

Desarrollo de un sistema de control reactivo para un robot enfocado en tareas de búsqueda de fuentes de calor

Development of a reactive control system for a robot search tasks focused on heat sources

Edwin Andrés Beltrán González*
Kristel Kristel Solange Novoa Roldán**

Fecha de recepción: 23 de marzo de 2013

Fecha de aprobación: 30 de abril de 2013

Resumen

Siguiendo el enfoque tomado por Brooks en el desarrollo de la arquitectura subsumida el desarrollo de este artículo muestra el diseño de un sistema de control comportamental en un robot, constituido por comportamientos básicos que son coordinados por medio de los mecanismos de inhibición y supresión. En cada uno de los comportamientos desarrollados se muestran las características más relevantes de cada uno y su función dentro de la tarea general de búsqueda y localización de una fuente de calor, así como los resultados y análisis de cada uno de los procesos de experimentación llevados a cabo.

Palabras clave

Arquitectura subsumida, comportamientos, paradigma reactivo, capas comportamentales.

* Grupo de Investigación ROMA Universidad Distrital Francisco José de Caldas Facultad Tecnológica 57 311 5475273. eabeltrang@correo.udistrital.edu.co Bogotá, Colombia

** Grupo de Investigación ROMA Universidad Distrital Francisco José de Caldas Facultad Tecnológica 57 310 3147440. ksnovoar@udistrital.edu.co Bogotá, Colombia

Abstract

Following the approach taken by Brooks in developing subsumed development architecture of this article shows the design of a control system of a robot behavioral, comprising basic behaviors that are coordinated by means of inhibition and suppression mechanisms. In each of the behaviors developed shows the most relevant characteristics of each and their role in the overall task of search and location of a heat source, and the results and analysis of each of the processes of experimentation carried out.

Keywords

Architecture subsumed, behaviors, reactive paradigm, behavioral layers.

1. Introducción

Actualmente las aplicaciones en robótica móvil están orientadas al diseño de sistemas cooperativos que de gran manera potencian muchas de las habilidades que pudiera presentar por sí solo un solo robot.

Las arquitecturas clásicas de control imposibilitan de gran manera implementaciones en el campo de la robótica cooperativa ya que por su estructura requiere de modelos abstractos de su entorno [1], haciendo de estos lentos y altamente costosos en el ámbito computacional.

El paradigma reactivo [2] introdujo la idea de diseñar robots capaces de interactuar con el humano y que fuesen tan sencillos y tan asequibles como se lograra. Bajo este paradigma surgieron enfoques distintos que dieron origen a arquitecturas reactivas. Una de ellas es la arquitectura subsumida derivada del trabajo de Brooks [3] que por medio de mecanismos como la supresión y la inhibición logra la coordinación del sistema [4] haciendo de esta una jerarquía de capas comportamentales [5] que permiten diseños incrementales basados en estructuras tan simples y primitivas como el diseñador las

pueda imaginar .

Muchas de las tareas que hoy en día puede desarrollar un robot dentro de un equipo de búsqueda y rescate urbano USAR son bastante peligrosas y que por lo mismo requieren una alta capacidad por parte del prototipo de responder a los estímulos tan rápido como se presente la situación se llevo a cabo el diseño e implementación de un sistema cooperativo enfocado en la búsqueda y localización de fuentes de calor siguiendo el enfoque de Brooks.

2. Metodología

2.1. Descripción de tareas para la búsqueda y localización de una fuente de calor)

La descripción de tareas y técnicas utilizadas por los equipos USAR en zonas colapsadas es el punto de partida del cual se basa la propuesta de este trabajo. En este pequeño estudio existen labores de búsqueda y localización tanto para entornos abiertos como para entornos cerrados y en las cuales se desarrollaban una serie de técnicas características para cada una de ellas. La técnica

de rastillaje ambigua en ambas situaciones es la que se muestra en el desarrollo de esta sección [6].

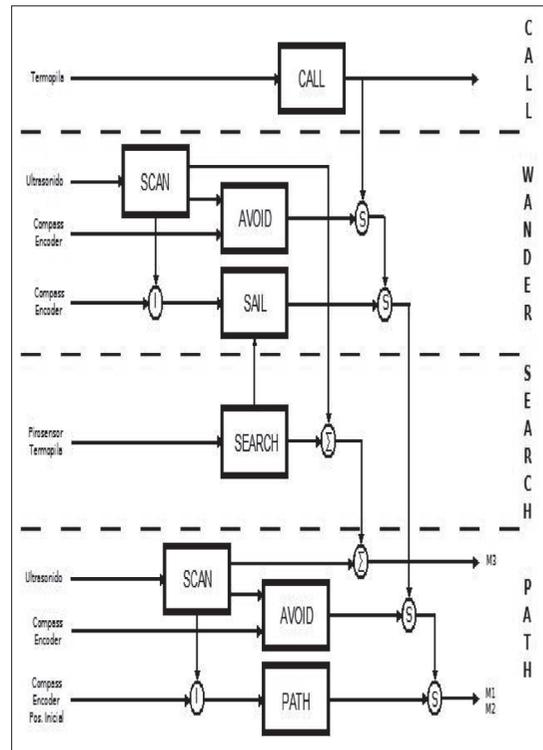
Algunas de las tareas específicas que se realizan en esta técnica son:

- Ubicación del equipo de búsqueda en un punto inicial.
- Enlistamiento de cada uno de los miembros del equipo.
- El líder del equipo ordena a cada uno de los miembros del equipo avanzar en línea recta una determinada distancia.
- Una vez recorrida dicha distancia, cada uno de los miembros realiza un barrido desde su punto.
- Si alguno de los rescatista ubica o localiza una posible víctima notifica al líder informándole la situación.
- De presentarse la situación anterior el liderazgo de la situación es entregado al rescatista que encontró la víctima.
- Luego se dirige libremente hacia la víctima, teniendo en cuidado con cada uno de los obstáculos y escombros que hallan en el camino. Mientras tanto los demás rescatistas mantienen su posición a la espera de nuevas órdenes.
- Una vez localizada la víctima se informa la ubicación a los demás rescatistas.
- Posteriormente cada uno de ellos se dirige hacia a baliza.
- De no encontrarse ninguna víctima se vuelve a realizar el avance previa orden del líder.

2.1. Descripción de los comportamientos reactivos básicos implementados en un robot

La anterior descripción de tareas es la que se ejecuta por parte de un equipo humano de búsqueda y rescate lo que aun para ellos

Figura 1. Descripción de comportamientos en un solo robot



Fuente: elaboración propia.

resulta ser una labor bastante dispendiosa y peligrosa. El desarrollo de comportamientos compatibles con este tipo de situaciones se puede lograr con la integración de tareas simples desde el punto de vista robótico [7] y a través de la integración de cada una de ellas en capas comportamentales con mecanismos de coordinación (supresión e inhibición). Esta descripción se tradujo en la estructura comportamental mostrada en la figura 1 y que está en un solo robot.

El objetivo de la descripción de los comportamientos básicos que serán implementados en el robot es poder establecer un punto de partida para el diseño de las máquinas de estado que serán diseñadas para el desarrollo de todo el proceso de un robot como objeto individual. A continuación se hace una descripción de cada una de las capas comportamentales y sus respectivos comportamientos.

2.2.1. Capa PATH

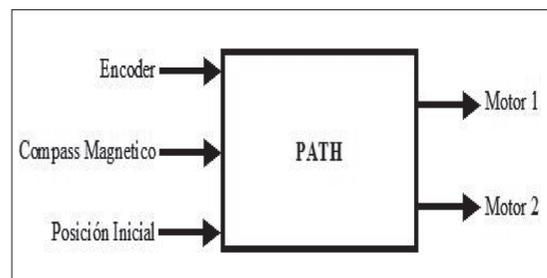
Es la capa que garantiza que el robot siga una trayectoria preestablecida dentro del proceso de búsqueda de víctimas evitando los obstáculos que se le presenten durante su recorrido. Esta capa está conformada por tres comportamientos básicos que son: PATH, SCAN y AVOID.

2.2.1.1. Comportamiento PATH

En el proceso de búsqueda de la víctima, se han definido cuatro movimientos típicos de los rescatistas en zonas colapsadas interiores cumpliendo la normativa internacional [6]. En este caso, el robot deberá seguir una trayectoria que será preestablecida, lo que permite el cumplimiento de una formación básica que con base en la información que es suministrada por los encoders sea implementada en ecuaciones cinemáticas del robot que permita conocer su velocidad lineal y angular.

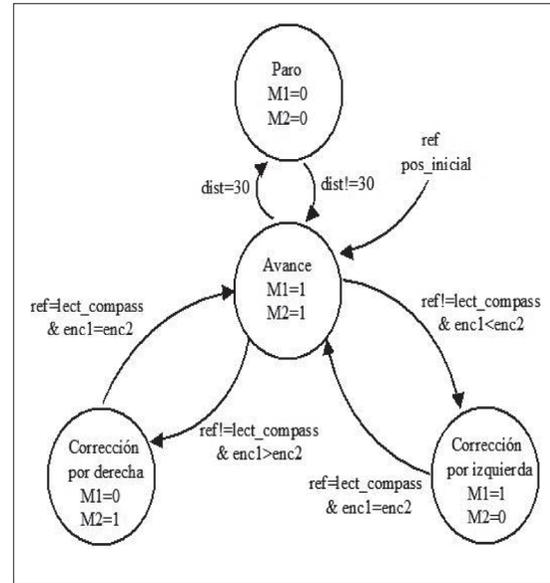
Un compás magnético establece el sistema de referencia del robot necesario para determinar su orientación y la posición inicial que se requiere para la estimación de posiciones y es punto de partida para la formación de todo el sistema multiagente. La salida del comportamiento serán las señales de estímulo a los motores M1 y M2 que deben ser entendidas como el porcentaje del ciclo útil de los PWM de cada motor (figura 2).

Figura 2. Descripción I/O para el comportamiento PATH



Fuente: elaboración propia.

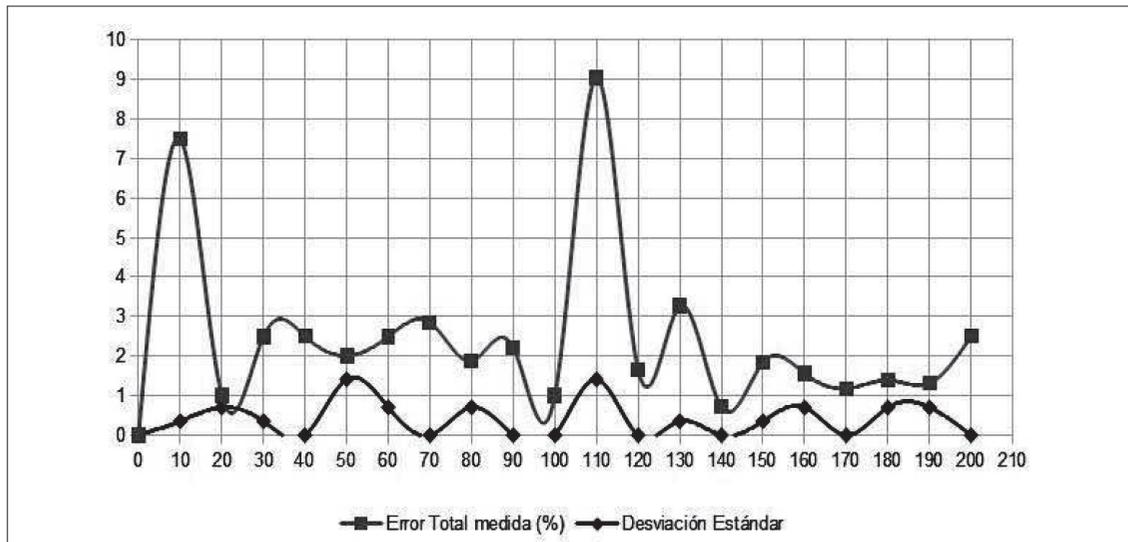
Figura 3. Diagrama de flujo para la descripción del comportamiento PATH



Fuente: elaboración propia.

Para la validación del comportamiento se implementó el algoritmo mostrado en la figura 3 en LabVIEW, en el que se tomaron como entradas al programa la señal de inhibición, el par coordenado (x, y) para predefinir la ruta, así como las coordenadas anteriores del robot, esto con el fin de realizar iteraciones con el robot y la lectura inicial del compás magnético. La información que entrega a la salida está compuesta por la posición (x, y) y el ángulo de orientación del robot, y una señal de tipo booleano que indica la ejecución actual del programa.

Durante el proceso de validación del comportamiento se registraron los valores de distancia recorridos por el robot. Durante la experimentación que consistió en la ejecución de 2 pruebas en la que se registraron los valores de distancia recorrida por el robot. En una trayectoria recta de 2 m se obtuvieron los errores porcentuales para intervalos de 10 cm y la variación en el cambio de distancia en esos mismos intervalos. En la figura 4 se muestra el análisis de las muestra tomadas, en ellas se observa que los

Figura 4. Resultados del proceso de validación para el comportamiento PATH


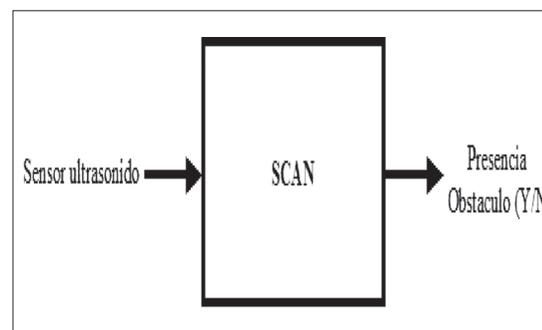
Fuente: elaboración propia.

valores medidos corresponden altamente a las medidas seleccionadas presentando en el mayor de los casos un error porcentual de 9,05 % y las variaciones entre las muestras tomadas de 1,41 %.

2.2.1.2. Comportamiento SCAN

Este comportamiento que tiene como objetivo principal la búsqueda y detección de un obstáculo mientras se realiza el comportamiento *PATH*. Se activa después de realizar un recorrido preestablecido para el robot (para el caso es de 30 cm) y que es una señal enviada por el comportamiento *PATH*. El comportamiento *SCAN* toma la señal del sensor de ultrasonido y determina la distancia y ubicación del obstáculo respecto al robot. A su vez este comportamiento controla el motor M3 (motor dedicado para la implementación de un radar básico). Activa el comportamiento *AVOID* e inhibe el comportamiento *PATH* permitiendo que el robot pueda sortear el obstáculo y seguir la trayectoria una vez sea superado el obstáculo (figura 5).

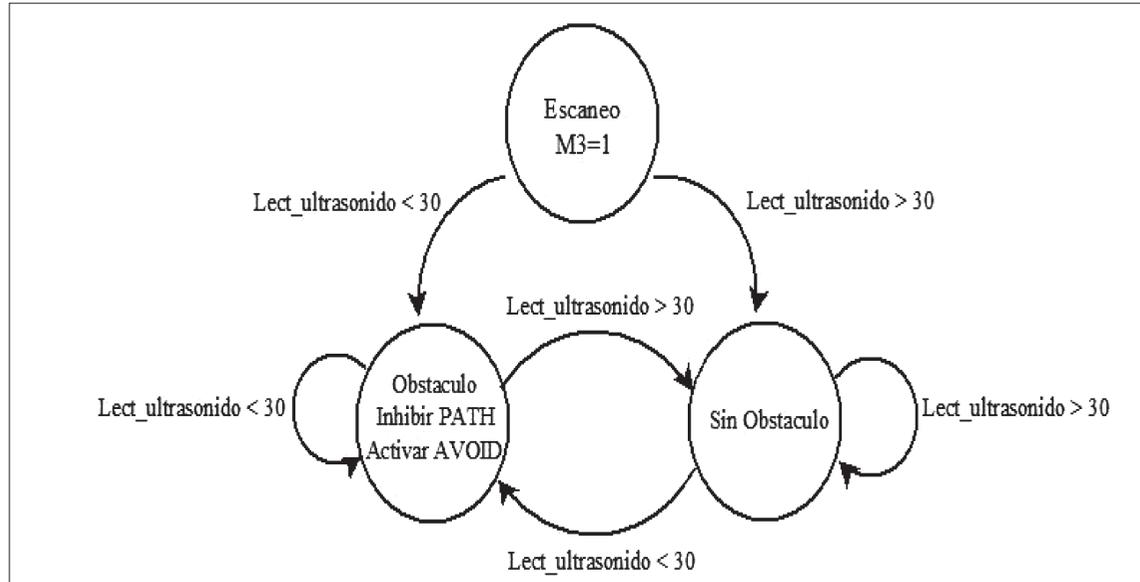
Para la validación del comportamiento se llevó a cabo la implementación de máqui-

Figura 5. Descripción I/O para el comportamiento SCAN


Fuente: elaboración propia.

na de estado mostrado en la figura 6 en LabVIEW en el cual el robot toma muestras con la ayuda del motor M3 en un rango de apertura de ± 90 grados y compara los valores entregados por el sensor de ultrasonido para así determinar la dirección y la distancia del obstáculo. De encontrar obstáculo, el comportamiento genera una señal de tipo booleano que permite la inhibición del comportamiento *PATH* a la vez que genera una señal que permite activar el comportamiento *AVOID* (figura 8).

Figura 6. Diagrama de flujo para la implementación del comportamiento SCAN



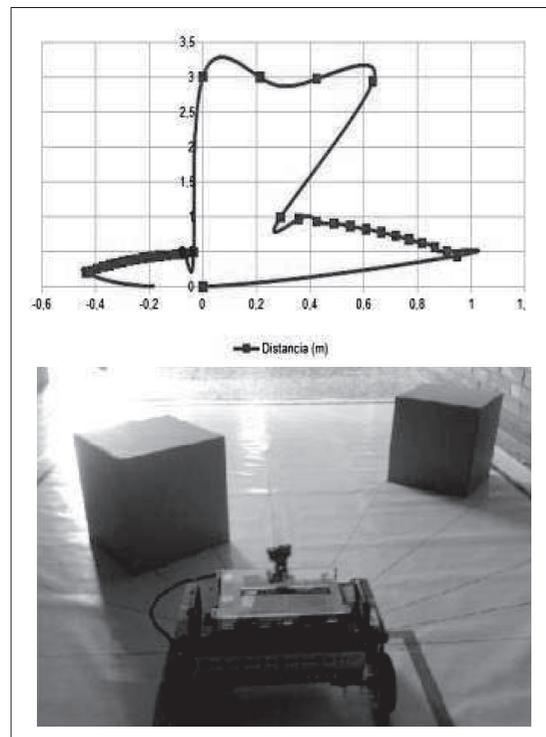
Fuente: elaboración propia.

Durante el proceso de experimentación y validación del comportamiento se programó el robot para que realizara una pequeña reconstrucción de su entorno con la información suministrada por el sensor ultrasonido. Estos registros permitieron conocer el desempeño de este comportamiento frente a distintas situaciones. A la izquierda de la figura 7 se muestra como alrededor de los 50 cm se ubican una serie de puntos y después una zona en blanco, esto representa la superficie de un obstáculo detectado. El mismo análisis para los puntos que se ubican a la derecha de la figura 7; en estos se refleja la existencia de un obstáculo aproximadamente a un metro de distancia.

2.2.1.3. Comportamiento AVOID (t4)

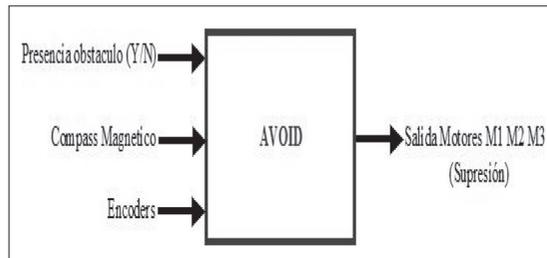
Con la activación dada por el comportamiento SCAN el comportamiento AVOID tiene como objetivo principal evadir el obstáculo, pero sin olvidar que debe regresar a la ruta original. Para hacer esto, este comportamiento suprime el comportamiento PATH y debe calcular el error de su movimiento con respecto a la ruta preestablecida

Figura 7. Experimentación y validación del comportamiento SCAN



Fuente: elaboración propia.

Figura 8. Descripción I/O para el comportamiento AVOID



Fuente: elaboración propia.

para garantizar el retorno a la ruta original. Para este fin utilizó el algoritmo VFH+. Las entradas de este comportamiento son las señales de los encoders y del compás necesarias para calcular las posiciones del robot y sus salidas son estímulos a los motores M1 y M2. Una vez se regrese a la ruta original este comportamiento debe liberar el comportamiento PATH que será el que sigue ejecutándose (figura 8).

Para la implementación de este comportamiento se hizo uso del algoritmo VFH+ [7] el cual está basado en el método de campos de fuerza virtuales, lo cual es una combinación del concepto de los campos de potencial con otro concepto que son las celdas de certidumbre. El método de celdas de certidumbre para la representación de obstáculos permite agregar y retirar datos durante la ejecución y permite una fácil integración de múltiples sensores. Para crear una cuadrícula de certidumbre el área de trabajo del robot se divide en varias celdas con las cuales se forma la cuadrícula general. Cada celda (i,j) contiene un valor de certidumbre $C(i,j)$ que indica el grado de confianza de que un obstáculo se encuentre dentro del área de la celda. Mientras más grande sea el valor de certidumbre $C(i,j)$ mayor será el nivel de confiabilidad de que exista un obstáculo en la celda.

Basado en el concepto de celdas de certidumbre VFH (Vector Field Histogram) [8] el cual trabaja buscando huecos en histogramas polares construidos localmente. Los va-

lores de los histogramas son inversamente proporcionales a las distancias de los obstáculos alrededor del robot. Lo que en el entorno es una ruta libre de obstáculos en el histograma es reflejada como un valle, así el algoritmo funciona tomando el valle más amplio para determinar la nueva dirección que va a seguir el robot.

El algoritmo se diseñó para buscar aberturas que pudiesen representar un espacio libre por el cual el robot tuviese la capacidad de moverse libremente, por lo general, el algoritmo asigna valores o pesos a cada una de las posibles rutas o caminos libres, la que menor valor de peso tenga será la que el robot elige para seguir, se divide en cuatro fases [9], [10]. La primera de ellas realiza la construcción de un primer histograma H^p través de (1). a

$$H_k^p = \sum_{i,j \in C^*} m_{i,j} \cdot h'_{i,j} \quad (1)$$

En donde: k es el valor numérico que representa el sector.

i, j son las coordenadas de la celda activa en la ventana activa de acción C^*

$m_{i,j}$ es el valor que se afecta el movimiento robot y está dado por:

$$m_{i,j} = (c_{i,j}^*)^2 (a - b d_{i,j})$$

En donde:

$c^{i,j}$ es el valor de la celda activa en la posición i, j e influye directamente en $d_{i,j}$

$d_{i,j}$ es la distancia que hay entre la celda activa y el robot.

a, b Son constantes positivas para configurar un valor de cero en celdas inactivas.

El proceso de validación para el comportamiento se realizó con la implementación del algoritmo VFH + en LabVIEW. En las figu-

ras 9 y 10 se evidencia una serie de datos y gráficas que representan la evolución del comportamiento en ejecución a través del tiempo. Inicialmente para entender un poco la terminología de estas gráficas se traducen algunos términos propios de esta, esto con el fin de hacer más asequible el entendimiento de la información suministrada, a saber:

Datos sin procesar de LIDAR (Raw LIDAR): está representado por pequeños cuadros de color negro, se traduce como la proyección de puntos que hace el algoritmo para representar el obstáculo detectado por el robot.

Umbral interior (Inner Threshold): está representado por una línea delgada de color rojo la cual se interpreta como el radio interno efectivo del robot considerando este como el radio crítico de acción del robot y al cual se presentaría un inminente choque con los obstáculos en el entorno.

Umbral exterior (Outer Threshold): está representado por una línea delgada de color verde la cual se interpreta como el radio externo efectivo del robot. Sobre este valor la navegación del robot se considera segura.

Histograma (Histogram): está representado por pequeños cuadros de color azul claro, esta información es como tal el histograma o reconstrucción de zonas libres y zonas ocupadas que realiza el algoritmo. En esta se consideran tanto las proyecciones de los obstáculos como la de los posibles caminos que puede tomar el robot para evadir el obstáculo.

Objetivo (Target): está representado por una línea gruesa de color verde sobre una circunferencia media que denota en frente de robot, cuando esta circunferencia media se torna de color rojo se refiere a la presencia de uno o varios obstáculos sobre la trayectoria del robot.

Comando de dirección (Command Heading): está representado por una línea gruesa de color rojo sobre una circunferencia media; esta denota la dirección en la cual el algoritmo detecta un obstáculo y a la cual no se

dirige el robot. Esta información es útil a la hora de considerar hacia qué dirección no se debe tomar un futuro camino o ruta ya que es altamente probable una colisión.

Rumbo actual (Current Heading): está representado por una línea delgada de color amarillo la cual denota la dirección tomada por el robot sobre el camino. Esta información es útil para escoger la trayectoria escogida por el robot.

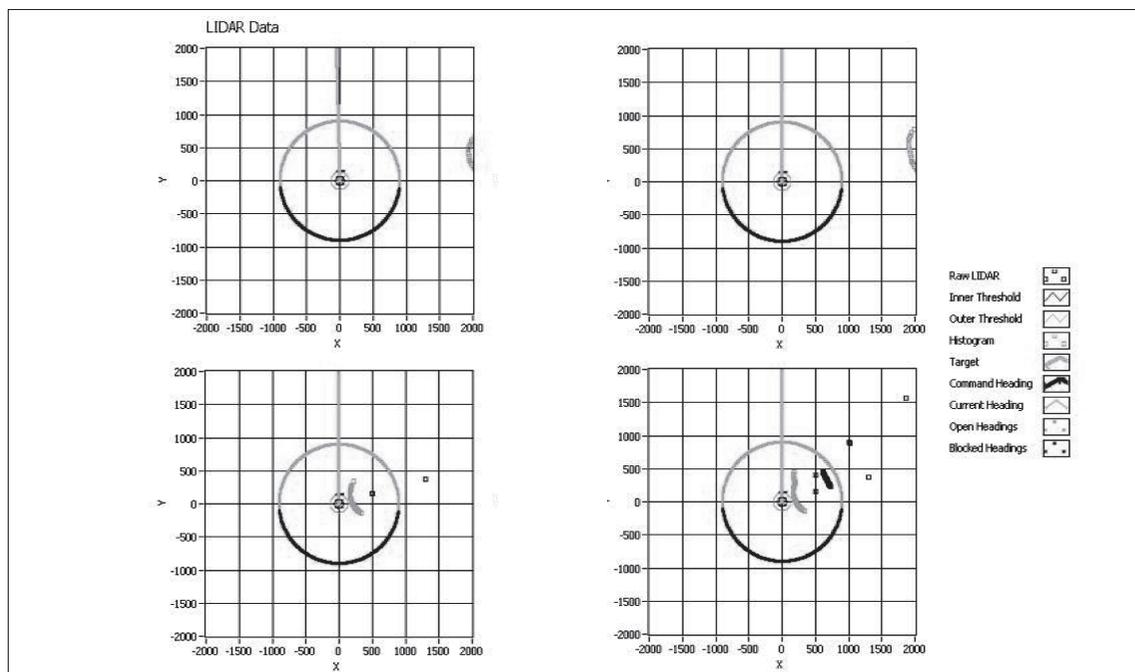
Rumbo despejado (Open Headings): está representado por pequeños cuadros de color verde y se interpretan como aquellas zonas libres a la cuales el robot puede dirigirse sin ningún riesgo de una posible colisión contra un obstáculo.

Rumbo bloqueado (Blocked Headings): está representado por pequeños cuadros de color rojo y se interpretan como aquellas zonas inaccesibles a las cuales el robot no puede dirigirse debido a la inminente posibilidad de una colisión con un obstáculo.

Una vez conocida la terminología respectiva, en la figura 9 se muestra la primera secuencia de imágenes compuesta por cuatro fotografías que retratan la información entregada por el comportamiento al usuario, en el primer cuadro (parte superior izquierda) el robot se encuentra ejecutando una rutina de avance en línea recta sobre un entorno sin obstáculos. En el siguiente cuadro (parte superior derecha) de hace un poco más evidente la información del histograma detectando una posible presencia de obstáculo. En el siguiente cuadro (parte inferior izquierda) se muestra cómo el robot está cada vez más cerca de un obstáculo proyectando sobre la imagen una serie de pequeños cuadros negros. Ya sobre el último cuadro el algoritmo está confirmando la presencia de un obstáculo muy cerca del umbral externo del robot.

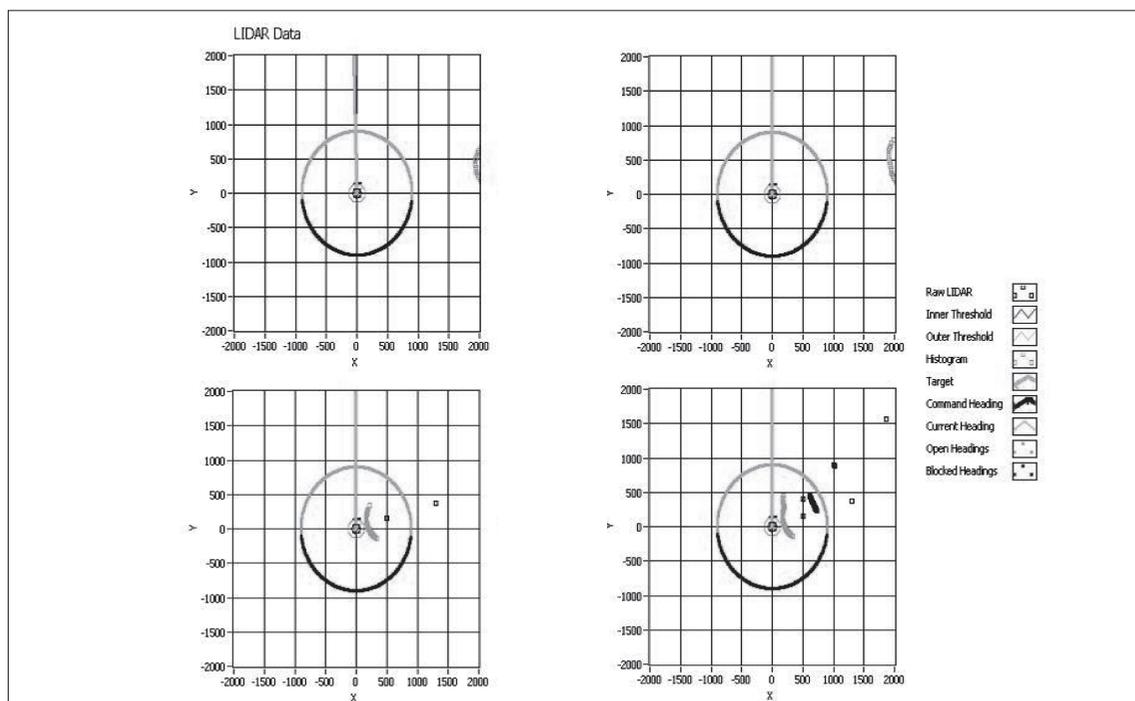
La figura 10 es la continuación de la secuencia de imágenes y en ella se sigue la evolución del comportamiento AVOID sobre el robot. En esta el primer cuadro (parte superior izquierda) muestra en instante el cual la

Figura 9. Resultados del proceso de validación del comportamiento AVOID



Fuente: elaboración propia.

Figura 10. Resultados del proceso de validación del comportamiento AVOID



Fuente: elaboración propia.

información del histograma refleja presencia de un obstáculo sobre el umbral mínimo de detección a lo cual el comportamiento responde con el inicio de la rutina de evasión como tal. En los siguientes cuadros se observa cómo la información del histograma refleja la cercanía con la que el robot evade el obstáculo y cómo mediante esta información reconstruye una silueta aproximada de obstáculo que está superando.

2.2.2. Capa SEARCH

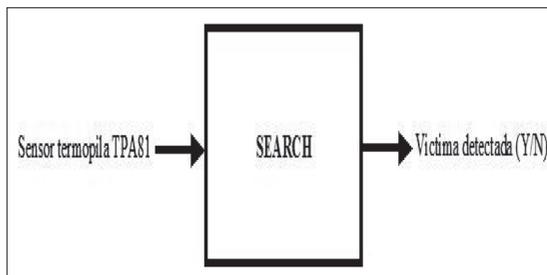
Es la capa que garantiza que el robot efectúe la búsqueda de la víctima dentro de la zona de prueba establecida. Esta capa está conformada por un comportamiento básico que es SEARCH.

2.2.2.1. Comportamiento SEARCH

Este comportamiento tiene como objetivo principal la búsqueda y detección de la víctima que para la aplicación es representada como una fuente de calor. Este comportamiento toma la señal del sensor TPA81 para determinar la dirección hacia donde se encuentra la víctima y la distancia a ella (figura 11). A su vez, permite el control sobre el motor M3 (motor dedicado para la implementación de un radar).

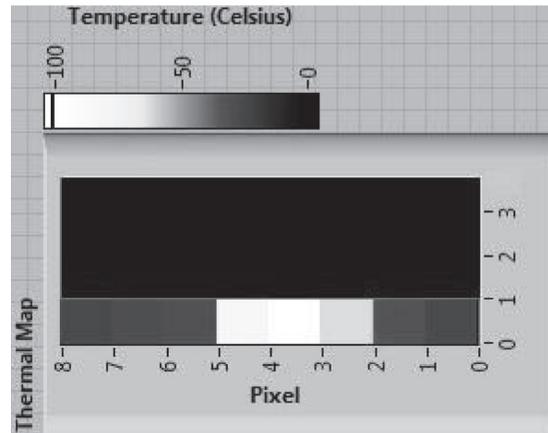
Durante el proceso de validación para el comportamiento implementado en LabVIEW se realizó un barrido de la zona y a través de la lectura del sensor termopila TPA81

Figura 11. Descripción I/O para el comportamiento SEARCH



Fuente: elaboración propia.

Figura 12. Resultado del proceso de validación de comportamiento SEARCH 1



Fuente: elaboración propia.

de la cual se obtuvo una gráfica en la que se refleja tanto la dirección como la intensidad de la fuente de calor. En la figura 12 se muestra un pequeño mapa termal muy sencillo que retrata los aspectos mencionados anteriormente mediante pequeños cuadros sobre el eje vertical, cada uno sobre 9 regiones que representan las zonas activas de detección del robot. Los colores de cada uno de estos cuadros determina el valor de temperatura en cada una de las regiones siguiendo las pautas a continuación mencionadas:

Tonos en escala de negros y rojos: representan temperaturas en el rango de los 0°C y los 35°C.

Tonos en escala de naranjas y amarillos: Representan temperaturas en el rango de los 36°C y los 75°C.

Tonos en escala de blancos: representan temperaturas superiores a los 75°C.

Siguiendo estas definiciones en el mapa termal de la figura 12 se evidencia la presencia de una fuente de calor en las zonas 3, 4 y 5 respectivamente, y si se toma el valor de intensidad dado por la tonalidad de los colores de los cuadros se concluye que la dirección en la que se detecta una fuente de calor

está sobre la región 4 del sensor. Este tipo de información es de valiosa cuantía para el proceso de localización de la fuente de calor ya que de una vez se puede conocer la dirección en la cual el robot debe dirigirse para llegar al punto deseado.

Referencias

- [1] K. H. Guadarrama, *Estrategias de coordinación para la exploración con multi-agentes robóticos*. Universidad Autónoma de Puebla. 2010.
- [2] R. C. Arkin, "Chapter 3. Robot Behavior". In M. Dorigo (Ed.), *Behavior Based Robotics* (pp. 65-120). London.England: The MIT Press. 1998.
- [3] R. Brooks, "A Robust Layered Control System for a mobile robot", *IEEE Journal on Robotics and Automation*, vol. 2, núm. 1: 14-23. 1986.
- [4] R. C. Arkin, "Chapter 4. Behavior Based -Architectures". In M. Dorigo (Ed.), *Behavior Based Robotics* (pp. 123-173). London.England: The MIT Press. 1998.
- [5] S. Bermejo, *Desarrollo de robots basados en el comportamiento* http://www.hernandezsola.com/telecos/sam/Llibre_SAM/EE05400C.pdf (pp. 1-226). 2003.
- [6] <http://www.fema.gov/national-flood-insurance-program/national-flood-insurance-program-community-rating-system> FEMA, www.fema.gov/prole-rescue, 2012.
- [7] L. E. Parker, "Distributed intelligence: Overview of the field and its application in multi-robot systems". *Journal of Physical Agents*, vol. 2, núm. 1: 5-14. 2008. [en línea] disponible en <https://www.aaai.org/Papers/Symposia/Fall/2007/FS-07-06/FS07-06-002.pdf>