Comenius University in Bratislava

Faculty of Mathematics, Physics and Informatics

Department of Applied Informatics

Learning Multisensory Integration and Coordinate Transformation in a Simulated Humanoid Robot

Master thesis

2019

Bc. Martin Kellner

Univerzita Komenského v Bratislave

Fakulta matematiky, fyziky a informatiky

Učenie multisenzorickej integrácie a transformácie súradníc v simulátore humanoidného robota

Diplomová práca

Študijný program:	Aplikovaná informatika
Študijný odbor:	Aplikovaná informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	prof. Ing. Igor Farkaš, Dr.

Bratislava, 2019

Bc. Martin Kellner

27797918

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Bc. Martin Kellner
Študijný program:	aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor:	aplikovaná informatika
Typ záverečnej práce:	diplomová
Jazyk záverečnej práce:	anglický
Sekundárny jazyk:	slovenský

Názov: Learning Multisensory Integration and Coordinate Transformation in a Simulated Humanoid Robot
Učenie multisenzorickej integrácie a transformácie súradníc v simulátore humanoidného robota

Anotácia: Spracovanie senzorických podnetov v mozgu zahŕňa multisenzorickú integráciu (kombinovanie vodítok do jedného spoločného podnetu) a transformácie súradníc (referenčných rámcov, napr. z retinotopického na telo-centrický, ovplyvnenej modulačnou premennou, napr. pozíciou očí). Táto schopnosť je kľúčová v kognitívnej robotike, ak chceme vybaviť robota schopnosťou operovať autonómne v 3D priestore.

Anotácia: 1. Naštudujte si problematiku z kognitívnej neurovedy o multisenzorickej integrácii a referenčných rámcoch (súradnicových systémoch).
2. Implementujte a natrénujte model umelej neurónovej siete, ktorá sa naučí integrovať vizuálnu a proprioceptívnu informáciu, vykonávajúc prepočet súradníc v úlohe týkajúcej sa ruky a očí, s využitím simulovaného robota iCub.
3. Vyhodnoťte a analyzujte správanie natrénovaného modelu.

Literatúra: Makin J., Fellows M., & Sabes P. (2013). Learning multisensory integration and coordinate transformation via density estimation. PLOS: Comput. Biol., 9(4).
Švec M., Farkaš I. (2014). Calculation of object position in various reference frames with a robotic simulator. In Proceedings of the 36th Annual Conference of the Cognitive Science Society, Quebec, Canada.
Tikhanoff V., Fitzpatrick P., Nori F., Natale L., Metta G., & Cangelosi A. (2008). The iCub humanoid robot simulator. Advanced Robotics, 1(1), 22-26.

Vedúci:	prof. Ing. Igor Farkaš, Dr.
Katedra:	FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry:	prof. Ing. Igor Farkaš, Dr.
Dátum zadania:	16.10.2017

Dátum schválenia: 20.10.2017 prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

....
študent vedúci práce

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Martin Kellner
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: aplikovaná informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Learning Multisensory Integration and Coordinate Transformation in a Simulated Humanoid Robot
Učenie multisenzorickej integrácie a transformácie súradníc v simulátore humanoidného robota

Anotácia: Spracovanie senzorických podnetov v mozgu zahŕňa multisenzorickú integráciu (kombinovanie vodítok do jedného spoločného podnetu) a transformácie súradníc (referenčných rámcov, napr. z retinotopického na telo-centrický, ovplyvnenej modulačnou premennou, napr. pozíciou očí). Táto schopnosť je kľúčová v kognitívnej robotike, ak chceme vybaviť robota schopnosťou operovať autonómne v 3D priestore.

Anotácia: 1. Naštudujte si problematiku z kognitívnej neurovedy o multisenzorickej integrácii a referenčných rámcoch (súradnicových systémoch).
2. Implementujte a natrénujte model umelej neurónovej siete, ktorá sa naučí integrovať vizuálnu a proprioceptívnu informáciu, vykonávajúc prepočet súradníc v úlohe týkajúcej sa ruky a očí, s využitím simulovaného robota iCub.
3. Vyhodnoťte a analyzujte správanie natrénovaného modelu.

Literatúra: Makin J., Fellows M., & Sabes P. (2013). Learning multisensory integration and coordinate transformation via density estimation. PLOS: Comput. Biol., 9(4).
Švec M., Farkaš I. (2014). Calculation of object position in various reference frames with a robotic simulator. In Proceedings of the 36th Annual Conference of the Cognitive Science Society, Quebec, Canada.
Tikhanoff V., Fitzpatrick P., Nori F., Natale L., Metta G., & Cangelosi A. (2008). The iCub humanoid robot simulator. Advanced Robotics, 1(1), 22-26.

Vedúci: prof. Ing. Igor Farkaš, Dr.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 16.10.2017

Dátum schválenia: 20.10.2017

prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

....
študent

....
vedúci práce

DECLARATION: I hereby declare that this thesis is my own work and that all the sources I have used or quoted have been indicated and acknowledged as complete references.

. . . .

Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

Kľúčové slová: jedno, druhé, tretie (prípadne štvrté, piate)

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords:

Contents

List of Figures

List of Tables

Introduction

Cognitive focus

2.1 Reference of Frame

To better understand the term of *reference frame* or *frame of reference*, consider the following situation: In a moving train is one passenger which lets a book fall. Another person is watching the case and standing on the ground. From the perspective of the passenger, the book is falling straight down, but the observer (the man standing on the ground) sees the book dropping down along a curved path because of the movement of the train. The situation mentioned above has shown that the description of a physical phenomenon depends on the position from where is the case observed. Using reference frames on the previous illustrating example, we can say the book is falling straight down in the train's reference frame but is dropping along a curved path in the earth's frame of reference. About the state of motion of the passenger, we might say he is stationary in the train's reference frame and moving in the earth's frame of reference (Soechting and Flanders, 1992).

Reference frames have often been used by physicists and engineers, but neuroscientists have also adopted the term. In the mathematical sense, the frame of reference is similar to a coordinate system that is defined by a set of axes and an origin. The origin can be anywhere in space, and the orientation of axes can be chosen arbitrarily because a reference frame is only characterized by the state of motion relative to an object. Therefore, the train's reference frame that was mentioned in the illustrated example has the same state of motion as the train, and any point in the reference frame can be defined by its position along each of coordinate axes. Alternatively, the location of a point can be defined by way of vectors. A vector has two properties: a magnitude and a direction; the magnitude is the length of a line segment between the origin and the point, and its direction is from the origin to the point (Soechting and Flanders (1992), Kumar and Barve (2002)).

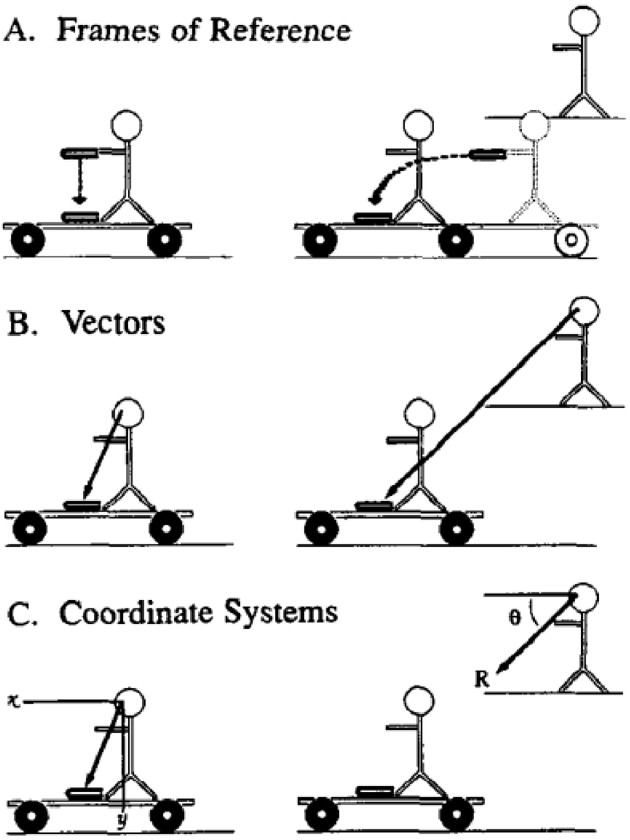

Figure 1: Schematic illustration of the spatial representations of objects in frames of reference (A), vectorially (B) and by coordinate systems (C). On the left, the frame of reference moves with the passenger; on the right, the observer's frame of reference is fixed to the earth.

To switch the reference frame into a term in neuroscience we have to make some changes against the mathematical definition because of neural systems do not report the position of an object as a vector or coordinates, but neurons of the visual system encode information on a restricted area of space. The cells will respond to its *receptive field* or *response field.* That means the firing of the neurons will change depending on stimuli coming from a special area of visual space (Batista 2002). If we want to identify in which frame of reference, for example, neurons encoding spatial information about the location of an object, then the firing activity of the neurons should stay the same as long as the image of the object falls on the same locus on the retina and stay constant. Once that happened we can say the location is encoded in a *retinocentric* frame of reference (Soechting and Flanders, 1992).

Different brain regions encode spatial locations in different RFs. In neuroscience are RF mostly divided to two main groups: *ego-centric* and *allocentric* reference frames. In an egocentric RF, the location of an object referenced to an observer. At a neural level, an egocentric RF means the receptive field, or the response of the neural population is attached to a reference point (Committeri, G., et al (2004)). For example, an eye-centred RF moves with the eyes (Colby, 1998). Spatial locations are mainly encoded in egocentric reference frames, especially space coding neurons in parieto-frontal cortex are associated with reference frames centred to the eye, the head or the hand (Colby, 1998; Hagler et al., 2007; Sereno and Huang, 2014). Also in the monkey's posterior parietal cortex and in connected regions of the premotor cortex has found neurons that encode space in RFs centred to parts of the body (Cohen & Andersen, 2002; Colby, 1998). Coding of space in allocentric RFs is not so well-studied like egocentric RFs, but it has shown (Fink, Marshall, Shah, et al., 2000; Galati, Lobel, et al., 2000; Honda, 1998; Fink 1997) that in the posterior parietal cortex and the dorsal premotor cortex (PMd), and early visual areas, neurons report during object-based spatial judgement. Also, it has been proven that during solving of a complex task, such as landmark knowledge, orientation in large-scale space and navigation, some operation referred to both types of reference frame, allocentric and egocentric, which are difficult to untangle (Committeri G., et al (2004)).

2.2 Sensorimotor transformations

When humans, aminals, or insect want to carry out basic operations to interact with the world, then they use their bodies and senses. To manipulate with an object, moving in space, or doing other daily completion, the brain must continuously update the internal representation of the world, deal with restrictions that can occur while performing an operation and change body configuration to achieve the goal of the action. For instance, during the reaching of an object, the human brain must take into account the location of the target, the current position of the particular hand to generate the right movement of the hand. One non-neuroscientist might say the operations as above mentioned are not very complicated because we do not need to spend a lot of effort to get the goal, but when we look closely at the neural level, the task is more complicated like it would seem. Information about the location of an object can be produced by stimuli from different modalities, each of these modalities provides neural information that is encoded in a different frame of reference. (Moving in ...) Therefore, neuroscientists focus their research to clarify to the following questions: How is the information represented in each reference frame? How is the information from different modalities combine to encode the correct representation of space? How is information in particular reference frame transform to another one? *Coodinate transformation*, in neuroscience, is a term to call transformation from one reference frame to another. We closely discuss eye-hand coordinate transformation, that is central to many human daily activities, in the next section.

2.2.1 Spatial Transformations for Eye–Hand Coordination

Eye-hand coordination is reviewed by Crawford J., et al. (2004) in detail. The authors also bring assumptions about using gaze-centred representations during the process of coordinate transformation that needed to perform a task such as reaching or grabbing an object. In this chapter, we closely discuss their conclusions.

To reach an object the brain needs information about the location of the reaching object, and the position, for the purpose, is encoded relative to a body part. When we realise the fact that arms are anchored to shoulders, and spatial information on object comes mainly from projections on retinas, then we can realise the brain must perform coordinate transformation between at least two representation. First of them is the representation specifying the object's location relative to retinas, and the second representation within witch the activation of arm's muscles must be determined to reach the target. Therefore we can say the purpose of eye-hand coordinate transformation (or system that is responsible to do that) is to guide movements of hands using vision. This section discusses how the brain deals with the transformation of visual stimuli into hand motion for controlling and grabbing a target.

A robot system for controlling the robot's arm to reach or to grab an object is usually in comparison with the human control system is simpler. Such a robot system is usually reduced by using visual feedback, basically, to drive the hand to a point seen by the visual system comparing the current position of the hand with the target position. This can work in such a robotic system because the speed of visual feedback is not so limited as a brain control system. The robotic system visual feedback is limited by computer processing time and speed of electrical flow, whilst the speed of neural conduction and processing time in a real primate brain is not adequate to perform fast hand movement that would be way off the field of vision before new visual feedback would arrive to accurately update the movement. Consequently, a primate brain cannot use only visual feedback to drive hand movement and must use another way to perform correct and fast movements: to use internal models of the physical system and world based on initial conditions and control eye-hand coordinate transformation in feedforward way. Despite this, we cannot say that visual feedback does not contribute to guiding hand movements. Visual feedback helps to achieve the best performance of grabbing a goal, and are essential to deal with unexpected events or a danger that can occur during performing a particular movement. To better understanding of such a feedforward transformation is helpfull to divide it into concentual steps.

Visual Representations of Reach Space

Internal models, for such operations as coordinate transformations, are described by using the concept of reference frames (2.1). Final reference frames for eye-arm coordination associated with points in upper arm and shoulder, and not hand-centrered representation like we could wrongly consider of. The first stimuli are controlled in a retinal frame, and there is a problem with gaze shifting. Every time when eyes (or/and head) shift(s) gaze the visual relationship between sensory apparatus and the external world (Hallett and Lightstone 1976). One solution, how the brain could deal with that problem, is waiting until the gaze shifting gets finished and then update its visual information (O'Regan and Noe 2001), but that could cause that the original target of interest could be moved into less-sensitive peripheral retina or even out of the visual field. It Is good to realise that would also produce long lags in processing time and also would introduce redundant visual computations, and thus, to perform visual guidance of arm performing a sequence of saccades (note) would be impossible. Therefore, there must the necessary representations for future action be stored, and either in an eye-independent form or in such form that is internally updated when the gaze is shifting whether caused by movements of eyes or head. (Duhamel et al. 1992).

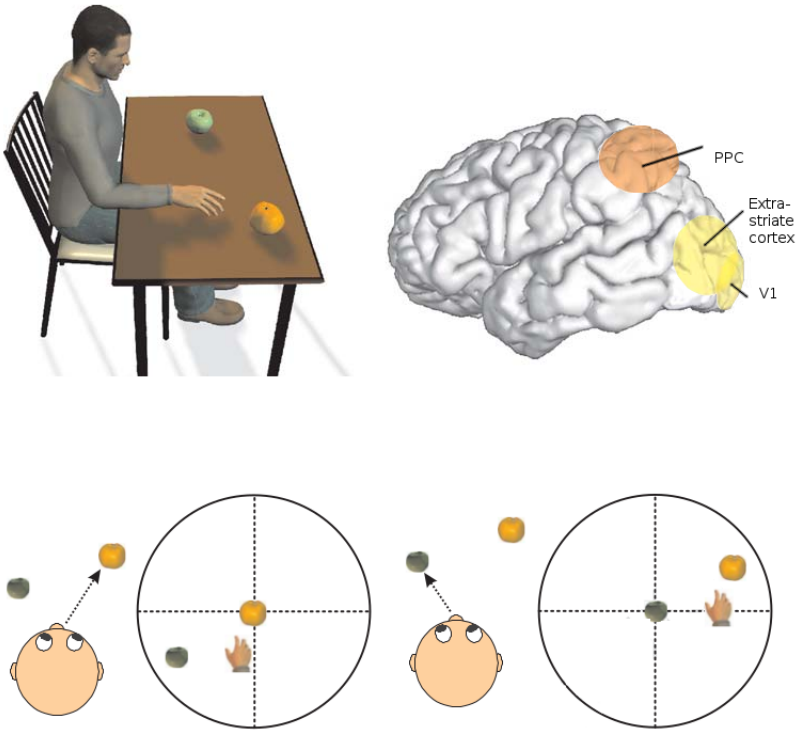

Figure 2: Gaze-centered encoding of reach space. (a) Drawing depicting the egocentric visual directions of the hand, an orange, and an apple, as shown by the gray arrows. (b) Side view of a human brain showing areas (highlighted in yellow and orange) that encode reach space in gaze-centered coordinates: V1, striate cortex; PPC, posterior parietal cortex. (c) The consequence of an eye movement on the gaze-centered representation of the visual field. The head/eye diagrams depict current gaze position; the circles represent the gaze-centered representation of this visual scene (dotted lines represent visual horizontal and vertical axes and intersect at the fovea). If the person in the upper diagram looks at the orange, the hand and apple are represented in the left visual field (upper circle). In contrast, if the person fixates the apple (lower diagram), the orange and hand are now represented in the right visual field (lower circle). If the orange and the hand were no longer visible when the eye movement occurred, the brain would need to remap their position by taking the intervening eye movement into account (Crawford J., et al. (2004)).

Henriques et al. (1998) were asking themselves the question: How the system in the human brain, which appears to be used for early planning of pointing and reaching a target (McIntyre et al. 1997; Vetter et al. 1999), stores these early motor representations during eye movements? In order to answer the question, they performed the experiment described by FIGURE ??. Human subjects, which were performing the experiment, was asked to point toward the location using hand in total darkness. At the start of the experiment the target location, that was pointed, was centrered to subject's fovea, then subjects shift the gaze by moving of eyes, and in the end, perform mentioned pointing toward the remembered location.

Henriques et al. (1998) tried to determine whether the responses of such a pointing task was affected by an intervening eye displacement by comparing them to pointing to remembered foveal targets or to retinally peripheral targets. When subjects ware pointing toward using an updated gaze-centered representation, pointing behaviour would echo pointing to peripheral targets, whereas pointing using a nonretinal representation would cause no effect. This experiment has cleared up the question about early motor representation and has supported the idea of using an updated gaze-centered frame of reference. Other studies have brought findings that during pointing of auditory and proprioceptive targets such a gaze-centered updating has been also recorded (Pouget et al. 2002). Evidence of coding the spatial information of an object in a gaze-centred frame of reference can be also found in both early visual pathway (i.e., retina, lateral geniculate nucleus, striate cortex) and also later in the visual pathway (e.g., extrastriate and parietal cortex).

The gaze-centered representation moves depending on the gaze because the retinal projection of the project depends on the gaze. Therefore, the positions of a reaching target can be defined by the target's direction (horizontal and vertical angular eccentricity) relative to *fovae*, and by the distance between the goal and eyes. Early visual areas carry out the computation of target direction as seen by a virtual eye which we can imagine placed between right and left eye, whereas the required distance is determined by taking into account monocular information (accommodation, relative object size, shading, perspective, etc.), and binocular information (retinal disparity and convergence). Later during early movement planning, signals that contain the information seem to be merged into a single gaze-centred representation of the space in the posterior parietal cortex.

How we has mentioned above gaze-centerered representation must be updated to maintain the stable representation of the world. The brain must update this representation all time while eyes are moving, because the gaze changed. This process which is performing during eyes rotation is named *updating* or *remapping*, and it also occurs during movements of the head and the body. An illustrating example is in FIGURE 1(c), the observer originally looks at the orange, and the apple is placed to left-down relative to the orange. The orange is at the centre of the gaze-centred representation and the apple located on the lower-left quadrant of the visual field. A fast orientation movement approaching the apple cause that the objects are remapped by the same rotation as the movement, but in the opposite direction, what means, the apple is mapped at the centrum, and the orange is located on the upper-right quadrant. It has shown the updating can be achieved even if vision is removed, only by using information about eyes movements.

Developing the reach plan

To correctly generate a reach plan the information about initial hand position must be linked to. (Buneo et al. 2002) have brought suggestions that comparison between initial hand location and gaze-centered representation of the visual target is done earlier and in a gaze-centered reference frame, even when the hand is out of visual field, what means, proprioceptive signals that carry the information about hand location must be transformed into gaze-centered coordinates. (Buneo et al. 2002) found such a kind of signals in the process of a gaze-centered transformation in parietal area 5. Findings of the mentioned earlier comparison do not want to say that a next reference frame transformation is not required to reach a target.

The next important problem, which the brain must solve to accuracy generate and perform a reach plan is how ego-center in translocated during head rotation, because rotation of head cause eye's translation with respect to the shoulder. Ignoration of eye translation component would cause erroneous reach pattern at noncentral head position.

A summary of conceptual and physiological models of visually guided reach is described in the figure 3. The first stage of figure corrections to the fact that 3D representations of target direction are stored and maintained in a retinal frame. The second stage shows an illustration of the transformation according to above-mentioned Buneo et al. (2002) schema to compute the hand displacement in a retinal reference coordinates. There are other reference frames transformations from gaze coordinates into shoulder representations that are not included in the figure and are necessary to reflect the eye-head-shoulder system, but such models like these illustrated can be useful in telling us what signals are required and used for a transformation, but obviously, all of such kinds of explicit intermediate representations are not used by brain directly, and also do not tell us how and where mentioned signals are coded.

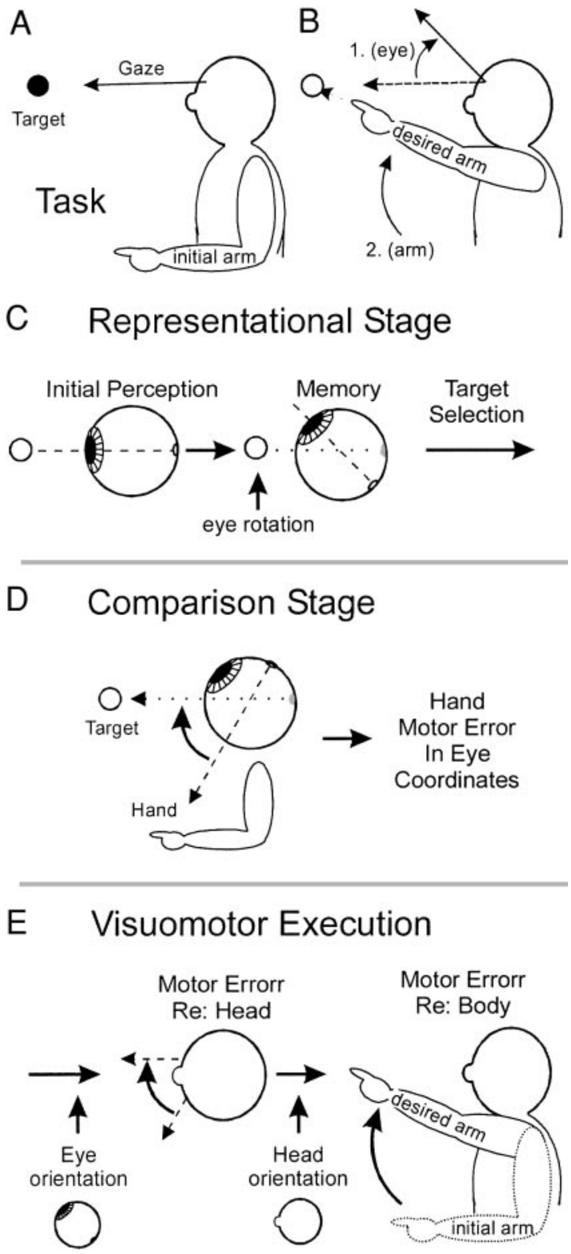

Figure 3: Conceptual scheme for spatial transformations in eye– hand coordination. To illustrate the model, consider the following "task": a subject looks at a briefly flashed target (F) with the arm at resting position (A). Then (B) the subject makes 1) an upward eye movement, followed by 2) a reaching or pointing movement toward the remembered target location (E). We hypothesize that the brain uses the following stages to do this. C: an early representational stage. Target location is stored in eye coordinates, such that this representation (E) must be counterrotated (updated) when the eye rotates. D: comparison stage. Updated target representation (E) is compared with an eye-centered representation of current hand location to generate "hand motor error" in eye coordinates (Buneo et al. 2002). E: visuomotor execution stage. "Hand motor error" signal is rotated by eye orientation and head orientation (or perhaps by gaze orientation) to put it into a body coordinate system appropriate for calculating the detailed inverse kinematics and dynamics of the movement. This last stage would also have to include internal models of the geometry.

To design a complete 3D reach plan, the brain must know the amplitude of the desired movement and the direction. It has shown that in the dorsal premotor cortex the direction and the amplitude are encoded together inside the same neuron, but there is also evidence that direction and amplitude might also be encoded independently. One explanation of this contradiction might be that, at the neural level, amplitude and direction are encoded together, but are used by different mechanisms to generate muscle activation. In account must be taken the rotation of head and body and also values of correct forces for specific muscles. An amplitude and a direction are properties of vectors, therefore, we can reformulate the problem of designing a reach plan into terminology of vectors as it is often done.

The conclusion of the problem of developing reach plan brain must design a hand movement vector, which depends on the difference between the hand location and the location of the target. The areas in the brain that are responsible for calculating the vector must have spatial information of the hand and the goal position, and the spatial information must be represented in the same reference frame. To determine the movement vector the particular areas could use information from different sources and also perform multisensory integration to get the most likely estimation of both locations, the hand and the target. One of approaches that allow us to examine how populations perform the task of coordinate transformation is using artificial neural networks. During studying of neural encoding of a cognitive function, we very often focus on changing response magnitude of neurons as known as *gain field* what we are discussing in the next section 2.3. (Crawford J., et al. (2004),).

2.3 Gain fields

The term of *gain field* comes from Andersen and Mountcastle (1983). The authors tested the visual receptive field for a specific neuron at different eye positions. The receptive field of a neuron or a neural population can be described as a specific region in sensory space which stimuli can have an influence on the activity of a single neuron or a neural population. For instance, activations of neurons encoding the gaze-centered representation of a target's location should be the same while gaze angles stay constant, contrariwise, firing activity of the neurons change with every eye motion. During the mentioned study the neuron's action potential frequency was changing in a manner of multiplication the frequency by gaze angle, that is scaled by some constant. Both of the shape and the locations of the visual receptive field was unchanged, only the receptive field was scaled by some *gain* factor. This phenomenon was the first time characterized in neuron within LIP and visual area 7A. Another study (Zipser and Andersen (1988)) which has dealt with the task of coodinate transformation from visual target position and signals of gaze position into the position representing by the target location in a space-fixed reference frame, trained a neural network to perform the transformation. Detailed analysis of their neural model has shown the network was able to develop visual receptive fields modulating by eye position in a very close way compared to modulation in parietal cortex. The figure 4 illustrates and can help to understand better the result of the study. At the neural population level gain fields modulate responses of single neurons, therefore, the brain can increase or decrease the population output of a neural population, that can allow comparing something like the output of the above mentioned artificial neural network with information about current hand location, and that might use the process of computation the hand movement to reach a target. It has been shown that also other types of signals produce gain modulation and the modulations, and gain fields were found in many other brain areas. (Blohm G. and Crawford J.D (2009))

2.3.1 Links between Gain Modulation and Coordinate Transformations

In order to demonstrate the role of gain modulation with performing the task of coordinate transformation in the brain, we will discuss again results that Zipser and Andersen (1988) has brought. They trained a neural network to solve coordinate transformation from gaze signal and retinal location to stimuli in body-centered reference frame. During the artificial neural network, which was training to learn the transformation had as the input information on mentioned sensory stimuli and the output was the body-centered representation of locations. The model was trained using the one of major-used training in machine learning called *backpropagation.* This algorithm allows to *learn* values of *hidden* units (neurons) in order to map inputs to corresponding outputs with the effort to achieve as low error as possible. All of output and input representation of stimuli were composed to match the measured response of neurons in the brain, thus the model has been able to perform a coordinate transformation similar to what is made by the visual system (Figure 5). After successfully training the authors has shown that hidden units developed gaze gain field as in PPC. This experiment has shown the potential role of gain modulation

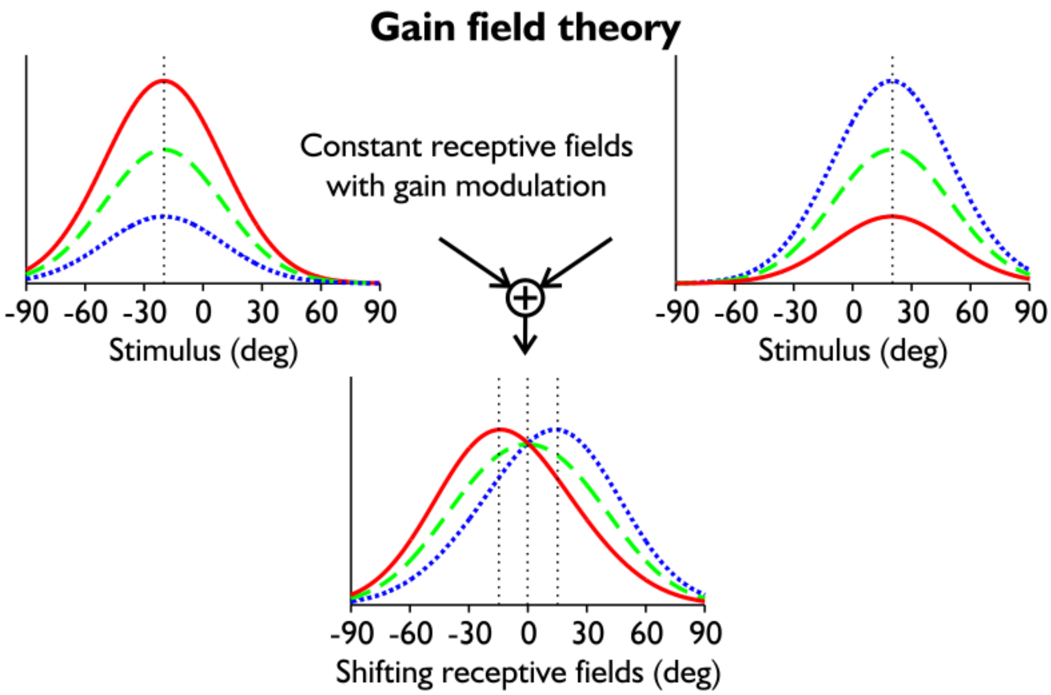

Figure 4: Working principle of gain fields, based on Zipser and Andersen (1988). The upper part of the panel shows the hypothetical receptive fields of two neurons that are gain modulated (e.g., by eye or hand position) in opposite ways without shifting. For example, the three lines in each graph could represent visual receptive fields mapped relative to gaze at a leftward eye position (red solid line), a central eye position (green dashed line), and a rightward eye position (blue dotted line). Here, eye position modulates the strength of response of two neurons, but does not cause them to shift. However, summation of these two gain-modulated neural responses results in shifting receptive fields in the output, e.g., eye position (or in other cases hand position) has shifted the receptive field. (Blohm G. and Crawford J.D (2009))

in solving coordinate transformations (Salinas E. and Sejnowski T. (2001)).

Other researches also looked into similarities between real neurons and artificial, especially their responses. Xing and Andersen (2000) trained more complex neural model to represent two consecutive saccades separated by a delay interval. The authors used a recurrent neural network because recurrent neural networks are able to learn a sequential task through their internal *memory*. Input data combine a visual map in retinal coordinates, an auditory map in head-centered coordinates, and eye-position units. The model was learned to produce outputs that encode simultaneous movements of both eyes (Xing and Andersen (2000)). After training, they have demonstrated that the network developed gain fields, and sensory and memory response matched those of LIP neurons.

Salinas and Abbott (1995) also studied the relationship between the coordinate

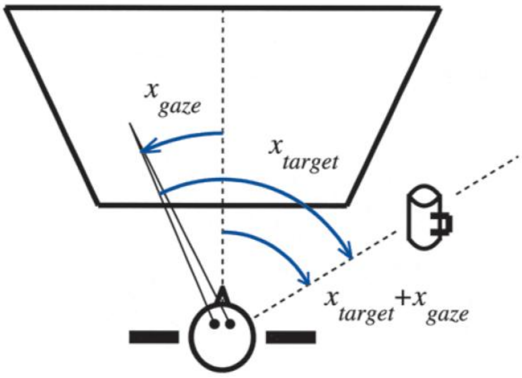

Figure 5: A coordinate transformation performed by the visual system. While reading a newspaper, you want to reach for the mug without shifting your gaze. The location of the mug relative to the body is given by the angle between the two dashed lines. For simplicity, assume that initially the hand is close to the body, at the origin of the coordinate system. The reaching movement should be generated in the direction of the mug regardless of where one is looking, that is, regardless of the gaze angle xgaze. The location of the target in retinal coordinates (i.e., relative to the fixation point) is xtarget, but this varies with gaze. However, the location relative to the body is given by xtarget + xgaze, which does not vary with gaze. Through this addition, a change from retinal, or eye-centered, to body-centered coordinates is performed. (Salinas E. and Sejnowski T. (2001))

transformation described by Figure 5 and gain modulation. Their research was centred to gain-modulated neurons by gaze those drive downstream neurons in another neural population included in the process of generating arm movements. They considered a population of neurons that respond to visual stimulus at retinal location x_{target}, and are modulated by gaze angle x_{gaze} what constitutes gain modulation, then all response r of a population like that can be described by the following equation:

$$r = f\left(x_{target} - a\right) g\left(x_{gaze}\right), \tag{2.1}$$

where $g\left(x_{gaze}\right)$ is the function which result is the gain field of particular neuron, and $f\left(x_{target} \breve{} a\right)$ represent the response of neurons as a curve with a single pick (Figure 6). Then, downstream neurons driven by a population such as the above mentioned must have responses R theoretically formed like the following function of x_{targer} and x_{gaze}:

$$R = F\left(c_1 x_{target} + c_2 x_{gaze}\right), \tag{2.2}$$

where both constants c_1 and c_2 depend on the synaptic weight. The receptive field of the downstream neuron is represented like the peaked function R. This mathematical formulation is only a simplification of the problem, but the authors

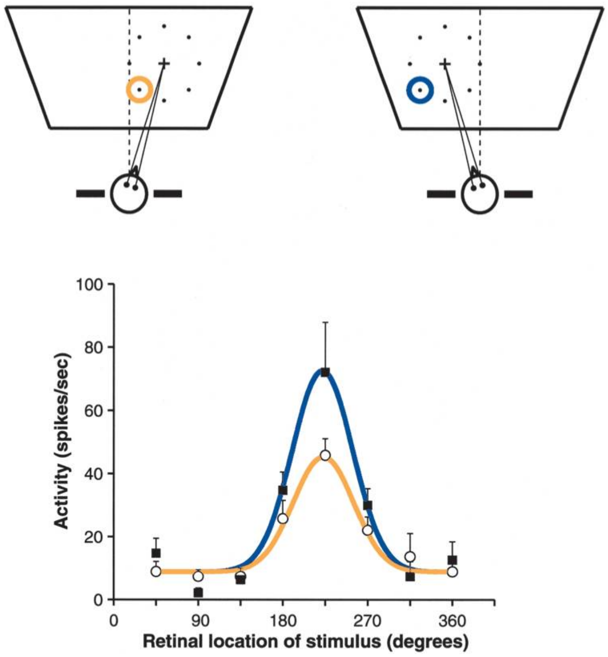

Figure 6: Visual responses that are gain-modulated by gaze angle. The response of a parietal neuron as a function of stimulus location was measured in two conditions, with the head turned to the right or to the left, as indicated in the upper diagrams. In these diagrams, the cross corresponds to the location where gaze was directed, called the fixation point; the eight dots indicate the locations where a visual stimulus was presented, one location at a time; and the colored circles show the position of the recorded neuron's receptive field. This was centered down and to the left of the fixation point. In the diagrams, the rightmost stimulus corresponds to 0 degrees, the topmost one to 90 degrees, and so forth. The dashed line indicates the direction straight ahead. The graph below plots the neural responses in the two conditions, indicated by the corresponding colors. The continuous lines are Gaussian fits to the data points. When the head is turned, the response function changes its amplitude, or gain, but not its preferred location or its shape. (Data redrawn from Brotchie and others 1995.)

found conditions under those this can happen and experimentally confirmed that the downstream neurons encode the sum of $x_{target} + x_{gaze}$, what means, the downstream neurons those must encode target in a body-centered reference frame respond as a function of x_{target} swifts when x_{gaze} changes (Salinas and Abbott (1995)). Similar swift was demonstrated in several cortical areas associated with a variety of coordinate transformations.

Releated researches

Artificial neural networks are mathematical model inspired by real brain neural circuits and used in studying cognitive functions of humans, whether in order to get to understand how are performed particular brain's tasks, or to implement human behaviours into a robotic humanoid. We will not discuss artificial models in detail in this section. This section is focused on significant research works that have been dealt with coordinate transformation, we will discuss their results and methods.

3.1 Learning using Back-Propagation

Zipser and Andersen (1988)[1] have been first who trained an artificial neural model and has brought findings that artificial neural networks are able to decode spatial transformation.

Authors have been concerned with the question: How the brain perform such as coordinate transformations which translate sensory inputs into motor commands. They mainly have focused on the brain's area named 7a and neural processes that are performed by this area. Eariel research findings have proposed that the area 7a is most likely used to perform spatial transformations. This area contains cells that to receive a convergence of both retinal signals and eye-position as a non-linear interaction, and visual responses have seemed as modulated by a function of ey position multiplied by the response of retinal receptive fields, what basically means, that visual receptive field remains retinotopic, but the intensity of response is modulated by position of eyes.

To facilitate a comparison of artificial neural cells with cells of area 7a, they anal-

[1] All information in the section 3.1 was received from Zipser and Andersen (1988)

ysed experimental data collected in studies with awake, unanaesthetized monkeys. The experimental data was collected by recording neuronal activity during performing various visuospatial tasks. They have focused on three major kinds of neurons in 7a area: the cells responding to eye-position only, the visual cells responding to retinal stimuli, and the cells that respond to both of mentioned cell classes those interactions produce an eye-dependent representation of visual targets. In order to get to test eye-position sensitivity the following experiment was performed: The animal fixated on a small point at different eye positions, their heads were fixated, and the experiment took place in total darkness. A linear increase was recorded in activity for a range of horizontal or vertical eye positions, but some cells showed more complex coding of eye position. To test the receptive fields of the visual cells a flashing spot stimulus was used at different locations in the visual fields while the animal fixated on a target at a single eye position. The neurons, that were specially tuned, were the largest group and showed a convergence of eye position and retinal position, what means, the visual response of the spatially tuned neurons alters as a function of eye position, what was observed by collecting data under the condition in which the visual stimulus appears at the peak locations in the retinal receptive field with animal fixating at different eye positions (mozno doplnit).

3.1.1 Model and algorithm

They used a three-layer neural network. The network was trained in order to get the mapping of visual targets to head-centred coordinates. The input data had two parts, an array of units that represent the visual stimulus, and representation of eye position via a set of units. Both of the retinal and eye position inputs was formed using characteristics of the cells in PPC that response to visual stimuli, respectively, to eye stimuli. As the output layer of the network was used two representations: the first one was a gaussian format in which units had gaussian receptive fields coding location in head-centered reference frame. The other one was a monolonic format in which the activity of each neuron is a linear function of the location of the stimulus in head-centered representation. These two formats were chosen because they represent the most common types of coding formats discovered in brain cells. The hidden layer was trained using algorithm named *backpropagation.*

The backpropagation algorithm (Rumelhart et al., 1986a) is used to train hidden layers of a feedforward neural network. The information that is provided by the input x is propagated up to the hidden units at each layer to produce output $\hat{y}$, this part is called forward propagation. Each artificial neuron consists of two properties, a weight and an activation function, therefore the output $\hat{y}$ depends on these properties of each neuron in hidden layers. The backpropagation is an algorithm that is used to find minimum of a cost function of mapping x to y through updating the weights. The second part consists from backward flow through the network to compute gradient, this is named backpropagation. The gradient which is calculated is the gradient of the cost function that expresses the error between the excepted output value and the output value that is the result of the current state of hidden layers during the learning procedure. The gradient is computed by *chain rule* what allows to compute derivatives of function (cost function) formed by composing other functions whose derivates are known. Suppose that $x \in \mathbf{R}^m, y \in \mathbf{R}^n$,function g maps from $\mathbf{R}^m$ to $\mathbf{R}^n$, and f maps from $\mathbf{R}^n$ to $\mathbf{R}$, then if $y = g\left(x\right)$ and $z = f\left(y\right)$, then

$$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_i} \frac{\partial y_j}{\partial x_i}. \tag{3.1}$$

The same equation only in vector notation:

$$\nabla_x z = \left(\frac{\partial y}{\partial x}\right)^T \nabla_y z, \tag{3.2}$$

where $\frac{\partial y}{\partial x}$ is the n x m Jacobian matrix of g.

The error E that is the result of the cost function is calculated through the composition of activation functions of neurons, what is a continuous and differentiable function of weights w_1, w_2, ..., w_l in the network, perforce, the minimalization of E using an iterative process of gradient descent, consist of calculating the gradient:

$$\nabla E = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, ..., \frac{\partial E}{\partial w_l}\right), \tag{3.3}$$

and following the *chain* rule mentioned above each weight is updated as following:

$$\Delta w_i = -\gamma \frac{\partial E}{\partial w_i} \text{ for } i = 1, ..., l, \tag{3.4}$$

where γ represent a learning rate variable, which defines the speed of learning (Goodfellow, I. et al (2017), Rojas, R (1996)).

3.1.2 Results

The model was trained by 1,000 trials. After successful training, the authors compared the experimental and trained receptive fields. This comparison has shown that the trained model generate single-peak receptive fields and also produce moderately complex these all are similar to those experimentally observed. The characteristics of generated respective fields of the trained network were obtained by holding the eye-position input into the network constant and simulating visual stimulation at the same retinal position as used in the above-mentioned experiment on area 7a.

By this comparison, Zipper and Andersen have shown that their model trained through backpropagation algorithms was able to learn receptive fields similar to those experimentally observed during performing the task of the coordinate transformation.

3.2 More Biologically Plausible Learning Rules Than Back-Propagation

Navarro at el (2018) have studied a coordinate transformation from eye-centered to head-centered coordinates (reference frames) using self-organising artificial neural networks. They have provided a biologically plausible training algorithm to demonstrate how head-centered response in the brain could be developed through an unsupervised process of visually-guided learning.

Coordinate transformation as one of the most studied domains in neuroscience has been examined by using several approaches. A lot of the approaches/works have been inspired by the early work of Zipper and Andersen (1988) (ref), and the models of those works was trained the synaptic weights matrix using a form of supervised. The problem of the usage of supervised learning is that such a training algorithm does not reflect the real learning process in the brain, because supervised learning algorithms oppose the Dale's law that says a given presynaptic neuron cannot be both excitatory (causing an increase of the likelihood of postsynaptic action potential) and inhibitory (causing a decrease of the likelihood of the postsynaptic action potential) across its efferent synapses.

In order to provide a more biologically plausible way, their laboratory has published a neural network of the visually-guided development of head-centered visual neurons those depend on associative learning rules and does not require a supervised approach. Their approach has consisted of an unsupervised competitive form of *trace learning*. Such an approach is similar to that in higher networks layers to bind together inputs that tend to occur close together in time, for instance, if a primate performs eyes movement more often compared to setting the position of the head, then retinal images that occur together in time will correspond to different positions of the eyes but to the identical position of the head. Trace learning, in this case, will learn neurons in higher layers to respond to the position of visual stimuli in an identical location in the head-centred reference frame across different retinal position.

3.2.1 Model and Algorithm

By authors proposed and used model is a composition of four main components to learn visual representation in the head-centered reference frame. The first part encodes both the retinal position of the visual target and the position of the eyes through a population of input neurons. The second part consists of a population of the output neurons those compete among each other through mutual inhibitory interactions. The third, the local synaptic *trace learning* rule, which is used to update feedforward synaptic connections between neurons of the input population and the output population. The last component of the model relates to experimental findings in which primates prefer to set their gaze by moving their eyes rather by moving the head, what means, there are periods when the eyes are moving and the head is stationary with respect to the visual environment, and thus, the visual target also remains stable in head-centered reference frame.

This model consisted of the four parts allow to train synaptic connections matrix through performing visually gained training, when the eyes move around a virtual environment that contains a stationary visual target, and the head is also stationary. In this case, the visual target falls on different retinal location, but the head-centered position of the target stays unchanged. Series of the positions of eyes and retinal locations of the visual target are represented by retinotopic neurons and their responses are gain modulated by eyes positions. The trace learning rule is able to bind together inputs corresponding to a visual target on the same position in the head-centered reference frame onto the same output neurons, although the locations of the visual target in eye-centered reference frame are different.

The network architecture o the model consists of two layers. The first layer is a population of input neurons that encode both the position of eyes and the retinal locations of visual targets. The second layer is a population of the output neurons that compete among each other to create a representation of patterns in the input layer.

During training, each period of an epoch consists of a visual target fixating in a given location in head-centered reference frame and the eyes perform a series of saccades to different eyes positions.

Trace learning rules use a temporal trace of recent neural activity to get postsynaptic neurons to bind together inputs that occur close together in time. The trace value $d_i(t)$ for the i^{th} neurons in the output layer is given by following equation:

$$\tau \frac{dq_i}{dt} = -q_i + v_i, \tag{3.5}$$

where v_i is the instant firing rate of the neuron, and τ_q is a time constant common for all output neurons. The strength of the synapse from the j_{th} input neuron to the i_{th} output neuron is expressed by:

$$\frac{dw_{ij}}{dt} = \alpha q_i v_j^I, \tag{3.6}$$

where α is the learning rate, v_j^I is the instant firing rate of the j_{th} input neuron, and q_i is the trace value of i^{th} output neuron. After each weight update, the lenght of the weight vector for each ouput neuron i, that is $w_i = (w_i 1, ..., w_{i,N_i})$ where there are N_I input neurons, is renormalised by:

$$w_i := \frac{w_i}{||w_i||} \tag{3.7}$$

Competitive learning in self-organizing maps consists of the comparison between a given input pattern and each weight vector w_i of output neurons. The weight vector that is the most similar to the given pattern is declared as the winner, and the winner and its neighbours are moved close to the input pattern by an update weight rules as 3.6.

3.2.2 Results

Two initial models were tested and trained on the same stimuli, the first model had peaked eye-position modulation in the input population, the input modulation for the other model had a sigmoid form. Training procedures for both models last 20 epochs and for each training epoch, the visual target acquires eight various head-centred training locations and while the visual target is in a fixed location, the eye position is changed as the model performs a series of saccades. These models were trained via using the stander trace learning rule, and the model with peaked gain-modulated input populations successfully developed head-centered outputs. The result of training the model with the sigmoid gain-modulated inputs using the standard trace learning rule brought an interesting finding; output neurons of this model after training procedure develop ed representations in eye-centered reference frame instead of estimated head-centred. These results imply the nature of competitive learning, thus the output neurons are learned to reply to the input neurons that tend to be co-active, and for peaked gain modulation model, the input neurons response very frequently and co-actively to circular cluster highly localised in both retinal and eye-position preference dimensions. Then, if a trace learning rule is implemented, output neurons can simply bind together such circular clusters along a diagonal line in the input space that corresponds to a particular head-centered location. Another situation occurs if sigmoid gain modulated input neurons are used; in such a case, the subsets of input neurons that tend to be frequently co-active are located in the retinotopic preference dimension but elongated in the eye-position preference dimension, that is caused by monotonic saturation of the response function of input neurons in eye-position dimension of the input space. That is a reason why using the standard Hebbian learning rule, the particular output neurons will learn to respond to these elongated clusters, and that causes eye-centered output responses.

After the introduction of a trace learning rule, the output neurons will not start to respond as head-centered but vice versa the output neurons tend to respond as eye-centered again.

In order to avoid the above-mentioned problem with sigmoid gain modulated input eye-position neurons, the authors investigated how to ensure developing head-centered visual output neurons and preserve the biologically plausible way. For this purpose, they explored several models with more sophisticated learning rules that include ways like temporally delayed memory traces, or a mixture of both trace learning and anti-Hebbian learning terms. Such modified learning rules preserve biological plausibility and are capable to drive self-organization of head-centered output neurons. These findings provide a view of how head-centered neurons can be developed in the brain and also demonstrate how to apply self-organized models for the task of a coordinate transformation.

Methods

4.1 Artifical neural networks

The human brain can be described as a highly complex, nonlinear, and parallel information-processing system capable to organize its information-processing units know as *neurons* to solve plenty of such cognitive tasks as, for example, coordinate transformation or multisensory integration. Investigations of the brain have brought the inspiration that was at the beginning of the history of *artificial neural networks.* In general, an artificial neural network is a machine, that is usually implemented by electronic components or is simulated in software, and is designed to perform a particular task or function of interest. The neural networks are characteristic by strong interconnectivity among neurons through those basic computations are performed. Sometimes a neural network is defined as an adaptive machine (Haykin, S., 2016):

> *A neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:*
>
> 1. *Knowledge is acquired by the network from its environment through a learning process.*
> 2. *Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.*

As already mentioned above neuron is a basic computation or information-processing unit of neural networks. In order to achieve a better understanding of neural networks may be helpful to elucidate the model of an artificial neuron. For this purpose, we introduce the Figure 7 that illustrates the model of a single neuron. Via the figure, we can observe three basic elements of a single artificial neuron. The first of them, the connecting links between every input and a specific neuron, those links are also known as *synapses* and are characterized by *strength* or *weight.* For example, the input x_1 is connected with the neuron k through the synapse characterized by the weight labelled as w_{k1} which determines the strength of the synapse, what basically means, that the input x_1 is multiplied by the weight w_{k1}.

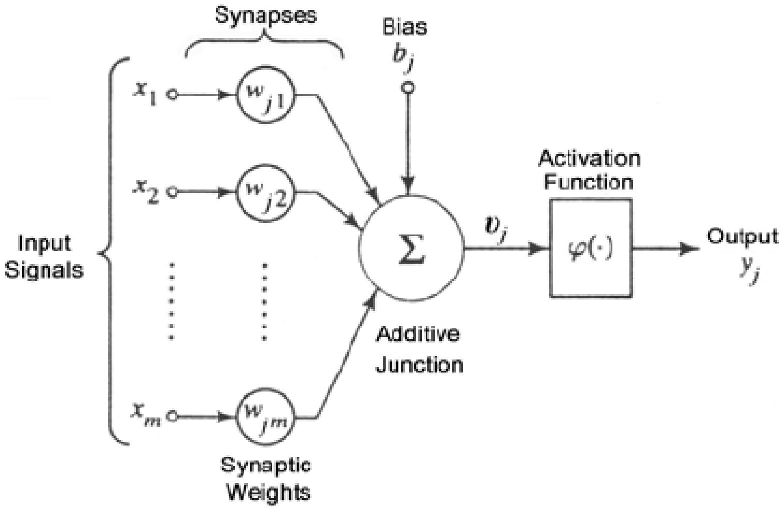

Figure 7: Nonlinear model of a neuron, labeled k. Typically, the normalized amplitude range of the output of a neuron is written as the closed unit interval [0,1], or, alternatively, [-1,1]. Adapted from Haykin, S. (2016)

The second element of a neuron is the summing junction that basically sums each of input signals weighted by the particular synapse strength, in mathematical terms, this operation may be described by the following equation:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j,$$

where u_k is the result of summing operation for a single neuron, this result is also named as *input net.* The last element is an *activation function* that formed the output of a neuron applying an elected function $\varphi\,(.)$ on the input net. On the figure, we can also observe the element labelled as b_j what express external *bias* that is responsible for increasing or decreasing the input net of the activation function (Haykin, S. (2016)). Finally, we can express the output of the neuron k as the following equation:

$$y_j = \varphi\left(\sum_{j=1}^{m} w_{kj}x_j + b_j\right)$$

There are a lot of different architectures of neural networks, where each architecture has its own unique properties and is designed for specific tasks, or better said, every architecture is more suitable for some kinds of tasks and less capable to perform other kinds of tasks. For illustrative purpose, we introduce the model named MLP (*multi-layer percepton*). The MLP is composed of an input layer, hidden layers, and output layer. The figure 8 provides a look at this layout.

The image illustrates 8 a type NN that is called *feed-forward*, this term groups neural models within that the input signals are transmitted in one direction, thus, there are not loop, so the output of a particular neuron does not affect the neuron itself. Feed-forward networks do not also contain any *backward* connections between two neurons, the kind of NN that allows such connections are called *feed-back networks*. The input layer of MLP is not composed of such neurons as we described above, but its job is just to transmit the input signals to the first hidden layer. Hidden layers are layers that have not a direct connection to the environment (Popescu, M. et al.).

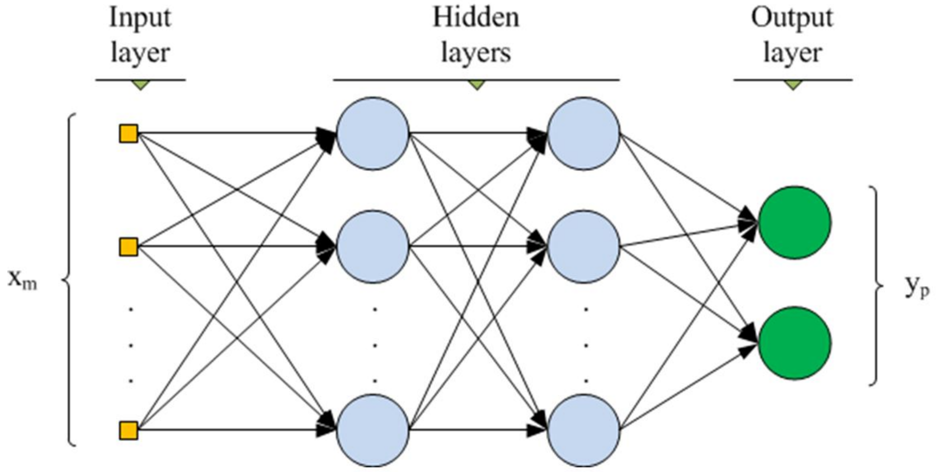

Figure 8: Typical architecture of an MLP with two hidden layers. (Geronimo, T. M. et al.)

Weights on all synapses between every two adjacent layers are *learned* by a procedure named *learning* or *learning algorithms*. During training of a feed-forward neural network, a learning algorithm drives hidden layers to acquire the best possible input-output mapping. In more mathematical terms, we can label the best-approximating function of real estimated outputs as $f^*(x)$ and let $f(x)$ be the function that is driven by a learning algorithm to match $f^*(x)$. The function $f(x)$ in our case on the Figure 8 composes together three different function to the form:

$$f(\mathbf{x}) = f^{(3)}\left(f^{(2)}\left(f^{(1)}(\mathbf{x})\right)\right),$$

where $f^{(1)}$ is the first hidden layer, $f^{(2)}$ is the second hidden layer, and $f^{(3)}$ is the output layer. Then, the learning algorithm is responsible to drive each of the functions $f^{(1)}$ - $f^{(3)}$ to acquire an approximation of $f^*(x)$. Artificial neural networks can be used to learn some task of regression or classification, where for classification, $y = f^*(x)$ maps an input x to a category y. For a regression problem, an input x is mapped to the particular numeric target y.

4.1.1 UBAL

In this section, we want to introduce the biologically-inspired algorithm named *UBAL* (Universal Bidirectional Activation-based Learning). UBAL is one of the answers to a biologically-more-plausible alternative to the BPROP algorithm that is based on the backward propagation of errors, and its biological plausibility is contentious (Farkas, I. Malinovská, K. (2013)).

In order to provide a more biological-plausible algorithm, O'Reilly (1996) introduced the model named *Generalized Recirculation* (GeneRec) that is designed to avoid the computation of error derivates. This model consists of three layers with full connectivity between layers, and the learning process is based on local differences of activations. Activations of model flow in two directions through the same weight matrix. The flow of activation from input to output is named *minus* phrase. In the second phrase, *plus* phrase, desired target is *clamped* on output layer and activations flow back from output to hidden layer. The learning rule is up to the equation:

$$\Delta w_{pq} = \lambda a_p^- \left(a_q^+ - a_q^-\right),$$

where λ is the learning rate, a_p^- labels the presynaptic and a_p^- postsynaptic activation of a unit in the minus phrase, and a_p^+ denotes the presynaptic activation from the plus phase. (Farkas, I. Malinovská, K. (2013))

Based on GeneRec, Farkas, the model named BAL was proposed by Farkas, I. Malinovská, K. (2013) that as well as GeneRec is based on the bi-directional flow of activation and is composed of three layers. The difference between GeneRec and BAL is that BAL does not use the same weight matrix for minus and plus phrase but uses different weight matrix for both directions. However, BAL failed to reach 100% convergence on basic tasks as the canonical 4-2-4 encoder task. Therefore, the authors have brought an improvement of BAL named *UBAL*.

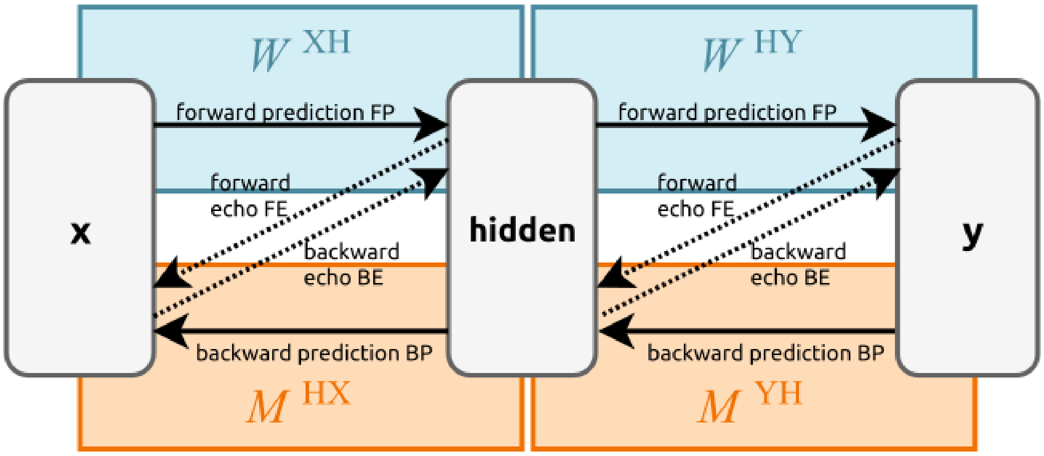

Figure 9: Activation propagation in a network with input-output layers x and y and one hidden layer.

The layer's architecture and flows of activations are illustrated in Figure 9. UBAL uses two separate matrices W and M for each direction of activation flow. In contrast with BAL, both flows contain two phases: *prediction* and *echo*. In the forward prediction phase, input signals are propagated to the layer p and subsequently, the activations are propagated to the layer q, the backward prediction phase is carried out in reverse. Echo activation phases consist of echoing previous outputs q_{FP} and p_{BP} back to p and q via matrices M and W. The table 1 depicts these phases in mathematical terms.

Forward prediction	FP	$q_j^{FP} = \sigma\left(\sum_i w_{ij} p_i^{FP} + b_j\right)$
Forward echo	FE	$p_i^{FE} = \sigma\left(\sum_j m_{ji} q_j^{FP} + d_i\right)$
Backward prediction	BP	$p_i^{BP} = \sigma\left(\sum_j m_{ji} q_j^{BP} + d_i\right)$
Backward echo	BE	$q_j^{BE} = \sigma\left(\sum_i w_{ij} p_i^{BP} + b_j\right)$

Table 1: Propagation of activations between two layers p and q.

The learning rules updating weights of matrices W and M are the following:

$$\Delta w_{ij} = \lambda t_i^B \left(t_j^F - e_j^F \right)$$

$$\Delta m_{ij} = \lambda t_j^F \left(t_i^B - e_i^B \right),$$

where λ is the learning rate and the rest of elements is described in the table 2.

Forward target	t_j^F	$\beta_q^F q_j^{FP} + \left(1 - \beta_q^F\right) q_j^{BP}$
Forward estimate	e_j^F	$\gamma_q^F q_j^{FP} + \left(1 - \gamma_q^F\right) q_j^{BE}$
Backward target	t_i^B	$\beta_p^B p_i^{BP} + \left(1 - \beta_p^B\right) p_i^{FP}$
Backward estimate	e_i^B	$\gamma_p^B p_i^{BP} + \left(1 - \gamma_p^B\right) p_i^{FE}$

Table 2: Definition of elements used in learning rules

Every ANN has attributes that are set before the training procedure. These attributes are called *hyperparameters* and their best configuration, for a particular learning problem, is searched experimentally. Hyperparameters of UBAL are described in table 3.

Learning rate	λ
Hidden layer size	h
Initial weights: mean ν and variance σ^2	$N(\nu, \sigma)$
Forward clamping strength	β^F
Backward clamping strength	β^B
Forward estimate strength	γ^F
Backward estimate strength	γ^B

Table 3: UBAL hyperparameters

4.2 The iCub: Humanoid Robot

Robotic platforms are being used by cognitive scientists with great popularity. One of these robotic platforms is the iCub, a humanoid robot replicating the size of a three-and-half years old child, the platform was developed within a collaborative project *RobotCub* funded by the Europan Commission. The height of the robot was proposed to correspond to the size of the 3.5 years old child, which is approximately 100cm, the overall weight of the is around 22kg. The iCub provides 53 degrees of freedom (DoF), that allow the robot's moving ability and the ability to manipulate with objects (Metta, G et al., 2018). The figure 10 show the iCub robot and its proportions.

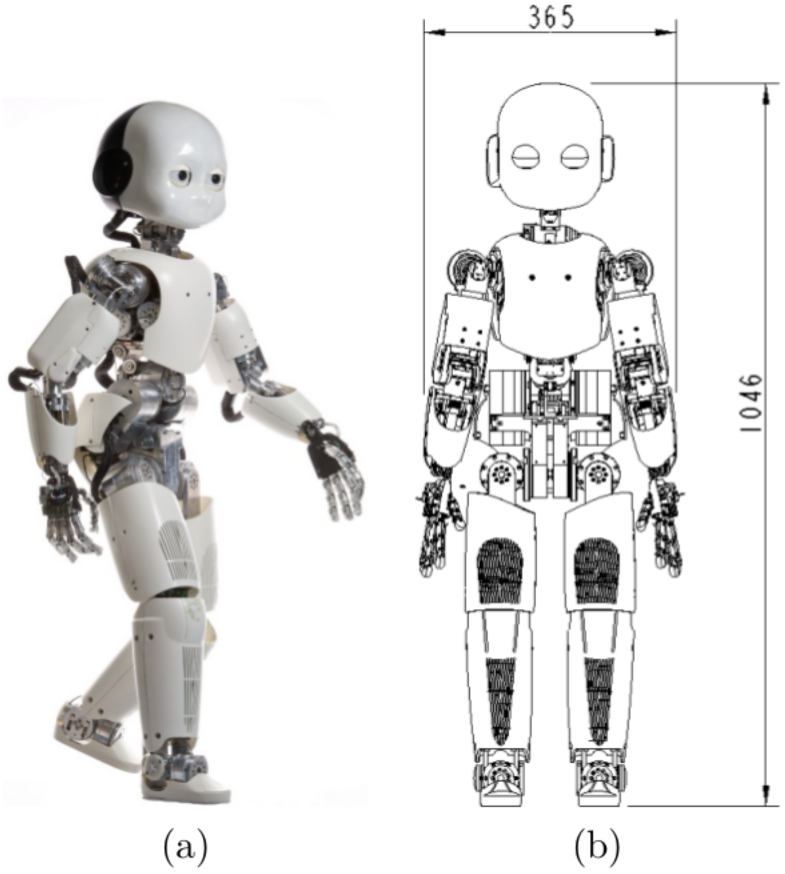

Figure 10: The iCub. The figure shows a photograph of the iCub robot (a). The overall robot dimensions are shown in (b) (Parmiggiani, A et al., 2012)

Such a kind of a real humanoid robot, as the iCub is, can be too expensive for some science departments. The cost of the one iCub is about 200,000$. Therefore, a lot of researchers uses a robotic simulator instead. A robotic system provides a biased model of the real environment, that although is not so fully complex as the real environment, but might be very useful, for example, in order to test algorithms performing cognitive functions or to collect suitable data for neural modelling via ANNs (Tinkhanoff, V. et al., 2012). The simulator of the robot iCub is distributed as Open-Source system following the GPL licenses, and it is supported on Windows, Linux, and also MacOS (Parmiggiani, A et al., 2012).

The iCub simulator is based on the widely used physics engine called Open Dynamics Engine. The ODE is used for simulating rigid bodies and for computing the physical interaction with objects. In order to render the scene, the combination of OpenGL and SDL is used, that enables to render the scene in such an easy way that is sufficient to carry out computationally-efficient simulation experiments. Another tool, that the iCub simulator is based on, is an open-source middleware called YARP. The YARP is very suitable for real-time applications interfacing with complex and changing hardware. From the view of the device API or network, the real robot and the simulator have the same interface and are interchangeable from the user perspective (Tinkhanoff, V. et al., 2012). The architecture of the simulator is illustrated in the figure 11.

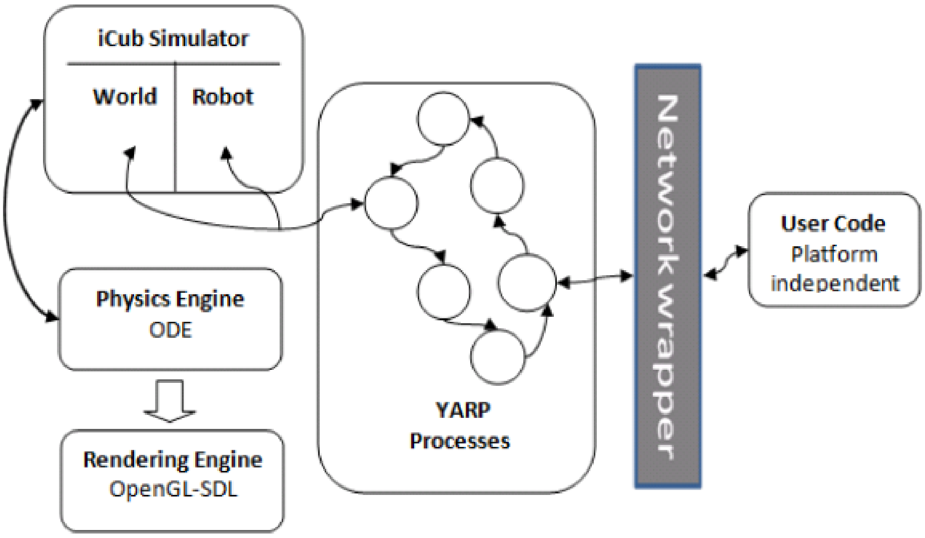

Figure 11: This figure shows the architecture of the simulator with YARP support. The User code can send and receive information to both the simulated robot itself (motors/sensors/cameras) and the world (manipulate the world). Network wrappers allow device remotization. The Network Wrapper exports the YARP interface so that it can be accessed remotely by another machine (Tinkhanoff, V el al., 2008)

4.2.1 The Cartesian Controller

The Cartesian Controller has been proposed in order to control arms and legs directly in operation space instead of commanding new configuration joints (Link iCubsim). In order to develop a Cartesian controller that is capable to deal with issues like a large number of DoF, computing of trajectories effectively and producing of smooth movements, the authors (Pattacini et al!!) designed a Cartesian Controller around the iCub robot.

Reaching the cartesian position of a target by the Cartesian Controller is performed in the two modules:

- the Solver module including *a nonlinear optimization technique*,
- the Controller module consisting of *a biologically inspired kinematic controller*.

The first module allows computing the arm joints configuration that reaches the desired position. The second stage, the biologically inspired kinematic controller, is responsible for performing a human-like trajectory of the end-effector by computing the velocities of motors (Pattacini et al!!).

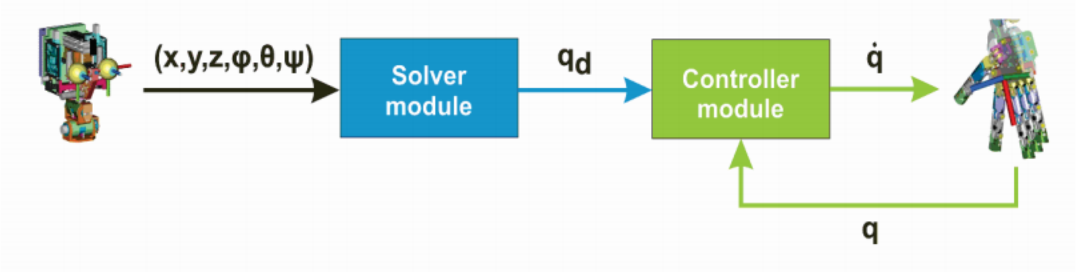

Figure 12: Diagram of proposed Cartesian controller (Pattacini et al!!).

In order to introduce an example of working with the Cartesian Controller that is used in our experiments, we introduce the usage of the methods `askForPosition(...)`. This method returns the joints configuration that achieves the desired position without performing movements. For instance, when we would want to know the joint configuration for the position that is determined by $x = -0.25$, $y = 0.0$ and $z = 0.1$, then the usage of the methods is the following:

```
Vector xd(3);
xd[0]=-0.25;
xd[1]=0.0;
xd[2]=0.1;
Vector xdhat, odhat, qdhat;
icart->askForPosition(xd,xdhat,odhat,qdhat);
```

Where `xd` is the vector determining the desired position, the parameters `xdhat`, `odhat` and `qdhat` are filled up inside the implementation of the method. `Xdhat` and `odhat` represents the results of the optimization process for the cartesian position and orientation, respectively. The parameter `qdhat` contains the joints configuration composed of three joints for the robot's torso and seven joints for the arm.

4.2.2 The Gaze Controller

A robotic system that is supplied with moving cameras has to be able to focus on the object of interest in order to perform such operations like the exploration of the surrounding environment, manipulation with seen objects or tracing an object. Performing of mentioned tasks includes movements of the neck and both eyes that must be performed as effectively and smoothly as possible. The iCub robot's head has six DoFs and that create such a complex kinematic that achieve the complexity of the human oculus system.

(Authors) designed a gaze control architecture that is implemented around the iCub's technologies. Their approach of how to solve controlling of a robot's gaze consists in the explicit control of the 3D point that is defined by both eyes of a generic binocular system. This *fixation point* can be defined as a virtual end-effector that is the result of the kinematic involving the eyes and neck (see the figure 13).

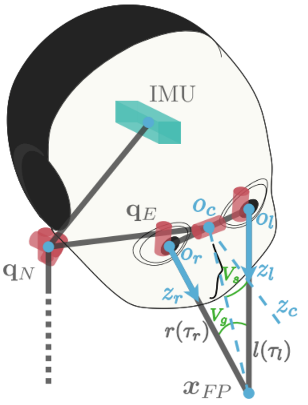

Figure 13: Kinematics of the head system for the iCub humanoid robot. It is composed of a 3-DoFs neck and a 3-DoFs binocular system, for a total of 6-DoFs (depicted in red). Each of these joints is responsible for the motion of the fixation point. The Inertial Measurement Unit (IMU) is the green rectangle, whose motion is not affected by the eyes' movements.

We made use of the gaze controller in order to find out where the robot is looking at. For this purpose, there is the method textttgetFixationPoint(...) that returns the current fixation point. The following example corresponds to the call of the method in C++.

```
Vector x;
igaze->getFixationPoint(x);
```

The vector x contains, after successful carrying-out, the fixation point corresponding to the current configuration of the neck and the eyes.

4.3 Neuronal Population Vector

The *neuronal population vector* has been introduced by Georgopoulos et al. 1983, who investigated the brain mechanisms coding the direction of the arm in 2D space. Their approach consisted in recording activities of cells in the monkey motor cortex while the monkeys were performing various arm movements of different directions. Particular neurons of the neural population had the orderly variation of neural activity corresponding to the directional tuning of rate of firing of action potentials shaped in the form that the highest firing was reached with a particular movement (named as the *preffered direction* of a cell), and activity was being decreased progressively as movements differed more from the preferred direction. The function describing the behaviour of activity of a particular cell against its preferred direction is known as the directional *tuning curve.* The example of a turning curve is illustrated in figure 14 (Mahan, M. et al., 2014).

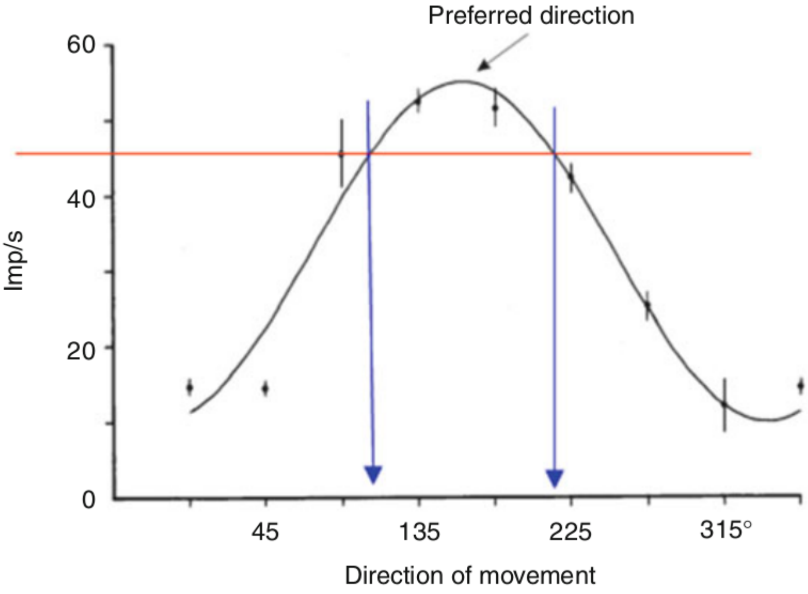

Figure 14: A fitted cosine curve with the preferred direction at the peak of the fitted curve. The ambiguity of the curve with respect to other movement directions is indicated by the red horizontal line at a given discharge rate: the line crosses the curve at two symmetric points corresponding to two different movement directions (blue arrows). This ambiguity points to the need for a unique, unambiguous coding of movement direction by a neuronal population. Adapted from Mahan, M. et al., 2014.

We introduce the example of the neuronal population vector where the activity tuning function of every single neuron is *gaussian* function, thus the activity of a particular neuron is the result of the following equation:

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}},$$

where the parameter a determines the height of the curve's peak, b determines where the center of the curve's peak is placed at, and the parameter c controls the width of the curve.

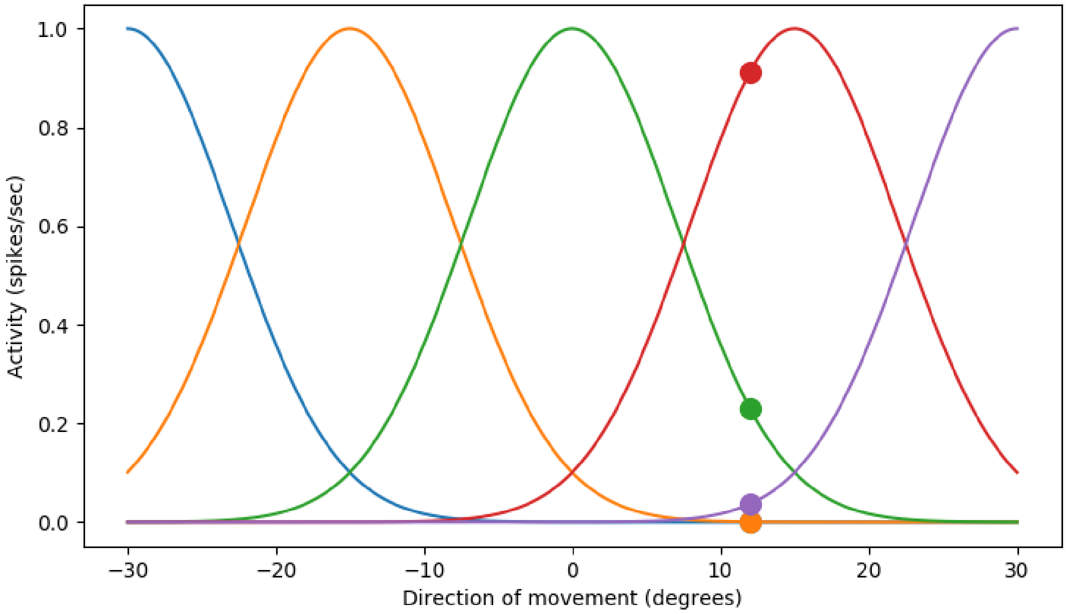

Figure 15: The example of a neuronal population vector with gaussian tuning curves. The population is composed of five neurons with preferred directions uniformly distributed over the interval $< -30, 30 >$. This figure illustrates the encoding of the value 12° that corresponds to the following activities of the neurons: $\{0, 0, 0.23, 0.912, 0.036\}$.

Consider the population of neurons that is composed of five neurons. Preferred directions of neurons are uniformly distributed on the interval $< -30, 30 >$, and the parameters of their Gaussian tuning curve are $a = 1$, $c = 7$ and values of b for each of the five neurons correspond to $[-30, -15, 0, 15, 30]$. Then, for instance, the value $x = 12°$ encoded by activities of the population's neurons corresponds to values: $\{0, 0, 0.23, 0.912, 0.036\}$, the same example is also illustrated in figure 15.

Experiments

We carried out three experiments of learning coordinate transformation via UBAL in order to demonstrate the capability of the model to learn such a complex task as coordinate transformation is, and to point out to properties of neurons of learned models in comparison with other experiments performed by learning an ANN, or with real neurons in the brain. The first experiment consists in learning of coordinate transformation where one modality is mapped on another one, concretely, the direction of the gaze is mapped on angles of the arm which correspond to the point where the gaze is focused on. The aim of this experiment had to test the capability of UBAL model to learn the mapping of data from the more complex dataset. The second and third experiment was designed to map two modalities on one modality what allow us to examine the presence of gain modulation in neurons of hidden layers. The mapping consisted of the gaze direction and the position of an object on retinal images mapped on angles of the hand. The difference between these two experiments was only in encoding input and output data, whereas the first experiment had only data scaled to the same interval, in the second was used the method of neuronal population vector described in 4.3. We describe each of the performed experiments in more detail in the following separate sections.

5.1 1-1

5.2 Training and Testing

5.3 Results

Bibliography

[1] ANDERSEN, Ra and MOUNTCASTLE, Vb. The influence of the angle of gaze upon the excitability of the light- sensitive neurons of the posterior parietal cortex. The Journal of Neuroscience. January 1983. Vol. 3, no. 3p. 532–548. DOI 10.1523/jneurosci.03-03-00532.1983.

[2] BATISTA, Aaron. Inner space: Reference frames. Current Biology. 2002. Vol. 12, no. 11., doi:10.1016/s0960-9822(02)00878-3.

[3] BLOHM, Gunnar and CRAWFORD, J. Douglas. Fields of Gain in the Brain. Neuron. 2009. Vol. 64, no. 5p. 598–600. DOI 10.1016/j.neuron.2009.11.022.

[4] BLOHM, G., KHAN, A.z. and CRAWFORD, J.d. Spatial Transformations for Eye–Hand Coordination. Encyclopedia of Neuroscience. 2009. P. 203–211. DOI 10.1016/b978-008045046-9.01102-5.

[5] COLBY, Carol L. Action-Oriented Spatial Reference Frames in Cortex. Neuron. 1998. Vol. 20, no. 1p. 15–24. DOI 10.1016/s0896-6273(00)80429-8.

[6] COMMITTERI, G., et al. Reference Frames for Spatial Cognition: Different Brain Areas are Involved in Viewer-, Object-, and Landmark- Centered Judgments About Object Location. Journal of cognitive neuroscience. November 2004. Vol. 16, p. 1517–1535. DOI 10.1162/0898929042568550.

[7] CRAWFORD, J. D., MEDENDORP, W. P. and MAROTTA, J. J. Spatial Transformations for Eye–Hand Coordination. Journal of Neurophysiology. 2004. Vol. 92, no. 1p. 10–19. DOI 10.1152/jn.00117.2004.

[8] FARKAŠ, Igor and REBROVÁ, Kristína. Bidirectional Activation-based Neural Network Learning Algorithm. Artificial Neural Networks and Machine Learn-

ing – ICANN 2013 Lecture Notes in Computer Science. 2013. P. 154–161. DOI 10.1007/978-3-642-40728-4_20.

[9] FINK, G. Space-based and object-based visual attention: shared and specific neural domains. Brain. January 1997. Vol. 120, no. 11p. 2013–2028. DOI 10.1093/brain/120.11.2013.

[10] FINK, G. R., MARSHALL, J. C., SHAH, N. J., WEISS, P. H., HALLIGAN, P. W., GROSSE-RUYKEN, M., ZIEMONS, K., ZILLES, K. and FREUND, H.-J. Line bisection judgments implicate right parietal cortex and cerebellum as assessed by fMRI. Neurology. 2000. Vol. 54, no. 6p. 1324–1331. DOI 10.1212/wnl.54.6.1324.

[11] GALATI, Gaspare, PELLE, Gina, BERTHOZ, Alain and COMMITTERI, Giorgia. Multiple reference frames used by the human brain for spatial perception and memory. Experimental Brain Research. 2010. Vol. 206, no. 2p. 109–120. DOI 10.1007/s00221-010-2168-8.

[12] GALATI, Gaspare, LOBEL, Elie, VALLAR, Giuseppe, BERTHOZ, Alain, PIZZAMIGLIO, Luigi and BIHAN, Denis Le. The neural basis of egocentric and allocentric coding of space in humans: a functional magnetic resonance study. Experimental Brain Research. May 2000. Vol. 133, no. 2p. 156–164. DOI 10.1007/s002210000375.

[13] GEORGOPOULOS, A., SCHWARTZ, A. and KETTNER, R. Neuronal population coding of movement direction. Science. 1986. Vol. 233, no. 4771p. 1416–1419. DOI 10.1126/science.3749885.

[14] GERONIMO, Thiago M., CRUZ, Carlos E. D., CAMPOS, Fernando De Souza, R., Paulo and C., Eduardo. MLP and ANFIS Applied to the Prediction of Hole Diameters in the Drilling Process. Artificial Neural Networks - Architectures and Applications. 2013. DOI 10.5772/51629.

[15] GOODFELLOW, Ian, BENGIO, Yoshua, COURVILLE, Aaron and UNDEFINED, undefined undefined. Deep Feedforward Networks. In : Deep learning. Cambridge, MA : MIT Press, 2017. p. 200–203.

[16] HAYKIN, S. Introduction. In : Neural networks and learning machines. 3. Delhi : Pearson, 2016. p. 1–38.

[17] HONDA, M. Cortical areas with enhanced activation during object-centred spatial information processing. A PET study. Brain. January 1998. Vol. 121, no. 11p. 2145–2158. DOI 10.1093/brain/121.11.2145.

[18] HAGLER, D. J. Jr., RIECKE, L., and SERONO, M. I. Parietal and superior frontal visuospatial maps activated by pointing and saccades. Neuroimage. 2007 May 1; 35(4): 1562–1577. DOI 10.1016/j.neuroimage.2007.01.033

[19] KUMAR, Arvind and BARVE, Shrish. Newtonian Relativity. In : How and why in basic mechanics. Hyderabad : Universities Press, 2002. p. 114–117.

[20] MAHAN, Margaret Y. and GEORGOPOULOS, Apostolos P. Neuronal Population Vector. Encyclopedia of Computational Neuroscience. 2014. P. 1–7. DOI 10.1007/978-1-4614-7320-6_401-1.

[21] MALINOVSKÁ, Kristína, MALINOVSKÝ, Ľudovít and FARKAŠ, Igor. Towards More Biologically Plausible Error-Driven Learning for Artificial Neural Networks. Artificial Neural Networks and Machine Learning – ICANN 2018 Lecture Notes in Computer Science. 2018. P. 228–231. DOI 10.1007/978-3-030-01424-7_23.

[22] METTA, G; SANDINI, G; VERNON, D; NATALE, L and NORI, F. iCub: the open humanoid robot designed for learning and developing complex cognitive tasks (2019).

[23] O'REILLY, R. Biologically plausible error-driven learning using local activation differ-ences: The generalized recirculation algorithm. Neural Computation 8(5), 895–938 (1996)

[24] PARMIGGIANI, Alberto, MAGGIALI, Marco, NORI, Francesco and SCHMITZ, Alexander. The Design of the iCub Humanoid Robot. International Journal of Humanoid Robotics. December 2012. Vol. 9, no. 4. DOI 10.1142/S0219843612500272.

[25] POPESCU, Marius and BALAS, E., Valentina adn PERESCU-POPESCU, Liliana and MASTORAKIS, Nikos. Multilayer perceptron and neural networks. WSEAS Transactions on Circuits and Systems. 2009 July 8.

[26] ROJAS, R. The Backpropagation Algorithm. In : Neural Networks. Berlin : Springer-Verlag, 1996. p. 155–157.

[27] SALINAS, Emilio and SEJNOWSKI, Terrence J. Book Review: Gain Modulation in the Central Nervous System: Where Behavior, Neurophysiology, and Computation Meet. The Neuroscientist. 2001. Vol. 7, no. 5p. 430–440. DOI 10.1177/107385840100700512.

[28] SOECHTING, J. and FLANDERS, M. Moving In Three-Dimensional Space: Frames Of Reference, Vectors, And Coordinate Systems. Annual Review of Neuroscience. January 1992. Vol. 15, no. 1p. 167–191. DOI 10.1146/annurev.neuro.15.1.167.

[29] SERENO, Martin I and HUANG, Ruey-Song. Multisensory maps in parietal cortex. Current Opinion in Neurobiology. 2014. Vol. 24, p. 39–46. DOI 10.1016/j.conb.2013.08.014.

[30] TIKHANOFF, V and FITZPATRICK, P and NORI, F and NATALE, L and METTA, G and CANGELOSI, A. The iCub Humanoid Robot Simulator. IROS Workshop on Robot Simulators (2012).

[31] TIKHANOFF, V., CANGELOSI, A., FITZPATRICK, P., METTA, G., NATALE, L. and NORI, F. An open-source simulator for cognitive robotics research. Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems - PerMIS 08. 2008. DOI 10.1145/1774674.1774684.

[32] XING, Jing and ANDERSEN, Richard A. Memory Activity of LIP Neurons for Sequential Eye Movements Simulated With Neural Networks. Journal of Neurophysiology. 2000. Vol. 84, no. 2p. 651–665. DOI 10.1152/jn.2000.84.2.651.

[33] ZIPSER, David and ANDERSEN, Richard A. A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. Nature. 1988. Vol. 331, no. 6158p. 679–684. DOI 10.1038/331679a0.