Real Time Decoding for Brain-Machine Interface Applications

Jonathan Becedas[1]; Rodrigo Quian Quiroga[2]

Centre for Systems Neuroscience, University of Leicester, Leicester, United Kingdom

[1]jbr6@leicester.ac.uk; [2]rqqg1@leicester.ac.uk

***Abstract*-There is substantial evidence that it is possible to predict movement intentions from single cell recordings in monkeys and since more recently, in humans. Such predictions, using decoding algorithms, have a large potential for clinical applications in order to drive robotic devices to be used by paralyzed patients or amputees. In spite of these advances, it is still not clear how accurate and practical the movements obtained from real neuronal devices could be.**

In this work, an original decoding method to perform movements to different locations was proposed and studied in realistic simulations. The method provides a high level control command to a Brain-Machine Interface device, which is a precise estimation of the target location as a function of the number of recorded neurons. Finally the method was applied to a 7 Degrees of Freedom (DOF) anthropomorphic robotic arm for reaching and grasping an object.

Keywords- Brain-Machine Interface; Neuroprosthetic Devices; Neural Decoding; Parietal Cortex

I. INTRODUCTION

Given the exquisite accuracy of neurons to control the whole body and thoughts, there is clear hope that appropriate measurements in specific parts of the brain and a correct decoding of the recorded information can transform these bioelectrical signals into high level commands to control artificial devices [1, 2]. In particular, a great scientific effort has been devoted to provide clinical advances to assist paralyzed patients with Brain-Machine interfaces (BMI) such as prosthetic devices driven by neural signals [3-5].

The first success in BMI was the cochlear implant [6]. These prostheses were clinically applied in patients with profound sensorineural deafness to restore functional hearing. Since that breakthrough, the BMI and neuroprostheticsfields advanced at very high speed [7], providing extraordinary scientific results. Some of the most important advances in BMI applications are the following: i) real time control of a cursor on a computer screen [8, 9], ii) opening the email or control a television with neural control [10], iii) control of robotic systems [11-19], as well as the development of communication systems to help paralyzed patients to interact with the external world [20-25]. For a comprehensive review on BMI and neuroprosthetics, see [1-4, 26, 27]. Furthermore, a discussion about the most used techniques in Brain-Machine Interface can be found in [28] and [3].

Many contributions in neuroprosthetics and BMI have been made by using invasive approaches based on extracellular recordings [29]. Implants of micro-electrode arrays in the brain can record single-cells and provide signals to drive artificial devices [30]. While some studies with invasive approaches focus on the motor cortex, some others selected the Parietal Cortex as input to the neuroprosthetic devices:

- Recordings from primary motor cortex: some of the works in the scientific literature [2, 4, 8, 9, 13, 31, 32] focused on the use of recordings from the primary motor cortex (M1). Motor cortical areas control the movements of the body, thus the activity related to hand trajectories can be used to control artificial devices.
- Recordings from Posterior Parietal Cortex: other works [33, 34] focused on recordings from the Parietal Reach Region (PRR) in the Posterior Parietal Cortex. Posterior Parietal Cortex encodes high level signals such as the intention of movement.

Despite of the area used to obtain appropriate signals from the brain to drive artificial devices, all these works require the implementation of a neural decoder that translates the bioelectrical signals recorded from the neural activity to high level signals that could be used to drive artificial prostheses or machines.

Different neural decoding methods have been proposed to control neuroprosthetic devices:

- In [35] a forward method was proposed to decode the next state of the trajectory motor planning. Forward methods require the computation of a model in which the parameters must be estimated and continuously recalculated to update the estimate, which increases the computational costs of the process.
- In [13] a real time prediction of hand trajectory was applied in primates. Recorded units from multiple cortical areas (premotor cortex, primary motor cortex and Parietal Cortex) were used in the trajectory decoding.

Other decoding methods were developed to simplify the decoding task:

- In [9] a method was developed to reduce the number of features to characterize from the neural activity. This was used to do real-time brain control of a 3D computer cursor by predicting upper limb movements from motor maps of the cortex. This algorithm required 32 to 40 days of training to adjust the parameters.
- In [36] a method to obtain an estimation of the trajectory was proposed, but this implies to carry location errors that increase with the time from the very beginning of the task.
- In [19] a population vector decoder [37-39] was applied in real time to control the task of self-feeding of a monkey with a robotic arm. The decoding method was based on the contribution to the movement of the whole recorded population of neurons in motor cortex. The method had implicit estimation of the robotic arm velocities, which were the input to control the artificial device. It provided well directed but unnatural movements of the robotic arm by using point to point small movements in short periods of time. The method, however, required a long training time to provide accurate results (several days). This occurred because continuous feedback between the brain and the device required a strong plasticity of the brain to control the artificial prosthesis and produce point to point movements, as reported in that work.
- In [40] a closed loop decoding method was proposed to improve the performance of BMI systems through a learning process.

While the previous works dealt with trajectory decoding, which involves recordings from motor cortex and requires high brain plasticity, other scientists proposed a decoding method to decode the target location of the movement intention:

- In [34] the methodology focused on the decoding of the intended goal rather than in the manner of moving the hand and plan the movement, which is very attractive to neuroprosthetic applications, since the decoded signals from the brain are high level signals that can be used as inputs of a prosthetic device. The time required to obtain an estimation of the target location was between 200 and 1100 ms; time limits the implementation of the method in real time tasks, mainly if a continuous feedback is required.

A review of decoding methods used in cognitive prosthetics can be found in [41].

In this work a decoding method for single cells recording in Posterior Parietal Cortex (PPC) was studied. Parietal Cortex was chosen to be decoded because in this area of the brain, high level commands such as the intention of movement of the arms is encoded, as demonstrated in [42]. The decoding method is based on the estimation of the receptive fields of the neurons [43], which directly encode the exact location of the movement intention. Thus, a fast, intuitive movement could be carried out with a prosthetic device without the requirement of small movements, which makes success difficult in the completing tasks.

The proposed method is simple and easy to implement in realistic applications since the training times required are short (10-20 minutes) and the estimation times were found to be short enough (4 ms) to be applied in real time with a robotic device.

Preliminary simulation results of this methodology were presented in [44]. In the present work, new results are shown and the methodology was applied to a 7 Degrees of Freedom (DOF) anthropomorphic robotic arm to study the performance of the method in real grasping tasks in real time. To encode the object location inside the robot workspace, an object recognition system constituted by a monocular camera was used and a pattern recognition software tool was implemented.

II. MATERIALS AND METHODS

A simulation environment was studied with the objective of developing a new and efficient neural decoding methodology to be used for real time control of neuroprosthetic devices and Brain-Machine Interfaces. The method proposed was evaluated in terms of the number of neurons in Posterior Parietal Cortex that would be necessary for reaching and grasping tasks in realistic applications. The activity of a population of PPC neurons was generated with characteristics similar to those found in real data, such as the firing rate and the receptive fields among others.

There is evidence that the Posterior Parietal Cortex has a special role in the planning and execution of movements in primates [43, 45, 46] and humans [47]. In addition, many researchers have shown that it is possible to predict the movement plans from cortical neurons [33, 34, 41, 43, 45, 46]. Based on these two considerations, the firing of neurons from PPC was simulated to decode the intention of movement into a 2-dimensional workspace [43]. The receptive fields of the PPC neurons were randomly distributed along the workspace. Those neurons spontaneously fired following a Poisson distribution and increased their mean firing rate whenever the target laid within their receptive fields. The firing rate was recorded from the simulated neurons and the target location decoded by means of a neural decoder.

A simulation environment was studied with the objective of analyzing the decoding methodology proposed to be applied in real time neuroprosthetics and Brain-Machine-Interface applications. The method proposed was studied as a function of the number of available neurons. The simulated environment consisted of a vision camera, a pattern recognition algorithm to extract the exact location of the target, a simulated population of neurons, and a neural decoding method to estimate the target location (see Fig. 1).

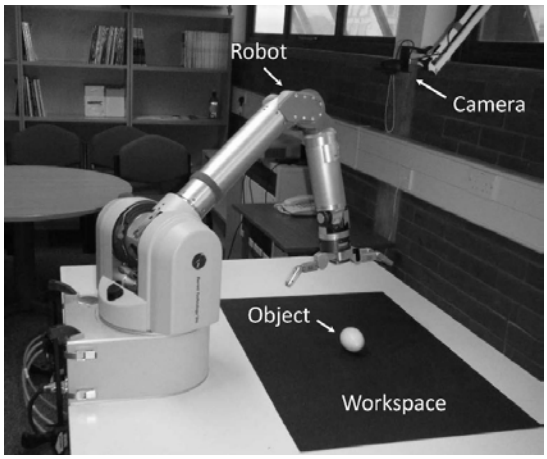


Fig. 1 System setup and 7DOF robotic arm

A. Page Layout

This section describes the system setup. Two different platforms were used: one for the simulations and another one for the experiments.

1) Simulations:

The simulations were run with a 2.67 GHz PC installed Windows XP Operative System. The software used to carry out all the simulations was Matlab 7.3.0.

2) Experiments:

The real time experiments were performed with a 2.4 GHz PC installed Linux 2.6.24.2 Operative System. This PC was used to control a 7DOF anthropomorphic arm, a vision pattern recognition system and the neural simulations of Posterior Parietal Cortex. The real time software was completely developed in C/C++ programming language.

B. 7DOF Anthropomorphic Robotic Arm

The robotic arm used in the experiments was a 7DOF anthropomorphic WAM arm from Barrett Technology Inc., which is able to reproduce natural human movements. This arm was selected because of its lightweight characteristic, which reduces the whole system's power consumption and makes the human-machine interaction easy and safe. A review of the characteristics of lightweight robots can be found in [48], and some applications and control strategies in [49].

The end-effector was a Barret Hand BH8-Series. The grasping torque was controlled through strain gauges sensors located inside the mechanical fingers. The robotic arm and the hand were controlled with an external PC, with an Intel Core 2 Duo E6600 Conroe, 2.4GHz, and runs Real Time Linux Operative System. A picture of the robotic system is depicted in Fig. 1.

C. Image Processing

The vision system was a monocular Logitech camera, model QuickCam 3000, with an image resolution of 480x640 px. The captured image was processed to obtain the exact Cartesian location of the object [50]. The image processing stage consisted of the following steps:

- Conversion of the RGB image to grayscale: This process allowed us to obtain an image in grayscale with 256 intensity levels.
- Bottom-Hat transform: This transform was applied because the uniformity of illumination can be deteriorated due to disturbances in the light sources, and also the boundaries of the captured image are usually darker than the center of the image. The bottom-hat transform provides brightness uniformity in the complete image.
- Borders detection: The Sobel borders detector was used to extract the borders of the object.
- Dynamic thresholding and image inversion: It was used to make the image binary. Then, the image was inverted to have black background.
- Noise filtering: To eliminate small noise patterns the opening and closing transforms were used.

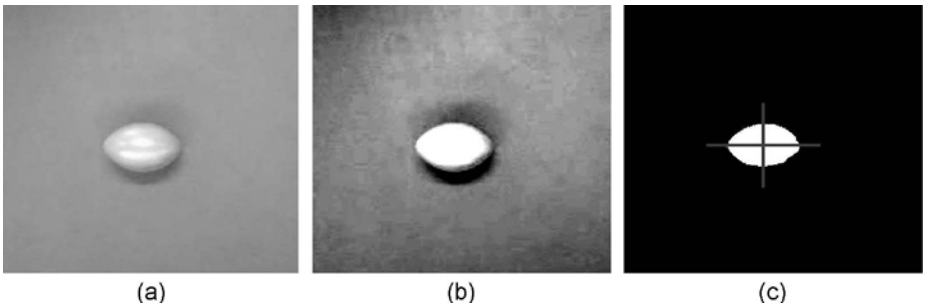


Fig. 2 Image processing from the capture with the camera to the detection of the exact object location. (a) Original image. (b) Intermediate processed image. (c) Final binary image and detection of the object location

- Holes filling: Small holes that appeared in the detected object background were filled to obtain a homogeneous surface by using the opening and closing morphologic operations.
- The borders of the detected object were smoothed by applying an erosion morphologic operation to the detected object.

For more details about the image processing operations, see [50]. Once the image was clear and the object detected, its centroid was calculated to obtain the exact location. Fig. 2 depicts a scheme of the image processing and location detection algorithm.

D. Neuron's Receptive Fields Computation and Distribution

To construct the neural population simulation environment, it was considered that the firing rate of the neurons in Posterior Parietal Cortex followed a random Poisson-like process [51]. Thus, the firing rate of a neuron could be represented by the following expression:

$$f\left(k,\lambda+\lambda_0\right)=\frac{e^{-(\lambda+\lambda_0)}(\lambda+\lambda_0)^k}{k!}, \tag{1}$$

where k is the period of the Poisson density function and λ_0 the baseline firing rate. λ stands for the average firing frequency in response to stimulus, which can be adjusted by a gaussian distribution [52-54]. Given that the goal was to decode the 2-dimensional location into a delimited workspace, the receptive field of each simulated neuron (i.e. the place in space that activates the neuron) was modelled as a 2-dimensional gaussian function:

$$\lambda = Ae^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2}+\frac{(y-y_0)^2}{2\sigma_y^2}\right)}, \tag{2}$$

where A represents the height of the gaussian, σ its width (if a gaussian shape with circular base is considered, then $\sigma_x = \sigma_y = \sigma$), and x_0 and y_0 the coordinates of its center.

An example of a neuron receptive field with gaussian shape and circular base and parameters σ = 12:5 cm, A = 56 Hz, $\lambda 0$ = 6 Hz, centered in the position (28, 33) cm is depicted in Fig. 3(a).

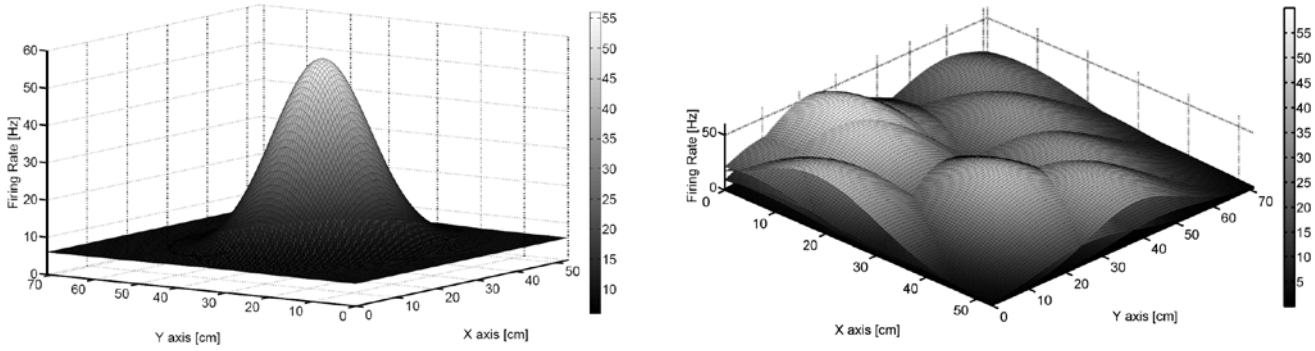
1) Setup for the Receptive Fields Simulations:

The neurons' receptive fields were randomly distributed along the complete workspace region. Based on a gaussian distribution, the receptive fields were considered to be circular shaped, then $\sigma x = \sigma y = \sigma$ = 12:5 cm, and the peaks of the gaussian shapes were randomly selected to have firing rates between 20 and 50 Hz with a baseline firing between 5 and 10 Hz. A distribution of 10 neurons with different receptive fields' shapes is depicted in Fig. 3(b).

E. Neural Decoder Characteristics

The neural decoder was designed to predict target locations based on the firing of simulated PPC neurons with the following characteristics:

- The decoder was based on the estimation of the single neurons' receptive fields.
- It was very fast and therefore suitable for on-line and real time applications.



(a) (b)

Fig. 3 Simulated receptive fields. (a) Simulated receptive field of a neuron, centered in x = 39 *cm*, y = 46:87 *cm*. (b) Distribution of the receptive fields of 10 simulated neurons covering the workspace

III. DECODING METHODOLOGY AND ALGORITHMS

This methodology was designed to obtain as much information from the simulated recorded neurons as possible with a very low computational cost, in order to make possible its implementation in real time brain-machine interfaces.

The method includes a training or calibration task to properly estimate the locations that the neurons' receptive fields encode. Thus the calibration task consists of two tests: the first one consists of the estimation of the baseline firing rate of the neurons in order to eliminate its noise effect, and the second one consists of the specific estimation of the locations of the receptive fields.

The estimation of the baseline firing rate consists of measuring neural activity from PPC with a hypothetical subject relaxed and without intending to reach any location in 30 seconds. This is enough time to have a representative measurement of the neurons' baseline activity. After measuring the baseline, it is filtered.

After that, the hypothetical subject is invited to reach different strategically distributed locations in a plane in front of him (perpendicular to the subject). The task consists of reaching every exact location ten times to acquire a representative sample of recordings for each location. Note that because of the gaussian nature of the receptive fields, if a target location is in the center of a receptive field or near it, the neuron will fire at a high firing rate (peak of the gaussian), and that if the target location is outside the receptive field, the neuron will fire at baseline.

With the data acquired, a receptive field estimation method is applied to estimate the location that the receptive fields of the recorded neurons encode. This provides a map of the receptive fields encoding the workspace. Then, implementing the decoding algorithm proposed, any location inside the workspace can be accurately estimated. Such estimation depends on the number of recorded neurons: the higher the number of neurons the better the estimation.

The whole process is next explained.

A. Baseline Calibration Task

The first step of the calibration task consisted of calculating the baseline of all the recorded neurons.

1) Baseline Estimation:

The baseline firing rate is the rate at which the neurons are firing when they are not excited by the location that they encode, i.e. a neuron is firing at a low rate when it is not excited, and at a high rate when it is excited. In this case, PPC neurons encode a movement intention to a specific location, if the subject intends to reach the location that a neuron encodes; such a neuron will fire at a high firing rate. On the contrary, if that location is not encoded by that neuron, the neuron fires at a lower rate, we say that this neuron is silent.

This baseline is a background noise that affects the estimation of the receptive fields, and therefore it has to be filtered. For that purpose, we firstly measure the baseline firing rate and define a strategy to eliminate it.

In this experiment, neurons firing at baseline firing rates during 30 s were simulated. From the generated data during that period of time, the mean and the standard deviation of the firing rates were estimated. These measures were used to calculate a threshold of activation, Te, which was used to eliminate the baseline activity. The threshold Te was set as the mean value plus the standard deviation of the measured baseline, i.e. the firing rate recorded when the stimulus was outside the neuron's receptive field (see Equation (2)):

$$T_e = M_e(f(k,\lambda_0)) + std(f(k,\lambda_0)). \quad (3)$$

2) Baseline Compensation:

After obtaining the threshold, it was considered that if the measured firing rate of a neuron had a higher value than the threshold, such a neuron was firing to the stimulus, otherwise it was silent:

$$f(k,\lambda) = \begin{cases} f(k,\lambda+\lambda_0) - T_e, \ for\, f(k,\lambda+\lambda_0) - T_e > T_e; \\ 0, \quad otherwise \end{cases} \tag{4}$$

B. Receptive Fields Estimation Task

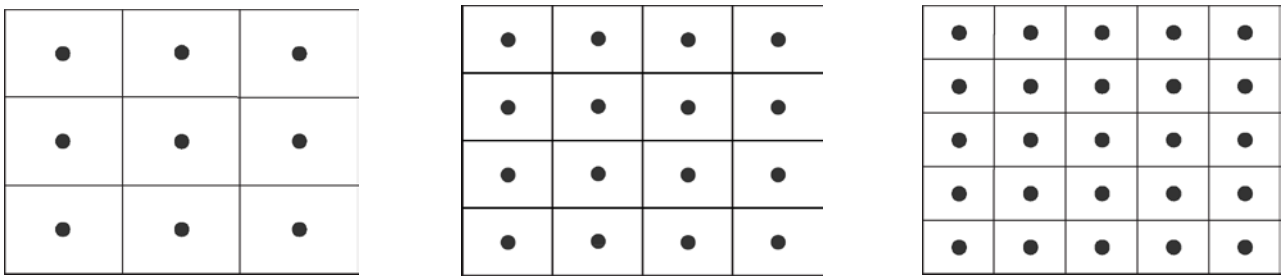
The estimation of the receptive fields was carried out by processing the signals recorded from the simulated population of neurons. In order to provide an efficient estimation of them, 3 possible workspaces were defined to study the performance of the methodology proposed with 9, 16 and 25 calibration points (Fig. 4).

A subject reaching a target at different locations was simulated. The reaching targets were at the centers of each grid, represented with a circle. The location of each reaching target *j* is denoted as (x_j, y_j).

The task consisted of simulating $n_t = 10$ reaches to the target locations (n_t stands for the number of trials for each location). The simulated data was recorded from the neurons to proceed with the estimation of the receptive fields.

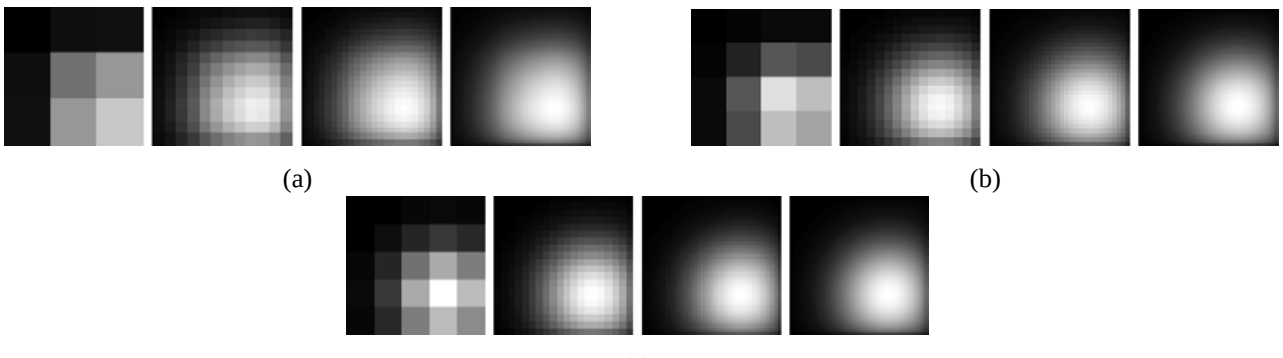
1) Receptive Fields Estimation Method:

The estimation of the receptive fields was carried out by estimating their centers (the location of the gaussian peaks). Ten trials were performed for each calibration location in which the firing rates were obtained. With the obtained data, the mean firing rate for each neuron at each location *j* was calculated in order to have a mean firing rate value f_{tm}^{j}. To obtain the peak of the gaussian, the matrix obtained with the calibration points was sampled and after that it was convolved with a gaussian filter to obtain a smoothed solution. Examples of different resamplings are shown in Figs. 5(a) to 5(c) for the three calibration grids previously proposed. The smoothing was carried out with a gaussian filter for the horizontal resampling of 0.75 in width and for the vertical resampling of 1, due to the fact that the camera used in the experiments had a resolution of 480*x*640, which was a ratio of 3x4. Thus, the difference in the width of the resampling from a square samples matrix was compensated.



(a) (b) (c)

Fig. 4 Target distribution to estimate receptive fields of the neurons. (a) 9 target points. (b) 16 target points. (c) 25 target points



(a) (b)

(c)

Fig. 5 Examples of resampling and smoothing of the sample matrices. (a) From 9 calibration points. (b) From 16 calibration points. (c) From 25 calibration points. From left to right the following results are presented respectively: no magnification, x16 magnification, x64 magnification, x4096 magnification

Then, with the mean firing rate of each neuron for each location f_{tm}^{j}, the center of the receptive field of the neuron $c_e(i)$ =

$(c_{ex}(i), c_{ey}(i))$ was estimated by calculating the center of mass of the f_{tm}^{j}. By denoting $c(i) = (c_x, c_y)$ as the exact center of the i's neuron receptive field, the error in the estimation $\varepsilon_n(i)$ could be obtained:

$$\varepsilon_e(i) = \sqrt{(c_x - c_{ex})^2 + (c_y - c_{ey})^2}\,. \tag{5}$$

This provides a map of the receptive fields covering the workspace. Once such a map is obtained, it is easy to understand that the recorded neurons will fire with different intensities when a subject is trying to reach different locations inside the calibrated workspace. The following methodology provides an estimation of any location inside the workspace when a hypothetical subject intends to reach such a location.

C. Neural Decoder

It has been shown that neurons in PPC encode pending reach intentions at determined target positions [55]. The firing of a neuron in PPC depends on the location of such a movement intention [56]. Therefore, if the neuron keeps firing at baseline levels it means that the location in which the subject had the intention to move is not in its *preferred location* [57]. Based on these assumptions, we could implement an online decoder in which the location of the movement intention could be decoded by computing the contribution of the whole population of recorded neurons.

Taking into account that all the neurons contribute to the decoding of the target location, and that their preferred locations were estimated by means of a calibration task as previously explained, the decoding method was designed as follows:

- First, the firing rates were normalized by subtracting the estimated threshold $T_e(i)$, as previously explained.
- Second, the weighted contributions of the neurons' firing rate with respect the maximum $F_{ce}(i)$ were obtained.

The resultant value obtained after these two operations was a normalized weight with respect to the maximum recorded firing rate obtained in the calibration task, $F_{ce}(i)$, which was the contribution of each neuron to the decoding of the target location.

Different weighting functions to measure the contribution of the firing neurons to encode the target location were studied. These were linear $\omega_i = m\, f_t(k, \lambda, i)$, with a slope $m = 1/F_{ce}(i)$; or nonlinear: parabolic and hyperbolic functions among others $\omega_i = m^p f_t(k,\lambda\ i)^p$, where p is the power of the nonlinear term; or a trigonometric function $\omega_i = 1\text{-}\cos((\pi\, f_t(k, \lambda, i))/(2F_{ce}(i)))$. A linear weighting function would give the same weight to firing rates near the maximum and to those far from it. A nonlinear function as the ones described above would give more importance to firing rates near the maximum, and less importance to those that are far from it, i.e. near the baseline.

To use an intermediate solution between linear and hyperbolic functions, a trigonometric function was chosen:

$$if \begin{cases} f_t(k,\lambda,i) \le F_{ce}(i), & \omega_i = 1 - \cos\left(\dfrac{\pi f_t(k,\lambda,i)}{2F_{ce}(i)}\right); \\ f_t(k,\lambda,i) > F_{ce}(i), & \omega_i = \dfrac{F_{ce}(i)}{F_{ce}(i)} = 1. \end{cases} \tag{6}$$

By calculating the center of mass of the estimated receptive field centers' distribution, using the weights ωi, an accurate estimation of the target location was obtained:

$$L_{ex} = \frac{\sum_{i=1}^{n} \omega_i \cdot c_{ex}^{i}}{\sum_{i=1}^{n} \omega_i}, L_{ey} = \frac{\sum_{i=1}^{n} \omega_i \cdot c_{ey}^{i}}{\sum_{i=1}^{n} \omega_i}, \tag{7}$$

where n stands for the number of simulated recorded neurons and L_{ex} and L_{ey} are the x and y coordinates of the estimated location respectively.

Denoting by Lx and Ly, the exact x and y locations of the movement intention, the error in the estimation is given by the following expression:

$$\varepsilon_e(i) = \sqrt{(L_x - L_{ex})^2 + (L_y - L_{ey})^2}\,. \tag{8}$$

IV. RESULTS

This section is devoted to show the main results obtained with the proposed methodology applied to a 7DOF anthropomorphic robotic arm.

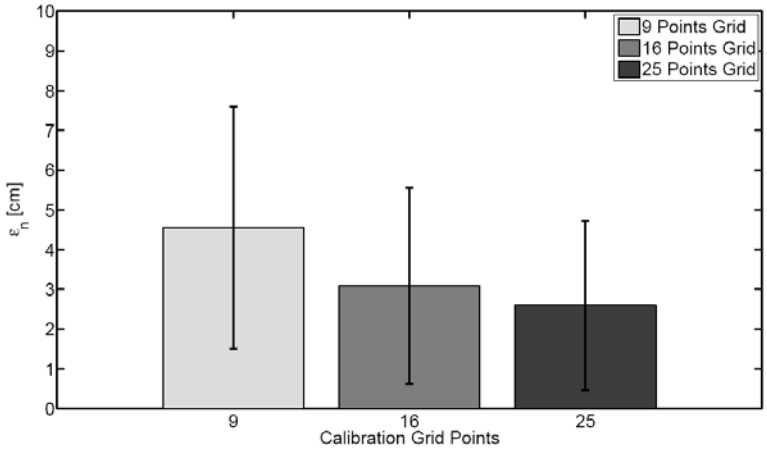


Fig. 6 Estimation errors of the 200 receptive fields preferred location. Vertical lines denote the standard deviation

A. Receptive Fields Estimation Results

Two hundred receptive fields were randomly distributed in the delimited workspace of 71x53.25 cm^2 to simulate the behavior of single neurons in Posterior Parietal Cortex. The receptive fields of the whole population of neurons were estimated by calibrating the neural decoder with 9, 16 and 25 points grids, as previously explained.

The estimation errors (in the unit of cm) of the receptive field centers (preferred locations) calculated with Equation (7) are depicted in Fig. 6. As expected, the estimation with 9 calibration points was less accurate than that with 16 points, which was less accurate than the one with 25 points. However, the training time increased with the number of calibration points. Table 1 depicts an estimation of the training time for each calibration grid, considering that every trial would require between 6-7 *s*. As previously explained, in every calibration point 10 reaches were effectuated.

TABLE 1 TRAINING TIMES

Grid	9 points	16 points	25 points
Time (min)	9-10	15-20	25-30

Thus, depending on the available time to do the training, a different number of calibration points can be used. Nine calibration points was considered to be reasonable to get accurate estimations of the receptive fields at a reasonable calibration time. Notice that the mean estimation errors with all grids (4.55 *cm*, 3.09 *cm* and 2.59 *cm*) were much smaller than the standard deviation σ = 12.5 *cm* selected to generate the receptive fields, and much smaller than the dimensions of the workspace 71 *cm x* 53.25 *cm*.

B. Decoding Results in Simulation

Once the receptive fields of the population of neurons were estimated, the target location was estimated with the decoding algorithm previously explained, and compared with the three calibration grids. To simulate different recording conditions, the decoding was carried out with different numbers of neurons.

To carry out a realistic simulation, the target location was defined as random locations inside the workspace. The number of neurons was varied from 10 to 200 in steps of 10: [10, 20, ..., 200] (neurons with different response properties were randomly selected), and in each case 100 reach trials were carried out at random positions in the calibrated workspace, i.e. 100 trials for each set of simulations with [10, 20, ..., 200] neurons, thus a total of 2000 trials. The errors ε_e were fitted with a curve of the following form:

$$y(x) = a \cdot e^{bx} + c\,. \tag{9}$$

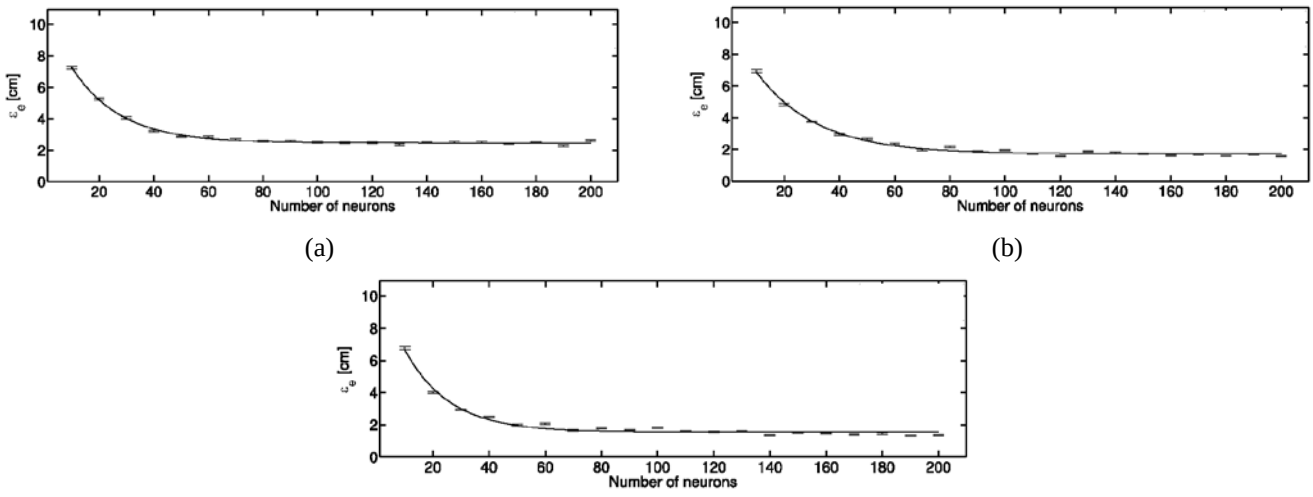
The parameters *a*, *b* and *c* that minimized the squared error are depicted in Table 2 and the fitted error curves are depicted in Fig. 7 for each of the calibration grids. These graphs show that the obtained curves accurately fitted the distribution of the errors.

Notice that there was a relatively large decrease of the error when going from 10 to about 60 neurons, reaching an

asymptotic value when considering a larger number of neurons. In all cases, the time spent by the decoder to obtain an estimation of object location was very small: 4.6 *ms* s.d. 0.5 *ms*. These results demonstrated the suitability of the decoder to be used in real time tasks.

TABLE 2 PARAMETERS OF THE FITTED ERROR CURVES

Grid	a	b	c
9	8.43	$-5.65\cdot10^{-2}$	2.47
16	8.12	$-4.56\cdot10^{-2}$	1.72
25	9.60	$-6.30\cdot10^{-2}$	1.55



(a) (b)

(c)

Fig. 7 Decoding errors of the target location as a function of the number of neurons. a) Calibration with a 9 points grid. b) Calibration with a 16 points grid. c) Calibration with a 25 points grid. Bars denote the standard mean error

C. Implementation with the WAM Robotic Arm

The decoding algorithm previously described was connected in real time with the anthropomorphic manipulator described in Section B, and with the vision pattern recognition system described in Section C. The whole system operated in the following steps:

- The object image was captured by the camera.
- The object recognition software described in Section C recognized the object location.
- The activity of the simulated neurons was determined based on the actual object position.
- The neural activity produced by the simulated neurons was decoded by the neural decoder, which provided an estimation of the location of the object.
- The estimated location was given to the robotic arm to execute the reaching and grasping of the object.
- The reaching and grasping of the object was carried out.

Figs. 8(a) to 8(d) depict four experiments in which a target was placed at random locations of the workspace. The object to be grasped in these experiments was a toy rugby ball that is 6 *cm* in diameter and 10 *cm* in length. The calibration was carried out with 9 calibration points because this was the most unfavorable case, which however required a relatively short training time. In these figures, the reaching and grasping was executed with 10, 20, 60 and 100 simulated neurons, respectively. Note that for a small number of neurons (10-40), the ball could not be reached in that location. On the other hand, when the population of simulated neurons was over 60, the reaching and grasping were done satisfactorily, with about 60 % success rate.

(a) (b) (c) (d)

Fig. 8 End locations of the robotic arm with different numbers of neurons. (a) With a neural network with 10 neurons. (b) With a neural network with 20 neurons. (c) With a neural network with 60 neurons. (d) With a neural network with 100 neurons

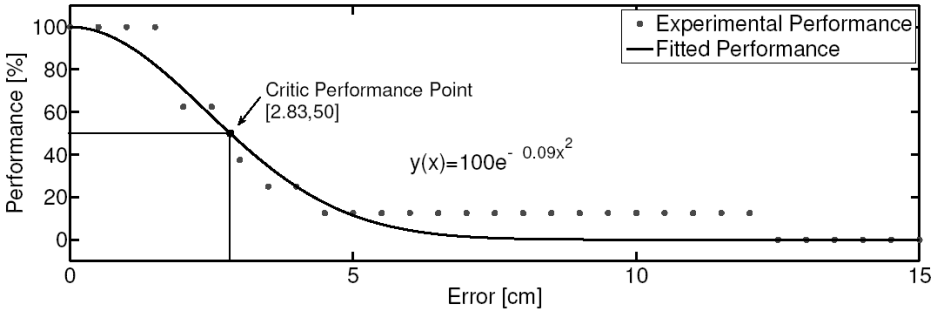


Fig. 9 Robotic arm grasping performance test

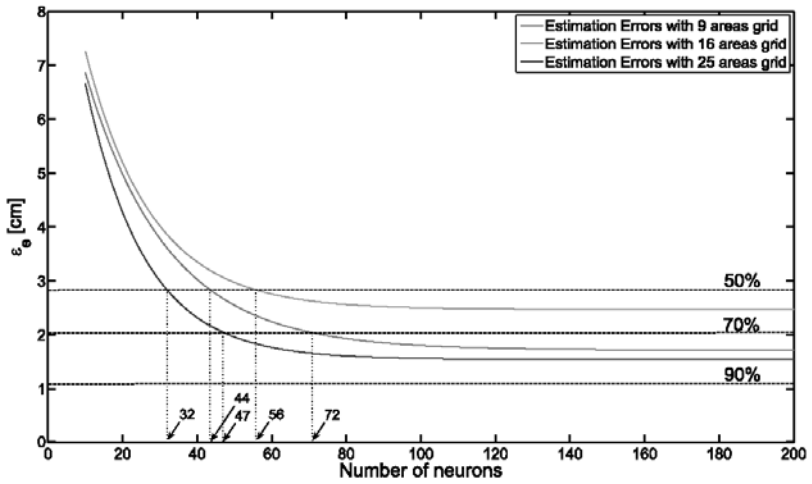


Fig. 10 Estimation errors of the decoder as a function of the number of neurons for the 3 calibration grids

To study the rate of success of the robotic hand in grasping the object, a performance test was carried out. 256 trials were done with variable errors obtained by performing reaches at different locations with variable number of neurons. If the object was grasped, it was considered a success, and otherwise it was a failure. Fig. 9 depicts the results. As expected, the higher the error, the lower the performance. However, this test gave us a quantitative realistic measure of the hand performance.

The errors were fitted with a Gaussian curve with the parameters adjusted to minimize the square error, providing an estimation of the performance in different grasping tasks. The results are given in percentage. The fitted performance curve was the following:

$$Performance\% = 100 \cdot e^{-8.68 \cdot 10^{-2} x^2}, \tag{10}$$

in which the variable x represents the error in centimeters. The performance critic point, at which the number of successes is the same with the number of fails, was obtained with an error of 2.83 *cm*. Fig. 10 shows a graph in which the calibration grid, the number of recorded neurons, the decoding errors and the robot arm performance are related. The three straight lines show the 50%, 70% and 90% performances. The 50% performance line cuts the curve obtained with 9 calibration points at 56 neurons, the curve of 16 calibration points at 44 neurons and the curve of 25 calibration points at 32 neurons. This means that those are the numbers of neurons required to grasp an object of similar dimensions to that previously described. Taking into account that the calibration task required time to be carried out (see Table 1), the higher the number of calibration points, the

more the time required. The 70% performance curve only cuts the curves of 16 and 25 calibration points, and the numbers of neurons required for grasping an object with this performance were 44 and 32 respectively. Finally, the curve of 90% performance does not cut any of the curves depicted in the graph, which means that this performance cannot be obtained with any of the grids proposed. The decoding errors could be improved by increasing the number of calibration points, for example to 36 or 49, which would imply training time of 35-45 minutes in the former case, and of 50-60 minutes in the latter case; and the time may be too long for realistic applications. Another way to improve the performance is by including visual feedback and a remapping of receptive fields, considering a first approximation to the object and then a final, more detailed approach. This will be subject of future research.

V. MAIN ASSUMPTIONS AND FUTURE WORK

In this section, the main assumptions that were made for the design of the neural decoding method explained here are described and some directions are given for future work.

For the design of the decoder, it was considered that the receptive fields of the neurons were Gaussian, as in previous works [52-54]. However, it is always possible to calculate a maximum firing rate and use it as input to the decoder even if the receptive field is not strictly Gaussian. It is likely though, that more errors will be observed in the presence of complex receptive fields with more than 1 local maximum. In relation with the neuron's firing, it was also assumed that it follows a Poisson-like process [51].

It was also assumed that the neurons have an object representation instead of a retinotopic representation, i.e. the neurons represent the position of the object with the effector (the robot hand) at the center of coordinates. However, even in the presence of a retinotopic representation, it is possible to use a similar strategy as the one described here. In this case, it is possible to first look at the hand and then the object, to provide the reference of the movement with respect to the initial location of the hand.

It was also assumed that there is no dependency between the neurons, so each of them only codifies one receptive field. Redundancy would increase the number of required neurons to decode them with good performance. However, this can only be determined with real experiments.

The tests were reduced to a two-dimensional space, but it is in principle possible to extend them to three-dimensional reaches since it was shown that posterior parietal cortex neurons also have three-dimensional selectivity (see [58]).

More realistic movements could also be used to reach the object. For example, it is conceivable to first perform a ballistic-type reach to the rough direction of the object and then introduce a final approximation with a finest tuning.

Finally, higher speeds could also be tested for the reaches. This would reduce the required time for the training and task execution. However, this would also increase the receptive fields estimation error and performance accuracy. The suitability of these strategies and to what extend they outperform the approach described here are subjects of future work.

VI. CONCLUSIONS

In this work, a method was proposed to decode the intention of an arm movement from the firing rates of the neurons in parietal cortex. The decoding method was based on the estimation of the receptive fields of the neurons.

The results presented demonstrated a good performance of the method by decoding accurately the location of the movement intention in the plane. The method was found to be fast: a mean of 4.6 *ms* was required to decode the target location.

The method was tested by executing reaching tasks inside a defined workspace of $0.71x0{:}53\ m^2$, and with different numbers of neurons. As expected, the performance increased with the number of neurons.

This study demonstrated that with more than 56 neurons, the reaching of an object like a tennis ball (6.7 cm in diameter) in the most unfavourable case can be carried out with an accuracy of higher than 50% and a calibration grid of 9 points, which required a training time lower than 10 minutes. The experiments presented in this work also demonstrated that the neural decoder studied here provides suitable high level signals to control a robotic arm.

ACKNOWLEDGMENTS

The authors would like to thank the support given by the Consejería de Ciencia y Tecnología of the Junta de Comunidades de Castilla-La Mancha (Spain), by the European Social Fund, Engineering and Physical Sciences Research Council (EPSRC), Biotechnology and Biological Sciences Research Council (BBSRC) and Medical Research Council (MRC). The authors would also like to thank Carlos Pedreira and Juan Martínez for their helpful comments and to Zaira Pineda and Emanuele Perugia for their help in the vision system setup.

REFERENCES

[1] R. A. Andersen, S. Musallam, and B. Pesaran, "Cognitive neural prosthetics," TRENDS in Cognitive Sciences, vol. 8, iss. 11, pp. 486–

493, 2004.

[2] M. A. Nicolelis, “Action for thoughts,” Nature, vol. 409, pp. 403–407, 2001.

[3] J. Becedas, “Brain-machine interfaces: Basis and advances,” IEEE Transactions on Systems, Man, and Cybernetics, Part C, vol. 42, iss. 6, pp. 825–836, 2012.

[4] J. P. Donoghue, “Connecting cortex to machines: recent advances in brain interfaces,” Nature Neuroscience, vol. 5. pp. 1085–1088, 2002.

[5] M. A. Lebedev and M. A. Nicolelis, “Brain-machine interfaces: past, present and future,” Trends in Neuroscience, vol. 29, iss. 9, pp. 536–546, 2006.

[6] G. E. Loeb, “Cochlear prosthetics,” Annual Review on Neuroscience, vol. 13, pp. 357–371, 1990.

[7] R. M. Rothschild, “Neuroengineering tools/applications for bidirectional interfaces, brain-computer interfaces, and neuroprosthetic implants - A review of recent progress,” Frontiers in Neuroengineering, vol. 3, iss. 112, pp. 1–5, 2010.

[8] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, “Instant neural control of a movement signal,” Nature, vol. 416, pp. 141–142, 2002.

[9] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, “Direct cortical control of 3d neuroprosthetic devices,” Science, vol. 296, pp. 1829–1832, 2002.

[10] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, Da. Chen, R. D. Penn, and J. P. Donoghue, “Neuronal ensemble control of prosthetic devices by a human with tetraplegia,” Nature, vol. 442, pp. 164–171, 2006.

[11] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. Nicolelis, “Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex,” Nature Neuroscience, vol. 2, pp. 664–670, 1999.

[12] A. B. Schwartz, X. T. Cui, D. J. Weber, and D. W. Moran, “Brain-controlled interfaces: movement restoration with neural prosthetics,” Neuron, vol. 52, pp. 205–220, 2006.

[13] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and M. A. Nicolelis, “Real-time prediction of hand trajectory by ensembles of cortical neurons in primates,” Nature, vol. 408, pp. 361–365, 2000.

[14] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O’Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis, “Learning to control a brain-machine interface for reaching and grasping by primates,” PLoS Biology, vol. 1, iss. 2, pp. 193–208, 2003.

[15] I. Obeid and P. D. Wolf, “Evaluation of spike-detection algorithms for a brain-machine interface application,” IEEE Transactions on Biomedical Engineering, vol. 51, iss. 6, pp. 905–911, 2004.

[16] J. d. R. Millan, F. Renkens, J. Mourino, and W. Gerstner, “Non-invasive brain-actuated control of a mobile robot by human eeg,” IEEE Transactions on Biomedical Engineering, vol. 51, iss. 6, pp. 1026–1033, 2004.

[17] H. K. Kim, S. J. Biggs, D. W. Schloerb, J. M. Carmena, M. A. Lebedev, M. A. Nicolelis, and M. A. Srinivasan, “Continuous shared control for stabilizing reaching and grasping with brain-machine interfaces.,” IEEE Transactions on Biomedical Engineering, vol. 53, iss. 6, pp. 1164–1173, 2006.

[18] F. Galan, M. Nutting, E. Lew, P. W. Ferrez, G. Vanacker, J. Philips, and J. d. R. Millán, “A brain-actuated wheelchair: Asynchronous and non-invasive brain-computer interfaces for continuous control of robots,” Clinical Neurophysiology, vol. 119, pp. 2159-2169, 2008.

[19] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, “Cortical control of a prosthetic arm for self-feeding,” Nature, vol. 453, pp. 1098–1101, 2008.

[20] N. P. Birbaumer, N. Ghanayim, T. Hinterberger, I. H. Iversen, B. Kotchoubey, A. Kbler, J. Perelmouter, E. Taub, and H. Flor, “A spelling device for the paralysed,” Nature, vol. 398, pp. 297–298, 1999.

[21] J. R. Wolpaw, N. P. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, “Brain-computer interfaces for communication and control,” Clinical Neurophysiology, vol. 113, pp. 767–791, 2002.

[22] J. R. Wolpaw, “Brain-computer interfaces (bcis) for communication and control: a mini-review,” Supplements to Clinical Neurophysiology, vol. 57, pp. 607–613, 2004.

[23] B. Obermaier, G. R. Mller, and G. Pfurtscheller, “Virtual keyboard controlled by spontaneous eeg activity,” IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 11, pp. 422–426, 2003.

[24] T. Hinterberger, R. Veit, B. Wilhelm, N. Weiskopf, J. -J. Vatine, and N. Birbaumer, “Neuronal mechanisms underlying control of a brain-computer interface,” European Journal of Neuroscience, vol. 21, pp. 3169–3181, 2005.

[25] N. Birbaumer, “Brain-computer-interface research: coming of age,” Clinical Neurophysiology, vol. 117, pp. 479–483, 2006.

[26] M. A. Nicolelis and M. A. Lebedev, “Principles of neural ensemble physiology underlying the operation of brain-machine interfaces,” Nature Reviews Neuroscience, vol. 10, pp. 530–540, 2009.

[27] M. A. Lebedev and M. A. Nicolelis, “Brain-machine interfaces: past, present and future,” Trends in Neuroscience, vol. 29, pp. 536–546, 2006.

[28] J. d. R. Millan and J. M. Carmena, “Invasive or noninvasive: Understanting brain-machine interface technology,” IEEE Engineering in Medicine and Biology Magazine, vol. 29, iss. 1, pp. 16–22, 2010.

[29] F. E. M. -Ivaldi and L. E. Miller, “Brain machine interfaces: Computational shorter demands and clinical needs meet basic neuroscience,” Trends in Neuroscience, vol. 26, iss. 6, pp. 329–334, 2003.

[30] E. M. Schmidt, “Single neuron recording from motor cortex as a possible source of signals for control of external devices,” Annals of Biomedical Engineering, vol. 8, pp. 329–349, 1980.

[31] S. Shoham, L. M. Paninski, M. R. Fellows, N. G. Hatsopoulos, J. P. Donoghue, and R. A. Normann, “Statistical encoding model for a primary motor cortical brain-machine interface,” IEEE Transactions on Biomedical Engineering, vol. 52, iss. 7, pp. 1312–1322, 2005.

[32] J. Zhuang, W. Truccolo, C. V. –Irwin, and J. P. Donoghue, "Decoding 3-d reach and grasp kinematics from high-frequency local field potentials in primate primary motor cortex," IEEE Transactions on Biomedical Engineering, vol. 57, iss. 7, pp. 1774–1784, 2010.

[33] B. Pesaran, J. S. Pezaris, M. Sahani, P. P. Mitra, and R. A. Andersen, "Temporal structure in neuronal activity during working memory in macaque parietal cortex," Nature Neuroscience, vol. 5, iss. 8, pp. 805–811, 2002.

[34] S. Musallam, B. D. Corneil, B. Greger, H. Scherberger, and R. A. Andersen, "Cognitive control signals for neural prosthetics," Science, vol. 305, pp. 258–262, 2004.

[35] R. C. Miall and D. M.Wolpert, "Forward models for physiological motor control," Neural Networks, vol. 9, iss. 8, pp. 1265–1279, 1996.

[36] C. Kemere, G. Santhanam, B. M. Yu, A. Afshar, S. I. Ryu, T. H. Meng, and K. V. Shenoy, "Model-based decoding of reaching movements for prosthetic systems," In Proceedings of the 26th Annual International Conference of the IEEE EMBS, pp. 4524–4528, San Francisco, Sept. 2004.

[37] A. B. Schwartz, R. E. Kettner, and A. P. Georgopoulos, "Primate motor cortex and free arm movements to visual targets in three-dimensional space. i. relations between single cell discharge and direction of movement," The Journal of Neuroscience, vol. 8, iss. 8, pp. 2913–2927, 1988.

[38] A. P. Georgopoulos, R. E. Kettner, and A. B. Schwartz, "Primate motor cortex and free arm movements to visual targets in three-dimensional space. ii. coding of the direction of movement by a neural population," The Journal of Neuroscience, vol. 8, iss. 8, pp. 2928–2937, 1988.

[39] R. E. Kettner, A. B. Schwartz, and A. P. Georgopoulos, "Primate motor cortex and free arm movements to visual targets in three-dimensional space. iii. position gradients and population coding of movement direction from various movement origins," The Journal of Neuroscience, vol. 8, iss. 8, pp. 2938–2947, 1988.

[40] R. Heliot, K. Ganguly, J. Jimenez, and J. M. Carmena, "Learning in closed-loop brain machine interfaces: Modeling and experimental validation," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 40, iss. 5, pp. 1387–1397, 2010.

[41] R. A. Andersen, E. J. Hwang, and G. H. Mulliken, "Cognitive neural prosthetics," Annual Review of Physiology, vol. 61, pp. 169–190, 2010.

[42] R. A. Andersen, S. Musallam, and B. Pesaran, "Selecting the signals for a brain-machine interface," Current Opinion in Neurobiology, vol. 14, pp. 720–726, 2004.

[43] R. Q. Quiroga, L. H. Snyder, A. P. Batista, H. Cui, and R. A. Andersen, "Movement intention is better predicted than attention in the posterior parietal cortex," The Journal of Neuroscience, vol. 26, iss. 13, pp. 3615–3620, 2006.

[44] J. Becedas and R. Q. Quiroga, "Implementation of a real time decoder for real neuroprosthetic applications," In Proceedings of The IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications, pp. 1337–1345, Liverpool, United Kingdom, 2010.

[45] R. A. Andersen and C. A. Buneo, "Intentional maps in posterior parietal cortex," Annual Review on Neuroscience, vol. 25, pp. 189–220, 2002.

[46] R. Q. Quiroga and S. Panzeri, "Extracting information from neuronal populations: information theory and decoding approaches," Nature Reviews Neuroscience, vol. 10, pp. 173–185, 2009.

[47] J. D. Connolly, R. A. Andersen, and M. A. Goodale, "FMRi evidence for a parietal reach region in the human brain," Experimental Brain Research, vol. 153, pp. 140–145, 2003.

[48] E. Pereira, J. Becedas, I. Payo, F. Ramos, and V. Feliu, "Control of flexible manipulators. Theory and practice," Robot Manipulators Trends and Development, INTECH, pp. 267–296, 2010.

[49] J. Becedas, J. R. Trapero, V. Feliu, and H. S. –Ramirez, "Adaptive controller for single-link flexible manipulators based on algebraic identification and generalized proportional integral control," IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, vol. 39, iss. 3, pp. 533–545, 2009.

[50] R. C. Gonzalez and R. E. Wood, "Digital Image Processing," Prentice Hall, 2002.

[51] F. Rieke, D. Warland, R. d. R. v. Stevenink, and W. Bialek, *Spikes: exploring the neural code*, MIT press paperback edition, 1999.

[52] A. Pouget and T.e Sejnowski, "Spatial transformations in the parietal cortex using basis functions," Journal of Cognitive Neuroscience, vol. 9, iss. 2, pp. 222–237, 1997.

[53] S. Ben Hamed, J. –R. Duhamel, F. Bremmer, and W. Graf, "Visual receptive field modulation in the lateral intraparietal area during attentive fixation and free gaze," Cerebral Cortex, vol. 12, iss. 3, pp. 234–245, 2002.

[54] R. A. Andersen, G. K. Essick, and R. M. Siegel, "The encoding of spatial location by posterior parietal neurons," Science, vol. 230, iss. 4724, pp. 456–458, 1985.

[55] L. H. Snyder, A. P. Batista, and R. A. Andersen, "Coding of intention in the posterior parietal cortex," Nature, vol. 386, pp. 167–170, 1997.

[56] C. A. Buneo, M. R. Jarvis, A. P. Batista, and R. A. Andersen, "Direct visuomotor transformations for reaching," Nature, vol. 416, pp. 632–636, 2002.

[57] L. J. Toth and J. A. Assad, "Dynamic coding of behaviourally relevant stimuli in parietal cortex," Nature, vol. 415, pp. 165–168, 2002.

[58] M. Hauschild, I. H. Mulliken, Fineman, G. E. Loeb, and R. A. Andersen, "Cognitive signals for brain-machine interfaces in posterior parietal cortex include continuous 3d trajectory commands," Proceedings of the natural academy of sciences, vol. 109, iss. 42, pp. 17075–17080, 2012.