

Avances en el Desarrollo y Construcción del Robot de Servicio SerBot II

Camberos Hernández Alan Ramón, Tovar García Carlos Antonio, Medrano Aguilar José, Jesús, Arriaga García Andrea Lizeth, Avilés Arriaga Héctor Hugo

Universidad Politécnica de Victoria
Av. Nuevas Tecnologías 5902
Parque Científico y Tecnológico de Tamaulipas
Carretera Victoria – Soto la Marina Km 5.5
Cd. Victoria Tamaulipas, México C.P. 87138

Resumen

El desarrollo de robots de servicio capaces de apoyar a los seres humanos en tareas cotidianas ha despertado gran interés en los últimos años. Sin embargo, el costo económico para adquirir sistemas robóticos comerciales suele ser prohibitivo para muchas Universidades en el país, lo cual limita la capacidad de investigación y desarrollo en este tipo de robots. Por tanto, en este documento se propone el desarrollo del robot de servicio SerBot II de bajo costo y de propósito general con capacidades de navegación, percepción visual del ambiente, interacción por voz y manipulación de objetos tomando en cuenta las reglas de la competencia RoboCup@Home. Se presentan el diseño 3D, los avances en su construcción y de la implementación para las tareas “Presentación” y “Sígueme”. Los experimentos preliminares muestran la factibilidad de la propuesta general y la efectividad de nuestro robot para presentarse y seguir a un usuario en un ambiente real de oficina con resultados prometedores.

Palabras clave: robots de servicio, navegación, percepción visual, interacción natural humano-robot.

1. Introducción

El desarrollo de robots de servicio autónomos ha recibido considerable atención en la última década. Para realizar efectiva y eficientemente sus tareas estos robots deben incluir una serie de capacidades perceptuales y de comportamiento como son: a) localización y navegación, b) percepción visual c) interacción verbal (reconocimiento y síntesis de voz) y d) manipulación de objetos en el medio ambiente.

Para hacer más eficiente el desarrollo de estas capacidades y puesta en operación de los robots es común preferir bases robóticas comerciales [1,2]. No obstante, esta estrategia presenta algunas desventajas: i) la inversión inicial y de mantenimiento resultan prohibitivas para muchas instituciones del país, ii) genera una dependencia tecnológica hacia las compañías comercializadoras y iii) dificulta la adaptación de los robots originales a nuevas necesidades. En México pocas instituciones han abordado el diseño y construcción física de estos robots en conjunto con el desarrollo de las habilidades necesarias. Aunque la curva de aprendizaje aumenta en la etapa de arranque, lo anterior permite considerar las actividades del robot en etapas tempranas del diseño y puede reducir los costos de adquisición del equipo y de mantenimiento. La consecuencia directa es la disponibilidad de robots accesibles a un mayor número de instituciones y estudiantes en el país.

Por lo anterior, en este documento se presenta el avance en la construcción de SerBot II, un robot de servicio de propósito general y de bajo costo. El objetivo de este robot es realizar actividades que requieran navegación, percepción visual del ambiente, interacción por voz y manipulación de objetos. El diseño del robot se realizó mediante herramientas de modelado 3D tomando como referencia las tareas definidas en el concurso RoboCup@Home [3]. Las capacidades del robot se implementaron usando librerías de software libre. Los resultados preliminares con las pruebas “Presentación” y “Sígueme” muestran la factibilidad de la propuesta general y la efectividad de nuestro robot para realizar tareas simples como presentarse y seguir a un usuario en un ambiente de oficina con resultados prometedores.

2. Trabajos Relacionados

En la industria pueden encontrarse diversas bases robóticas comercializadas internacionalmente [4,5]. Un ejemplo de robot de servicio construido en México a partir de estructuras de este tipo es Golem II+ del IIMAS-UNAM [1]. Este robot cuenta con capacidades avanzadas de interacción por voz y gestos y de localización de usuarios por sonido. El control está dirigido mediante un modelo de diálogo y por contexto. Otro ejemplo es Sabina del INAOE [2] que posee habilidades de localización global y local, navegación y evasión, reconocimiento de personas y de gestos. Las acciones se definen por procesos de decisión de Markov. Aunque los robots mencionados anteriormente poseen las habilidades esenciales requeridas en la competencia @Home u otras características deseables como el uso de software libre, el diseño y desarrollo de bases robóticas propias proporciona ventajas como la colaboración natural de grupos multidisciplinarios, la capacitación de profesores y estudiantes y la simplificación del mantenimiento preventivo y correctivo de los robots. Un ejemplo es Donaxi de la UPAEP [6] capaz de detectar personas en el suelo por caídas o accidentes y de transportar un usuario sentado. Por otro lado, Justina de la FIC-UNAM [7] hace énfasis en la interacción efectiva por lenguaje natural con sus usuarios, reconocimiento de personas por ropa y rostro, navegación y reconocimiento y manipulación de objetos.

3. Metodología de desarrollo

En esta sección se presentan el proceso de modelado 3D del robot, los avances en su construcción, las capacidades implementadas hasta el momento y las arquitecturas software y hardware propuestas.

3.1 Diseño 3D

Una vista frontal del diseño 3D en SolidWorks de la base del robot sin dispositivos periféricos se muestra en la figura 1a. El diseño corresponde a una base móvil diferencial de 2 ruedas (de 27 cm de diámetro) con motores y una rueda de castor, para ambientes de interiores. El modelo incluye un mástil a manera de torso con dos brazos manipuladores de 6 grados de libertad (ambos brazos se encuentran actualmente en manufactura). La altura del robot y la longitud de brazos se eligieron considerando proporciones antropométricas [8]. Se considera además un soporte ajustable para la colocación de una computadora laptop en la parte inferior trasera.

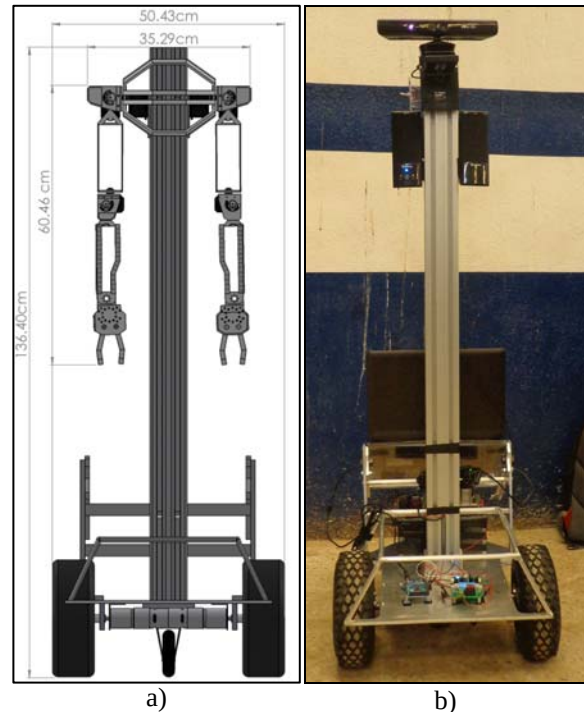


Fig. 1. Imágenes del desarrollo de SerBot II: a) diseño 3D y b) avance actual del robot real.

En la figura 1b se muestra el avance del robot real hasta el momento. La estructura está conformada de aluminio y acrílico. Los dispositivos principales son una laptop ASUS iCore 7, 6GB RAM, dos motorreductores EMG49, una tarjeta controladora MD49, una tarjeta Arduino Mega, 4 sensores infrarojos (10-80cm), un MS Kinect y bocinas USB. Además, el robot cuenta con una tarjeta de alimentación que distribuye la energía de 3 baterías de 12V-7Ah al sistema.

3.2 Capacidades del robot

3.2.1. Navegación por odometría

Para desplazarse un robot debe conocer su posición en el ambiente (frecuentemente usando una representación 2D del mundo). Una de las estrategias más utilizadas es la estimación de posición (x , y , θ) por odometría de acuerdo a las lecturas de encoders en los motores. La velocidad de giro de nuestros motores es especificada por valores enteros de -127 (velocidad máxima en reversa) a 127 (velocidad máxima hacia adelante). Para determinar el *número de revoluciones* por segundo de la flecha del motor, se dividió el *número de pulsos* generados por el encoder en un segundo entre el número de pulsos máximo del encoder por giro completo de la flecha

(en nuestro caso, 980). Este cálculo se describe mediante la siguiente fórmula:

$$\frac{\text{Núm. de revoluciones}}{\text{segundo}} = \frac{\frac{\text{Núm. de pulsos}}{1 \text{ Seg}}}{980} \quad (1)$$

$$\frac{\text{Núm. de revoluciones}}{\text{segundo}} = \frac{\text{Núm. de pulsos}}{980 \text{ 1 revolución}}$$

La Figura 2 muestra los resultados de una serie de repeticiones del procedimiento anterior aplicando valores de 0 a 127 a ambos motores. La pendiente m de esta línea es 0.01558. A partir de estos datos, el tamaño de la rueda y las fórmulas descritas en [9] se estima la posición de nuestro robot. Además, con m es posible establecer la correspondencia entre las velocidades lineal o angular requeridas en la tarea y las velocidades de motor que se deben aplicar.

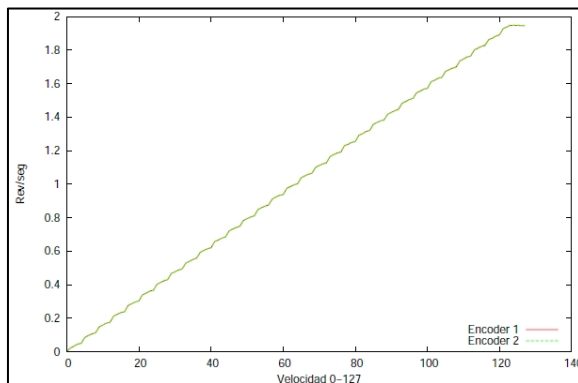


Fig. 2. Número de revoluciones por segundo de los motores a diferentes velocidades.

Actualmente nuestro robot es capaz de realizar movimientos lineales y giros sobre su propio eje. A partir de esto se desarrolló un módulo simple de navegación que funciona como sigue. Al iniciar su operación, la posición del robot se supone (0, 0, 0). Esta posición se actualiza de acuerdo al movimiento del robot. Para alcanzar una posición final (x_f, y_f, θ_f) , el robot gira para orientarse a la coordenada (x_f, y_f) y posteriormente ejecuta un movimiento hacia adelante en línea recta. Una vez en (x_f, y_f) , el robot hace un segundo giro para alcanzar la orientación θ_f solicitada.

3.2.2. Habilidades de interacción por voz

SerBot II cuenta con síntesis y reconocimiento de voz en inglés. Para generar la voz del robot se utiliza sintetizador *Festival*. El reconocimiento de voz usa la herramienta CMU *PocketSphinx*. La interpretación de la voz se realiza

identificando las palabras que son relevantes a un contexto de la interacción usuario-robot de manera similar a [2]. En los robots de servicio es común instalar uno o más micrófonos direccionales para capturar la voz del usuario. En nuestro caso, hemos desarrollado una interfaz entre las librerías *libfreenect* y el *PocketSphinx* para aprovechar los micrófonos disponibles en el MS Kinect.

3.2.3. Seguimiento visual

El seguimiento visual utiliza los datos de profundidad obtenidos del MS Kinect para estimar la distancia y posición de un usuario con respecto al robot en ambientes de interiores. El sistema transforma la imagen “cruda” de profundidad a una imagen de color RGB asignando a cada pixel uno de tres colores posibles de acuerdo a su profundidad. La Tabla 1 muestra intervalos considerados, sus valores de profundidad “crudos”, la distancia que cubren y su color correspondiente.

Intervalo	Valores “crudos” de profundidad	Distancia cubierta (cm)	Color
Intervalo 1	[350,653]	[50, 80]	Rojo
Intervalo 2	(653,772]	(80, 110]	Morado
Intervalo 3	(772,841]	(110, 140]	Amarillo

Tabla 1. Intervalos de profundidad considerados por el sistema de seguimiento usando el MS Kinect.

La relación entre los valores de profundidad y distancia real del objeto se obtuvieron por experimentación. Los valores de profundidad menores de 50cm y mayores a 1.4m no son considerados. La imagen RGB resultante es segmentada por regiones homogéneas de color usando un algoritmo simple de crecimiento de regiones [10]. La posición de la persona es determinada por la posición de la ventana de atención en la imagen. La distancia al usuario se establece por el color predominante en esta ventana de atención. La Figura 3 muestra un ejemplo de la aplicación de este sistema. Los movimientos del robot son definidos por reglas simples de acuerdo a la distancia y a la posición del usuario.

3.3 Arquitectura de software y hardware

La arquitectura de software del robot se describe gráficamente en la figura 4. El nivel superior

corresponde al módulo de coordinación general que define las acciones a ejecutar por medio de procesos de decisión de Markov [11].



Fig. 3. Ejemplo de la ejecución del sistema visual de seguimiento: a) imagen original, b) segmentación y localización del usuario.

Estos modelos permiten calcular una política de acciones (o capacidades) para completar una actividad como traer un vaso de agua o limpiar una mesa. El cálculo de odometría, la interacción por voz y el seguimiento visual están organizados en los módulos de navegación, de interacción y de visión, respectivamente. Estos conforman la segunda capa de la arquitectura. El tercer nivel lo componen las librerías utilizadas para la codificar los módulos. Finalmente, en el último nivel se encuentra el *firmware* del robot que provee acceso a los actuadores y sensores. Este firmware está basado en el *driver p2os* del servidor Player. El sistema está codificado en C/C++ en el sistema operativo Ubuntu 12.04. La comunicación entre módulos se realiza mediante *sockets* TCP/IP.

El flujo de operación de nuestra arquitectura es principalmente secuencial y se describe a continuación. Las decisiones de acción son tomadas por el módulo de coordinación. Cada módulo de capacidades en el segundo nivel recibe una acción específica (como detectar un rostro o navegar) de acuerdo a la actividad. Dicha instrucción es ejecutada por los actuadores a través de las librerías definidas en la tercera capa. Finalmente, los resultados de la acción son reportados al coordinador con lo cual se genera una nueva acción para el robot hasta completar la tarea.

La organización del hardware del robot se presenta gráficamente en la figura 5. Excepto por el firmware del robot que se ejecuta en la tarjeta Arduino, el resto del procesamiento de los módulos descritos en la sección anterior se realiza en la laptop del robot.

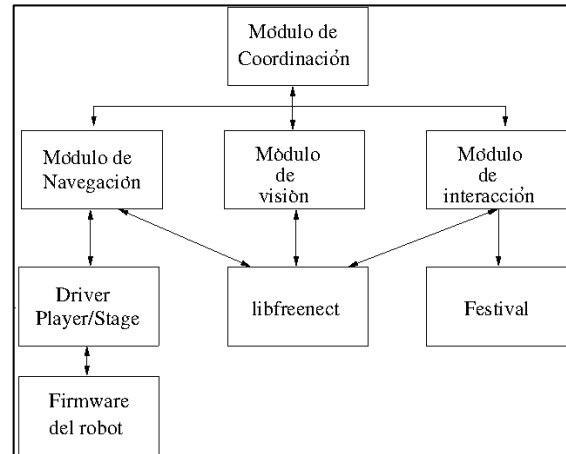


Fig. 4. Arquitectura de software.

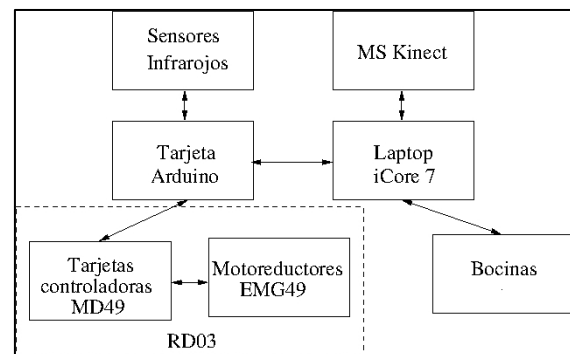


Fig. 5. Comunicación del hardware del robot.

4. Experimentos y resultados

Para evaluar nuestra aproximación se realizaron un conjunto de experimentos preliminares de navegación por odometría, de reconocimiento de voz y de las pruebas “Sígueme” y “Presentación”. Estos experimentos y sus resultados se describen brevemente a continuación.

4.1 Cálculo de odometría y navegación

Para evaluar la odometría y la navegación se definió una ruta cuadrada unidireccional de 3×3m en un piso plano sin obstáculos a una velocidad lineal de 15cm/seg y velocidad angular de 10°/seg. El robot inicia en (0,0,0), moviéndose de izquierda a derecha y girando 90° en dirección contraria a las manecillas del reloj. La figura 6 muestra el cálculo de la posición en 5 repeticiones continuas de esta ruta. La posición real del robot marcada manualmente se muestra en la figura 7. Como puede observarse, el error odométrico

es considerable hacia el final de la prueba pero es consistente en su dirección. Es bien sabido que este error puede deberse al tipo de superficie, su inclinación, al diámetro de las ruedas, derrapes, el retraso de comunicación entre módulos, etc. Para identificar las posibles fuentes de error se requieren experimentos posteriores con otros recorridos sugeridos en la literatura y se añadirá un sistema de control para regular la velocidad del robot cuando se acerque a su objetivo [12].

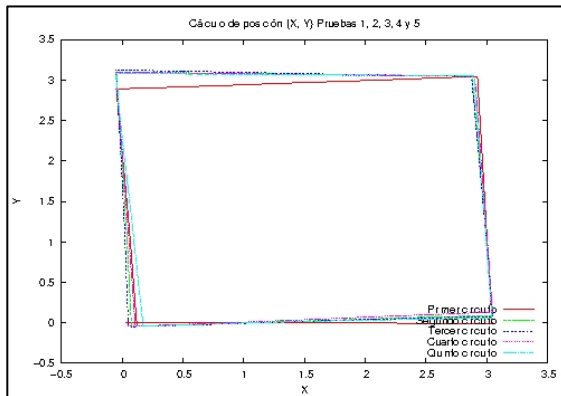


Fig. 6. Cálculo de odometría del robot. Las coordenadas están dadas en metros.

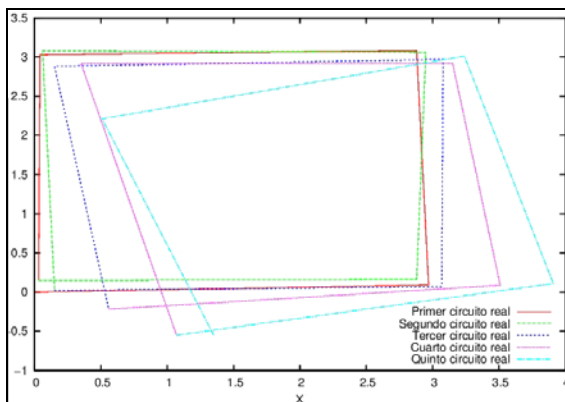


Fig. 7. Posición real del robot en el experimento de seguimiento unidireccional de una ruta cuadrada.

4.2 Reconocimiento de voz usando MS Kinect

Para probar el reconocimiento de voz se realizó un experimento informal con cuatro estudiantes de la Universidad. A cada uno de ellos se le asignó 5 palabras diferentes en inglés del tipo “sí”, “no”, “cocina”, “recámara”, o “ve”. Los participantes

repetieron las 5 palabras frente al MS Kinect, a 45°, 90° y 135° (medidos de izquierda a derecha del usuario frente al Kinect) y a una distancia de un metro, en un ambiente de laboratorio. Para el reconocimiento se utilizó únicamente el micrófono número 1 del Kinect (el primero de izquierda a derecha frente al dispositivo). Los mejores porcentajes de clasificación, que en dos de los casos fueron de 100%, se obtuvieron a 45° del dispositivo (i.e., en la posición más cercana al micrófono utilizado). También se observó que el volumen de la voz y la distancia del usuario afectan al reconocimiento. Los autores consideramos que es posible incrementar la eficacia de la clasificación fusionando dos o más micrófonos y aplicando técnicas de cancelación de ruido.

4.3 Integración del sistema

La integración del sistema se realizó mediante la implementación de versiones básicas de las pruebas Sígueme y Presentación. La figura 8 muestra una ejecución de la tarea Sígueme¹.

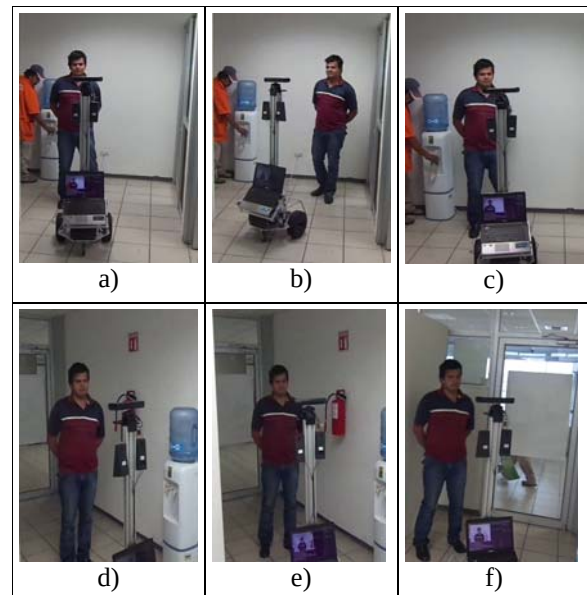


Fig. 8. Secuencia de imágenes (de a) a f)) de la ejecución de la prueba ‘Sígueme’.

Estas dos tareas han sido mostradas como exhibición en diversos eventos regionales y nacionales mostrando la capacidad del robot para ejecutar estas actividades. No obstante, para incrementar la eficacia del robot en el seguimiento, es

¹ Un video demostrativo se encuentra disponible en: http://www.youtube.com/watch?v=RdMog_4rfjM

necesario incluir una predicción de la posición del usuario, así como información de color y forma y habilitar el movimiento panorámico para MS Kinect. Al mismo tiempo, es necesario incrementar la velocidad de comunicación entre los actuadores y el módulo de coordinación del robot.

5. Conclusiones y trabajo futuro

En este documento se presentaron los avances en el desarrollo del robot de servicio SerBot II. Este robot considera habilidades de navegación por odometría, percepción visual, interacción por voz y manipulación de objetos. Se presentaron el diseño 3D y la configuración física actual del robot, así como su arquitectura software y hardware. Los resultados preliminares en las pruebas Sígueme y Presentación muestran la factibilidad de este robot para realizar diferentes tareas con resultados positivos.

Como trabajo futuro se propone incorporar el sistema de navegación basado en el MS Kinect del *Robotic Operating System (ROS)* y la inclusión de una cabeza de movimiento panorámico para el sistema visual. Se propone también estimar la posición de un usuario por la dirección de la fuente de voz usando los cuatro micrófonos disponibles en el Kinect. Igualmente se implementará un sistema de reconocimiento de gestos de un usuario. Finalmente, se plantea el ensamblado de los brazos del robot para sujeción de objetos.

Agradecimientos. Este trabajo está financiado por el proyecto PROMEP /103.5/12/3620. Parte del sistema visual fue desarrollado por uno de los autores en una estancia de investigación en el Laboratorio de Robótica en el INAOE entre marzo y junio del 2011.

Referencias

- [1] Avilés H., Alvarado M., Venegas E., Rascón C., Meza I., y Pineda L. Development of a Tour-Guide Robot using Dialogue Models and a Cognitive Architecture. *Advances in Artificial Intelligence - IBERAMIA 2010*. Springer, págs. 512-521, Bahía Blanca, Argentina. 2010.
- [2] Avilés A., Sucar L.E., Morales E.F., Vargas B.A., Corona E. Markovito: A Flexible and General Service Robot. *Design and Control of Intelligent Robotic Systems*. Volumen 177. Springer, Berlin-Heidelberg. ISBN: 978-3-540-89932-7, págs. 401-423. 2009.
- [3] RoboCup@Home: Rules and Regulations. 2013. Disponible en: <http://www.robocupathome.org/rules>
- [4] Sakagami, Y., Watanabe R., Aoyama C., Matsunaga S., Higaki N., Fujimura K. The Intelligent ASIMO: System Overview and Orientation. *IEEE/RSJ International Conference on Intelligent Robots and Systems EPFL*, págs. 2478-2483.
- [5] Willow Garage. Turtlebot 2. Disponible en: <http://turtlebot.com/>. Última visita: 1 de septiembre del 2013.
- [6] Vargas H.S., Olmedo E., Martínez D., López M., Medina E., Pérez J.L., Linares D., Peto C., García E., Poisot V., Robles J., Beristain G., Santos A., Xochimitl M. Donaxi @Home Project. Disponible en: <http://wla.orgfree.com/pruebas/PDF/DonaxiTD P.pdf>. Última visita: 2 de agosto del 2013.
- [7] Savage J., Negrete M., Matamoros M., Cruz J., Contreras L., Pacheco A., Figueroa I., Márquez J. Pumas@Home 2013 Team Description Paper. Disponible en: <http://biorobotics.fi-p.unam.mx/>. Última visita: 2 de agosto del 2013.
- [8] Gómez G. Con la vara que midas: Antropometría para el Diseño Regional. Colección Rajuela 3, México, Universidad de Colima, 1987.
- [9] Olson, E. Reporte Técnico. A Primer on Odometry Control. 2004.
- [10] Avilés H., Sucar E., Mendoza C., y Vargas B. Visual Recognition of Gestures using Dynamic Naive Bayesian Classifiers. *IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*, págs. 133-138. 2003.
- [11] Hoey J., St-Aubin R., Hu A.J., Boutilier C., SPUDD: Stochastic Planning using Decision Diagrams. *Uncertainty in Artificial Intelligence*, págs. 279-288. 1999.
- [12] Borenstein J., Feng L. UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots. *SPIE Conference on Mobile Robots*. Filadelfia. 1995.