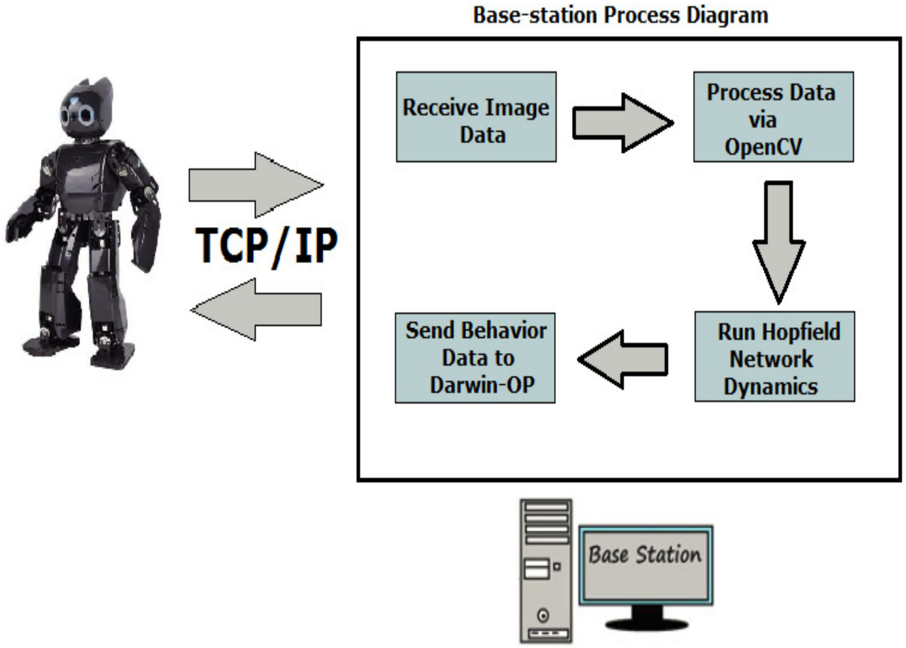

Base-station Process Diagram
TCP/IP
Receive Image Data
Process Data via OpenCV
Run Hopfield Network Dynamics
Send Behavior Data to Darwin-OP
Base Station

**Figure 11:** Data Flow between Darwin-OP and PC

The data flow between base-station and Darwin-OP is illustrated in Figure 11 In that, the base-station receives image RGB values via TCP packages over network from USB Camera which is located in the front face of the robot. In the base-station, these obtained values are scaled and converted into binary format via OpenCV libraries. Then, trained Hopfield Neural Network is initialized with these binary values to perform the network dynamics in asynchronous update mode until convergence occurs or iteration conditions terminated. In parallel the number of flipped bits due to network dynamics is recorded. That number is used as the output of consumed energy function which will be compared to a predetermined *threshold* value. If the total number of flipped bits is above the threshold, the robot is commanded to display a negative behavior, otherwise the robot is commanded to display a positive response.

### 3.3.2 Darwin-OP and behavior execution process

In this study all software architectures are implemented on both on a PC and Darwin-OP (Dynamic Anthropomorphic Robot with Intelligence- Open Platform). Darwin-OP is designed in modular and extensible structures in order to facilitate humanoid robot research. Darwin-OP has been built with various sensors and hardware structures with standard PC architecture which are 3-axis Gyroscope, 3-axis Accelerometer, Atom Z530 CPU, USB Camera. The detailed information about software and hardware of Darwin-OP can be found at [27]. Darwin-OP's software framework has been designed in a hierarchical way to be modular and independent. In experiments, the robot runs the behavior presenting program by using software development kit's, namely open-DarwIn-SDK, and existing structures run on a GNU/Linux based operating system.

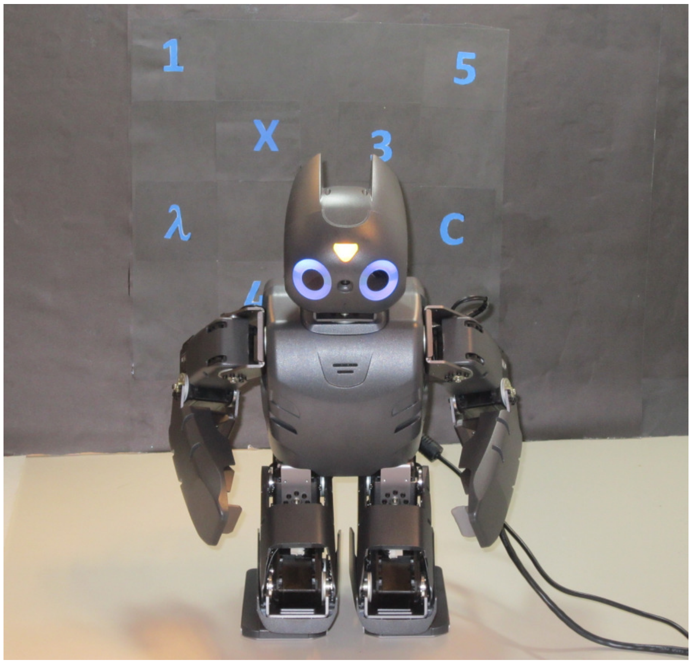


**Figure 12:** Darwin-OP

In the experiments, the robot is connected with a PC via wireless and wired communication to send and receive data over TCP/IP and the wired Ethernet in order to use graphical interface of the operating system and run programs for actuations and vision processing. In this system, the robot sends RGB values sensed from its camera to remote computer and receives data for representing behavior. The received data enable robot to activate its actuators to execute these behaviors. The behaviors consist of two states which are categorized into positive and negative states. In the positive state, the robot moves its head around and claps with voice announcement. On the other hand, the robot will consecutively sit down and stand up to present the negative state's behavior. Note that these states are selected in the base-station through calculating the total number of changed bits while running Hopfield Neural Network dynamics.

## *3.4 Results and Discussions*

This section presents the results obtained during the PC and robot experiments in detail. These results are evaluated to indicate how our approach may be seen as a biologically realistic model of what is observed in biological systems.

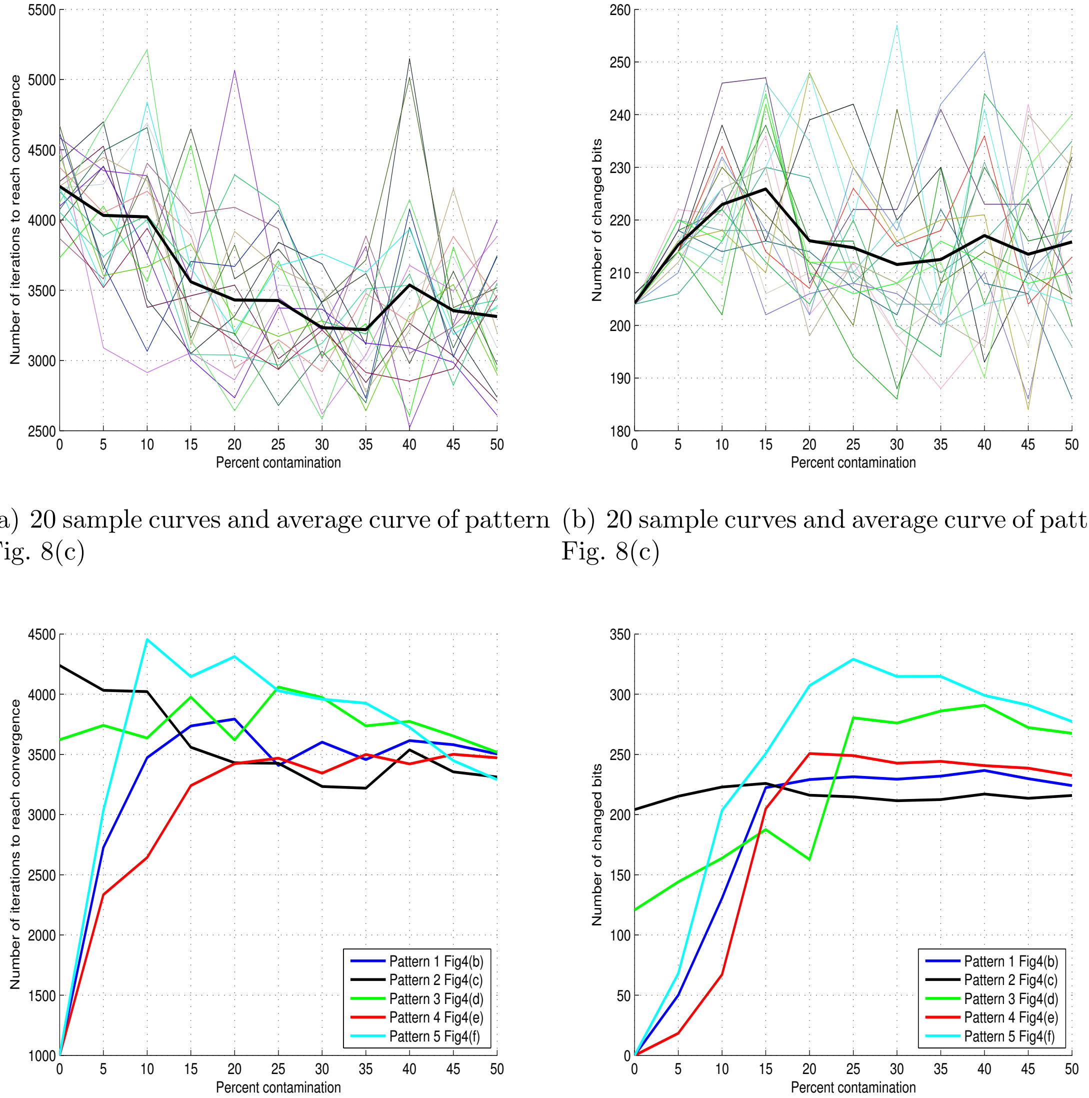
The dynamics of the Hopfield Neural Network coupled with energy self-monitoring provides a neuron-based framework to implement. The number of changed pixel values are used as an output of the consumed energy by neurons. In this implementation we use a threshold value to determine whether the robot will display a positive or negative response towards a given stimuli. In other words, this value is applied to modulate the behavior of the robot for the received input.

### 3.4.1 Analysis of energy conservation

In the PC experiments reported in this study, the maximum number of iterations that the Hopfield Neural Network was allowed to run was 10000. If no unit changes its state in response to the application of asynchronous update rule for a total of 1000 iterations, the network is deemed to have converged. This state of the network is thus considered as the pattern recalled by the network. In general the converged pattern may or may not be one of the patterns that has been learned by the network.

It should be stated that Figure 13(a) and Figure 13(b) are obtained for the second trained pattern to get 20 samples and in each time contamination rate is increased by 5 up to 50 through PC experiments. These random values can be considered as the contamination rate of the patterns which are taken by the robot's retina and contamination factors can be listed as the light in the environment, the camera focus and the robot's position towards images. This procedure is applied for all trained patterns and the obtained data are plotted in Figure 13(c) and Figure 13(d). The number of iterations and changed bits while converging varies for each patterns and these values depend on the similarities of the trained patterns, consider bottom part of the patterns in Figure 8(c) and 8(d), and the location of randomly changed bits. Due to the characteristics of the Hopfield Neural Network, the network can converge on the inverse of the trained patterns. To get an understanding of how the noise in the input pattern affects the convergence time and the pseudo-energy (that we defined) it consumed we run systematic noise contamination experiments. As can be seen in Figure 13(a) and 13(c), the network does not reach the maximum iteration which is 10000 in all of the simulation runs indicating that with 10000 iteration limit we do not prematurely interrupt the evolution of the network dynamics. Furthermore, the network converges in different number of steps and require more flips, i.e. require more energy depending on the level of noise contamination (see Figures 6(c,d)). Figures 6(a,b) are included to give an indication of the variability of the network dynamics for a typical input pattern. This variability is the result of the the stochastic update rule of the Hopfield Network as well as due the randomness in noise contamination.

These experiments provide considerable amount of data to evaluate the network behavior and facilitate deriving a parameter, namely *threshold*, which will be used in the robot experiments. In the all PC experiments, the network dynamics lead the network state to the convergence state since there is no such energy *threshold* value



(a) 20 sample curves and average curve of pattern Fig. 8(c)

(b) 20 sample curves and average curve of pattern Fig. 8(c)

(c) Average Curves of all patterns

(d) Average Curves of all patterns

**Figure 13:** Converged pattern curves for Fig. 8(c) and average curves of all patterns

| Robot Behavior | Trained Pattern | Untrained Pattern |
|---|---|---|
| Positive | $\iota$ | 1 |
| Negative | $\chi$ | $\lambda$ |

**Table 11:** Pattern and behavior matrix

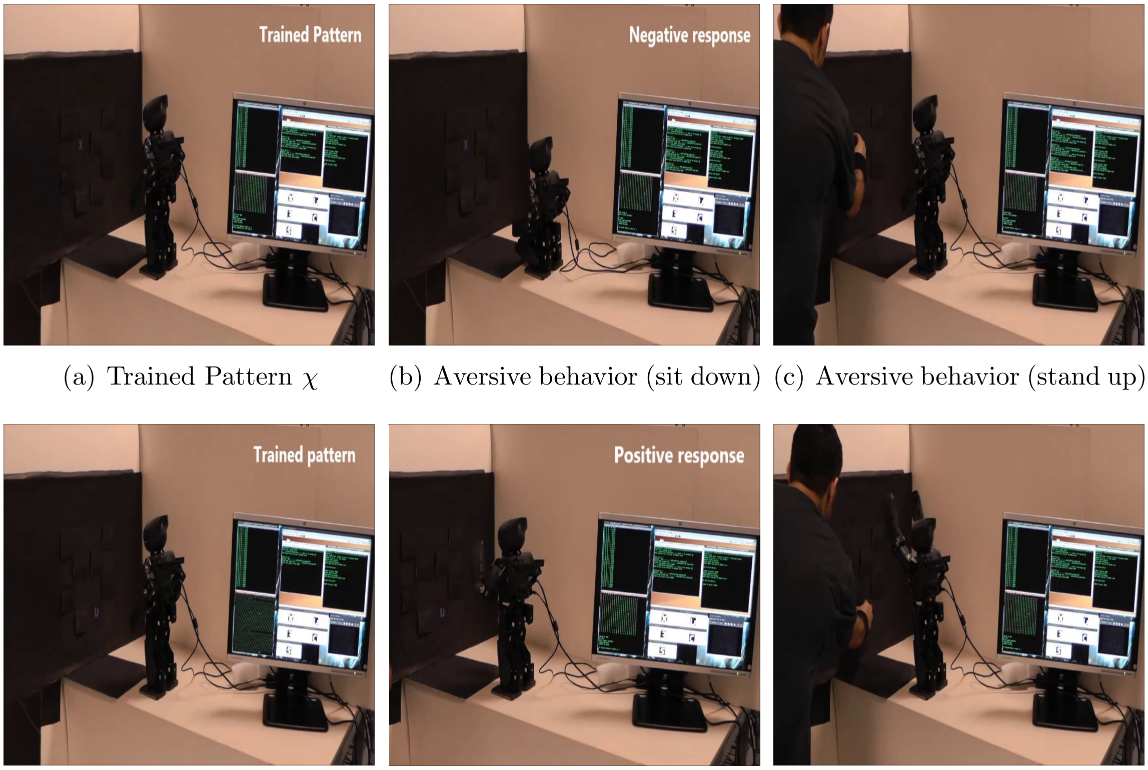
to escape the recognition process. Yet, in the robot experiment the quantitative comparison of the total number of changed bits and energy *threshold* value is applied in deciding whether to display a negative or positive behavior.

### 3.4.2 Emotion based network dynamics

For the robot experiments, the energy threshold value is chosen to be 230 which corresponds approximately of maximum number of the changed bits (see Figure 13(d)). The implemented energy regulation is very simple: for a given input stimulus the number of changed bits is counted and when the threshold is reached the processing (Hopfield Dynamics) is halted and an aversive behavior is generated. Otherwise, the processing continues until convergence is reached. At this time the robot displays a positive engaging behavior. Note that, in a more general setting and, the energy threshold should be considered as an adaptive parameter that can be learned by the organism.

Nevertheless, the current robot implementation captures the basic idea that neural energy might be one of the determinants of behavior. Moreover, the implemented behavior regulation can be considered biologically realistic since under the unpredictable conditions of the environment, an organism should decide the line of action to take in order to satisfy its metabolic needs while minimizing the amount of neuronal energy. Thus, this energy consumption and decision making trade off yields the adaptability of the agent for undertaking complex tasks in the unknown environment.

Moreover, the attached media which can be seen in Figure 14 includes the robot experiment which shows the robot has self-assessment mechanisms to decide the line of



(a) Trained Pattern $\chi$ (b) Aversive behavior (sit down) (c) Aversive behavior (stand up)

(d) Trained Pattern $\iota$ (e) Engaging behavior (clapping) (f) Engaging behavior (clapping)

**Figure 14:** Experiment media snapshots

action to take while receiving trained or untrained stimuli (see Table 11 pattern/behavior matrix). As mentioned before, the robot may display negative or positive behaviors for both trained and untrained stimuli which indicates that there is no deterministic output in the proposed system. In other words, the robot uses its own internal dynamics to conclude whether it *"likes"* the received stimuli or not. The conducted experiments' processes highlight that the robot can regulate its own states to minimize energy consumption while successfully executing recognition process and presenting a behavior.

# CHAPTER IV

# CONCLUSIONS

This chapter provides the concluding remarks of our studies on the brain mechanisms of action recognition (e.g. mirror neurons) and emotions. In addition, future directions and key insights of the ongoing studies are emphasized to pave the road for gaining answers to open questions in the literature.

## *4.1 Action Recognition (Mirror Neurons)*

The obtained object level decoding results are explained in two subsections since detecting mirror neurons among the neuron population and evaluating decoding performance are based on execution-observation comparison and cross decoding methods. In short, we developed an automatic procedure for detecting mirror neurons and classifying them according to their decoding characteristics.

### 4.1.1 Execution and observation decoding

As shown in Table 4 and Table 5, the neuron ids placed in a colored cells are performed well according to their own absolute specificity values in both observation and execution conditions. This finding indicates strong supporting remarks for these neurons to be mirror neuron candidates. While mirror neuron candidates are detected in a mechanistic way via adapted software frameworks, neural decoding types are also marked in relatively large neuron population. Based decoding characteristics of these mirror neuron candidates, three different decoder neuron types can be labelled to classify neurons decoding behavior for a given object. It is worthwhile to state that this conclusion is also compatible with the selective type of the neurons which are highlighted in the literature [12].

1. **General decoder neuron:** decodes all objects in observation-execution conditions (e.g. neuron 54).

2. **Multi-object decoder neuron:** decodes more than one object in observation-execution conditions but not all of them (e.g. neuron 93).

3. **Object specific neuron:** decodes only one object in observation-execution conditions (e.g. neuron 131 and 168).

Labeling neurons with their decoding specificity paves the way for further analysis of grasping ability of the ventral premotor cortex since obtaining any promising results will lead to considerable contributions in different areas such as computational neuroscience and medical robotics. One of the possible implementations is to design a grasping prosthesis for physically disabled people which receives motor command trough neural activation.

### 4.1.2 Cross decoding

The cross decoding results presents an important fact that the neural representation of the mirror neuron candidates have almost same features in both execution and observation conditions. The success of a decoder trained with data from one condition, on the data obtained from another condition is very important as it indicates that the representations in both conditions are the same. Our analysis with weight transfer on available neuron set indicated that with single neurons object-specific decoders can be constructed, i.e. the type of the object being grasped in both execution and observation conditions can be decoded (max 70% success rate, the chance level is 25%). As expected, it was not always the case that a good decoder in execution condition, became a good decoder in the observation condition. These decoder neurons in fact correspond to selective type mirror neurons reported in the mirror neuron literature. The critical question was whether there would be neurons that can become good decoders for say execution condition even though their decoder was formed using observation data. We found a few neurons that can do this for the Sphere (3 neurons, with mean cross-decoding rate above 60%) and Ring (2 neurons with mean cross-decoding rate above above 50% correct) in the observation-to-execution transfer. It is also observed the the reverse similar results were obtained. These results are very encouraging as even at the single neuron level we showed the existence of transfer.

To extend this analysis to population level we examined all pair performance of a 10-neuron set, of 4 were picked from the best solo decoders and the rest were picked randomly. Out of the 45 pairs we found the best performing seven pairs (in cross-decoding). The most successful neuron pairs gave 70% success on the average in transfer performance. More interestingly, perfect pair that does the cross-decoding well for all the objects did not appear and none of the 4 initial manually picked neurons paired up to make up strong decoder. Yet, they all appeared in high-performing pairs with a "helper" neurons, which were not good solo performers. These results, strongly points to a population based representation: while a neuron itself may not look important for decoding but it can become the best assisting partner to make a robust 2-pair mirror neuron system.

These results bring about the idea of "temporal mirror neuron" term in which the neuron can change its own decoding ability vary for each execution-observation conditions. To extent event-wise neural representation transferring, we are planning to perform same transfer between two monkeys to find whether same representation exist in different neural system.

## *4.2 Emotions*

The proposed emotion mechanism that the regulation of neural energy for cognitive processes may lead to emergent behaviors that may be associated by the emotional state of the agent. For example instead of trying to search for the best stimuli, an agent can settle for a less energy requiring sub-optimal solution. This behavior then may be interpreted as an emotional behavior displayed by the agent. To show that emotion-like behaviors may emerge through simple computational energy conserving self-regulatory mechanisms, the proposal is realized on a humanoid robot with simple cognitive architecture. The interaction experiments with the robot indicate that this is a promising view that can place the concept of emotion on firm grounds. Combining this with the novelty of the proposal, we think this is a very rich new research area that needs to be attacked from multiple disciplines.

To extend this study, we will develop a bio-inspired emotion framework to model specific regions of the brain including Prefrontal Cortex and Amygdala. This framework will capture the reciprocal connections of the Prefrontal Cortex and Amygdala and simulate agent's emotions as introduced in this paper. Moreover, the framework will provide a complex self-assessment mechanism to regulate an agent's internal dynamics to adapt environmental changes and display behaviors accordingly. We predict that the robot will be seen as having affections for certain patterns, even for those seen for the first time, which puts less computational burden for the cognitive system of the robot.

# Bibliography

[1] D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, 1982.

[2] G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi, Premotor cortex and the recognition of motor actions. Brain Res. Cogn. Brain Res. 3, 131141, 1996.

[3] G. Rizzolatti and L. Craighero, Mirror neuron: a neurological approach to empathy, in: Neurobiology of Human Values, Springer, New York (2005), pp. 107-124.

[4] M. A. Arbib, From monkey-like action recognition to human language: An evolutionary framework for neurolinguistics. Behavioral and Brain Sciences, 28 (2), 105124.

[5] G. Buccino, F. Binkofski and L. Riggio, The mirror neuron system and action recognition. Brain and Language, 89 (2), 370376, 2014.

[6] M. Dapretto, M.S. Davies, J.H Pfeifer, A.A Scott, M. Sigman, S.Y. Bookheimer, and M. Iacoboni, Understanding emotions in others: mirror neuron dysfunction in children with autism spectrum disorders. Nature Neurosci. 9, 2830, 2006.

[7] E. Oztop, M. Kawato and M.A. Arbib, Mirror neurons: functions, mechanisms and models, Neuroscience Letters, 540, 43-55, 2013.

[8] J. Carpaneto, V. Raos, M.A Umilt, L. Fogassi, A. Murata, V. Gallese, S. Micera, W Zhang, J.A. Johnston and M.A Ross, Continuous decoding of grasping tasks for a prospective implantable cortical neuroprosthesis, NeuroEngineering and Rehabilitation, 9(1):84, 2012

[9] Y. Hao, Q. Zhang, S. Zhang, T. Zhao, Y. Wang, W. Chen and X. Zheng, Decoding grasp movement from monkey premotor cortex for real-time prosthetic hand control Chin. Sci. Bull. 58 251220, 2013.

[10] M. Kirtay, P. Vassilis, R. Vassilis and E. Oztop, Detecting mirror neuron candidates in F5 area via object level decoding. 13rd National Neuroscience Congress, 2015, Konya, Turkey.

[11] M. Kirtay, P. Vassilis, R. Vassilis and E. Oztop, Neural representation in F5: cross-decoding from observation to execution. 24th Annual Computational Neuroscience Meeting, 2015, Prague, Czech Republic.

[12] A. Murata, L. Fadiga, L. Fogassi, V. Gallese, V. Raos, and G. Rizzolatti, Object representation in the ventral premotor cortex (area F5) of the monkey, Journal of Neurophysiology 78 (1997) 22262230

[13] J. E. Ledoux and E. A. Phelps, Emotional Networks in the Brain, 1982.

[14] M. Arbib and J.-M. Fellous, Emotions: from brain to robot., Trends in cognitive sciences, vol. 8, no. 12, pp. 554-561, Dec. 2004.

[15] J.-M. Fellous and R. Suri, The Handbook of Brain Theory and Neural Networks, MIT Press, pp. 361-365, 2003.

[16] S. Jung and K. An, Development of a facial expression imitation system, IEEE/RSJ International Conference on Intelligent Robots and Systems, no. 4, pp. 31073112, 2006.

[17] M. Kirtay and E. Oztop. Emergent Emotion via Neural Computational Energy Conservation on a Humanoid Robot. IEEE-RAS International Conference on Humanoid Robots (Humanoids 2013), pp.450,455, 15-17, 2013

[18] D. Canamero and N. Munkegade, Designing Emotions for Activity Selection, 2000.

[19] D. Parisi and G. Petrosino, Robots that have emotions, Adaptive Behavior, vol. 18, no. 6, pp. 453-469, Nov. 2010.

[20] T. Shibata, T. Tashima, and K. Tanie, Emergence of emotional behavior through physical interaction between human and robot, IEEE International Conference on Robotics and Automation, vol.4, no., pp.2868-2873 vol.4, 1999.

[21] M. A. Salichs and M. Malfaz, A New Approach to Modeling Emotions and Their Use on a Decision-Making System for Artificial Agents, IEEE Transactions on Affective Computing, vol. 3, no. 1, pp. 56-68, Jan. 2012.

[22] S. C. Gadanho and J. Hallam, Robot Learning Driven by Emotions, Adaptive Behavior, vol. 9, no. 1, pp. 42-64, Mar. 2001.

[23] D. C. Geary, The origin of mind: Evolution of brain, cognition, and general intelligence, American Psychological Association (APA) Press, 2004

[24] S. C. Gadanho, Reinforcement Learning in Autonomous Robots : An Empirical Investigation of the Role of Emotions, Ph.D Dissertation, Univ. of Edinburgh, 1999.

[25] S. B. Laughlin, R. R. de Ruyter van Steveninck, and J. C. Anderson, The metabolic cost of neural information., Nature neuroscience, vol. 1, no. 1, pp. 36-41, May 1998.

[26] S. B. Laughlin, Energy as a constraint on the coding and processing of sensory information., Current Opinion in Neurobiology, 11(4):475-80, 2001.

[27] I. Ha, Y. Tamura, H. Asama, J. Han, and D.W. Hong. Development of open humanoid platform DARwIn-OP. In SICE Annual Conference (SICE), pages 2178-2181. IEEE, 2011.

[28] J. Hertz, A. Krogh and R. G. Palmer, Introduction to the Theory of Neural Computation, Addison-Wesley Publishing Co., Redwood City, CA, 1991