Probabilistic Scan Mode of a Robot Manipulator Workspace using EEG signals. Part II

Fernando A Auat Cheeín¹, Fernando di Sciascio¹, Teodiano Freire Bastos² and Ricardo Carelli¹

¹Institute of Automatic, National University of San Juan. San Martin, 1109 - Oeste 5400 San Juan, Argentina.
²Electrical Engineering Department, Federal University of Espirito Santo. Fernando Ferrari, 514 29075-910 Vitoria-ES, Brazil.

E-mail: fauat@inaut.unsj.edu.ar

Abstract. In this paper a probabilistic-based workspace scan mode of a robot manipulator is presented. The workspace is divided into cells. Each cell has its own probability value associated with it. Once the robot reaches a cell, its probability value is updated. The updating process is governed by a recursive Bayes algorithm. A performance comparison between a sequential scan mode and the one proposed here is made. Mathematical derivations and experimental results are also shown in this paper.

1. Introduction
Bayesian algorithms are becoming a new trend in the robotics field [1]. Applications such as robot localization ([1]), SLAM (Simultaneous Localization and Mapping) ([2]), environment representation and intelligent decisions are examples where Bayes theory is applied [2].

In Human-Machine Interfaces and Brain-Computer Interfaces (BCI), Bayes theory is used to recognize patterns in signals, for classification and decision ([3] and [4]).

In this work, a bayesian algorithm is proposed to govern a probabilistic scan mode of a robot manipulator’s workspace. The main objective is to improve the time needed to reach the most frequently accessed positions at that workspace.

Recent works on BCIs and robot manipulators can be found in [5] and [6]. In those works, a sequential scan mode type is used. This paper shows the advantages of using a probabilistic scan mode to anticipate user’s decision based on workspace’s probabilistic information.

The work is organized as follows. In companion work (Part I), the state of art among with a complete system description is introduced. In this Part II, mathematical derivations and experimental results are shown. Thus, Section II of this work presents both scan mode systems: the sequential and the probabilistic ones, among with the mathematical derivation of the bayesian algorithm used; Section III shows the results for a Montecarlo experimentation, where the probabilistic evolution of the whole workspace and of a specific cell is presented. Section IV shows the conclusions of this work.

2. System Presentation
As it was mentioned in another work (Part I), the whole system connects a BCI (Brain Computer Interface) with a robot manipulator. The objective is that, those cells that have more probability to be
accessed be reached by the BCI’s user in less time. Thus, cells with higher probability are scanned first. On the other hand, cells with lower probability are left to the end of the scan process. Figure 1 shows the graphical interface used. Each position at robot’s workspace is represented by a cell.

![The graphical interface.](image1)

**Figure 1.** The graphical interface.

2.1. Sequential Scan Mode

As a brief introduction, the sequential scan mode of the robot workspace developed in [5] is presented here.

The workspace is previously divided into three main zones as it can be seen in figure 2. The system iteratively scans from zone 1 to zone 3 until one of them is selected by the user. Once it is so, the selected zone is scanned row by row until one is selected. This situation is shown in figure 3. Once a row is selected, the system scans cell by cell iteratively inside the selected row. After a cell is selected by the user, the robot reaches the position given by that cell. Figure 1 shows this situation.

![Main zone division at robot’s workspace.](image2)

**Figure 2.** Main zone division at robot’s workspace.
2.2. Probabilistic Scan Mode

In order to generate a Probabilistic Scan Mode of the robot’s workspace, each cell of it has a probability weight. This value answers the question of how often a cell is reached. If the probability value of a cell is near one, then that cell is frequently accessed. A cell with weight near zero means that it is rare to be accessed by the user.

As was pointed out in Part I, the robot’s workspace is divided into three zones, each one of them represents a probability level and the scan is made according this level. Thus, cells with higher probability form zone 1; cells with lower probability form zone 3 and cells between form zone 2. The scan starts at zone 1. The details of zone’s partitioning can be found in Part I.

Figure 4 shows a workspace distribution for a right-handed user. As it is shown, cells to the right of the middle point are more often accessed (higher probability level) than cells to the left. Figure 4.a shows each cell’s probability value while figure 4.b shows the three primary zones that they conform. It can be seen from figure 4 that set connectivity is not preserved in this work. Thus, two or more non-connected cells may belong to a same zone. Once a primary zone is selected by the user, this set of cells is then divided into three sub-zones under the same rule used before: sub-zone 1 formed by cells with higher probability value and so on. After a sub-zone is selected, the scan is made cell by cell according to its weight. Thus, the sub-zone with the highest probability value cells is scanned first.

2.3. Recursive Bayes Rule

In another work (Part I), the Bayes rule equation implemented was introduced. That expression is transcribed below, in equation (1).
Though equation (1) is mainly used in very simple applications [2], it fits as an updating rule for the purpose of this work.

Equation (1) can be re-written in (2), where a scale factor was used.

\[
P_k (C \mid G) = \eta P_{k-1} (C \mid G) \frac{P_k (G \mid C) P_{k-1} (C \mid G)}{P_k (G \mid C) P_{k-1} (C \mid G) + P_k (G \mid \bar{C}) P_{k-1} (\bar{C} \mid G)}
\]  

(1)

According to the Total Probability Theorem [7], \( \eta \) is the scale factor, which represents the total probability of \( P(G) \). In (1), \( P_{k-1} (C \mid G) \) is the prior probability of a cell given the primary set to which it belongs at time \( k-1 \). \( P_k (G \mid C) \) is the transition probability which represents the probability that a given cell \( C \) belong to a set \( G \). Finally, \( P_k (C \mid G) \) is the posterior probability -at instant \( k \)- of the cell used given the zone to which it belongs.

In order to make sense to the use of the recursive Bayes algorithm, an initial probability value must be given to all cells at the workspace.

Figure 5 shows the evolution of a cell’s probability when it is accessed successively by the user.
The cell used in figure 5, for example, has an initial value of 0.05 but it is increased each time the cell is accessed by the user. As was expected, the maximum value a cell can reach is one. When this situation occurs, the whole workspace is scaled. This scaling does not change the scan mode because the relative probability information remains without changes, i.e., if a cell $p$ has the maximum probability over all cells, after scaling, $p$ will continue being the cell with the highest weight.

3. Experimental Results

This section is entirely dedicated to compare both scan types: sequential and probabilistic and to show the benefits of using the last one. For this purpose, a Montecarlo experiment was designed [8]. This experiment shows the performance of both methods by measuring the time needed to reach different cells at the robot’s workspace.

3.1. Montecarlo Experiment

The robot’s workspace consists of 72 cells. It also can be considered as a $4 \times 18$ matrix. According to this, a cell’s position is defined by a number of row and a number of column at that matrix. The number of a row and a column can be considered as a random variable. To generate a random position of a cell destination, the following algorithm was implemented.

i. An uniform random source generates two random variables: $x$ and $y$.

ii. The random variable $x$ is mapped into the rows of the $4 \times 18$ matrix workspace.

iii. The random variable $y$ is mapped into the columns of the $4 \times 18$ matrix workspace.

iv. When a position is generated, both scan types begin. The time needed to reach the cell is recorded.

v. After the system reaches the position proposed, a next process point generation is settled -the algorithm returns to point i-.

3.2. Mapping functions

Let $f_x$ be a mapping function such as:

$$f_x : A \rightarrow B$$

$$x \rightarrow m$$

where,

$$A = \{x : x \in [0, 1) \subset \mathbb{R}\}$$

$$B = \{m : m \in \{1, 2, 3, 4\} \subset \mathbb{N}\}$$

Figure 5. Evolution of Cell’s probability when successive accessed.
and let $f_y$ be another mapping function such as:

$$f_y : A \rightarrow C$$

$$y \rightarrow n$$

where,

$$A = \{ x : x \in [0,1] \subset \mathbb{R} \}$$

$$C = \{ n : n \in \{1,2,3,4,\ldots,18\} \subset \mathbb{N} \}$$

Equations (3) and (4) show the domain and range of the mapping functions. Finally, the mapping is made according to the following statements.

i. Let $\delta$ be the sum of all weights at robot’s workspace, that is, $\delta = \sum_{i \in B} \sum_{j \in C} P_{ij}$, where $P_{ij}$ is the probability value of a cell located at the $i$ - row and $j$ - column.

ii. Let $x \in A$ be an outcome of the uniform random source for $f_x$.

- If $0 \leq x < \frac{\sum_{i=1, j \in C} P_{ij}}{\delta}$ then $f_x(x) = i = 1$. This means that the value of $x \in A$ should be lower than the sum of all cell’s values in row one -over $\delta$ - to $f_x(x)$ be equal to one.

- If $\frac{\sum_{i=1, j \in C} P_{ij}}{\delta} \leq x < \frac{\sum_{i=2, j \in C} P_{ij}}{\delta}$ then $f_x(x) = i = 2$. This means that $x \in A$ should be greater or equal to the sum of all cell’s values in row one and lower than the sum of all cell’s values in row 2.

- The same process continues up to the last row, which expression is: if $\frac{\sum_{i=3, j \in C} P_{ij}}{\delta} \leq x < \frac{\sum_{i=4, j \in C} P_{ij}}{\delta}$ then $f_x(x) = i = 4$.

- Each time a cell is selected, the mapping functions vary. It is so because they are dependent with the probability value of the cells.

- For the mapping over the columns, the procedure is the same only that in this case, the sum is made over the set $B$ (four rows).

Concluding, the mapping presented here is dynamic because it is updated each time a cell varies its probability value. For the case implemented in this work (a right-handed user) the initial mapping functions are represented in figures 6.a and 6.b. In figure 6.b is also possible to see that column 10 has higher probability than column 1. It is also important to see that, if all cells at robot’s workspace have the same probability weight, then the mapping functions would be uniform. Thus, each row or column would have the same probability to be generated.
3.3. Montecarlo Simulation Results

The objective of Montecarlo Experiments was to test the performance of both scanning methods: probabilistic and sequential ones. The performance is measured in function of the time needed to access a given position. This position is generated by the uniform random source. After 500 trials the mean time needed to access a random position by the probabilistic scan was of 20.4 seconds instead of 19.8 seconds of the sequential scan. Both results are in the same order.

Consider now only the right side of the workspace, which is, according to figure 4, the most accessed side. The mean time of access for all points belonging to the workspace right side is of 8.4 seconds under the probabilistic scan. Under sequential scan, the mean time is of 14.8 seconds. The probabilistic scan mode is 57% faster than the sequential scan for cells over the right side of the workspace.

In all tests, no considerations were made in case the user do not access to the position proposed.

Figure 7 shows how a low probability valued cell evolves after successive callings. The cell passes through the different zones of cells according to its actual probability value. After 240 iterations -or callings-, the cell has passed through three zones and its performance has also been improved as long as its weight. In figure 7, one can see that at the beginning, 32 seconds were needed to access that cell. After 240 iterations, only 14 seconds were needed. This time is smaller than the one needed on the
sequential scan mode which is of 18 seconds. Figure 7 also shows when the cell changes zones. Thus, if its probability increases, the cell passes from, for example, primary zone 2 to primary zone 1, as it is described in Part 1 of this work. Though a cell could be the first in being scanned in the primary zone 2, if it increases its value and passes to primary zone 1, it could be the last scanned element in this zone. That is the reason of the two time increments in figure 7.

![Scan Mode Time evolution of a low Probability Cell after 240 iterations](image)

**Figure 7.** Evolution of a cell access time.

Figure 8 shows the workspace state after 500 iterations generated by the Montecarlo experiment. Figure 8.a shows the probability state of each cell at the workspace while figure 8.b shows the new three zones of the scan mode algorithm. One can see that the non-connectivity tends to disappear.

![Probability values of the workspace's cells](image)

**Figure 8.** Workspace state after 500 iterations.

4. Conclusions
The work presented here showed the implementation of a probabilistic scan mode, based on a recursive Bayes algorithm, of a robot manipulator’s workspace. A comparison between this method and a sequential scan mode showed that the probabilistic scan improves the access time of the most frequently accessed cells. Although this system could be implemented in several Human-Machine Interfaces, it was primary designed for a Brain-Computer Interface.
Experimental results show that the time needed to access a specific position at the workspace is decreased each time the position is reached. This is so because the recursive Bayes algorithm implemented updates the probability value of that position once it is reached. A decrement of the access time means that the user of the Interface needs less effort to reach the objective.

In this work, a right-handed workspace distribution case was presented. This case showed that all cells to the right of the middle point -half of the main workspace- have the higher probability and the lower time needed to be accessed. Finally, it is possible to say that the system learns the user’s workspace configuration. It pays special attention to those cells with the highest probability minimizing the time needed to access them.

References
[1] Siegwart R and Nourbakhsh I R 2004 Autonomous Mobile Robots (Massachusetts: The MIT Press)
[2] Thrun S, Burgard W and Fox D 2005 Probabilistic Robotics (Massachusetts: The MIT Press)
[3] Marqués de Sá J P 2001 Pattern Recognition (Springer)
[4] Duda R O, Hart P E and Store D G 2000 Pattern Classification (Wiley-Interscience)
[5] Ferreira A, Auat Cheein F A, Postigo J, Bastos Filho T F, Carelli R and Sarcineli Filho M 2006 Teleoperação de um Manipulador para uma Cadeira de Rodas Usando Sinais de EEG 1V Congresso Ibero-Americano Sobre Tecnologias de Apoio a Portadores de Deficiência Anais do IV Congresso Ibero-Americano Sobre Tecnologias de Apoio a Portadores de Deficiência 1 MA23-MA27
[6] Ferreira A, Freire Bastos-Filho T, Sarcinelli-Filho M, Auat Cheein F A, Postigo J F, Carelli R 2006 Teleoperation of an Industrial Manipulator Through a TCP/IP Channel Using EEG Signals International Symposium of Industrial Electronics Montreal Proceedings of ISIE2006 1 3066-3071
[7] Papoulis A 1980 Probabilidad, variables alatorias y procesos estocásticos (Barcelona: Eunibar)
[8] Fishman G S 1996 Monte Carlo Concepts, Algorithms, and Applications (New York: Springer-Verlag)