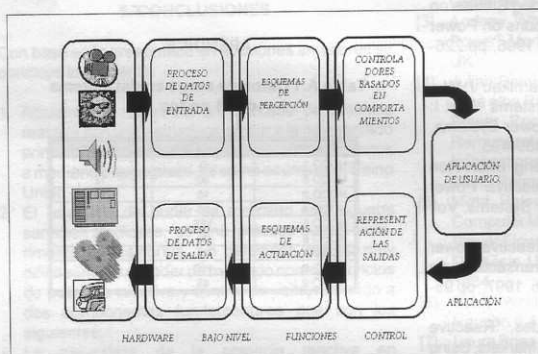


IMPLEMENTACIÓN DE UNA ARQUITECTURA DE CONTROL SOBRE UN ROBOT MÓVIL MODULAR



RESUMEN

Este artículo presentará la implementación de una arquitectura de control [1] orientada a la programación basada en comportamientos [2] sobre un robot móvil modular (Gonzo) [3]. Se incluye un caso de estudio de una aplicación, que basada en un conjunto de comportamientos, permite al robot móvil deambular en un entorno estructurado del cual el robot no tiene previo conocimiento.

PALABRAS CLAVES

Robótica móvil, arquitectura de control, programación basada en comportamientos.

ABSTRACT

A control architecture [5] implementation oriented to behavior-based programming [2] on a mobile robot (Gonzo) [3] is presented. It includes a case of study of an application which, based on a behaviors set, performs a Wandering-behavior on a mobile robot in a structured environment without previous robot knowledge.

KEYWORDS:

Mobile robotics, control architecture, behavior-based programming.

• **BACCA CORTES EVAL BLADIMIR**, Ing. Estudiante de Maestría en Automática, Programa de Posgrado en Ingeniería Eléctrica, Electrónica y de la Computación, Universidad del Valle.

• **CAICEDO BRAVO EDUARDO**, Ph.D. Profesor Titular. Ingeniero Electricista Escuela Ingeniería Eléctrica Electrónica. Universidad del Valle.

Grupo de Investigación PSI, Percepción y Sistemas Inteligentes.

48

• Recibido : Diciembre 2002
• Aceptado: Septiembre 2003

1. INTRODUCCIÓN

El diseño e implementación de aplicaciones sobre plataformas de uno o varios robots debe ser llevado a cabo bajo el amparo de una serie de herramientas y especificaciones que permitan la implementación de una solución que cumpla con los objetivos establecidos. Este conjunto de herramientas, requerimientos y especificaciones lleva al estudio de las arquitecturas de control [4][5], de las cuales, una propuesta fue seleccionada para el presente trabajo y como se verá se adapta muy bien a la plataforma de desarrollo que se tiene en el laboratorio.

Primero que todo, este artículo presenta una resumida introducción sobre los tipos y los objetivos que intenta cumplir una arquitectura de control; luego, se presenta la plataforma de desarrollo que se tiene en el laboratorio; a continuación se describe la arquitectura de control seleccionada y su implementación en la plataforma mencionada; y finalmente se muestran los resultados obtenidos a través de la implementación de un conjunto de comportamientos los cuales producen un comportamiento emergente de "robot errante" o Wandering Behavior.

2. MARCO TEÓRICO

Una arquitectura de control está constituida por un sistema software y un conjunto de especificaciones que proveen herramientas y métodos para la construcción de sistemas basados en robots; ésta define la manera de organizar un sistema de control, sin embargo, además de proveer una estructura, la arquitectura de control impone restricciones en la forma de resolver un problema de control [6][4]; también, se refiere a la manera como una aplicación o sistema es subdividido en subsistemas y como estos subsistemas interactúan [5]. La arquitectura de control como parte esencial de un sistema robot, debe cumplir con ciertos criterios, de los cuales varios pueden llegar a ser antagónicos, sin embargo existen distintas aproximaciones, que se explican más adelante, que permiten un compromiso entre ellos [6]:

- Soporte de paralelismo.
- Buena relación arquitectura-hardware.
- Buena relación arquitectura-ambiente.
- Soporte de modularidad.
- Robustez.
- Tiempo de desarrollo de aplicaciones.
- Flexibilidad en tiempo de ejecución.

Con el fin de cumplir con estos criterios, existen ciertas aproximaciones de las arquitecturas de control [4]: deliberativas, reactivas, híbridas y basadas en comportamientos. La deliberativa emplea un modelo

del entorno el cual se usa extensivamente para producir la mejor secuencia de acciones; las reactivas, por el contrario, enfatizan en la relación estrecha entre las percepciones y acciones, por lo tanto trabajan sin un modelo del entorno; las híbridas, combinan lo mejor de los dos mundos anteriores, típicamente, los procesos de más bajo nivel y más dinámicos son manejados por aproximaciones reactivas, mientras que a más alto nivel se emplean las deliberativas; finalmente, las basadas en comportamientos, trabajan en la relación entre condiciones de entrada y acciones, teniendo de esta forma distintas representaciones del entorno para cada controlador o comportamiento, los cuales se agrupan en paralelo para producir un comportamiento emergente deseado [6][2].

Con esto en mente, existen ciertos requerimientos que se deben tener en cuenta para la plataforma de desarrollo en cuestión:

- Su capacidad de cómputo es reducida.
- Las especificaciones de hardware son modestas.
- Las herramientas de desarrollo están restringidas en el tamaño de código compilado producido.
- Es deseable un adecuado compromiso entre los objetivos antes citados, con el fin de emplear al robot móvil como herramienta de desarrollo en el laboratorio.
- Se está restringido a una metodología de programación estructurada (debido al tipo de herramienta de desarrollo).

Estas restricciones junto con los objetivos mencionados, justifican la selección de la arquitectura de control que se presentará más adelante; además, también justifican la selección de la programación orientada a comportamientos debido a su flexibilidad de concepción, facilidad de relacionarla con la plataforma de desarrollo, su inherente paralelismo, modularidad y robustez basada en la redundancia [6].

3. EL ROBOT

Gonzo, es un robot móvil modular el cual se compone de 3 subsistemas: sensorial, control de movimientos y sensores y control general. Desde el punto de vista de percepción posee 6 sensores de proximidad infrarrojos, 4 sensores de contacto, 4 fotoceldas, 4 micrófonos y la capacidad de recibir mensajes de 8 bits a través de 4 sensores infrarrojos en formato RS-232 [9]; como actuadores posee la capacidad de transmitir mensajes a través de LEDs infrarrojos en formato RS-232, tanto hacia delante solamente o en todas las direcciones y por último el robot posee un sistema de desplazamiento diferencial impulsado por dos servomotores S3003 modificados acoplados a encoders incrementales. El

subsistema de control de sensores y movimiento se encarga de ejecutar los algoritmos de sensado, de controlar la velocidad de desplazamiento, de estimar la posición y orientación y de comunicarse con el sistema central para la actualización de los datos o la ejecución de comandos de movimiento. Por último el sistema central se encarga de interactuar con el PC y la aplicación de usuario con el fin de descargar el programa y 48kb de RAM) a través de un enlace serie RS-232. Cabe resaltar que el robot es autónomo, posee un banco de 8 pilas AA [3].

4. LA ARQUITECTURA DE CONTROL

La arquitectura de control propuesta por [1] se resume gráficamente en la figura 1.

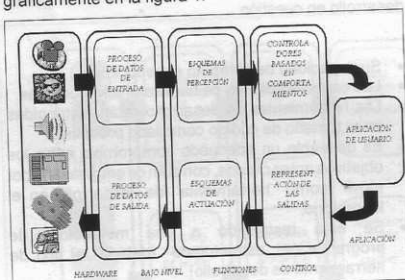


Figura 1. Arquitectura de control

4.1 Breve descripción de los componentes

La figura 1 muestra 4 capas básicas, al nivel más bajo se encuentran un conjunto de manejadores tanto de entrada como de salida que se encargan de operar el hardware del sistema, estos manejadores de los dispositivos son dependientes del hardware. Seguidamente, las funciones del sistema, las cuales pueden ser de percepción como de actuación. Los esquemas de percepción [8] (actuación) son diseñados con el fin de depender del tipo de sensor(actuador) y de esta manera proveer entradas(salidas) adecuadas a los controladores o comportamientos. A nivel de controladores se encuentran un conjunto de comportamientos (rama de entrada) que emplean los esquemas de percepción para tomar decisiones y lograr los objetivos deseados; además, se encuentra un módulo de representación que permite construir entradas adecuadas a los esquemas de actuación e incluso realizar actividades de "fisión" de actuación [1]. Al nivel más alto se encuentra la aplicación que reunirá dos o más conjuntos de comportamientos (controladores) con el fin de lograr un comportamiento emergente deseado [6][2]. Nótese

que no existe restricción alguna en el emplear desde el más alto nivel funciones propias de nivel bajo, pero no al revés.

4.2 Implementación

A bajo nivel, se distinguen el manejo de sensores y actuadores, las tablas 1 y 2 resumen las utilidades de cada uno respectivamente.

Tabla 1. Manejadores de bajo nivel para los sensores del robot.

Dispositivo	Descripción del manejador
Infrarrojos	Obtención proximidad óptica.
Infrarrojos	Recepción de mensajes.
Contacto	Obtención proximidad por contacto.
Microfonos	Estado de 1 o todos los microfonos.
Fotoceldas	Estado de 1 o todas las fotoceldas.
Motores	Obtención de una medida de las corrientes en los motores.
Encoders	Posición y orientación actual.
Encoders	Velocidad lineal de las ruedas.
Baterías	Alarma de baterías bajas.
--	Utilidades generales (inicialización, conversión A/D de 8 bits, transmisión y recepción de datos serie).

Tabla 2. Manejadores de bajo nivel para los actuadores del robot.

Dispositivo	Descripción del manejador
Motores	Velocidades lineales de las ruedas.
Motores	Valores PWM para cada rueda.
Infrarrojos	Transmisión de un mensaje deseado.

A nivel funcional, se distinguen esquemas de percepción y actuación, resumidos en las tablas 3 y 4 respectivamente.

Tabla 3. Esquemas de percepción

Sensor	Descripción del esquema
Comunicaciones	Filtro para la recepción de un determinado mensaje.
Comunicaciones	Detección de proximidad de robots.
Luz o sonido	Presencia de estímulos.
Proximidad	Proximidad de robot en seguimiento.
Luz o sonido	Detección de la posición objetivo.
Comunicaciones	Detección indirecta de la posición objetivo.
--	Generación de orientaciones aleatorias.
Luz o sonido	Detección de la orientación de la posición objetivo.
Proximidad	Evasión de obstáculos a través de sensores infrarrojos.

Continuación

Proximidad	Evasión de obstáculos a través de sensores de contacto.
--	Utilidades generales (inicialización, rotación de ejes, normalización de vectores, potencias de 2, conversiones de binario a vector, inicialización de arreglos, etc).

Tabla 4. Esquemas de Actuación

Actuación	Descripción del esquema
Movimiento	Velocidad lineal y angular basadas en restricciones cinemáticas.
Movimiento	Desplazamiento hacia delante x cm.
Movimiento	Desplazamiento hacia atrás x cm.
Movimiento	Giro de x grados.

Por último, a nivel de controladores se tiene implementada un conjunto de comportamientos que usa los esquemas de percepción antes descritos, sin embargo, el módulo de representación no se ha implementado para esta plataforma de desarrollo. En la tabla 5 se describe el resumen de controladores.

Tabla 5. Conjunto básico de comportamientos.

Nombre	Descripción
Evasión de robots	Orienta al robot hacia un sector libre de robots.
Escape	Evade obstáculos no detectados por los infrarrojos.
Homing (básico)	Orienta al robot hacia la fuente de un estímulo.
Encontrar	Detecta la presencia de un determinado robot alrededor.
Alcance de posición objetivo (directo).	Detiene al robot en el momento de alcanzar la posición objetivo.
Alcance de posición objetivo (indirecto).	Detiene al robot en el momento de alcanzar la posición objetivo.
Orientación aleatoria	Genera una orientación aleatoria para el robot.
Evasión de obstáculos.	Evade obstáculos detectados a través de los infrarrojos.
--	Utilidades generales (definición del tiempo de muestreo, adición de vectores, adición de ruido blanco a un vector y funciones booleanas de uso específico).

Las tablas anteriores contienen un resumen de las diferentes herramientas y especificaciones implementadas, sin embargo, existen herramientas de computo tales como funciones de uso general para la

actualización de información, configuración de periféricos del procesador principal y macros que facilitan la programación cuyos valores responden a las restricciones sensoriales, cinemáticas y geométricas del robot.

5. CASO DE ESTUDIO, "ROBOT ERRANTE"

Desde el punto de vista práctico, esta sección pretende mostrar el empleo de la arquitectura de control para el diseño e implementación de una aplicación específica.

5.1 Descripción del experimento

"Robot Errante" o Wandering Robot es una aplicación típica en robótica, cuyo objetivo es hacer que el robot vague en un ambiente estructurado, del cual el robot no tiene conocimiento previo, evadiendo los obstáculos y explorando la mayor cantidad de lugares dentro de este ambiente. Particularmente, se ha dispuesto un campo de trabajo que se muestra en la figura 2, donde cada cuadro posee un lado de 30cm, el diámetro aproximado del robot es de 21cm y los obstáculos son cajas dispuestas sobre el piso del laboratorio.

5.2 La aplicación

Esta aplicación fue diseñada teniendo en cuenta la aproximación de la programación orientada a comportamientos [2], para la cual existen varias formas de expresar los comportamientos, a saber, diagramas de estímulos-respuesta, máquinas de estado finitas, notaciones funcionales, etc [6][2]. En este artículo se empleará para esta aplicación un diagrama de estímulo-respuesta el cual se muestra en la figura 3.

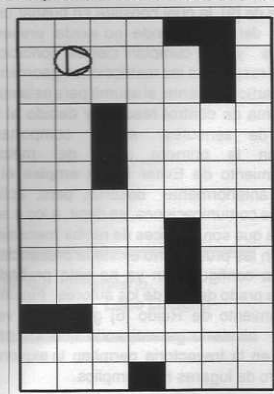


Figura 2. Ambiente de trabajo para la aplicación

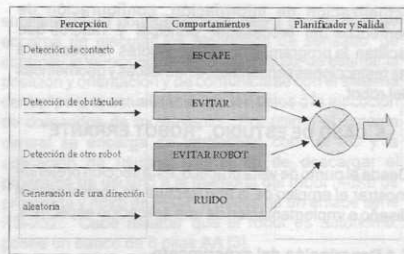


Figura 3. Diagrama estímulo-respuesta de la aplicación de "robot errante"

Se observa que los comportamientos empleados son los descritos, abreviadamente, en la arquitectura de control. En la programación orientada a comportamientos existe el problema de definir que comportamiento posee la salida más adecuada dada una situación específica, sin embargo no existe una respuesta fácil a este interrogante, por lo tanto el esquema seguido aquí es el propuesto por [8] y que consiste en realizar una suma ponderada de las salidas de cada comportamiento y este resultado es aplicado a los actuadores. Ahora, para esta aplicación se ha optado que cada comportamiento arroje una orientación como salida, esta orientación se representa por un vector normalizado y referido al plano coordenado del robot. Esta salida es multiplicada por el valor de prioridad asignado de forma estática, se suman las contribuciones, se normaliza el resultado y se actúa en base al mismo. La estrategia de control de los comportamientos de Escape y Evitar se basa en una simplificación de la propuesta de [9], la cual consiste en buscar un sector alrededor del robot donde no exista presencia de obstáculos y se cumplan ciertas condiciones de seguridad respecto a las restricciones geométricas del mismo. Particularmente, al asumir para esta aplicación un esquema de control reactivo y debido al limitado número de sensores estos comportamientos desarrollan la primera parte del método. El comportamiento de Evitar Robot emplea el mismo modelo anteriormente descrito, pero, aplicado al sistema de comunicaciones, es decir, a los 4 sensores infrarrojos que son capaces de recibir mensajes serie, aunque en las pruebas no existe la presencia de otro robot esta configuración ya ha sido probada en el trabajo de grado de uno de los autores. Finalmente, el comportamiento de Ruido [6] genera un vector de dirección aleatorio garantizando que los pequeños cambios en la trayectoria permitan la exploración de un número de lugares más amplios.

5.3 Análisis de resultados

La implementación de aplicaciones basada en la programación orientada a comportamientos empleado una aproximación reactiva posee ciertos inconvenientes, entre ellos, que no existe una adecuada forma basada en los objetivos finales de la aplicación, restricciones geométricas y cinemáticas del robot para poder escoger un conjunto de prioridades, esto se revela en el vector de salida arrojado por la aplicación en el momento de la activación de ciertos sensores en particular; la suma ponderada es una buena ayuda, sin embargo no lo resuelve del todo. Debido a que se ha tomado una aproximación vectorial para el modelado de los sensores según [9], este tipo de aproximación (en esta aplicación) tiende a presentar oscilaciones en la presencia de corredores estrechos y una trayectoria menos oscilatoria para corredores amplios cuyas paredes estén en el rango de acción de los sensores de proximidad, tal y como se reporta en [6].

6. CONCLUSIÓN

La implementación de una arquitectura de control en el robot móvil modular Gonzo, ha resultado en una mejora considerable del tiempo de desarrollo de una aplicación debido a la organización de la información y herramientas, además, permite que el usuario se concentre en el desarrollo de la aplicación como tal y no distraiga su esfuerzo en la solución de problemas de implementación ajenos al objetivo de ésta. Aprovechando la modularidad, desde el punto de vista de hardware que posee la plataforma en cuestión y la implementación de la arquitectura de control descrita en este artículo, es más fácil la ampliación sensorial o de actuación de Gonzo. Finalmente, con la arquitectura de control que opera sobre Gonzo, es factible la implementación de otros esquemas de control basados en comportamientos más sofisticados, próximos al campo deliberativo con el fin de obtener soluciones de tipo global más que local, como son las obtenidas hasta el momento.

7. REFERENCIAS BIBLIOGRÁFICAS

- [8] MINGUEZ J., Montano L., Nearness Diagram Navigation: A New Real Time Collision Avoidance Approach, Computer science and systems engineering, University of Zaragoza, 2000.
- [7] ARKIN C. Ronald, Motor Schema-Based Mobile Robot Navigation, International journal of robotics research, Vol 8, No. 4, pp 92-112 1987.
- [6] ARKIN C. Ronald, Behavior-Based Robotics, second printing, MIT press 1998.
- [5] COSTE E., Simmons R., Architecture, the Backbone of Robotics Systems, Proceedings of the 2000 IEEE international conference on robotics

- and automation, Abril 2000.
- [4] MATARIC Maja, Behavior-Based Control: Main Properties and Implications, Proceedings of workshop on intelligent control systems, International conference on robotics and automation, Nice, France, May 1992.
 - [3] BACCA E., Paz Eddy, "Diseño e Implementación de un Robot Móvil Modular", Universidad del Valle Tesis de grado para optar el título de Ingeniero Electrónico, 1999.
 - [2] BROOKS Rodney, A Robust Layered Control System for a Mobile Robot, IEEE Journal of robotics and Automation, Vol RA-2 No. 1 pp 15-23 1986.
 - [1] SALIDO J., Dolan J., Hampshire J., Khosla P., A Modified Reactive Control Framework for Cooperative Mobile Robots, The robotics institute, Carnegie Mellon University, Pittsburgh, 1997.

AUTORES



BACCA CORTES EVAL BLADIMIR, Ingeniero Electrónico de la Universidad del Valle, actualmente se encuentra realizando la Maestría en Automática en la misma entidad.
evbacca@eiee.univalle.edu.co



CAICEDO BRAVO EDUARDO Ph.D., Ingeniero Electricista de la Universidad del Valle, actualmente es profesor titular de esta entidad y director de la Escuela de Ingeniería Eléctrica y Electrónica.
ecaicedo@eiee.univalle.edu.co

Errar es humano; abstenerse de reír, humanitario.

Lane Olingho