

Implementing the Imprint in Robot by Calculating Probabilities

Rodolfo Romero Herrera ^{a*}, Manuel Mejía Perales^b and Francisco Gallegos Funes^c

^a Instituto Politécnico Nacional, ESIME-ESCOM Unidad Zacatenco, Sección de Estudio de Posgrado Investigación, Ciudad de México, México.

^b Instituto Politécnico Nacional, ESCOM, Unidad Zacatenco, Ciudad de México, México.

^c Instituto Politécnico Nacional, ESIME, Sección de Estudios de Posgrado e Investigación, Unidad Zacatenco, Ciudad de México, México.

Abstract

The purpose of this work is to implement the simulation of learning imprint in a Robot. One of the main problems in learning machines is how robots acquire knowledge of the environment that surrounds them. A bird can begin its learning by following its mother. This mechanism is called imprint. This training is incorporated into a Robot so that he can follow objects. To achieve this, an algorithm based on probabilities is used. In this case, a robot followed a red ball in motion; to make it, a Kinect camera was used and the characteristics of the RGB pixels were determined in changing lighting conditions. This can mimic the intelligent behavior of a bird on a machine. The method can be used to start autonomous learning in a system; in such a way, that this can be increased his knowledge using few resources.

Keywords: Robot control, mobile robot, computer vision, visual guidance, automatic learning, intelligent control techniques.

INTRODUCTION

The first learning that occurs in many species is to follow a relatively large object in motion. The filial imprint is defined as a learning that leads an animal to focus on an object, in a brief critical period that affects behavior patterns when they still do not develop [1]. Following the behavior of the chosen object develops an innate preference for some characteristic; for example, the color.

The imprinting learning is mainly seen in birds; and is based on the attention and evolution of the species; as well as, in the linear movement regardless of the characteristics such as the shape of the object to be followed [2].

In this way without greater knowledge the birds begin their learning until they reach complex behaviors. From here, the importance of implementing the imprint in Robots; since doing so is possible to develop machine learning with a minimum consumption of resources, through evolution; where gradually, as in birds, complex behavior can be achieved. Despite this, there are few machines that base their learning on simple behaviors like that of birds; with the advantage of having an apprenticeship practically without teacher, In addition, to use few material resources. However, to achieve this, one must start with artificial intelligence techniques such as reinforcement learning [3].

In the learning by reinforcement, an agent lives in an environment in which there is not a person who carries out a

teaching. The agent detects the environment and performs actions that modify its status. Thus the agent receives a reward, which can be positive, negative or void. Thus; the system must learn to generate actions that make its reward maximum [4].

To maximize the reward, a model must be found that relates the actions to the environmental conditions, the reward, and the optimization of actions [5].

The agent does not control the environment; but this does influence, so, a Markov chain with reward associated with each state is possible to obtain [6]. Where there is an initial state and terminal states.

In the present article the imprint is implemented using the probability of occurrence of a movement; in order that a robot follows a ball to simulate the imprint. It's not just about performing the tracking technique [7]. The behavior is intended to be more similar to that of a bird; that is why, probability is used and not a deterministic algorithm or one based on image processing.

TRACKING

The tracking is based on the assignment of values to the state - action pairs; forming chains like the one in figure 1 [8]. The characteristics of learning are:

- The environment may or may not be accessible.
- The probabilities can be received in any state and be part of the real utility or suggestions that the agent maximizes.
- The agent can be passive or active.
- The algorithm has a state-action pair

The probabilities are important at all times and are obtained when passing between states [9]. See figure 1.

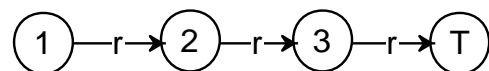


Figure 1. Markov chain for the learning process.

The algorithm only updates the values of the states through which it has passed. However, the agent can select an action with the highest probability value for that state, or choose a random action; either way, it is feasible to assign a probability to each action.

The robot in a 3x3 grid

To explain in a practical way how the algorithm works, a 3x3 grid is proposed, because of its simplicity. In this example, there is a robot that moves in a two-dimensional grid and a red ball in a final position that we will call Meta. The goal is for the robot to reach the goal from any position in the grid; See figure 2. The movements are in four directions: up, right, down, left. The robot has a reward when it approaches the goal, and 0 in the rest of the cases as shown in table 1. So to move from the position $s(1, 1)$ to $s'(1, 2)$ the robot has to follow the direction indicated by column "a" with reward "r" of zero. Table 1 indicates the route that the robot follows to reach the goal.









(1,3)	(2,3)	(3,3) 
(1,2)	(2,2)	(3,2)
(1,1) 	(2,1)	(3,1)

Figure 2. Grille for robot movement

Table 1. Rewards, positions and movements

s	a	s'	r
(1,1)		(1,2)	0
(1,2)		(2,2)	0
(2,2)		(2,1)	0
(2,1)		(3,1)	0
(3,1)		(3,2)	0
(3,2)		(3,3)	1

To choose a position, actions are executed randomly until the objective is reached, and couples are registered.

It is important to explore other possibilities [10]; for example, executed the route indicated in figure 3 by the blue line.



(1,3)	(2,3)	(3,3) 
(1,2)	(2,2)	(3,2)
(1,1) 	(2,1)	(3,1)

Figure 3. A more cut path.

This procedure solves the problem to find the route to the goal. But, what happens if the red ball is in motion? A bird depending on the experience acquired predicts the direction and speed of the moving object. This prediction can be implemented by calculating probabilities.

PROBABILITY OF POSITION OF THE BALL.

Now a grid of 5x5 is proposed as in table 1. When passing any object on this surface, then there will be different passing

frequencies in the boxes. The frequency and probability matrices are constructed considering the positions and changes from one period to another between them. This is how table 2 is obtained 3.

Table 2. Frequency matrix.

	A	B	C	D	E
A	408	62	0	0	0
B	61	406	68	0	0
C	0	63	402	60	0
D	0	0	59	301	55
E	0	0	0	55	256

In this matrix we find that the row cell A with column B tells us how many times the ball is in B since it was previously in A (61). Transition probability matrices handle probabilities as shown in table 3.

Table 3. One-step or first-order transition probability matrix.

	A	B	C	D	E
A	0.8680	0.1319	0	0	0
B	0.1140	0.7588	0.1271	0	0
C	0	0.12	0.7657	0.1142	0
D	0	0	0.1421	0.7253	0.1325
E	0	0	0	0.1612	0.8387

The previous matrix is called a one-step transition matrix, and it gives the probability that the system moves from one state to another in one step or in a transition. The following example shows how the Bayes networks are related and answer the question: If the system is in state B, What will be the probability that it is in state C after two stages? Figure 4 shows the calculations.

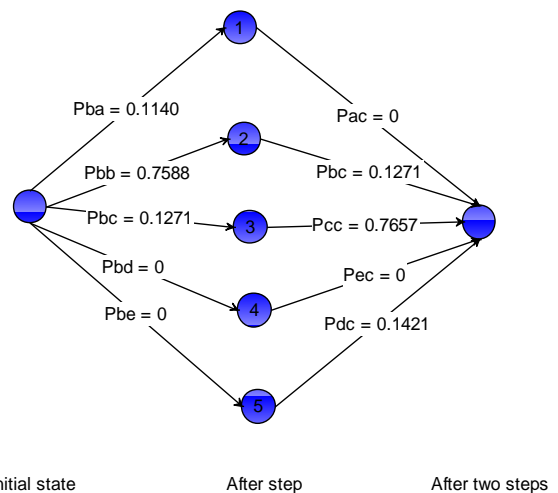


Figure 4. Movement from b to c.

The number of each arrow represents the probability of taking that particular step; For example, in one of the arrows the system goes from B to B and then from B to C. The probability of the first step is 0.7588 and that of the second step is 0.1277. Then the two quantities are multiplied to obtain the probability that the system travels along that route. The probability of any possible route can be calculated in the same way.

$$P^2_{B,C} = (0.1140 \times 0) + (0.7588 \times 0.1271) + (0.1271 \times 0.7657) + (0 \times 0.1421) + (0 \times 0) = 0.1937 \quad (14)$$

If P_{ij} is the probability that the system goes from the state i to the state j in one step and $P^2_{i,k}$ is the probability that the system goes from i to k in two steps, then:

$$P^2_{i,k} = \sum_{\text{todo } j} (P_{i,j})(P_{j,k}) \quad (15)$$

Equation (15) can be used to calculate $P^2_{i,k}$ for all possible combinations of i and k . Table 4 is a two-step transition matrix.

Table 4. Matrix of probabilities of a Markov chain of 2nd order.

	A	B	C	D	E
A	0.7685	0.2146	0.0168	0.0000	0.7685
B	0.1855	0.6061	0.1938	0.0145	0.1855
C	0.0137	0.1829	0.6178	0.1703	0.0137
D	0.0000	0.0171	0.2119	0.5636	0.0000
E	0.0000	0.0000	0.0229	0.2521	0.0000

The procedure shown in (19) is used to calculate the three-step transition matrix, and continue until, as much as possible, there are steady states; that is to say, the probability that the system

is in state i after many transitions will be the same without depending on the start state.

Once the probability matrix has been established, it is implemented within the algorithm as a policy for the robot to take the most likely direction; instead of proposing the arrows as in table 4, even when the goal (ball) is in motion.

IMPRINTING SIMULATION

A Kinect camera was used to obtain the depth of the image. Once the data are obtained are entered into the program; which has the inverse kinematics equations, this allows to calculate the necessary rotation of each engine to reach the point captured by Kinect [11]. This is how the imprint based on probabilities is made with the aim of following the movements of another object. The model of figure 5 satisfies the system:

The robot used is from the brand lego mindstorms, which with some modifications is communicated by the serial port of the computer to obtain data of its position. See figure 6.



Figure 6. Robot used for the proposed system.

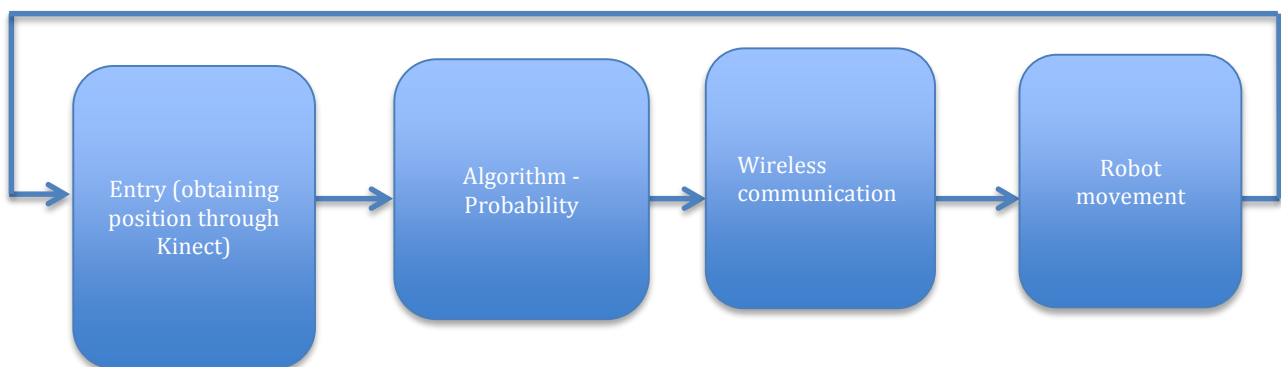


Figure 5. Model for information processing.

IDENTIFICATION OF THE BALL BY ITS COLOR

A ball was recognized within the visual field; that is, the course of a red ball. Remember that learning by imprinting is that birds follow the first moving object they observe [12]. The ball is identified by its red color. The value of its RGB components is compared between the sample and each pixel of the image, for which use is made of the Euclidean distance, which must be less than 10 [13]. See (16). Where: p and q are the pixels with values (x, y) and (s, t) .

$$D_E(p, q) = \sqrt{(x - s)^2 + (y - t)^2} \quad (16)$$

Table 5 shows some values of each saturation component. The Euclidean distance between the value of the sample and those found inside the ball was calculated. To avoid the effects of changes in luminosity, as caused by the shadow of the ball; a light intensity sensor was used.

Table 5. Variation of light intensity captured by Lego Robot.

Deseados			Obtenidos <10			LUMINOSIDAD	DISTANCIA
R	G	B	R	G	B		
177	3	33	174	5	33	27	3.6055512
177	3	33	178	3	30	27	3.162277
177	3	33	177	4	34	27	1.41421
185	2	42	191	1	45	43	6.78233
185	2	42	185	4	48	43	6.3245
185	2	42	188	2	47	43	5.8309
219	4	54	219	4	54	41	0
219	4	54	219	4	54	41	3.162277
219	4	54	222	4	55	41	4.690416
150	1	25	150	3	33	47	8.2462
150	1	25	147	4	31	47	7.34
150	1	25	152	3	25	47	2.8284
86	16	47	87	19	43	52	5.09
86	16	47	89	17	44	52	4.35
86	16	47	88	18	44	52	4.123

A mathematical relationship between the RGB values, distance and luminosity is obtained by means of multiple linear regression [14]:

$$\begin{aligned} \text{DISTANCE} = & -1.74036 + 0.44856 * R1 - 0.66288 * G1 \\ & - 0.44396 * B1 - 0.42494 * R2 \\ & + 0.94657 * G2 + 0.46801 * B2 \\ & - 0.02486 * \text{BRIGHTNESS} \end{aligned} \quad (22)$$

The ratio cm vs Pixel allows to know what the size of the visual field is and handle speed relationships. Figure 7 shows the relation cm vs Pixel in a plane x , and considering the distance between the plane and the Kinect sensor on the z axis with a value of 1 meter.

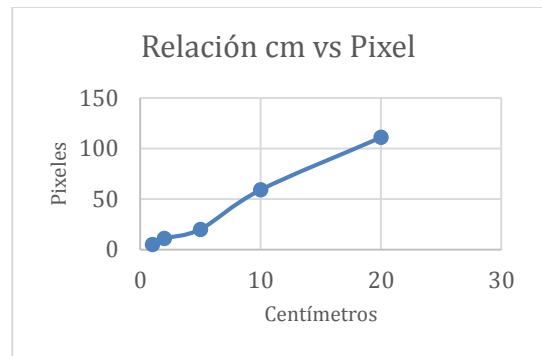


Figure 7. Pixel vs Cm Ratio

$$y = 5.52x \quad (17)$$

However, when the distance between the sensor and the plane changes this relationship also; the graph of figure 8 and the equations (17) show it:

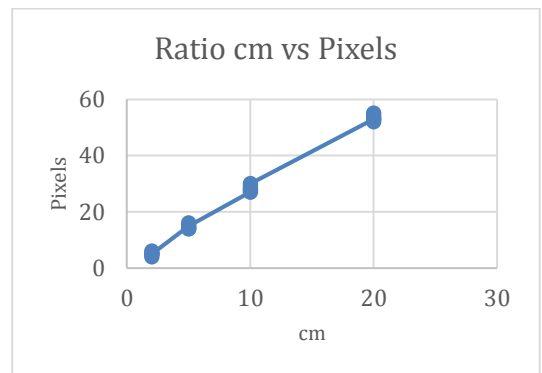


Figure 8. Pixel vs Cm relation with distance change between the sensor and the plane, 2 meters:

$$y = 2.72x \quad (18)$$

From the graphs and equations it is deduced that the relationship between pixels and centimeters changes with the distance between the sensor and the plane in which measurements are made. Thus, the relationships of figure 7 and 8 are not functional; it is necessary to build a relationship between the centimeters variables, depth and pixels. By multiple linear regressions it is obtained:

$$\text{cm} = 5.6272 + 0.15107 * \text{pixels} + 0.56252 * \text{Depth (in centimeters)} \quad (19)$$

The relation allows to know the size of the objects in centimeters depending of the value in pixels, which is obtained from the camera at different depths; with an error of 4 centimeters at a distance of two meters Deep; however, since our field of vision is 1.20 meters, the error is 1 cm and is not relevant to our experiment, so the expression is valid.

RESULTS AND DISCUSSION

The graph of Figure 9 was obtained and illustrates the tracking of the ball. This graph shows the time that the target is reached, which is about 7 seconds.

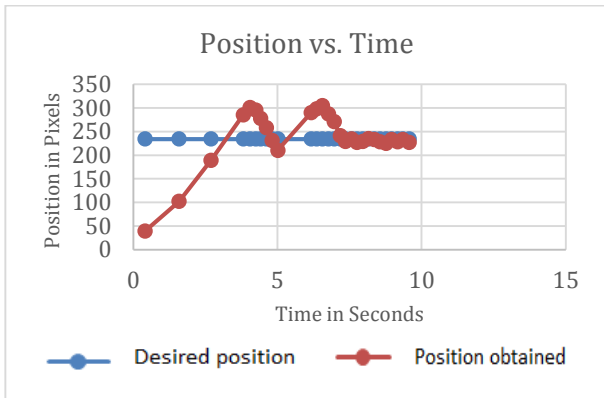


Figure 9. Position tracking.

In a second test the graph of figure 10 was obtained. In this graph is distinguished that the robot reaches the ball in a time of approximately 4 seconds, also variations of 10 pixels are distinguished.

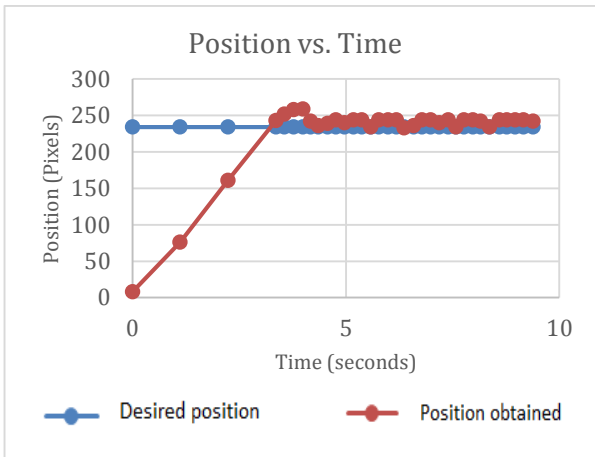


Figure 10. Second follow-up test

In a third test the graph of figure 11 was obtained:



Figure 11: Third follow-up test.

This graph shows that the desired point was reached in less than 4 seconds while keeping track of the moving object. Figure 12 shows multiple changes of direction of the ball. The robot likewise continues the monitoring so the imprint is made.

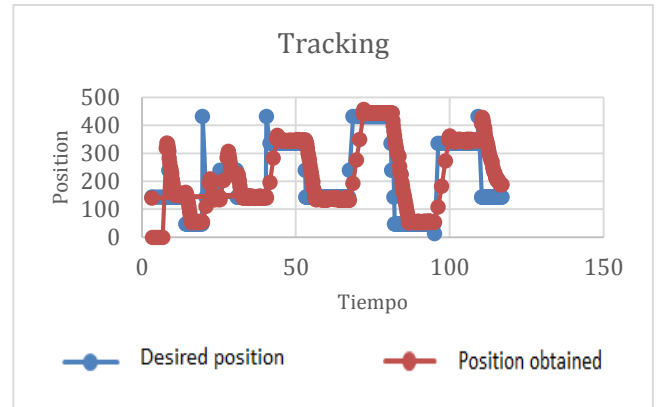


Figure 12. Set of position graphs.

These graphs show that the tracking of the ball is carried out, even with abrupt changes in direction and distance. See figure 13.

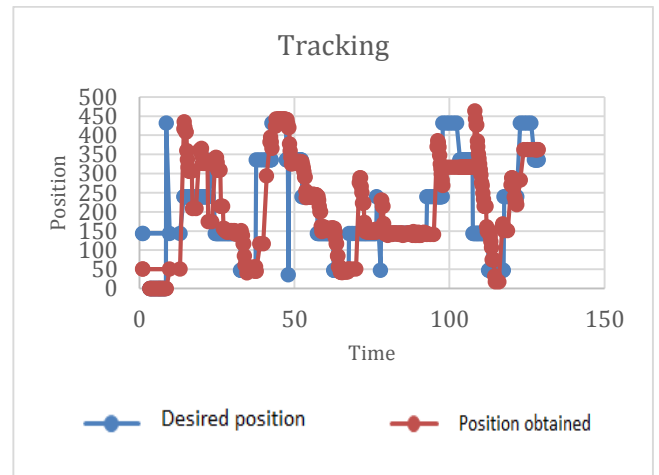


Figure 13. Mobile tracking with abrupt changes.

CONCLUSION

A system was designed that allows the learning of the imprint behavior, based on the calculation of probabilities. It is important to emphasize that algorithms can be reproduced in different fields and areas, since its operation is based on the ability to collect data and its interpretation. The system provides information that allows a better understanding of the context in which it is presented.

The relationship to determine the position of the ball and its tracking with changes in lighting solves the problem of similar colors like orange with shadows; so the imprint can be implemented even considering objects with different colors or shapes. Even when the ball makes abrupt changes is possible to follow it.

The algorithm is possible to use for any start of learning activities since the imprinting occurs at any stage of life. For example, it can be applied to the recognition of faces or in the study of phobias that influences the design of robots that allow the control of stress levels. That is, any robot can learn autonomously if executes the imprint learning; and no need to invest in large resources or knowledge databases.

This type of interaction with the robots allows them to perform a large number of tasks. In our case it starts with an imprint with the final goal of following objects in movement.

ACKNOWLEDGMENTS

The authors are grateful for the support received from the Instituto Politécnico Nacional, ESIME - ESCOM, and the CONACYT of Mexico.

REFERENCES

- [1] Pérez M., 1998. *Psicología II*, Universidad de Barcelona, España.
- [2] Kent, A; Williams, J and Kent, R, 1991. *Encyclopedia of Microcomputer; Geographic information system to hypertext*, CRC Press, USA.
- [3] Rebollo F., Barrojo D., 2009. *Aprendizaje por Refuerzo; Aprendizaje Automático*, Departamento de Informática, Escuela Politécnica Superior; Universidad Carlos III, Madrid.
- [4] Cabrera J., 2006. *Seguimiento visual en mecanismos robóticos con Q-Learning*; ITESM, Cuernavaca, Morelos.
- [5] Dos Santos D., Pañalver R., 2005. *Sistema Autónomo de navegación en robots con reconocimiento de patrones geométricos regulares*, *tekhne Revista de la Facultad de Ingeniería*; No. 8; Universidad Católica Andrés Bello; Caracas Venezuela.
- [6] Ruiz J., Mateos F., 2014-2015. *Procesos de Markov; Ampliación de Inteligencia Artificial*, Dpto. Ciencias de la Computación e Inteligencia Artificial.
- [7] Cruz V., Hidalgo E., Acosta H., 2012. *A line follower robot implementation using Lego's Mindstorms Kit and Q-Learning*. Vol. 22 (NE-1), RNC, Acta Universidad, Universidad de Guanajuato.
- [8] Cabrera J., 2017. *Seguimiento, Visual en Mecanismos con Q-Learning-Edición Única*; ITESM-Campus Cuernavaca.
- [9] Printista A., Errecalde M., Montoya C., 2000. *Una Implementación del algoritmo de Q-Learning basada en un esquema de comunicación con Cache*, Universidad Nacional de San Luis; Argentina.
- [10] Teknomo K., 2017. *Q-Learning by examples*, <http://people.revoledu.com/kardi/tutorial/ReinforcementLearning/>.
- [11] Ramos E., Castro C., 2011. *Arduino and Kinect Projects: Design, Build, Blow Their Minds*, Apress, New York.
- [12] Miura M., Matsushima T., *Biological motion facilitates filial imprinting*; *Animal Behaviour*, Volume 116, pp. 171-180, 2016.
- [13] Esqueda J., Palafox L., 2005. *Fundamentos de procesamiento de imágenes*, Universidad Autónoma de Baja California; pp. 15.
- [14] Torrelavega., 2017. *Ajuste por Mínimos cuadrados*, Escuela Politécnica de Ingeniería de Minas y Energía, <http://ocw.unican.es/enseñanzas-tecnicas/fisica-i/practicas-1/Ajuste%20por%20minimos%20cuadrados.pdf>.