

Localización y Reconocimiento de Obstáculos de un Robot Móvil Mediante SLAM-R

Sergio Díaz-Zagal, Citlalih Gutiérrez-Estrada, José Rafael Cruz-Reyes
Rafael Lemus-González y Juan José Rivera-Alemán

Instituto Tecnológico de Toluca, Toluca, Edo de Méx., México
sdiaz@ittoluca.edu.mx, citlalihg@ittoluca.edu.mx

Paper received on 23/07/12, Accepted on 27/08/12.

Resumen. Para un robot móvil, la capacidad de localizarse en un entorno y mapearlo es fundamental, lo cual los hace realmente autónomos, capaces de adaptarse a las circunstancias y al medio que los rodea. La autonomía y la adaptación de los robots móviles a su entorno es el problema principal a resolver. Estos robots autónomos no solo se aplicarían en la industria, sino se emplearía para el apoyo de los seres humanos en sus diversas tareas. El presente artículo describe un sistema de simulación, para la localización y reconocimiento de obstáculos, utilizando un algoritmo de localización y mapeo simultáneos (SLAM) en un robot móvil con las características del Pionner 3-AT, mediante técnicas de medición láser y un identificador de radiofrecuencia (RFID), para el reconocimiento de objetos. La integración del RFID y la técnica de navegación como resultado permitió disminuir el error de la posición de un robot móvil en un espacio interno, al mismo tiempo identificar lugares y objetos.

Palabras Clave: SLAM-R, RFID, Marilou, Telémetro Láser.

1 Introducción.

Una característica fundamental que debe tener un robot autónomo, utilizando la técnica del SLAM (Simultaneous Localization and Mapping), es la capacidad de explorar el entorno y construir un mapa del mismo entorno. En la técnica del SLAM los sensores más utilizados por muchos autores son los sensores SONAR [1] o LASER para 2D [2, 3] y 3D [4]. Por otra parte, también existe un gran interés por el uso de cámaras como sensores, técnica conocida como SLAM visual. Una de las ventajas que ofrece las cámaras, es la mayor información acerca del entorno, además que son menos costosas que los sensores previamente mencionados. Gran parte de la técnica SLAM visual, funcionan con mapas basados en características (landmark-based maps). Estos mapas representan la localización de un conjunto de puntos distintivos del entorno, respecto a un sistema de referencia global [5, 6].

En el estado del arte, los mapas “preconstruidos” e incluidos en los robots previamente, resuelven el problema de la localización de un robot móvil, actualmente estos mapas son elaborados a medida que el robot se va desplazando.

La pieza clave de la autonomía completa de un robot móvil, está en la búsqueda de la solución al problema del SLAM, lo que constituye el principal esfuerzo de los investigadores de robótica móvil en todo el mundo, sin embargo, actualmente se han conseguido avances importantes, pero se continúa buscando la mejor solución.

En la localización y la construcción de los mapas, el principal problema se deriva de la existencia del ruido y de las limitaciones en el rango de las características de los sensores. No obstante, a través de la herramienta computacional se ha tratado de corregir el problema del SLAM (y se continúa investigando) desde diversos puntos de vista.

Los algoritmos más utilizados en la robótica móvil actualmente en el campo del SLAM son: el SLAM-EKF (Filtro Extendido de Kalman), el basado en representaciones gráficas, aplica satisfactoriamente métodos de optimización no lineal (sparse nonlinear optimization methods) y el de técnicas de filtrado estadístico no paramétrico, conocidas como filtros de [7].

Una limitación importante que tiene el algoritmo SLAM-EKF, es el coste computacional, haciéndolo difícilmente escalable. Sin embargo, la construcción de sub-mapas son las versiones más prometedoras del SLAM-EKF, aunque los algoritmos resultantes se parecen demasiado a los basados en grafos.

El problema del SLAM completo, de naturaleza offline, es resuelto por los métodos basados en grafos. La estimación la realizan desde la observación, por ello el problema de SLAM puede ser modelado como un grafo no-denso de restricciones débiles, donde cada restricción corresponde a un movimiento o a la observación de un landmark. También hay versiones online que suelen eliminar del grafo las posiciones antiguas del robot.

Dentro del método que usa técnicas de filtrado estadístico no paramétrico, se encuentran las conocidas como filtro de partículas [7], dejan de lado la relación entre los objetos del mapa y el robot, representándolo como independiente. El algoritmo de FastSLAM computacionalmente presenta algunas ventajas, puede ser usado como un filtro, y su actualización requiere un tiempo lineal/logarítmico, mientras que el EKF necesita un tiempo cuadrático. Además, FastSLAM se salta el paso de asociación de datos (Porque no lo necesita), lo que lo convierte en un algoritmo ideal en casos en los que se desconoce cómo realizar asociaciones. Sin embargo, el número de partículas necesarias puede crecer en exceso, especialmente en robots que buscan realizar mapas de múltiples trayectorias cerradas.

Este artículo se centra en la implementación de un algoritmo de navegación en un robot móvil y su evaluación, a través de un estudio formal de simulación, que integra a la vez procesos de la Ingeniería del Software [8], en las etapas de obtención de requerimientos, programación y control de versiones; además de emplear el modelo de ciclo de vida conocido como Espiral, modelo que no se describió, debido a que no es el objetivo de este artículo, sin embargo, se utilizó para el diseño del prototipo del robot móvil, así mismo se empleó la ingeniería de software para darle la formalidad al trabajo, como el análisis de requerimientos, gestión y desarrollo del proyecto.

2 Diseño del prototipo virtual.

El modelado, el control así como la simulación, han sido desarrollados empleando el software de simulación Anykode Marilou Robotics 2010 [9], para el diseño de un robot móvil virtual, incluyendo un telémetro láser Hokuyo URG-04X [10] y un lector RFID ID-12 [11] de Innovations. Marilou Robotics una importante herramienta del software para el estudio, la creación y la evaluación de sistemas robóticos de alto desempeño.

Recordando que un modelo es una representación de un objeto o de un fenómeno [12], dentro del proyecto se busca modelar un robot móvil para lo cual es necesario primeramente obtener las dimensiones físicas del prototipo real tomadas de los requerimientos funcionales y no funcionales anteriormente mencionados, para poderlas plasmar de manera fidedigna con el software de diseño Marilou Robotics 2010 [13].

A continuación se detalla el modelo de un robot móvil, se deben integrar en el modelo los requerimientos funcionales, para toma de datos y control, que representen los sensores y actuadores que implementa el prototipo real, finalmente debido a que el tipo de simulación que se hace la cual es a través de una animación en 3D que toma en cuenta características de cinemática y dinámica, dentro del software de diseño, fue necesario establecer las restricciones físicas del modelo animado en el escenario 3D como son la gravedad, el peso y otras fuerzas que interactuaron sobre él. Este proceso se muestra en la Figura 1.

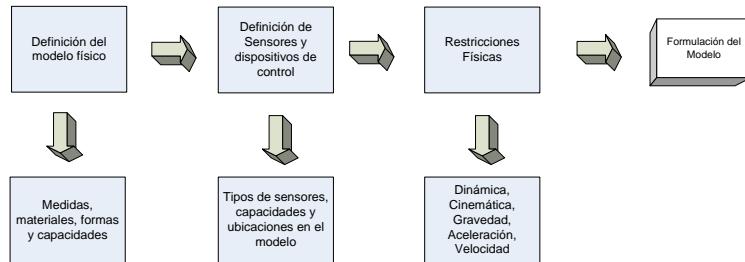


Figura 1. Proceso de Diseño del Modelo.

Con las definiciones y Especificación de Requerimientos, se logró el rediseño del algoritmo de navegación autónoma adaptado a escenarios con gran cantidad de obstáculos, se integró en el diseño el sensor laser y se configuró para realizar un barrido axial suficientemente amplio para implementar el algoritmo de mapeo (SLAM).

El prototipo final del robot móvil con las características del Pionner 3-AT [14], se muestra en la Figura 2.

La evaluación del prototipo virtual presentó una similitud con el modelo real del 95% en características visuales y de rendimiento contra el modelo real propuesto. El algoritmo de mapeo se implementó con resultados óptimos en escenarios sin gran cantidad de obstáculos. Finalmente el algoritmo de navegación autónoma se adaptó mejor en escenarios con gran cantidad de obstáculos.

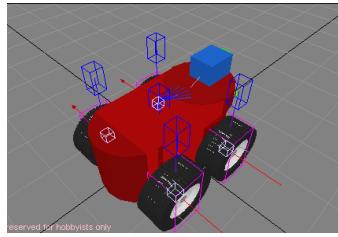


Figura 2. Prototipo Desarrollado.

2.1 Programación.

La existencia de robots que realicen de manera autónoma tareas de modo eficiente, depende fundamentalmente de su construcción mecánica y de su programación. Una vez construido el cuerpo mecánico del robot, conseguir que realice una tarea se convierte en la práctica en un problema de programación [15]. La generación de comportamiento en un robot consiste entonces en escribir el programa, que al ejecutarse en el robot, causa ese comportamiento cuando éste se encuentra en cierto entorno.

La autonomía y la “inteligencia” residen en el programa desarrollado. Por ejemplo, en los robots móviles el comportamiento principal se centra en su movimiento. Los programas que se ejecutan en el robot determinan cómo se mueve éste por su entorno, reaccionando ante obstáculos percibidos por los sensores, acercándose a algún destino, etc. Y para ello tienen que enviar continuamente las órdenes pertinentes a los motores. Este proceso se describe en la Figura 3.

El software de simulación Marilou Robotics 2010 [13], a través de su módulo MODA (Marilou Open Device Access), Permitió la interacción directa entre la lógica del programa y los sensores que éste controla, gracias a la integración directa entre los nombres de los dispositivos y su asociación con variables definidas dentro del programa que comunican la información entre sus dispositivos en tiempo real, durante la simulación.

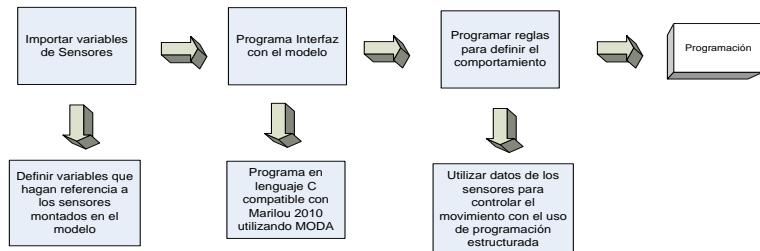


Figura 3. Proceso de Programación.

Una de las aportaciones del proyecto consistió en el empleo de técnicas, hasta ahora solo utilizadas en gestión de proyectos de software, a un proyecto de robótica, que además emplea también procesos de simulación. Con estas técnicas fue posible

formular y evaluar cinco prototipos distintos de robots móviles; las características en común de estos prototipos fue la inclusión de sensores ultrasónicos y laser, motores independientes en cada rueda y un modelado siempre emparentado con modelos físicos reales. Las características cambiantes de estos prototipos fueron sus configuraciones móviles, entre las cuales se probaron, principalmente las de tipo diferencial, con una rueda loca atrás o una combinación de dos diferenciales, una al frente y otra atrás; otras características que se modificaron frecuentemente fueron las relativas al software, aplicado a los algoritmos de navegación autónoma y mapeo implementados en la lógica del robot móvil.

Actualmente se han realizado otros proyectos con el software Marilou Robotics 2010 [13], en donde la simulación y la robótica van de la mano para evaluar y experimentar nuevas técnicas de navegación, modelos de robots y escenarios, en los que se pretende probar posteriormente, un sistema o conjunto de sistemas, como el trabajo presentado por Marcela Urdaniz [16], donde se describe una simulación realizada con el mismo software aquí descrito. Sin embargo poco se menciona acerca de la gestión del proyecto, la obtención de requerimientos, la forma en cómo se validaron los resultados y el control de versiones que se utilizó durante el proyecto.

En los últimos años ha crecido el interés por los robots móviles con capacidad de identificar su entorno (localización), y con capacidad de reaccionar respecto al mismo, sin la necesidad de intervención humana. El proceso de localización puede ser dividido de forma general en seguimiento de la posición, localización basada en mapas y SLAM (Simultaneous Localization And Mapping) [17]. Cada uno de estos niveles constituye un amplio campo de investigación que se ha extendido en función a la manera como la comunidad científica ha enfocado el problema. Usualmente los robots diseñados para operar en entornos interiores cuentan con sensores, que les permiten determinar la posición inicial para que el robot pueda estimar su posición a través del tiempo. Sin embargo, la estimación del proceso de localización se ve afectado por errores y factores del entorno que producen que el error de la posición estimada y la posición real del robot crezca con el paso del tiempo [18].

3 Método.

La navegación del robot móvil empleó el algoritmo fastSLAM que trabaja mediante filtros de partículas, para la estimación se implementó un sistema híbrido donde únicamente se utilizó la posición anterior $St-1$ y la actual St para determinar la posible posición del robot $St+1$ almacenando el mapa durante todo el periodo de la exploración. Este modelo permite que el procesamiento de los datos sea más rápido y se pueda generar un mapa en tiempo real (online), una de las modificaciones que se realizaron dentro del algoritmo fastSLAM es la inclusión de la variable Rt que aporta una nueva solución al problema de la asociación de datos mediante las etiquetas RFID usadas como referencias (Figura 4). La característica principal de los sistemas de navegación se basa en la dependencia de los datos que los sensores generan. Por tanto, si los sensores emiten lecturas erróneas, la posición estimada tendrá un alto grado de error. Debido a este problema, se propuso incluir un dispositivo auxiliar (RFID) para aumentar la fiabilidad de los datos del entorno.

El algoritmo es el encargado de estimar la posición actual del robot en un entorno semi-dinámico y utiliza las etiquetas RFID como información adicional para reconocer objetos dentro del mapa mediante un identificador único (ID).

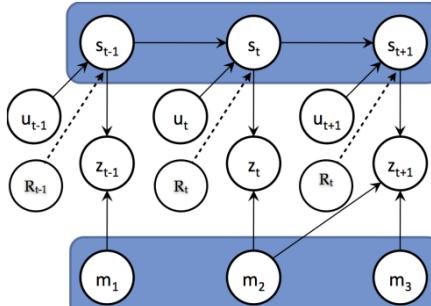


Figura 4. Red Bayesiana Dinámica SLAM-R.

En la Figura 4 se muestra el modelo de red bayesiano dinámico utilizado para representar las conexiones que existen entre el mapa del entorno (M), los datos del sensor laser (Z), el uso de las etiquetas RFID como referencia (R), la posición del robot (S) y la odometría (U) todos ellos dispuestos en un determinado tiempo (t). Para realizar la estimación de la posición del robot es necesario implementar la etapa de la predicción (Ec 1):

$$S_t^{[m]} \sim p(S_t | S_{t-1}^{[m]}, U_t, R_t) \quad \text{Ec 1}$$

Y posteriormente la de actualización de pesos y re muestreo (Ec 2),

$$W_t^{[m]} = \frac{p(S_t^{[m]} | Z^t, U^t, R^t)}{p(S_t^{[m]} | Z^{t-1}, U^{t-1}, R^{t-1})} \quad \text{Ec 2}$$

Además de emplear el uso del RFID para la identificación de los objetos del entorno, se mejoró el algoritmo de navegación, la identificación de los objetos del entorno se realiza mediante el ID, que contienen cada una de las etiquetas, registrando cada una de ellas en una base de datos, donde se registran las características físicas del objeto, para posteriormente incluirlo de forma virtual dentro del mapa generado y obtener una mejor representación gráfica del entorno real (Figura 5).

Para realizar las pruebas del funcionamiento del algoritmo propuesto, se programó en C++, por ser uno de los lenguajes que cuenta con soporte para implementar librerías SDL (Simple Directmedia Layer), que permiten la creación de gráficos en 2D y la generación del mapa. Como se mencionó anteriormente, para la simulación se utilizó el software Marilou, que permite modelar el funcionamiento de robots, dispositivos electrónicos y entornos para crear un ambiente similar al real, en el cuál se diseñó el robot móvil basado en las características físicas del robot Pioneer 3AT y de los sensores que éste incluye adicionando el telemetro láser Hokuyo URG04LX. Por otro lado, también se modelaron diversos entornos estáticos con y sin objetos para analizar los posibles cambios en el sistema de navegación.

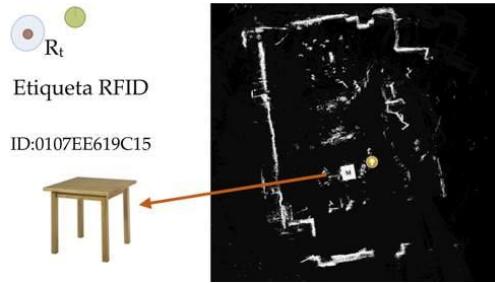


Figura 5. Reconocimiento de obstáculos mediante RFID.

Para la conexión de ambos sistemas (Algoritmo + Simulador) fue necesario utilizar un conjunto de librerías MODA (Marilou Open Devices Access), suministradas por el software Marilou, que sirven para utilizar métodos y tipos de datos que manipulan el funcionamiento de los dispositivos electrónicos simulados. Gracias a este conjunto de librerías, se logró la comunicación de ambos sistemas para transmitir la información del telémetro láser hacia el algoritmo de navegación. Uno de los problemas fue incluir el dispositivo RFID, debido a que el simulador Marilou no cuenta con dicho dispositivo, por tal motivo, se utilizó un puerto virtual para conectar el dispositivo RFID físico mediante el puerto serial de la computadora (Figura 6).

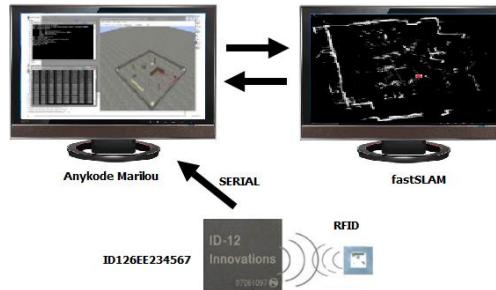


Figura 6. Sistema de interconexión entre el algoritmo de navegación y el simulador Marilou.

4 Pruebas y resultados.

El proceso de experimentación del proyecto consistió en el diseño de escenarios, las restricciones impuestas a éstos y el montaje del modelo del robot móvil dentro del escenario.

El diseño del escenario se realizó con el mismo software con el que se modela el robot Marilou Robotics2010 [13], la ventaja de este software fue la adaptación a las condiciones físicas como gravedad, masa de los objetos inmersos en el escenario, fuerzas que pueden inferir sobre el modelo, etc. Todas estas condiciones pueden ser definidas y cambiadas para observar el comportamiento del modelo dentro del escenario y corregir anomalías del sistema de navegación, que reside en la

programación del robot. Este proceso se detalla en la Figura 7.

La evaluación de los resultados obtenidos se logró mediante la generación de un mapa virtual con diversos objetos y se comparó el mapa virtual generado por el algoritmo de navegación, determinando la similitud entre ambos.

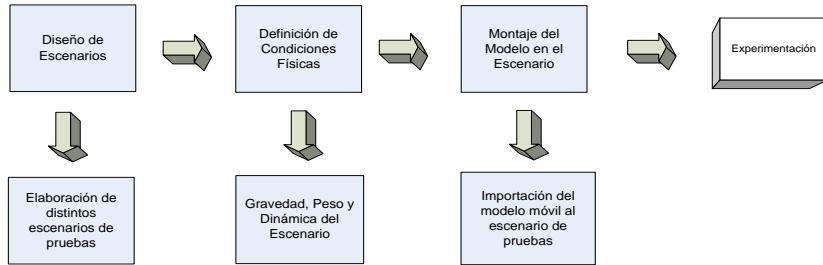


Figura 7. Proceso de Experimentación.

Se realizaron pruebas con 3 entornos diferentes dando como resultado una similitud de 95%, comparada con el mapa virtual, quedando únicamente incompletos los objetos no etiquetados mediante RFID. Así mismo para probar el algoritmo de SLAM, se utilizaron archivos repositorios (DataSet) internacionales, como dataset Málaga 2006 [19] o como los del laboratorio de robótica de mines paristech [20].

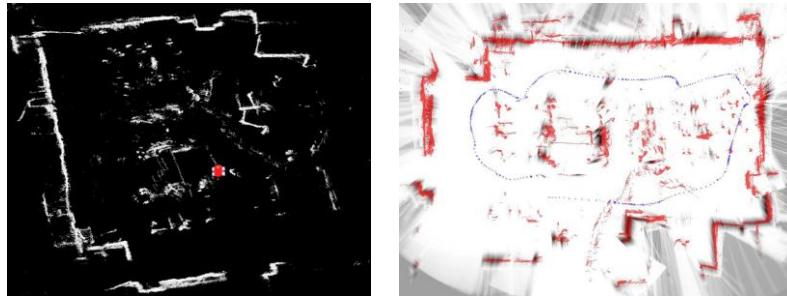


Figura 8. Resultado del algoritmo SLAM sin utilizar el RFID sin objetos en el entorno, utilizando el repositorio del laboratorio de robótica de mines paristech.

Una de las limitaciones del algoritmo propuesto fue la dependencia de las etiquetas RFID que fueron colocadas en el entorno para la identificación de objetos, sin embargo, esto no implicó que el algoritmo no pudiera continuar con la navegación ya que, en caso de que un objeto no se encuentre etiquetado, éste será representado como un obstáculo, sin las características propias del mismo, que le indicará al robot que no puede desplazarse por dicho lugar. La Figura 9 se muestra un entorno virtual sin objetos, con la finalidad de comprobar la independencia del algoritmo SLAM con el sistema de RFID.

Posteriormente se diseñó un entorno con diversos obstáculos para verificar si el algoritmo era capaz de interpretar las etiquetas y reconocer dichos objetos durante la navegación. En la Figura 10 se muestran los resultados de la prueba.

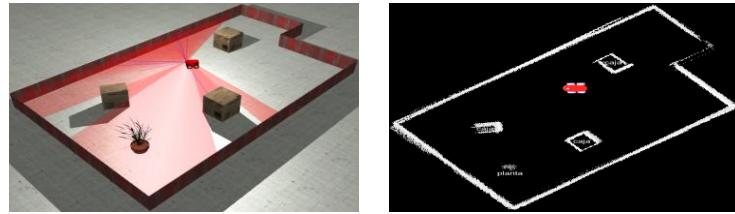


Figura 9. Resultados del algoritmo propuesto SLAM-R sin objetos en el entorno.

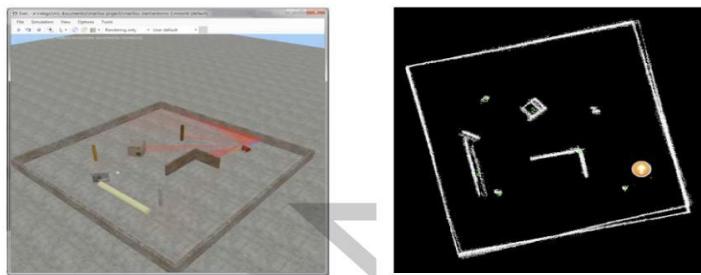


Figura 10. Resultados del algoritmo propuesto SLAM-R con objetos en el entorno y etiquetas RFID.

5 Conclusiones.

De acuerdo a los resultados obtenidos en las pruebas simuladas, se concluyó que las etiquetas RFID en un entorno interior son de gran utilidad para la identificación de objetos y para disminuir el error en la estimación de la posición del robot dentro del mapa, sin embargo, se recomienda etiquetar únicamente los objetos de interés necesarios para realizar determinadas tareas (ubicación de objetos, transporte de materiales, reconocimiento de áreas, etc.), ya que etiquetar todos los objetos del entorno, puede convertirse en proceso complicado y tedioso. Por otro lado, es importante mencionar que el trabajo realizado se probó únicamente en un modelo virtual (simulación), sin embargo, uno de los objetivos es comprobar que los resultados obtenidos corresponden al comportamiento de un modelo físico.

La utilización del modelo en espiral como respuesta a la necesidad de formalizar la gestión del proyecto, tuvo como principales objetivos, mejorar el análisis de la información, apoyar a la toma de decisiones, la re-utilización del código de programación, la evaluación y validación de cada uno de los prototipos; así como la reformulación de nuevos requerimientos que se adaptaron mejor a cumplir el propósito principal del proyecto; el cual estuvo enfocado al estudio e implementación de un algoritmo de navegación autónoma para un robot móvil, así como un algoritmo de mapeo y su representación gráfica en tiempo real de la simulación; con todas las aplicaciones que el empleo de estas técnicas pudieron aportar para proyectos futuros en forma de componente o como base de investigación para desarrollar otros proyec-

tos de robótica con ésta misma gestión, enfocados al aprovechamiento de la simulación como una técnica precedente a la manufactura eficiente de robots móviles y el ahorro de costos

Un objetivo más de la investigación como trabajo futuro es considerar la navegación del robot tanto en entornos interiores como exteriores.

Agradecimientos

Agradecemos el apoyo a la DGEST por aprobar el proyecto “Prototipo de un sistema virtual de un par de mini-actuadores de músculos artificiales neumáticos en antagonismo basados en el modelo de McKibben”, con clave TOL-DCIET-2010-102, de la convocatoria “Registro de proyectos 2010”. De igual manera agradecemos el apoyo a la DGEST por aprobar el proyecto “La Ingeniería de Sistemas como modelo integral para el desarrollo de aplicaciones adaptativas”, con clave 4389.11-P, que ha servido como base para continuar en investigaciones relacionadas a este proyecto.

Referencias

1. O. Wijk and H. I. Christensen. Localization and navigation of a mobile robot using natural point landmarkd extracted from sonar data. In *Robotics and Autonomous Systems*, 1(31), pp. 31-42, 2000.
2. J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(4), 1991.
3. S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. In *Int. Journal of Robotics Research*, 20(5), pp. 335-363, 2001.
4. R. Triebel and W. Burgard. Improving simultaneous mapping and localization. In *Proc. Of the National Conference on Artificial Intelligence (AAAI)*, 2005.
5. J. Little, S. Se, and D.G. Lowe. Global localization using distinctive visual features. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2002.
6. A. Gil, O. Reinoso, W. Burgard, C. Stachniss, and O. Martínez Mozos. Improving data association in rao-blackwellized visual SLAM. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2006.
7. J. A., “Sistema de navegación y modelado del entorno para un robot móvil”, noviembre 2006, UPM Autonomous Systems Laboratory.
8. Ian Sommerville, Ingeniería del Software, “Ingeniería de Requerimientos”, (Pearson, 7^a Ed., México), pags. 105-150 (2005).
9. AnyKode, “Marilou, simulate your robotics applications.” <http://www.anykode.com/index.php>, 2012.
10. Hokuyo, “Hokuyo urg-04lx datasheet.” http://www.hokuyo-aut.jp/02sensor/07scanner/download/data/URG-04LX_spec.pdf, Febrero 2012.
11. Sparkfun, “Rfid innovations id-12 datasheet.” <http://www.sparkfun.com/datasheets/Sensors/ID-12-Datasheet.pdf>, Febrero 2012.
12. Robert Shannon; Johannes J. D. Systems simulation: the art and science, (IEEE Transactions on Systems, Man and Cybernetics. Vol. 6(10), Inglaterra)pp. 723-724(1999).
13. Marilou Robotics Studio, anyKode, Francia2008-2010, www.anykode.com.
14. Adept, “Pioneer 3-at datasheet.” <http://www.mobilerobots.com/Libraries/Downloads/Pioneer3AT-P3AT-RevA.sflb.ashx>, 2012 Febrero.
15. Tortoise Hg, Mercurial, <http://tortoisehg.bitbucket.org>.

16. Marcela V. Urdaniz, María A. Ortiz, et al., Diseño y Simulación de una Órtesis Robótica para Rehabilitación de la Marcha, (Universidad Nacional de Entre Ríos, Revista Iberoamérica de Ingeniería Mecánica, Argentina)(2007).
17. D. H. and B. T, “Simultaneous localization and mapping: Part 1,” IEEE Robotics and Automation, 2006.
18. B. J. and F. L., “Correction of systematic odometry errors in mobile robots,” IEEE/RSJ International Conference, 1995.
19. Dataset: Málaga dataset 2006, http://www.mrpt.org/robotic_datasets
20. Dataset: ParisTech, <http://cogrob.ensta-paristech.fr/loopclosure.html>