

Software abierto para la navegación autónoma de una silla de ruedas inteligente.

C. Losada, J.C. García, M. Marrón, D. Pinedo y M. Mazo

Departamento de Electrónica. Universidad de Alcalá. Edificio Politécnico. Campus Universitario s/n. 28805 Alcalá de Henares, Madrid, España. Telf. 918856592, Fax: 918856591
e-mail: {losada, jcarlos, marta, mazo}@depeca.uah.es

Resumen. En esta publicación se justifica la necesidad de abordar la implantación de sistemas de *software* abierto y de *hardware* estándar para la urgente incorporación de los numerosos avances de la robótica móvil al ámbito de las tecnologías de apoyo, y más concretamente las sillas de ruedas inteligentes. Con este objetivo se describe una plataforma robótica abierta y estándar, tanto en hardware como en software, basada en librerías ROS (“*Robotic Operating System*”) buses CAN y sistemas empotrados Cortex. Esta estructura es de gran potencial por su fácil adaptación a los últimos avances en robótica móvil.

Palabras clave: Robótica móvil, Silla inteligente, *Intelligent Wheelchair*, “*Robotic Operating System*” (ROS).

1. Introducción

En las últimas décadas ha habido continuos y espectaculares avances en las áreas tecnológicas relacionadas con los sistemas de robótica móvil. Sensores, procesadores y actuadores cada vez mejores y más baratos, unidos a avanzadas estrategias de control, son los actores de estas mejoras. Gracias a estos avances se han venido proponiendo aplicaciones innovadoras a estos sistemas robóticos, algunas de ellas en el campo de las Tecnologías de Apoyo o de Asistencia.

A comienzos de los años 1980, ciertos trabajos de la Universidad de Stanford [1] proponen utilizar sensores ultrasónicos sobre una silla de ruedas adaptada. En la década de 1990, otros grupos comenzaron a desarrollar vehículos de asistencia a la movilidad innovadores [2], [3], [4]. En su concepto inicial, estos vehículos eran similares a cualquier robot móvil estándar, siendo el propósito de estos trabajos lograr un vehículo autónomo al que podemos denominar Silla Inteligente (IW - *Intelligent Wheelchair*).

Tras más de 30 años de investigaciones casi ninguno de estos prototipos ha llegado a los potenciales usuarios. Este trabajo trata de resolver parcialmente, al menos, los problemas que mantienen a estos vehículos dentro de los laboratorios. Bajo nuestro punto de vista es preceptivo reducir el coste de construir una IW, usando para ello elementos electrónicos y de comunicaciones estándar (USB, CAN, WiFi, etc.) con el objetivo de garantizar su compatibilidad con otros sistemas y servicios y facilitar su adaptación a los diversos perfiles de usuarios. Para alcanzar este objetivo, se toma como punto de partida la arquitectura de IW propuesta en [5], que se describe brevemente en el apartado 2.

Además es aconsejable diseñar los algoritmos de control de las IW usando paquetes software típicos en sistemas de navegación autónoma, ampliamente extendidos entre la comunidad científica como “*Robot Operating System*” (ROS) [6] u OpenCV. Se propone usar ROS, debido a su arquitectura modular y su naturaleza distribuida, para incorporar nuevas funcionalidades a la IW, como se explica en el apartado 3.

2. Arquitectura de la silla inteligente (IW)

En este trabajo, se denomina silla inteligente (IW) a una solución modular que permite construir sillas básicas (*Basic Wheelchairs*, BW), avanzadas (*Advanced Wheelchairs*, AW) y autónomas (*Smart Wheelchairs*, SW) empleando la misma plataforma, módulos y estructura (Figura 1). Se parte del trabajo [5], en el que se propone un marco de desarrollo común para IW, que se beneficia del bajo coste de producción de las BW, a las que se añade un módulo intermedio que controla comportamientos como la conducción asistida. Otras mejoras pueden instalarse en una etapa superior, permitiendo la evolución gradual desde una BW a una SW específica, idónea para el caso específico de usuarios especiales.

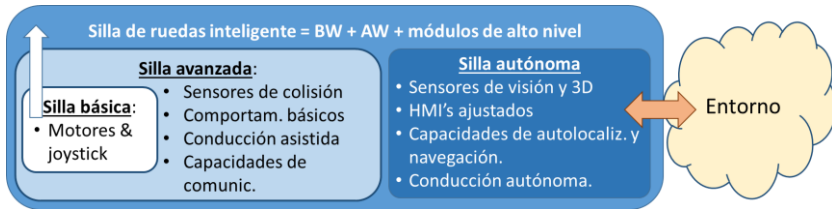


Figura 1. Modelo modular de la silla de ruedas inteligente.

La arquitectura hardware general de la IW se muestra en la Figura 2. Se tiene una estructura dividida en dos capas, para procesamiento de bajo y alto nivel respectivamente. Cada capa está formada por nodos funcionales (módulos de cómputo) interconectados mediante buses de comunicaciones estándar. El uso de estándares, reduce el coste de la IW. Esta estructura se ajusta al modelo modular objetivo (Figura 1): la BW puede construirse empleando únicamente el control de motores y el joystick (sistema mínimo), mientras que la AW y la SW se implementan añadiendo los módulos apropiados, tanto software como hardware.

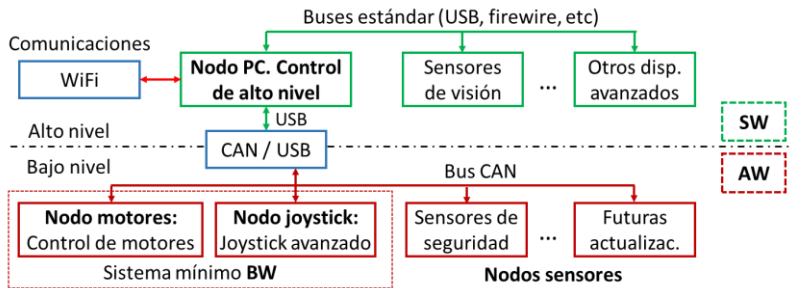


Figura 2. Diagrama de bloques general de la arquitectura hardware del prototipo de IW.

Esta estructura tiene como ventaja la facilidad para realizar modificaciones en el control y funcionalidad del vehículo sin más que conectar al sistema, mediante un driver CAN-USB, cualquier dispositivo informático en donde se pueden instalar aplicaciones exclusivas para un control más avanzado.

3. Diseño del software abierto para la navegación autónoma

La arquitectura hardware del sistema, posibilita utilizar elementos hardware (microcontroladores basados en arquitectura Cortex, LPC2129) y de comunicación (CAN, USB) estándares, es necesario diseñar los algoritmos de control de las IW usando paquetes software típicos en sistemas de navegación autónoma.

Para lograr el objetivo, se emplea ROS: un Framework de Desarrollo de Robótica ampliamente utilizado por la comunidad científica [6], [7], que ofrece herramientas y librerías para el desarrollo de sistemas robóticos. Entre sus características destacan: la posibilidad de trabajar en diferentes lenguajes de programación de alto nivel (C++, Python, etc.), y su naturaleza distribuida. Así, un sistema construido con ROS consiste en una serie de procesos que pueden encontrarse en máquinas diferentes, y que permanecen conectados en tiempo de ejecución mediante una topología *peer-to-peer*, sin necesidad de tener un servidor central.

La implementación de ROS está formada por nodos, mensajes, temas y servicios. Los nodos son las aplicaciones que se ejecutan para realizar una determinada función, y se comunican entre sí por medio de mensajes. Los temas o *topics* son canales de comunicación por los que circulan los mensajes. Los nodos pueden comunicarse entre sí mediante los temas, actuando uno como publicador (creando temas y difundiendo mensajes) y otro como suscriptor (accediendo a mensajes). Otra forma de comunicación entre nodos son los servicios: mensajes de solicitud/respuesta entre un nodo servidor y otro cliente. Los paquetes son la unidad principal de organización en ROS y permiten agrupar una serie de aplicaciones (nodos), servicios o bibliotecas, facilitando su portabilidad e instalación.

3.1. Sistema desarrollado

Partiendo de la plataforma hardware descrita en el apartado 2, actualmente el sistema desarrollado está formado por cuatro nodos que se comunican entre sí mediante un bus CAN a través de LPC2129.

- Nodo sensores:** es el encargado de leer los datos de los diferentes sensores de ultrasonidos (SRF02) usados para el sistema de evasión de obstáculos así como los valores del acelerómetro para detectar choques. Estos sensores se conectan con el microcontrolador mediante un bus I2C.
- Nodo joystick:** se trata de un joystick analógico de 2 ejes, un potenciómetro, display, teclado, bocina y luz de encendido. Este nodo proporciona la interfaz hombre-máquina a la silla de ruedas.
- Nodo motores:** está formado por los encoders ópticos, los motores y la tarjeta de potencia.
- Nodo PC:** en el que se trabaja con ROS para implementar algoritmos de navegación reactiva y representar en tiempo real las lecturas de los sensores de ultrasonidos (US), y el movimiento de la silla.

Una de las aportaciones del trabajo aquí presentado consiste en transformar la comunicación de campo basada en protocolo CAN en *topics* de ROS, convirtiendo así los distintos nodos descritos en nodos ROS.

Partiendo de esta estructura, se ha creado el modelo para ROS de la silla de ruedas (Figura 3). Este modelo se conoce como modelo URDF y en él se definen tanto las dinámicas de la silla como el modelo visual de ésta. Además, está estructurado en forma de árbol, formado por piezas, engranajes que unen las distintas piezas entre sí y relaciones TF entre los ejes de coordenadas de las diferentes piezas.

El modelo se incorpora a distintos simuladores 3D (GAZEBO) o 2D (STAGE, STDR), donde es posible incluir algoritmos de navegación reactiva a partir de la información de los sensores US. La herramienta RVIZ, junto a cualquiera de los simuladores, permite visualizar el modelo de silla, los datos de sensores reales o simulados y los *topics* que se están enviando en ROS al bajo nivel de la arquitectura descrita.

La incorporación de ROS permite evaluar diferentes algoritmos de navegación reactiva, mediante simulación, e implementarlos de forma sencilla en la silla de ruedas real. En este caso se ha incorporado un sistema de evasión de obstáculos autónomo a partir del comando manual de la IW con un joystick. Gracias al sistema abierto y estándar diseñado, el paso del sistema simulado al real ha sido rápido y ha mostrado un funcionamiento altamente fiable.

3.2. Resultados

Los resultados obtenidos permiten validar, a nivel de simulación, la arquitectura propuesta y el modelo URDF de la silla de ruedas descrito, tal y como se observa en la Figura 3.

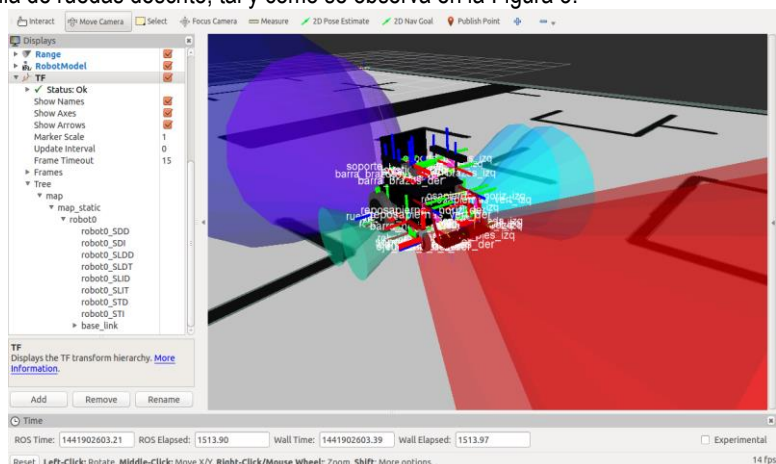


Figura 3. Simulación de la arquitectura software basada en ROS.

Esta herramienta es tremendamente útil para el diseño de distintas estrategias de navegación totalmente autónomas, ya puestas en marcha con herramientas software desarrolladas ad-hoc previamente para esta plataforma como las descritas en [5].

Por otro lado, la integración de la arquitectura ROS a la plataforma física arroja resultados muy positivos, pues, tal y como se muestra en la Figura 4, esta arquitectura permite acceder a todos los mensajes del bajo nivel que viajan por el bus CAN en tiempo real, a través de *topics* de ROS, corroborando la posibilidad de incorporar los algoritmos de navegación referidos con previsión de éxito en la nueva plataforma diseñada.

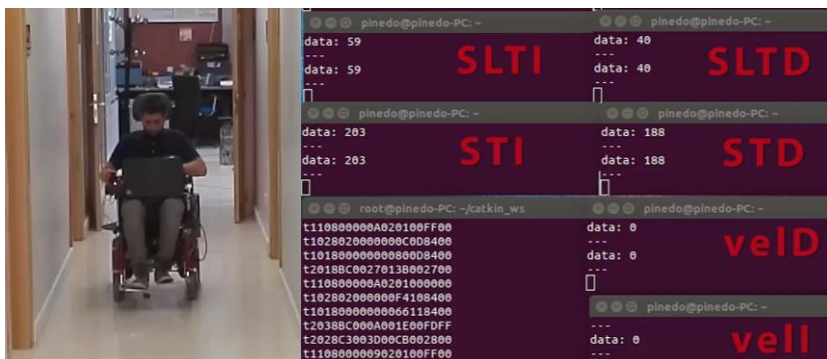


Figura 4. Integración física de la arquitectura software basada en ROS en la plataforma.

4. Conclusiones y trabajos futuros

En este trabajo se presenta una plataforma básica robusta y flexible para el diseño de una silla de rueda inteligente en la que se puedan incorporar de forma de fácil y rápida distintos tipos de sensores y algoritmos ya ensayados en arquitecturas software previamente diseñadas ad-hoc, para su navegación autónoma.

Concretamente, la IW diseñada incluye la estructura básica de una silla de ruedas semi-autónoma que incorpora movimientos reactivos determinados por un array de sensores ultrasónicos ubicados en su contorno. Con la plataforma diseñada se ha podido probar fácilmente primero en simulación y posteriormente en tiempo real, un algoritmo de navegación reactiva basados en sensores ultrasonidos.

Como trabajo futuro se plantea incorporar mayores niveles de inteligencia a la plataforma diseñada, de modo que muestre comportamientos técnicamente más avanzados ([5]) con un esfuerzo de implementación, y por tanto de puesta en el mercado, mínimos.

Referencias

- [1] David L. Jaffe. An ultrasonic head position interface for wheelchair control. *Journal of Medical Systems*, 6 (4) (1982), 337-342.
- [2] H.A. Yanco, et al. Initial Report on Wheelchey: A Robotic Wheelchair System. *Proc. of the Workshop on Developing AI Applications for the disabled*, Montreal, Canada (1995).
- [3] S.P. Levine, D.A. Bell, L.A. Jaros, R.C. Simpson, Y. Koren, & J. Borenstein. The NavChair assistive wheelchair navigation system. *IEEE Transactions on Rehabilitation Engineering*, 7(4) (1999) 443-451.
- [4] M. Mazo, J.C. Garcia, et al. Experiences in assisted mobility: the SIAMO project, *Proceedings of the 2002 IEEE International Conferences on Control Applications*, 2, (2002) 766-771.
- [5] J. C. Garcia, M. Marrón, J. Ureña, D. Gualda. *Intelligent Wheelchairs: Filling the Gap between Labs and People*. Assistive Technology Research Series. 33: Assistive Technology: From Research to Practice. 2013, Pp. 202-209.
- [6] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., & Ng, A. Y. (, May). ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, 2009. Vol. 3, No. 3.2, p. 5.
- [7] Hart, Stephen; Dinh, Paul; Hambuchen, Kimberly. The Affordance Template ROS package for robot task programming. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Pp: 6227 - 6234,