BEHAVIORAL CONTROL OF A LEGO NXT ROBOT ORIENTED BY SEARCHING TASKS AND AVOIDING OBSTACLES

CONTROL COMPORTAMENTAL DE UN ROBOT LEGO NXT ORIENTADO PARA TAREAS DE BÚSQUEDA Y EVASIÓN DE OBSTÁCULOS

.

Edwin A Beltrán González[1] **Miguel R Perez Pereira**[2] **Giovanni R Bermúdez Bohórquez**[3]

Abstract: This paper shows the design of a reactive architecture for a robot using LEGO NXT drawing Brooks' approach for the development of searching tasks and avoiding obstacles in a dynamic environment. One of the most important aspects in this work, is the implementation of two primary mechanisms of coordination mentioned by Brooks, inhibition and suppression. Reactive paradigm is one of the approaches used in the robotics field. This reactive paradigm emerged in the late 80´s as a result of researchers such as Brooks and Arkin´s work at Massachusetts Institute of Technology MIT, their proposal is based on the creation of strongly coupled systems of perception and action, which enables them to interact in dynamic environments. Subsumed Architecture SA is also one of the approaches based on this paradigm in that it proposes a parallel architecture layered behavioral, which runs asynchronously but in many cases, they have common goals.

Key Words: subsumed architecture, behaviors, LabVIEW, LEGO NXT, reactive paradigm.

---

[1] BSc. In Electronic Technology, and Control Engineering, Universidad Distrital Francisco José de Caldas, Colombia. Current position: Research group in Robotic Mobile Autonomous (ROMA), Colombia. E-mail: eabeltrang@correo.udistrital.edu.co

[2] BSc. In Control and instrumentation Engineering, Universidad Distrital Francisco José de Caldas, Colombia; Specialist in Teaching and Pedagocical University, Universidad San Buenaventura, Colombia. Current posititon: Professor Universidad Distrital Francisco José de Caldas and Research group in Robotic Mobile Autonomous (ROMA)- E-mail: mrperezp@udistrital.edu.co

[3] BSc. In Electricial Engineering, Universidad Nacional de Colombia; MSc. In Electronic and Computers Engineering, Universidad de los Andes. Current position: Professor Universidad Distrital Francisco José de Caldas, Colombia and Director research group in Robotic Mobile Autonomous (ROMA), Colombia.E-mail: gbermudez@udistrital.edu.co

Preparación de Artículos revista VISIÓN ELECTRÓNICA: algo más que un estado sólido
Fecha de envío: 5 de Marzo de 2016
Fecha de recepción: 15 de Marzo de 2016
Fecha de aceptación: 14 de Junio de 2016

Resumen: El paradigma reactivo es uno de los enfoques más utilizados en el campo de la robótica. Surge a finales de los años 80 como resultado del trabajo de algunos investigadores como Brooks y Arkin en el MIT, su propuesta está basada en la creación de sistemas que acoplan fuertemente la percepción y la acción, lo que los capacita para interactuar con entornos dinámicos. La arquitectura subsumida es una de los enfoques basados en este paradigma y propone una arquitectura paralela dividida en capas comportamentales, las cuales se ejecutan asincrónicamente pero que en muchos casos poseen objetivos en común. El presente trabajo muestra el diseño de una arquitectura reactiva para un robot LEGO NXT utilizando el enfoque de Brooks para el desarrollo de tareas de búsqueda y evasión de obstáculos en un entorno dinámico, uno de los aspectos más relevantes en este trabajo es la implementación de los dos mecanismos primarios de coordinación mencionados por Brooks, inhibición y supresión.

Palabras clave: Arquitectura subsumida, comportamientos, LabVIEW, LEGO NXT, paradigma reactivo.

1 Introduction

A behavior coined by [1] is defined as a response or reaction to a stimulus, from the pragmatic point of view that can be expressed as an interaction that arises between an individual and his environment. The biological sciences are responsible for studying these concepts and bring them to a number of formalisms that turn out to be the inspiration for many robotic systems based on animals behavior, which in turn are also based on the reactive paradigm.

1.1. Reactive Paradigm

One approach commonly used in robotic systems based on behaviors is the reactive paradigm [1] where perception and action without an abstract representation of time is strongly coupled. Taking into account this, a reactive robotic system has the following features:

- The building blocks are composed by primitive behaviors: behaviors are usually systems characterized like a basic response to a stimulus of a motor, that is enough to generate an action of low level applicable to layers' behavior slightly structured.
- The use of abstract representations of knowledge is restricted to the generation of basic responses of a stimulus: Since being purely reactive systems respond very quickly to stimuli generated by the environment. This is essentially the system, also it is focused on the production of reactions to inputs that there is no time for generating abstractions of the world, neither to the creation of possible models representing the dynamics of the world around the robot, in terms of computational demand, the reactive systems prove to be more efficient than others.
- The animal behavior turns out to be the basis of reactive systems: biological sciences such as ethology, neuroscience and psychology studies are provided by concerning the tasks applicable to robotic systems, by generating models, formalisms and mechanisms readily applicable to robots. It usually seeks to imitate or potentially generate degrees of similar behaviors of animals in both individually and collectively, where there are already multi-robot tasks systems.
- Reactive systems are found to be inherent in the algorithmic structure the designer to use. Since its high degree of modularity behavior can be described by adding various

Fecha de envío: 5 de Marzo de 2016
Fecha de recepción: 15 de Marzo de 2016
Fecha de aceptación: 14 de Junio de 2016

reactions, but if you get high levels of complexity, the designer is not obliged to discard or redesign your model to adopt new strategies.

1.2. Subsumed Architecture

Subsumed Architecture SA is proposed and developed by Rodney Brooks [2] in the mid-80s at the Massachusetts Institute of Technology (MIT), their approach is a method based on purely reactive behavior, which led to a revolution in its time, even when the deliberative paradigm controlled robotic designs.

Brooks argued that the paradigm sensing-plan-act was inefficient when building robots for real applications (robots were not capable of interacting with humans in everyday tasks), even if its architecture were complex and world models were built, it was rather expensive in computationally platforms, thus being rather slow in responses. The proposed model is divided into parallel layers of architecture behavioral which are executed asynchronously, but in many cases, have targets in common (Figure 1).

One of the most common ways to represent such interactions, occurs through asynchronous finite state machines [3], which are on the lowest level of abstraction and are able to build their own representations of the world without needing to know information of other behaviors. The coordination process coined by [1], [4]; between layers of behaviors is called subsumption [3], where the complex actions can be suppressed or subsumed and even are less complex. It is a hierarchical topology that defines upper layers and lower layers, the latter has the peculiarity that they do not have access to information from higher layers, whereas, this last one has mastery of the most relevant information of the lower layers.

The coordination in subsumption has two primary mechanisms of action, namely:

- Inhibition: Used to prevent transmission of a signal between behaviors, it is usually given to the entry thereof.

- Suppression: Used to temporarily replace or terminate the transmission of a signal, it usually operates while this active suppression sends a message to the actuators or behavioral outputs.

2. Methodology and Results

The approach undertaken contemplates the implementation of four behavioral layers: PATH, SEARCH, WANDER, CALL, each comprising a set of basic behaviors [4-6] that have allowed the execution of search and rescue tasks from heat sources in a dynamic environment (figure. 1). In this article only the PATH layer responsible for the execution of search tasks and obstacle avoidance is involved[1].

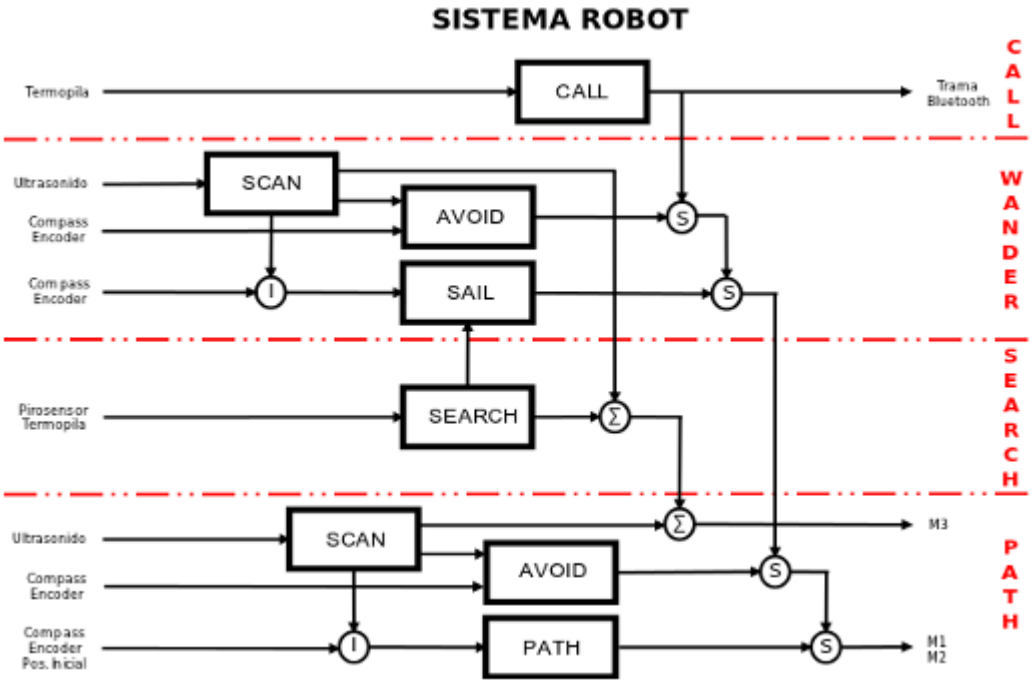

Figure 1. Objects of study in the development of behavioral layers to the robot. Source: own.

[1] Las capas SEARCH, WANDER y CALL no son objeto de estudio en este trabajo

Fecha de envío: 5 de Marzo de 2016
Fecha de recepción: 15 de Marzo de 2016
Fecha de aceptación: 14 de Junio de 2016

2.1 PATH Layer –PATH behavior trajectory

The objective of the implementation of the behavior was given to the robot´s ability to follow a straight path, and helped by the compass´ sensor and position encoder. Thus, initially, the robot is taking a baseline, after following a predetermined route (rect line) and once traveled 30 cm and through his position sensors determined that the error was introduced along the road, in other words, it was far from the desired trajectory, in order to send a stimulus to the motors (behavioral outputs) and to correct its course to meet again on the predetermined path (figure 2).

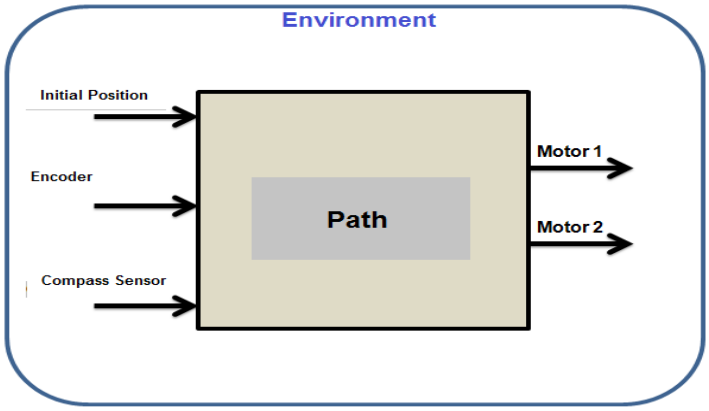

Figure 2. Description of the inputs and outputs of the PATH behavior according to their interaction with the work environment. Source: own.

Kinematic model development [7] is carried out after (1) and (2), in which an approximation to the model shown in a differential stage. This was necessary to implement in the physical parameters in the platform such as drive axle (11.5cm) radius wheels (2.8cm).

$$\omega(t) = \frac{\left(V_L(t) - V_R(t)\right)}{b} \tag{1}$$

$$v(t) = \frac{\left(V_L(t) + V_R(t)\right)}{2} \tag{2}$$

In the figure above, it is shown where the speed is in each wheel of the platform, as well as the main result of the implementation of the kinematic models that have been developed. The kinematic model of the robot platform figure (3) below is presented in the general expression in a simple form or in a vector form, by using these all the corresponding constants that were obtained in the platform.

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} -0.0244 \cdot sen\varphi & -0.0244 \cdot sen\varphi \\ 0.0244 \cdot \cos\varphi & 0.0244 \cdot \cos\varphi \\ -0.2856 & 0.2856 \end{bmatrix} \cdot \begin{bmatrix} \theta_L \\ \theta_R \end{bmatrix} \quad (3)$$

Through a compass sensor the robot reference system must be established to determine the orientation of the requirements to estimate positions and which one was the starting point for the formation of the whole multirobot system (4).

$$\varphi = \alpha - \theta \quad (4)$$

After studying the basic requirements for the location of the robot (position and orientation) an algorithm was developed to validate the PATH behavior (figure 3), which took as input signals inhibition and suppression of behavior, developed a coordinate pair (x, y) and the previous coordinates of the robot in order to perform the iterations with the robot platform and the initial reading of the compass sensor. The information delivered to the output consists of the (x, y) position and orientation angle of the robot, including a Boolean signal that indicates the current program execution.

Fecha de envío: 5 de Marzo de 2016
Fecha de recepción: 15 de Marzo de 2016
Fecha de aceptación: 14 de Junio de 2016

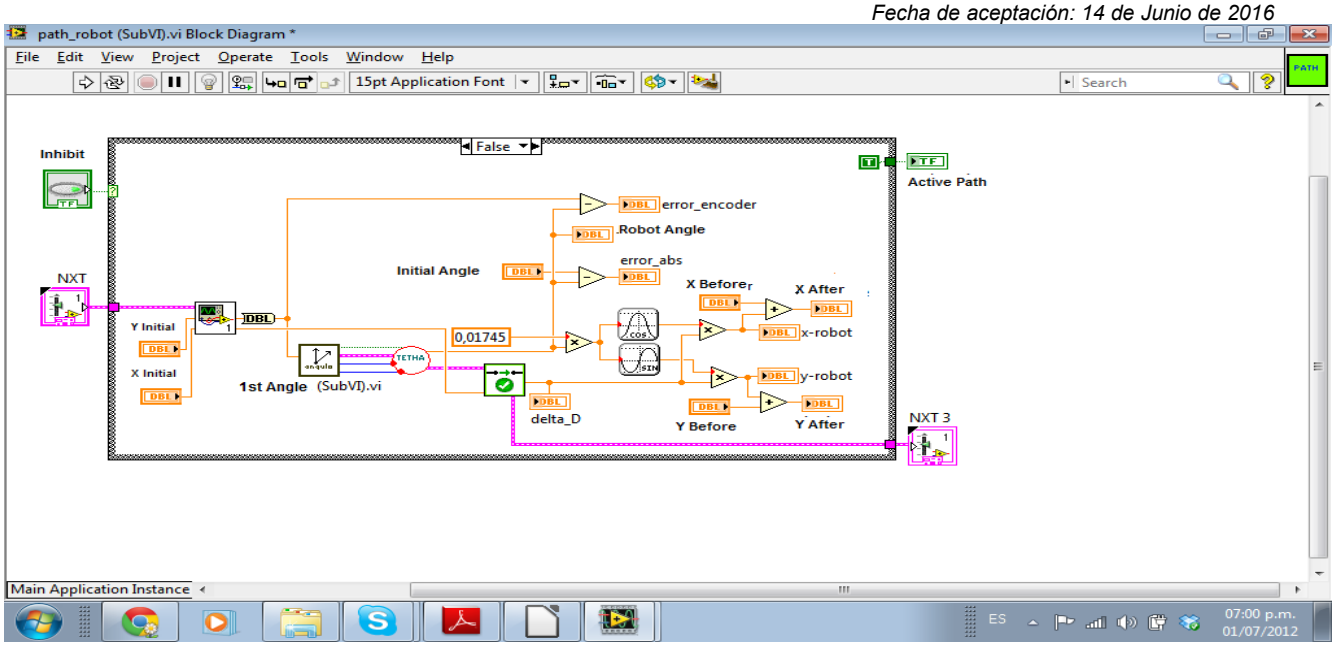

Figure 3. *LabVIEW Implementation behavior PATH.* Source: own.

The validation and experimentation of the process was carried out in the coliseum of Technological Faculty. The recording information about the processes of experimentation conducted, yielded a number of data which were tabulated and used to develop a process of characterizing systematic errors related to the mobile platform using quadratic approximation models (figure 4).

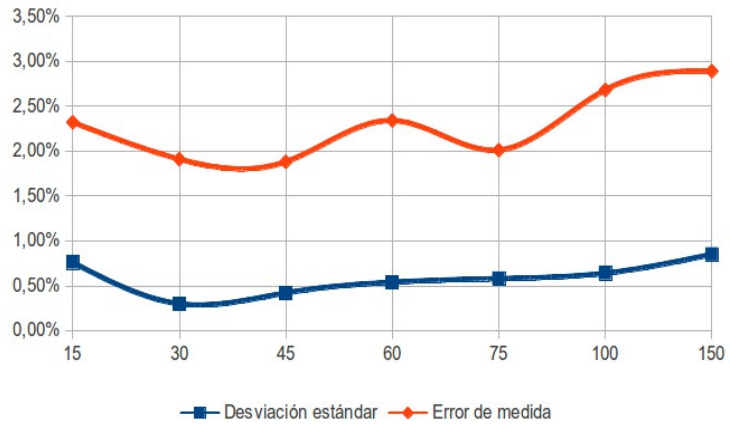

Figure 4. Measurement error for scrolling through the implementation of PATH behavior.

Source: own

The motion in a straight line was obtained by displacement of 15cm that diversion standard was ±0,76% with a measurement error ±2,32% and for displacement of 45cm was ±0,57% with ±1,88%. For the selected distance was obtained a standard deviation of ±0,30% and a measurement error of ±1,91%.

As part of this validation process, it was also taken into account the deviation of the robot relative to the reference system (figure 5), where the ideal route introduced an error of zero, additionally the values above zero show a deviation to the right, similar to the values below zero determining that the deviation of the robot was to the left, once it is presented, the behavior configured can show a correction in the opposite direction to the one performed previously.

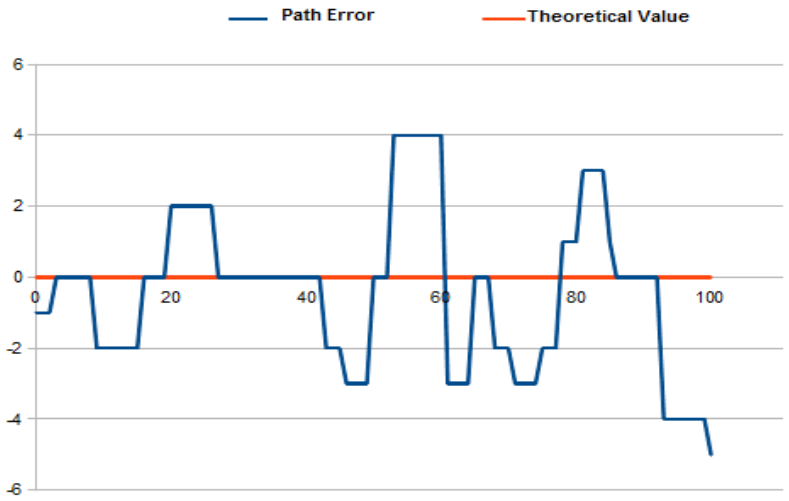

Figure 5. Deviation of the robot relative to the reference. Source: own.

2.2. Layer PATH – Seeks Behavior or SCAN

This behavior whose main objective is searching and obstacle detection while PATH behavior is performed. It is activated after performing a predetermined path for the robot (for this case is 30cm) will be a signal sent by PATH. In this sense, SCAN behavior takes the signal of the ultrasonic sensor and determines the location and distance of the obstacle with respect to the robot. In turn, this behavior M3 controls (dedicated engine for the implementation of a basic

Fecha de envío: 5 de Marzo de 2016
Fecha de recepción: 15 de Marzo de 2016
Fecha de aceptación: 14 de Junio de 2016

radar) motor. Besides, AVOID active behavior and inhibits PATH allow the robot to overcome the obstacle and track once it passes the obstacle (figure 6).

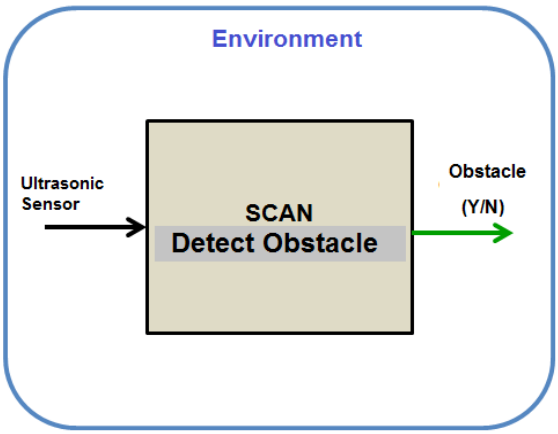

Figure 6. inputs and outputs description of behavior SCAN. Source: own.

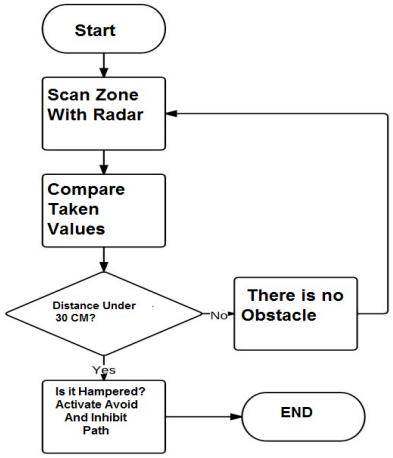

Figure 7. Flowchart for describing the behavior SCAN. Source: own.

To validate the performance, it was carried out the implementation of the algorithm shown in (figure 7) in LabVIEW, in which the robot takes samples every 30 degrees with the motor M3 in an aperture range of ± 80 degrees and it compares the values delivered by the ultrasonic sensor to determine the direction and distance of the obstacle.

Finding the behavior obstacle, it generates a Boolean inhibition signal to be sent to the PATH, and it also generates a signal for activating the same type of the following behavior, this is the AVOID layer (figure 8).

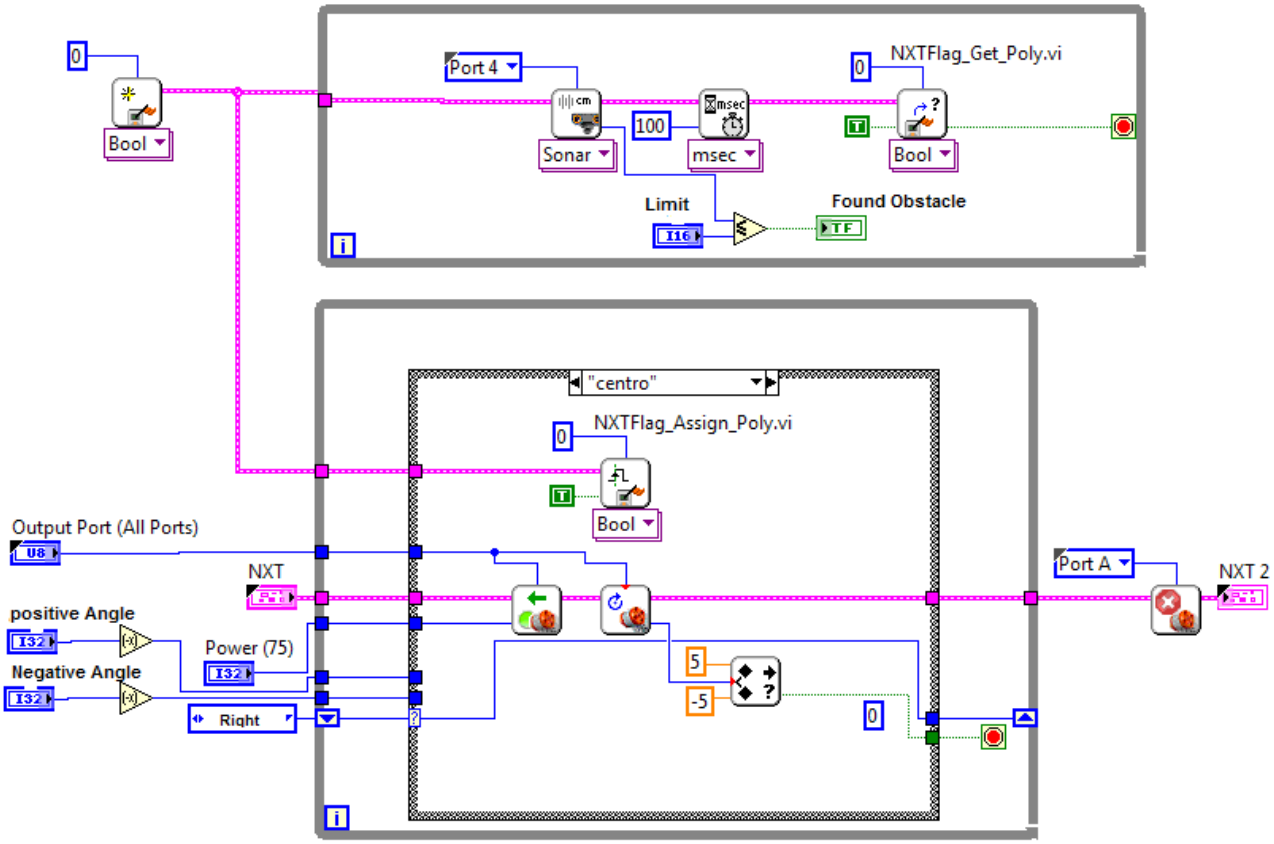

Figure 8. Implementation SCAN behavior in LabVIEW. Source: own.

2.3. Layer PATH - Behavior evade or AVOID

The layer PATH is activated by SCAN behavior, whereas the main objective of the AVOID behavior is circumvent to the obstacle considering that it must return to the original route. To do this, this behavior suppresses the behavior PATH and it should calculate the error of its motion relative to a predetermined route to ensure the return to the original route, once the obstacle has been overcome. The displacement of the robot is 15cm when running this behavior. The entries in this behavior are the signs of the encoder and compass necessary to calculate the positions of the robot and its outputs are stimuli to M1 and M2 engines. Once

Fecha de envío: 5 de Marzo de 2016
Fecha de recepción: 15 de Marzo de 2016
Fecha de aceptación: 14 de Junio de 2016

you return to the original route, this behavior should release behavior PATH which will be the following running (figure 9)

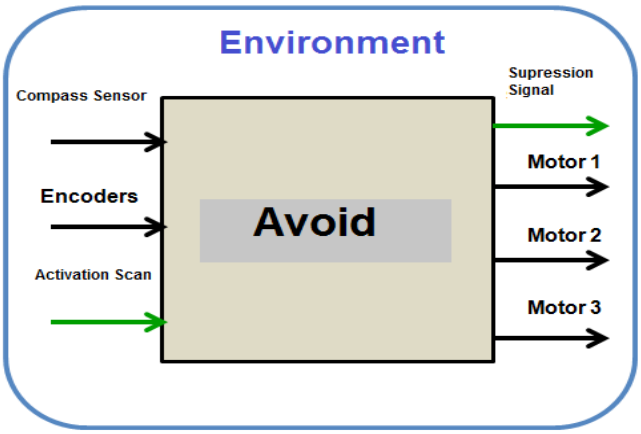

Figure 9. Description of the inputs and outputs AVOID behavior according to their interaction with the environment. Source: own.

To validate the behavior, it was implemented in the LabVIEW algorithm shown in figure10, where the behavior is activated with the SCAN activation signal which indicates the presence of obstacles on the way. At the moment of register the location of the robot, this is activated followed to this the robot when turning right until the compass indicates 90 ° of deviation from the path and it moves in a straight line, then through the radar, it determines the presence of the obstacle being detected just moving again in straight line. Otherwise, it rotates counterclockwise until the compass determines that is in line with the direction of the path, repeating the process to move forward and to determine through radars the presence of obstacles and it follows the same logic, once it overcomes the obstacle, it rotates clockwise.

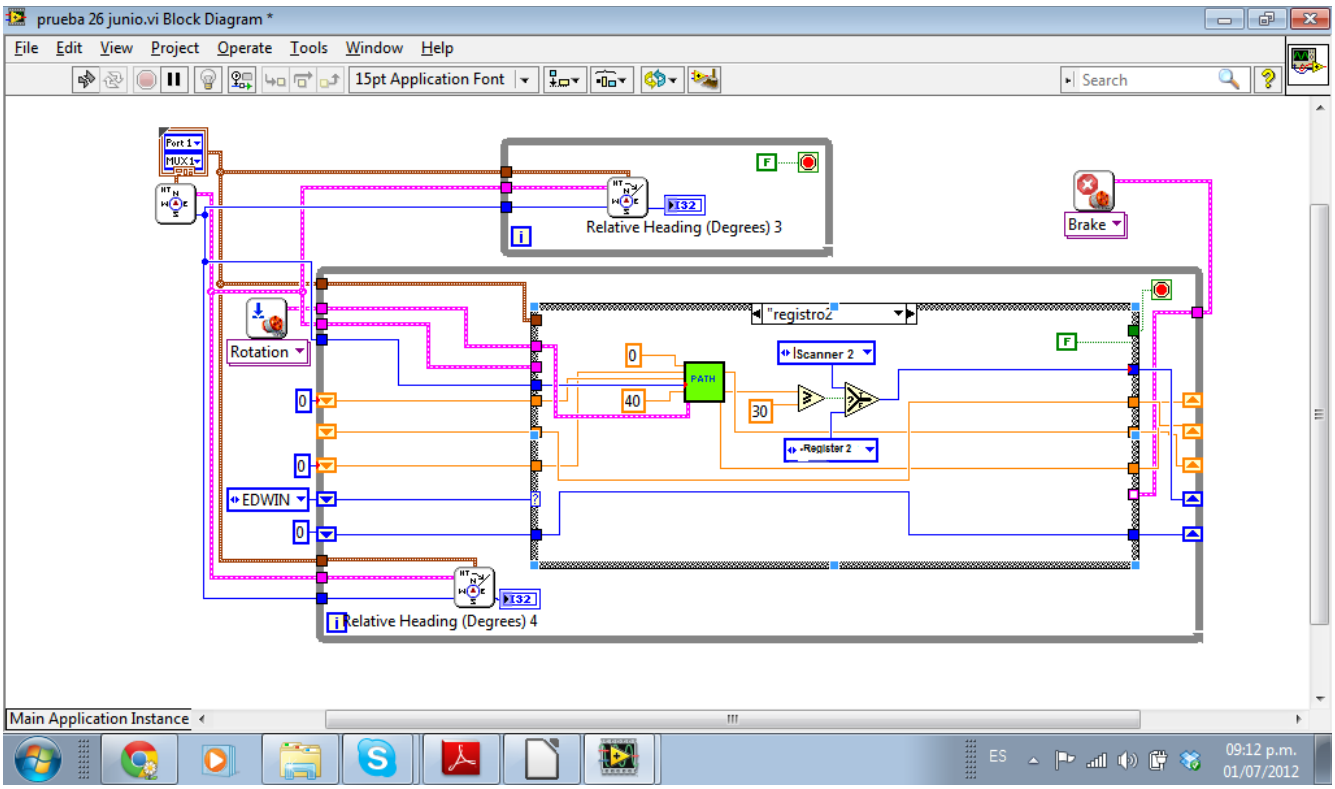

Figure 10. At this point calculation error in STI movement relative to the path is determined, and based on the registration of STIs position (x, y) and the angle of deviation given by the compass, so the mistake can be when the robot is near zero it is centered on the way, and while it increases its value it means that the robot is moving away from the road. **Source:** own

Fecha de envío: 5 de Marzo de 2016
Fecha de recepción: 15 de Marzo de 2016
Fecha de aceptación: 14 de Junio de 2016

At this point, calculation error in its movement relative to the path is determined and based on the registration of its position (x, y) and also because of the angle of deviation given by the compass, so when the error is near zero the robot is centered on the way, and while this increases its value, it means that the robot is moving away from the road (figure 11)

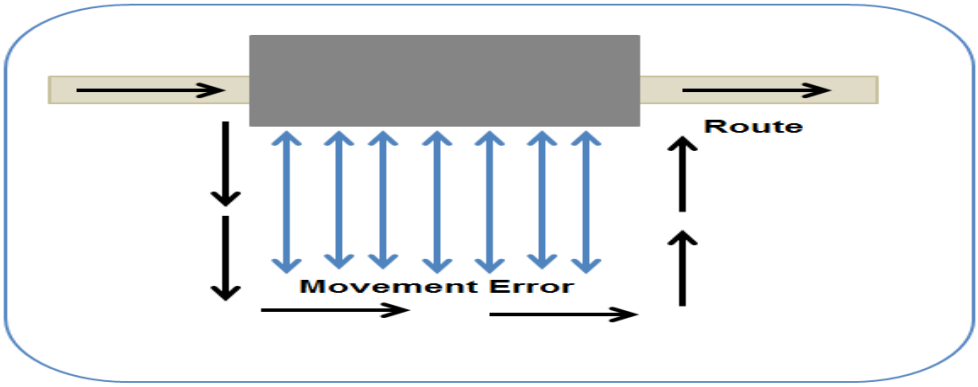

Figure 11. Nature of motion error regarding a route. Source: own

2.4. Integration and validation of behavior in the PATH layer

Once developed the three basic robot behaviors separately, they held their integration into LabVIEW, in order to validate the operation of the layer together, which was established as a cornerstone in the research project. Through the combination of each SubVI there is a possible integration of three behaviors, where the main objective was to observe the performance of the coating acting in conjunction with each of the signals from the inputs and outputs to the motors as well as the inhibition and suppression signals for each of the control algorithms.

In figure 12 the first integration between PATH and SCAN behaviors shown, the goal was to give the robot the ability to follow a straight-line path, also the correct movement and to detect

obstacles on its way, if the presence is determined by an obstacle, the robot will stop otherwise it stands straight forwards. Additionally, once developed subVI where through a Boolean command was ordered the algorithm execution, the information will return to the PC.

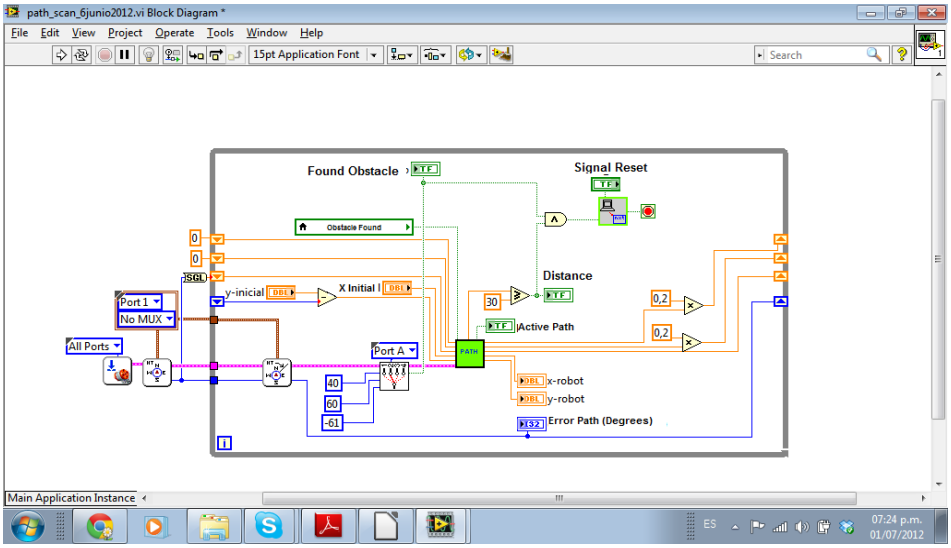

Figure 12. Integration of the PATH AND SCAN behaviors in a single algorithm. Source: own.

Once integrated the first two behaviors, we proceeded with the integration of AVOID, thanks to the modularity of each of the algorithms, the layer was successfully coupled. Thus, for the experimentation process it was followed with the same test protocol where it placed the robot on a line with several obstacles on the line, whichever it has taken the mistake of movement in relation to the straight line, this was recorded and therefore it was observed its evolution through the test execution.

Fecha de envío: 5 de Marzo de 2016
Fecha de recepción: 15 de Marzo de 2016
Fecha de aceptación: 14 de Junio de 2016

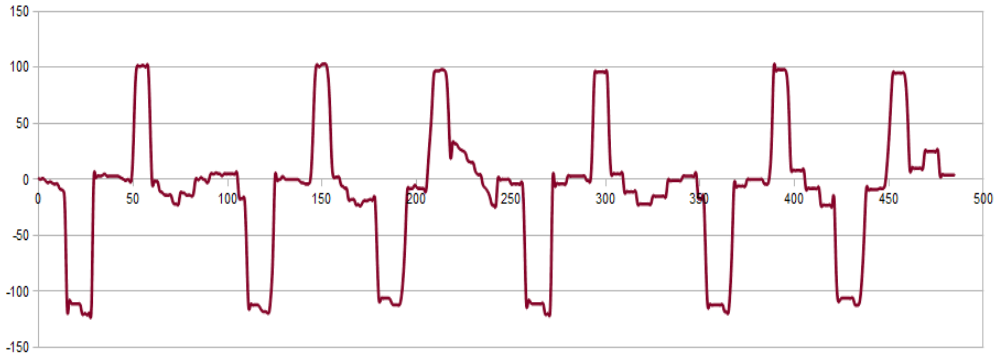

Figure 13. Robot registered movement in the implementation of the PATH layer. Source: own.

In figure 13, the record obtained for the movement of the robot shown evading five obstacles. It traces with values that start from zero and become negative these are taken as reading the error while the robot is turning to leave the road and avoid the obstacle. While values starting from scratch and become positive, it represents the rotation performed by the robot to return to the default path. As a result of this observation is that the orders made by the robot are pretty close to 90 ° and regardless of the number of obstacles encountered on the way the robot takes its path successfully (figure14).

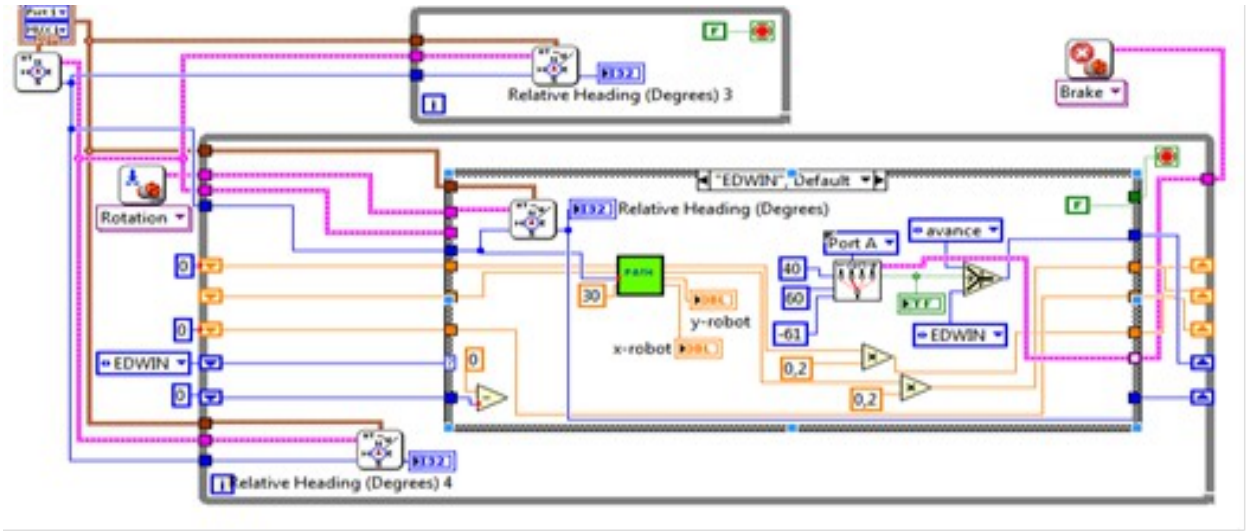

Figure 14. Three behaviors integration of PATH, SCAN and AVOID. Source: own.

3. Conclusions

The reactive paradigm emerged as a solution to the need for more efficient robots, its proposal is based on the life sciences, which takes several of its most important elements for the designing and construction of robotic most capable agent.

The subsumed architecture is characterized by an incremental system, where the sum of primitive behaviors can structure and control complex systems, since the implementation of several of them asynchronously leads to accomplish tasks and goals initially set.

One of the most important tasks within the research project was to give the robot the ability to enter and exit into appropriately environments, with the development of the layer PATH achieving this task, this is assured through integration of three primitive behaviors, which were implemented in LabVIEW and validated in a Lego NXT robotics platform.

The coordination of the three basic behaviors of the PATH layer is conducted through the primary mechanisms discussed by Brooks in his proposal (suppressants and inhibitors), getting a good platform performance by implementing its main task, recording values with very small errors regarding a desired trajectory.

References

[1] M. Amoretti & M. Reggiani, (2010). “Architectural paradigms for robotics applications”. *Advanced Engineering Informatics*, vol 24, no 1, pp. 4–13. doi:10.1016/j.aei.2009.08.004

[2] R. C. Arkin, Chapter 3. “Robot Behavior”. In M. Dorigo (Ed.), Behavior Based Robotics, pp. 65–120. London.England: The MIT Press, (1998a)

[3] R. C. Arkin, Chapter 4. Behavior Based -Architectures. In M. Dorigo (Ed.), Behavior Based Robotics, pp. 123–173. London.England: The MIT Press. (1998b).

[4] T. Balch, R. C. Arkin & S. Member “Behavior-based Formation Control for Multi-robot Teams”. *IEEE Transactions on robotics and automation*, vol XX, no 1, pp. 1–15, 1999

[5] G. Baldassarre, D. Parisi & S. Nolfi, “Distributed coordination of simulated robots based on self-organization”. *Artificial life*, vol 12, no 3, pp. 289–311, 2006 doi:10.1162/artl.2006.12.3.289

Fecha de envío: 5 de Marzo de 2016
Fecha de recepción: 15 de Marzo de 2016
Fecha de aceptación: 14 de Junio de 2016

[6] L. E. Parker, "Current research in multirobot systems". *Artif Life Robotics*, vol 7, no 1, pp.1–5, 2003, doi:10.1007/s10015-003-0229-9

[7] G. Bermudez, "Modelamiento cinemático y odométrico de robots móviles Aspectos matemáticos". *Revista Tecnura*, vol 1, no 1, pp. 1–12, 2003.