

Navegación Autónoma Asistida Basada en SLAM para una Silla de Ruedas Robotizada en Entornos Restringidos

Fernando A. Auat Cheein * Celso De la Cruz ** Ricardo Carelli *
Teodiano F. Bastos Filho **

* Instituto de Automática, Facultad de Ingeniería, Universidad Nacional de
San Juan, Av. Libertador Gral. San Martín 1109 Oeste, San Juan, Argentina
(email: fauat@inaut.unsj.edu.ar, rcarelli@inaut.unsj.edu.ar)

** Departamento de Engenharia Elétrica, Centro Tecnológico II,
Universidade Federal do Espírito Santo, Av. Fernando Ferrari 514, Vitória,
ES, Brasil (email: celsodelacruz@gmail.com, tfbastos@ele.ufes.br)

Resumen: En este trabajo se presenta una interfaz especialmente diseñada para la navegación de una silla de ruedas robotizada dentro de entornos restringidos. El funcionamiento de la interfaz se rige por dos modos: un modo autónomo y un modo no-autónomo. El manejo no-autónomo de la interfaz de la silla de ruedas se realiza por medio de un *joystick* adecuado a las capacidades del usuario el cual gobierna el movimiento del vehículo dentro del ambiente. El modo autónomo de la silla de ruedas se ejecuta cuando el usuario tiene que girar un determinado ángulo dentro del ambiente. La estrategia de giro se ejecuta mediante un algoritmo de maniobrabilidad compatible con la cinemática del vehículo y mediante el uso del algoritmo de SLAM (por sus siglas en inglés de *Simultaneous Localization and Mapping*). El modo autónomo se compone de dos módulos: un módulo de planificación de caminos y un módulo de control. El módulo de planificación de caminos usa la información del mapa provista por el algoritmo de SLAM para trazar un camino seguro y compatible con la silla de ruedas, que le permita al vehículo alcanzar la orientación deseada. El módulo de control gobierna el movimiento del vehículo en el seguimiento del camino trazado mediante un controlador de seguimiento de trayectorias. Las referencias del controlador son actualizadas mediante la estimación de la postura de la silla de ruedas dentro del ambiente, obtenida por el algoritmo de SLAM. Acompañan a este trabajo resultados experimentales utilizando una silla de ruedas robotizada real. Copyright © 2011 CEA.

Palabras Clave: Vehículos autónomos, Sistemas biomédicos, Navegación de robots.

1. INTRODUCCIÓN

En este trabajo se presenta el diseño y experimentación de una interfaz para la navegación autónoma y no-autónoma de una silla de ruedas robotizada. La navegación no-autónoma se ejecuta mediante un *joystick* manual. El *joystick* permite el control directo de la silla de ruedas en entornos que no presenten restricciones o dificultades de maniobrabilidad. La navegación autónoma permite la maniobrabilidad de la silla de ruedas robotizada dentro de entornos restringidos. La estrategia de maniobrabilidad permite el giro –cuyo ángulo es previamente especificado por el usuario– de forma segura, protegiendo la integridad del usuario y del vehículo.

La maniobrabilidad y gobernabilidad de vehículos de asistencia dependen actualmente de las habilidades del usuario. Aún más, los movimientos de los vehículos de ambientes internos se encuentran restringidos por la estructura y disposición del entorno. Los vehículos de asistencia, en hospitales o centros de rehabilitación, son usados para el acarreo de medicina, asistencia a los ancianos o a personas con capacidades motoras limitadas y para el transporte de usuarios (Park *et al.*, 2004; Dario *et al.*, 1999). Aunque estos vehículos constituyen una solución al servicio de asistencia al usuario, la falta de in-

teligencia de los mismos para sobrellevar de forma autónoma situaciones inesperadas, o moverse dentro de ambientes desconocidos hacen que su funcionalidad se vea limitada a comportamientos básicos dependientes del usuario. Así, un giro de 180 grados se puede volver una tarea frustrante para el usuario, quien, por las restricciones motoras de sus capacidades físicas, podría no estar en condiciones de mirar hacia atrás. Este tipo de giro requiere de habilidades extras por parte del usuario para garantizar un movimiento seguro en reversa. Aunque las sillas de ruedas robotizadas pueden girar sin usar movimientos en reversa (dado principalmente a que la mayoría de las sillas de ruedas robotizadas tienen modelos cinemáticos tipo unicycle (Martins *et al.*, 2008)), en ambientes restringidos el movimiento en reversa podría ser necesario de manera tal de alcanzar zonas con mayores espacios libres donde realizar los movimientos de giro sin colisionar con elementos del entorno.

El campo de la robótica ofrece un amplio espectro de soluciones que pueden ser aplicadas en las áreas de asistencia y rehabilitación. Más aún, las herramientas desarrolladas para la robótica móvil se pueden aplicar directamente en vehículos de asistencia para mejorar sus tareas asistivas. De esta manera, diversas tareas pueden ser relegadas del control del usuario para ser comandadas autónomamente, optimizando tiempos y recur-

del vehículo, como así también disminuyendo el esfuerzo realizado por el usuario.

Generalmente, los vehículos móviles se circunscriben por un círculo, tal que el vehículo se puede considerar como un punto y los obstáculos del entorno son evitados de acuerdo a la cercanía de los mismos con el límite del círculo circunscribiente (Siegwart and Nourbahsh, 2004). Esta consideración no puede ser utilizada en sillas de ruedas robotizadas destinadas a maniobrar en espacios restringidos, dado que el círculo que la circunscribiría podría llegar a ser demasiado grande y algunos pasillos o corredores se volverían innavegables (por ejemplo, la silla de ruedas presentada en Bourhis *et al.* (2001) tiene 0.7 metros de ancho por 1.2 metros de largo y debe navegar por pasillos de 1.5 metros de ancho). Debido a las dimensiones de una silla de ruedas, si la misma, por ejemplo, se encuentra cerca de una pared, no podría rotar libremente dado que su parte frontal o trasera podría colisionar con el muro.

El algoritmo de SLAM (por sus siglas en inglés de *Simultaneous Localization and Mapping*), se ha convertido en una de las áreas más desafiantes dentro del campo de la robótica (Thrun *et al.*, 2005; Whyte and Bailey, 2006b). La auto-localización de vehículos es una parte fundamental de la navegación autónoma: un vehículo debe ser capaz de estimar su posición y orientación (pose) dentro del ambiente por el cual navega (la Cruz *et al.*, 2010). Sin embargo, en varias aplicaciones de relevancia práctica, tales como tareas explorativas, no se dispone de un mapa del entorno, o la información previa del mismo contiene demasiada incertidumbre como para tomar decisiones autónomas seguras. En estos casos, el robot debe usar las mediciones obtenidas por su propio equipo sensorial para estimar un mapa del ambiente y, al mismo tiempo, localizarse dentro de ese mapa. Esto se conoce como el *problema de SLAM*.

Se pueden encontrar diversas soluciones al problema de SLAM, tales como el *filtro Extendido de Kalman*, o EKF (por sus siglas en inglés de *Extended Kalman filter*), donde la pose del vehículo y los elementos del entorno se modelan como una sola variable aleatoria de distribución Gaussiana. Debido a que el EKF trabaja con modelos linealizados del entorno, el UKF (por sus siglas en inglés de *Unscented Kalman Filter*) surge como una solución a dicha no-linealidad (Thrun *et al.*, 2005; Whyte and Bailey, 2006b,a). Para compensar el costo computacional asociado a la etapa de corrección del EKF, se presenta al filtro de Información (FI) como una posible alternativa (Thrun *et al.*, 2005). Además, el filtro de partículas (FP) es una solución a la restricción Gaussiana del SLAM. El FP permite tratar al entorno y al robot como variables aleatorias con cualquier distribución de probabilidad asociada a las mismas (Thrun *et al.*, 2005; Liu *et al.*, 2008; Dellaert *et al.*, 1999).

Las diversas soluciones al problema del SLAM comparten un mismo inconveniente: a medida que se construye el mapa del entorno aumentan las características extraídas del mismo, lo que ocasiona un aumento en el tamaño del vector de estados del sistema. Un aumento en la dimensión del vector de estados del sistema implica mayor costo computacional, lo que podría hacer inviable una implementación del SLAM en tiempo real (Bailey *et al.*, 2006), (Guivant and Nebot, 2001).

El principal aporte de este trabajo se concentra en brindar a los usuarios que tengan restricciones físicas para mirar hacia atrás (por ejemplo ancianos) una navegación automática en espacios reducidos, ya que numerosas veces es necesario un movimiento en reversa cuando la silla de ruedas tiene que

maniobrar en este tipo de entornos. El sistema propuesto es la combinación de varias técnicas avanzadas de navegación robótica y una interfaz gráfica. La interfaz gráfica propuesta tiene el objetivo de permitir al usuario el ingreso de comandos utilizando solamente el joystick sin la necesidad de botones extras, esto es más útil aún si se tiene pacientes que solo pueden comandar la silla de ruedas con movimientos de cabeza.

Así, en este trabajo se presenta una interfaz que combina un sistema de navegación y maniobrabilidad autónoma aplicado a una silla de ruedas robotizada. La estrategia de maniobrabilidad de la silla de ruedas está diseñada para la planificación de caminos dentro de entornos restringidos. La interfaz está comandada por un joystick manual, aunque el mismo puede ser adaptado según las capacidades del usuario. El joystick se usa para dirigir los movimientos de la silla de ruedas dentro del ambiente. Si el usuario necesita girar dentro del entorno, elige primero el ángulo de giro en la interfaz visual y luego un sistema de navegación autónomo, compuesto por un módulo de planificación de caminos y un módulo de control, gobiernan el movimiento de la silla hasta que alcanza la orientación deseada. Tanto el módulo de planificación como el módulo de control, usan la información provista por el algoritmo de SLAM. La planificación de caminos usa el mapa obtenido por el SLAM para una planificación segura, mientras que el controlador de trayectorias usa la estimación de la pose del vehículo dentro del entorno como realimentación de las referencias de control. El algoritmo de SLAM está implementado en un EKF y extrae líneas y esquinas –cóncavas y convexas– del ambiente por medio de un sensor de rango láser (construido por SICK). Considerando que la silla de ruedas robotizada no cuenta con sensores de rango en la parte de atrás de su chasis, el algoritmo de SLAM provee de la información del ambiente y de la pose del vehículo dentro del mapa como referencias para el controlador durante el movimiento de reversa. La estrategia de planificación provee de las velocidades lineal y angular a una silla de ruedas robotizada virtual, la cual genera la trayectoria deseada. Las velocidades lineal y angular se mantienen constantes hasta que la silla virtual alcanza un entorno del límite del espacio maniobrable y navegable. Cada vez que la silla virtual alcanza una vecindad del límite navegable y maniobrable, otro par de velocidades lineal y angular son provistas al vehículo virtual pero con sus direcciones opuestas. Este procedimiento se repite sucesivamente hasta que la silla virtual alcanza la orientación deseada. Una vez que se obtiene un camino factible, un nuevo camino es buscado con diferentes valores de velocidad lineal y angular. La magnitud de la velocidad lineal virtual es siempre una función de la velocidad angular virtual (así, a mayor velocidad angular, menor velocidad lineal, protegiendo la integridad del usuario). La estrategia de planificación se basa en el método de Monte Carlo para encontrar el camino mínimo dentro del ambiente restringido usando un vehículo virtual. Una vez que dicho camino mínimo es encontrado, un controlador de seguimiento de trayectorias gobierna a la silla de ruedas robotizada hasta que alcanza la orientación deseada. El sistema propuesto está implementado en una silla de ruedas robotizada real. Se presenta además el análisis de los resultados experimentales de cada una de las etapas de la propuesta. Se presentan también los resultados estadísticos de probar el sistema propuesto en una población de 7 usuarios cognitivamente normales.

2. TRABAJO RELACIONADO

Los sistemas de sillas de ruedas robotizadas integran un subsistema de sensoramiento, un módulo de navegación, un módulo de control y una interfaz Hombre-máquina para guiar la silla de ruedas en modo automático o semi-automático (Montesano *et al.*, 2010; Bourhis *et al.*, 2001; Zeng *et al.*, 2008; Parikh *et al.*, 2007). En el modo automático, la persona con capacidades motoras reducidas solo tiene que escoger en un monitor el destino deseado, usando cualquiera de las siguientes alternativas para la interfaz Hombre-máquina: *joysticks* controlados por la mano o barbilla, dispositivos de exhalación, comandos de voz, movimiento ocular o movimiento de cabeza. Otras alternativas para la interfaz Hombre-máquina pueden ser señales cerebrales (Ferreira *et al.*, 2008). Luego de escoger el destino, la silla de ruedas robotizada se dirige automáticamente al destino sin la participación del usuario en el control. Por otro lado, en el modo semi-automático el usuario comparte el control con la silla de ruedas robotizada utilizando cualquiera de las interfaces Hombre-máquina mencionadas. En este caso, son necesarias solo algunas habilidades del usuario. Por ejemplo, el sistema automático puede interactuar con el usuario para desviar obstáculos mientras que el usuario guía a la silla de ruedas hacia el destino. En este caso, se alivia al usuario de la tarea de desviar obstáculos, que puede ser dificultoso cuando tiene restringidas sus habilidades.

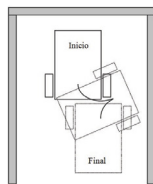


Figura 1. Planificación de trayectoria en espacio restringido.

Los trabajos (Auat Cheein *et al.*, 2010c; la Cruz *et al.*, 2010; Auat Cheein *et al.*, 2010a) tratan el problema de la navegación en espacios restringidos para vehículos tipo carro y sillas de ruedas. El enfoque que utilizan es basado en cambios de las direcciones del movimiento. Cada vez que el robot virtual, utilizado en la planificación de trayectoria, llega cerca del punto de colisión, se cambia la dirección del movimiento. Las subtrayectorias utilizadas son arcos de radio constante o variable. La Fig. 1 muestra el caso más simple, el cual consiste en planificar la trayectoria utilizando arcos de radio constante. En el ejemplo de la Fig. 1, la silla de ruedas no podría girar libremente sobre su eje para rotar 180 grados porque la parte delantera colisionaría con la pared. El uso de arcos con radio variable generaliza este método de planificación. El problema es elegir el radio ideal para cada arco de la trayectoria, el cual puede variar de cero a infinito, con el fin de optimizar la trayectoria de acuerdo a algún funcional de costo. Este problema se soluciona utilizando el método de Monte Carlo (Auat Cheein *et al.*, 2010c; la Cruz *et al.*, 2010).

Los trabajos relacionados que tratan la navegación en entornos restringidos, donde el espacio libre es reducido, (Auat Cheein *et al.*, 2010c; la Cruz *et al.*, 2010; Auat Cheein *et al.*, 2010a) no integran estos sistemas de navegación con una interfaz al usuario.

3. ARQUITECTURA GENERAL DEL SISTEMA

La Fig. 2 muestra la arquitectura del sistema de navegación propuesta en este trabajo. Se considera que se usa el control manual (mediante el *joystick*) de la silla de ruedas hasta que se requiere el control automático debido a las restricciones del ambiente. La estrategia de navegación y el funcionamiento de la interfaz se resumen de la siguiente manera:

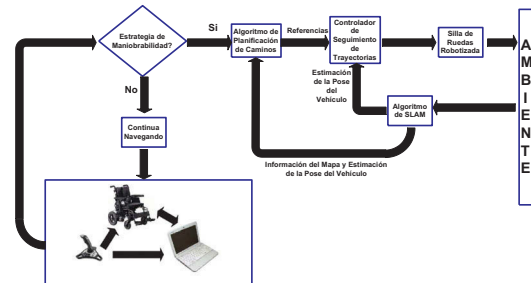


Figura 2. Arquitectura general del sistema.

- Mientras el vehículo es gobernado por el *joystick* dentro del ambiente, el algoritmo de SLAM es continuamente ejecutado, adquiriendo de esta forma las características del entorno –líneas y esquinas– y localizando al robot dentro de dicho mapa al mismo tiempo.
- Una vez que se alcanza un lugar del entorno donde el usuario debe maniobrar, primero escoge el ángulo de giro. Una vez elegido el ángulo de giro, se invoca el algoritmo de maniobrabilidad autónoma.
- El algoritmo de maniobrabilidad planifica el camino a seguir por la silla de ruedas para alcanzar la orientación deseada usando la información del mapa y la pose del vehículo provista por el algoritmo de SLAM. El camino final obtenido es libre de obstáculos. Los beneficios del SLAM surgen cuando la silla de ruedas debe maniobrar en reversa y necesita la información del mapa circundante (dado que el vehículo no posee sensores de proximidad o de rango en la parte de atrás de su chasis).
- Una vez encontrado un camino viable, se ejecuta un controlador cinemático de seguimiento de trayectorias, para guiar al vehículo hasta alcanzar la orientación deseada.
- Cuando la silla de ruedas robotizada alcanza la orientación deseada por el usuario, el algoritmo de maniobrabilidad autónoma detiene su ejecución. El control de vehículo retorna al modo no-autónomo. Durante la ejecución autónoma, el sistema permite una parada de emergencia gobernada por el usuario. Así, si el usuario decide cancelar la maniobra, dicha cancelación puede ejecutarse con el *joystick*.

En las secciones siguientes se explicarán en detalle cada bloque de la Fig. 2.

4. SILLA DE RUEDAS ROBOTIZADA

La silla de ruedas robotizada utilizada en los experimentos fue desarrollada en el Departamento de Ingeniería Eléctrica de la Universidade Federal do Espírito Santo - Brasil (ver Fig. 3). La arquitectura hardware de la silla de ruedas robotizada, mostrada en la Fig. 4, consiste de: 1) una silla de ruedas eléctrica comercial, del cual solo son utilizados la estructura mecánica y los motores; 2) dos encoders directamente conectados a los ejes de los motores; 3) un microcontrolador MSP430F1611 de

Texas Instruments, donde es implementado el control de bajo nivel; 4) una tarjeta de potencia que amplifica la señal PWM obtenida del microcontrolador y la envía a los motores; 5) una computadora, donde son implementados los algoritmos de alto nivel; y 6) un sensor láser de distancias. Los algoritmos de alto nivel son implementados bajo el sistema operativo Windows utilizando el lenguaje de programación Microsoft Visual C++.

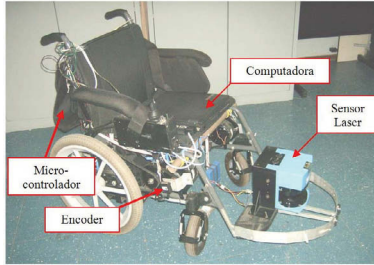


Figura 3. Silla de ruedas robotizada.

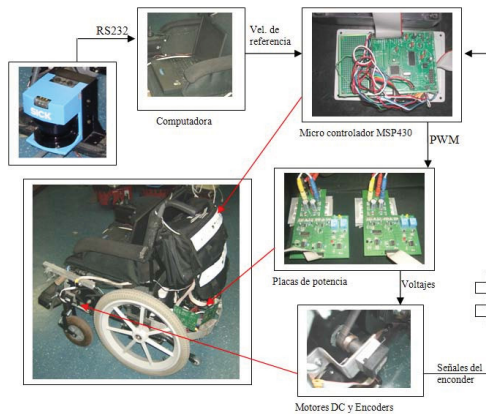


Figura 4. Arquitectura mecatrónica de la silla de ruedas robotizada.

Como se puede ver la silla de ruedas es del tipo tracción diferencial o uniclo. Las dos ruedas delanteras son ruedas castor. Por lo tanto, el giro del vehículo está dado por la diferencia de las velocidades de las ruedas traseras. La configuración de los vehículos tipo uniclo es de sistemas no-holonomicos. No-holonomía implica que, en el plano, la velocidad está restringida en una dimensión mientras que la posición no lo está. En efecto, la velocidad en estos vehículos está restringida lateralmente. Como consecuencia, el control de este tipo de vehículos se dificulta considerablemente.

5. MODO NO-AUTÓNOMO: INTERFAZ VISUAL

Debido a que el sistema propuesto en este trabajo brinda a los usuarios que tengan alguna restricción física para mirar hacia atrás una navegación automática en espacios reducidos para gobernar la silla de ruedas robotizada, la interfaz que usada entre los usuarios y el robot es una interfaz gráfica. Esta interfaz gráfica (o visual) se encuentra gobernada únicamente por un *joystick* lo que permite el ingreso de comandos sin la necesidad de usar botones extras. El *joystick* puede ser de cualquier tipo: comandado por movimientos de cabeza, comandado por la boca del usuario, un *joystick* manual, etc., dependiendo de las necesidades del paciente. En este trabajo, se usó para fines experimentales un *joystick* del tipo manual.

La interfaz visual permite al usuario escoger la orientación deseada utilizando el *joystick*. Por ejemplo, escoger un ángulo deseado como 90 o 180 grados. Dado que el *joystick* también cumple las funciones de comandar el movimiento de la silla de ruedas, es necesaria una máquina de estados especialmente diseñada para este sistema con el objeto de interpretar los comandos del *joystick*. Para tal fin, se definen las siguientes variables: sea la variable y_J igual al valor de la inclinación del *joystick* hacia delante; un valor negativo de y_J indica una inclinación del *joystick* hacia atrás; sea la variable x_J igual al valor de la inclinación del *joystick* hacia la derecha; similarmente, un valor negativo de x_J indica una inclinación del *joystick* hacia la izquierda. En la Fig. 5 se muestra el diagrama de transición de estados de esta máquina de estados, donde α_J y β_J son constantes positivas tal que $\alpha_J > \beta_J$, $m_J = \sqrt{x_J^2 + y_J^2}$ y $\theta_J = \text{atan2}\{-x_J, y_J\}$. Las variables m_J y θ_J se definen como la magnitud y orientación del *joystick*, respectivamente. Inclinaciones del *joystick* hacia la derecha resultan en valores negativos de θ_J y hacia la izquierda resultan en valores positivos de θ_J . Una inclinación del *joystick* hacia adelante resulta el valor cero de θ_J .

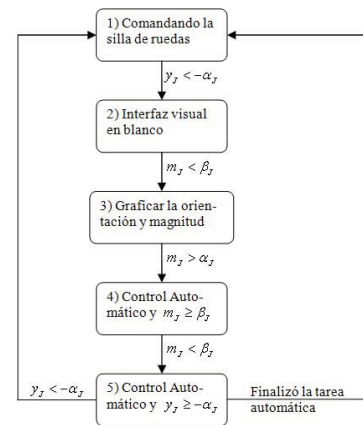


Figura 5. Máquina de estados de la Interfaz Visual.

La máquina de estados inicia en el estado "1". Dado que se necesita una señal del *joystick* para pasar del modo manual al modo automático, se utiliza el movimiento del *joystick* hacia atrás para cambiar de modo. Los demás movimientos comandarán la silla de ruedas de manera convencional cuando se está en el modo manual. El movimiento de la silla de ruedas hacia atrás se realizará de manera especial. El movimiento hacia atrás será de muy baja velocidad proporcional al *joystick* cuando $-\alpha_J \leq y_J < 0$. Cuando $y_J < -\alpha_J$ entonces se pasa del estado "1" al estado "2". El estado "2" es la interfaz visual en blanco indicando que se está en un estado que no produce ninguna acción. Este estado es necesario para dar tiempo al usuario de mover el *joystick* cerca del estado perpendicular y estar listo para dar otros comandos. Cuando $m_J < \beta_J$ (el *joystick* volvió a estar cerca del estado perpendicular) se pasa al estado "3", en el cual se grafica la orientación θ_J y magnitud m_J del *joystick* (ver Fig. 6). Luego, cuando la línea roja es más larga que la línea negra ($m_J > \alpha_J$), se pasará al siguiente estado "4" que es el estado donde se ejecuta el modo automático para llegar a la orientación seleccionada. La orientación deseada seleccionada es igual a la mostrada en la interfaz visual y es relativa a la orientación actual de la silla de ruedas. Cuando $m_J < \beta_J$ (el *joystick* volvió a estar cerca al estado perpendicular) se pasa al estado "5" donde está habilitado para cambiar

al modo manual una vez terminada la tarea automática o para cancelar el modo automático inclinando el *joystick* hacia atrás en un valor mayor a un umbral.

En el estado "4", el sistema automático no realizará ninguna tarea cuando la orientación deseada es cercana a cero en relación a la orientación actual; es decir, previamente se seleccionó la orientación marcada en gris (ver Fig. 6). Luego, el sistema volverá al modo manual inmediatamente después de pasar al estado "5". Por lo tanto, seleccionar una orientación cercana a cero equivale a cancelar el modo automático.



Figura 6. Interfaz visual. La línea roja muestra la magnitud y orientación del *joystick* y la línea negra muestra la magnitud que debe superar la línea roja para seleccionar la orientación deseada. La zona gris marca la zona donde la orientación es cercana a cero.

6. MODO AUTÓNOMO: ALGORITMO DE PLANIFICACIÓN

Una vez que el usuario elige el ángulo de orientación final que deberá tener la silla de ruedas, el algoritmo de planificación usa la información provista por el SLAM para generar un camino seguro y cinemáticamente compatible, que le permita a la silla de ruedas alcanzar dicha orientación. El algoritmo de SLAM provee información del mapa del entorno y de la pose de la silla de ruedas robotizada dentro de ese mapa. Dicho algoritmo de SLAM se encuentra implementado en un EKF, lo que permite además un modelado Gaussiano del entorno. Así, es posible asignar a cada punto del entorno mapeado un valor de probabilidad.

Con la información geométrica y probabilística del entorno, el algoritmo de planificación determina el espacio de trabajo seguro de la silla de ruedas. Según sea el espacio de trabajo y la distribución del mismo, el algoritmo de planificación genera, iterativamente, posibles caminos compatibles con la cinemática del vehículo. Estos caminos conducirían a la silla de ruedas a alcanzar la orientación deseada por el usuario. Tanto el algoritmo de SLAM como el de planificación serán explicados en detalle en las siguientes sub-secciones.

6.1 EKF-SLAM

El algoritmo de SLAM implementado en este trabajo se basa en el filtro Extendido de Kalman (EKF). El vector de estados del SLAM se compone de la estima de la pose del vehículo –posición y orientación dentro del entorno– y de las características extraídas del ambiente. Las características extraídas del ambiente corresponden a líneas –asociadas a paredes del entorno– y esquinas –cóncavas y convexas–. Para propósitos de visualización y reconstrucción posterior del mapa, al mismo tiempo que se ejecuta el algoritmo de SLAM, se mantiene operativo un segundo mapa. Este mapa secundario almacena las coordenadas globales de los extremos de los segmentos asociados a las líneas del entorno. De esta manera, el mapa

secundario permite una representación de paredes limitadas. Además, el mapa secundario se mantiene actualizado y corregido de acuerdo a la corrección del vector de estados en el algoritmo de EKF-SLAM. Así, si una nueva característica se adiciona al vector de estados del SLAM, se adiciona también al mapa secundario (Auat Cheein *et al.*, 2009).

Las ecuaciones (1) y (2) muestran la estructura del vector de estados del SLAM y su matriz de covarianza. Todos los elementos del vector de estados del SLAM están referenciados a un marco de coordenadas global al sistema.

$$\hat{\xi}_t = \begin{bmatrix} \hat{\xi}_{v,t} \\ \hat{\xi}_{m,t} \end{bmatrix} \quad (1)$$

$$P_t = \begin{bmatrix} P_{vv,t} & P_{vm,t} \\ P_{vm,t}^T & P_{mm,t} \end{bmatrix} \quad (2)$$

En la Ec. (1), $\hat{\xi}_t$ es el vector de estados del sistema; $\hat{\xi}_{v,t} = [\hat{\xi}_{x,t} \ \hat{\xi}_{y,t} \ \hat{\xi}_{\theta,t}]^T$ es la pose estimada del vehículo, donde $\hat{\xi}_{x,t}$ y $\hat{\xi}_{y,t}$ representan la posición global de la silla de ruedas dentro del ambiente y $\hat{\xi}_{\theta,t}$ es su orientación; $\hat{\xi}_{m,t}$ representa el mapa del entorno y está compuesto por las líneas y esquinas extraídas del ambiente (en este trabajo, las esquinas se definen en coordenadas Cartesianas, mientras que las líneas en coordenadas polares). El orden según el cual las líneas o esquinas aparecen en $\hat{\xi}_{m,t}$ depende del momento en que dichas características hayan sido detectadas. P_t es la matriz de covarianza asociada al vector de estados del SLAM; $P_{vv,t}$ es la covarianza de la estimación de la pose de la silla de ruedas robotizada y $P_{mm,t}$ es la covarianza asociada al mapa. $P_{vm,t}$ y $P_{vm,t}^T$ son matrices de correlación cruzada entre el vehículo y las características presentes en el mapa.

La inicialización de las variables involucradas en el algoritmo de SLAM como también las ecuaciones que lo desarrollan se pueden ver en (Whyte and Bailey, 2006b; Thrun *et al.*, 2005; Auat Cheein *et al.*, 2009).

Tal como se explicó en la sección 4, la silla de ruedas robotizada usada en este trabajo tiene un modelo cinemático equivalente al de un robot móvil tipo unicycle, tal como se muestra en la Fig. 7. La ecuación cinemática de este robot se muestra en la Ec. (3).

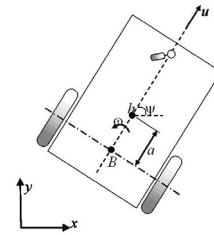


Figura 7. Representación gráfica de un vehículo con modelo cinemático tipo unicycle.

$$\begin{bmatrix} \xi_{x,t} \\ \xi_{y,t} \\ \xi_{\theta,t} \end{bmatrix}_G = \begin{bmatrix} \xi_{x,t-1} \\ \xi_{y,t-1} \\ \xi_{\theta,t-1} \end{bmatrix} + \Delta t \begin{bmatrix} \cos(\xi_{\theta,t-1}) & 0 \\ \sin(\xi_{\theta,t-1}) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_t \\ \omega_t \end{bmatrix} + \Phi_t \quad (3)$$

En la Ec. (3), $\xi_{x,t}$, $\xi_{y,t}$ y $\xi_{\theta,t}$ son las coordenadas del punto B de control del vehículo en la Fig. 7. Φ_t es el ruido Gaussiano asociado al modelo del vehículo; u_t es el comando de velocidad

lineal aplicado al robot mientras que ω_t es el comando de velocidad angular; Δt es el tiempo de muestreo del sistema (en este trabajo, se adoptó $\Delta t = 0,1$ segundos). El sufijo G en la Ec. (3) indica que las coordenadas del vehículo están representadas en un marco de referencia global del ambiente.

Los modelos asociados a las características del entorno –líneas y esquinas– se pueden consultar en otro trabajo de los autores (Auat Cheein *et al.*, 2009). En (Auat Cheein *et al.*, 2009) se muestra además que las esquinas están modeladas en el espacio Cartesiano mientras que las líneas se modelan en el espacio polar.

De forma tal de evitar el manejo innecesario de información que no se relaciona al problema de maniobrabilidad de la silla de ruedas en espacios restringidos, el vector de estados del SLAM, como así también su matriz de covarianza, se truncan cada vez que una característica queda fuera del rango de maniobrabilidad. Este método de truncamiento no afecta la consistencia del mapa tal como está demostrado en (Whyte and Bailey, 2006b,a). Así, sea por ejemplo la reconstrucción parcial de las instalaciones del *Departamento de Engenharia Elétrica* de la Universidade Federal do Espírito Santo, Brasil. La Fig. 8 muestra la navegación de la silla de ruedas y la reconstrucción del mapa correspondiente al entorno navegado (los segmentos negro sólidos se asocian con las paredes del entorno; los círculos verdes corresponden a esquinas mientras que los puntos magenta son los datos del sensor de rango láser). En la Fig. 8, el rectángulo rojo sólido indica la zona dentro de la cual el robot debe maniobrar. El tamaño de esta zona se determina por diseño. Toda característica del entorno que no se encuentre dentro del rectángulo rojo, es eliminada tanto del vector de estados del SLAM como de la matriz de covarianzas (este proceso se conoce como *truncamiento* (Thrun *et al.*, 2005)). De esta manera, las dimensiones del vector de estados del SLAM permanecen pequeñas en comparación a la dimensión del entorno por el cual navega la silla de ruedas robotizada.

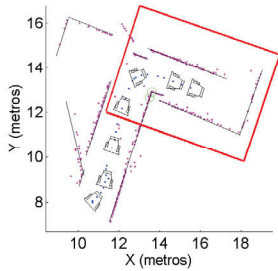


Figura 8. Ejemplo de mapa construido por el EKF-SLAM.

Los segmentos negro sólidos corresponden a paredes y puertas del entorno; los círculos verdes a esquinas y los puntos magenta son los datos del sensor de rango láser; el camino realizado por la silla de ruedas robotizada está en línea punteada azul. El rectángulo rojo indica la zona del entorno que se restringe para maniobrabilidad.

Mapa probabilístico del entorno

La reconstrucción geométrica del entorno navegado a partir de la información provista por el algoritmo de SLAM no es exacta. De hecho, el EKF-SLAM implementado en este trabajo sólo extrae las líneas y esquinas del ambiente. De esta manera, el mapeo de las paredes del entorno puede no ser

geométricamente preciso al, por ejemplo, no mapear toda la longitud de la pared. Este tipo de situaciones crea vacíos dentro del mapa geométrico que, cuando la silla de ruedas ejecuta un movimiento de reversa, pueden causar la colisión de la misma. Es por ello que en este trabajo se utilizó el mapa probabilístico asociado al mapa geométrico obtenido por el algoritmo de SLAM, como una forma de determinar el espacio libre de maniobrabilidad de la silla de ruedas.

El procedimiento empleado en este trabajo para obtener un mapa probabilístico a partir de un mapa geométrico, se basa en aprovechar la distribución Gaussiana de las características mapeadas del entorno. El procedimiento se resume de la siguiente manera:

- i Obtener el mapa geométrico a partir de la información provista por el vector de estados del SLAM y por el mapa secundario presentado en la sección 6.1.
- ii Dividir el entorno mapeado en celdas. Este procedimiento se conoce como *rejillado del mapa* (Thrun *et al.*, 2005).
- iii Considerando que las características mapeadas son funciones de distribución Gaussianas, entonces cada característica del ambiente influye en la probabilidad de ocupación de todas las celdas del mapa rejillado.
- iv Asignar al centro de masa de cada celda la probabilidad de ocupación de acuerdo a cada característica del entorno. Así, una celda cualquiera del mapa rejillado, tendrá un valor de probabilidad dado por la i -ésima esquina y otro valor dado por la j -ésima línea.
- v Aplicar la *suma de Gaussianas* para obtener la probabilidad total asociada a cada celda del mapa.

Así, por ejemplo, dado el punto $p_{x,y} = [p_x \ p_y]^T$ del espacio mapeado, la probabilidad de ocupación de ese punto con respecto a la i -ésima esquina viene dada por la Ec. (4). La Ec. (4) es la expresión general de una distribución Gaussiana, donde los valores medios son extraídos del vector de estados del SLAM. La varianza, se extrae de la matriz de varianzas del vector de estados del algoritmo de SLAM.

$$P(p_{x,y})_{esquina,i} = \frac{1}{\sqrt{(2\pi)^2 |\Psi_i|}} e^{-\frac{1}{2} \left(\begin{bmatrix} p_x \\ p_y \end{bmatrix} - \begin{bmatrix} \hat{\xi}_{i,x} \\ \hat{\xi}_{i,y} \end{bmatrix} \right)^T \Psi_i^{-1} \left(\begin{bmatrix} p_x \\ p_y \end{bmatrix} - \begin{bmatrix} \hat{\xi}_{i,x} \\ \hat{\xi}_{i,y} \end{bmatrix} \right)} \quad (4)$$

En la Ec. (4), $[\hat{\xi}_{i,x} \ \hat{\xi}_{i,y}]^T$ son los valores predichos de la i -ésima esquina adquirida por el algoritmo de SLAM; Ψ_i es la matriz de covarianza asociada a la i -ésima esquina.

La probabilidad de un punto respecto a una línea del entorno se puede escribir como se muestra en la Ec. (5).

$$P(p_{x,y})_{linea,j} = \frac{1}{\sqrt{(2\pi)^2 |\Psi_j|}} e^{-\frac{1}{2} (\Upsilon - \Gamma)^T \Psi_j^{-1} (\Upsilon - \Gamma)}. \quad (5)$$

En la Ec. (5), Γ contiene los parámetros de la j -ésima línea extraídos del vector de estados del SLAM, mientras que Ψ_j es su matriz de covarianza asociada. El vector Υ contiene los parámetros del punto del espacio $p_{x,y}$ convertidos a coordenadas polares. Esto es necesario dado que la función de distribución asociadas a las líneas se expande sobre el dominio polar, mientras que la función de distribución asociada a una

esquina, se expande sobre el dominio Cartesiano. Ambos valores de probabilidad, representan la probabilidad de ocupación del punto dentro del espacio mapeado (ver (Auat Cheein *et al.*, 2010c) para más información).

Una vez que se tienen los valores de probabilidad de una celda asociados a todas las características del mapa, se aplica el método de *suma de Gaussianas* (Miralles and Bobi, 2004) para obtener un solo valor representativo de dicha probabilidad de ocupación. Le Ec. (6) muestra la expresión completa del método.

$$P(p_{x,y}) = \sum_{k=1}^{N_L} \frac{\varepsilon_{L,k}}{\sqrt{(2\pi)^2 |\Psi_{L,k}|}} e^{-\frac{1}{2}(\gamma_{L,k} - \Gamma_{L,k})^T \Psi_{L,k}^{-1} (\gamma_{L,k} - \Gamma_{L,k})} + \sum_{k=1}^{N_E} \frac{\varepsilon_{E,k}}{\sqrt{(2\pi)^2 |\Psi_i|}} e^{-\frac{1}{2} \left(\begin{bmatrix} p_x \\ p_y \end{bmatrix} - \begin{bmatrix} \hat{\xi}_{i,x} \\ \hat{\xi}_{i,y} \end{bmatrix} \right)^T \Psi_i^{-1} \left(\begin{bmatrix} p_x \\ p_y \end{bmatrix} - \begin{bmatrix} \hat{\xi}_{i,x} \\ \hat{\xi}_{i,y} \end{bmatrix} \right)} \quad (6)$$

En la Ec. (6), N_L es el número de líneas detectadas en el entorno mientras que N_E es el número de esquinas. Los coeficientes $\varepsilon_{L,k}$ y $\varepsilon_{E,k}$ son los coeficientes de normalización (Miralles and Bobi, 2004). Como se puede ver, el método de *suma de Gaussianas* es una suma pesada de distribuciones. La *suma de Gaussianas* fusiona la información de probabilidad dada por las diferentes características del entorno respecto a un mismo punto del espacio. Finalmente, este método se aplica a todas las celdas del mapa. La Fig. 9 muestra la construcción del mapa de probabilidad basado en la información obtenida por el SLAM mostrado en la Fig. 8. En la Fig. 9, un valor de probabilidad bajo correspondiente a un punto dado dentro del espacio mapeado, implica que la probabilidad de que el punto se encuentre ocupado por algún elemento del entorno es también baja.

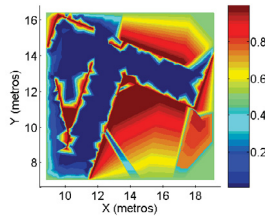


Figura 9. Mapa de probabilidad asociado al entorno reconstruido en la Fig. 8. Las zonas azules representan a las celdas del espacio con menor probabilidad de estar ocupadas.

La zona del espacio mapeado que será tenida en cuenta durante la planificación de caminos, será aquella zona *navegable*, es decir, cuyas celdas tengan una probabilidad de ocupación baja. En la Fig. 9, la zona navegable del espacio corresponde a la zona azul. En este trabajo se considerará que un punto (o celda) del mapa posee una probabilidad baja de ocupación, cuando la *suma de Gaussianas* asociadas a dicho punto sea menor o igual a 0.2, es decir, $P(p_{x,y}) \leq 0,2$.

6.2 Planificación de caminos

El método de planificación de caminos usado en este trabajo se basa en el método de Monte Carlo (Auat Cheein *et al.*, 2010c).

Este método encuentra un camino mínimo de acuerdo a una determinada función de costo. La función de costo usada en este trabajo está relacionada a la energía cinética que la silla de ruedas robotizada disiparía si siguiera ese camino.

El método de Monte Carlo permite generar los comandos de control de un vehículo virtual asociado a la silla de ruedas robotizada. Este vehículo virtual posee la misma cinemática que la silla de ruedas. Cada comando de control generado determina un arco. Este arco se calcula de acuerdo al modelo cinemático del vehículo (ver Eq. (3)). Como se explicó en la sección 4, se puede considerar a la silla de ruedas motorizada como un robot móvil tipo unicycle, no-holonómico (Siegwart and Nourbahsh, 2004).

El mapa probabilístico del entorno obtenido mediante el algoritmo de SLAM y el método de *suma de Gaussianas* se usan para generar un camino seguro y libre de obstáculos dentro del espacio navegable, de forma tal que le permita a la silla de ruedas alcanzar la orientación deseada. Como se explicó en la sección 6.1, el vector de estados del SLAM –y por lo tanto su covarianza– se ven truncados cuando una característica se encuentra lo suficientemente distante del robot como para poder ser descartada. En este trabajo, esa distancia fue establecida en 2 metros, aunque este valor depende del criterio de diseño. Así, toda característica no comprendida en un rango de dos metros desde la pose del robot, es eliminada del vector de estados del SLAM (si parte de una línea se encuentra dentro del rectángulo, entonces se conserva toda la línea). Este proceso de eliminación (ver Fig. 8) también restringe el espacio maniobrable del espacio navegable contenido dentro del rectángulo usado en la Fig. 8. La Fig. 10a muestra el rectángulo virtual aplicado al mapa de probabilidad del entorno presentado en la Fig. 8, mientras que la Fig. 10b muestra solamente el espacio navegable final que será tenido en cuenta para la planificación de caminos. En la Fig. 10, el robot se dibuja como un triángulo sólido blanco, mientras que el rectángulo que circunscribe el ambiente navegable, está representado por líneas sólidas blancas.

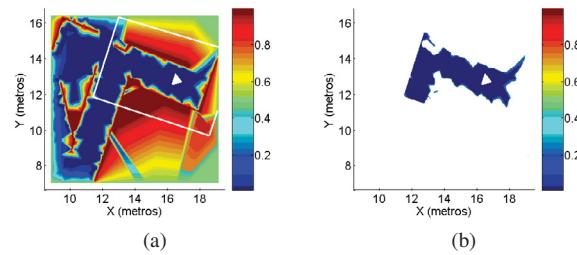


Figura 10. Determinación del espacio navegable y maniobrable a partir del mapa de probabilidad y del mapa geométrico generado por el algoritmo de SLAM. a) El mapa general es truncado cuando una característica se encuentra lo suficientemente alejada del robot; b) espacio navegable y maniobrable luego del truncamiento (en azul).

Cabe aclarar que la Fig. 10 es sólo representativa del proceso de truncamiento, dado que el mapa de probabilidad se calcula directamente en el mapa truncado. Caso contrario, los costos computacionales involucrados en el procesamiento del mapa probabilístico harían inviable su implementación en tiempo real.

Una vez que se dispone del espacio navegable y maniobrable, el algoritmo de planificación de caminos genera un primer arco basado en los comandos de control obtenidos mediante

el método de Monte Carlo y en base al modelo cinemático del vehículo. Si el arco alcanza una vecindad de la frontera del espacio navegable y maniobrable, un nuevo arco se genera a partir de este punto pero con una dirección opuesta a su dirección previa. El proceso de generación de arcos continúa hasta que el robot virtual alcanza la orientación deseada por el usuario. La Fig. 1 muestra un ejemplo en la generación de un camino basada en arcos. La orientación inicial del vehículo es de $\xi_{\theta,t} = \pi/2$ y debe alcanzar una orientación final de $\xi_{\theta,t} = 3\pi/2$.

Como se puede observar en la Fig. 1, una vez que el primer arco –que parte de la pose inicial del vehículo–, alcanza la frontera del espacio navegable y maniobrable, el próximo arco generado cambia de dirección. El proceso de generación de arcos se repite iterativamente hasta que el robot virtual alcanza la orientación deseada o hasta que se haya excedido en el número de arcos admisibles por cada camino. En este trabajo, ese número fue fijado en 10, es decir, si una planificación no alcanza la orientación final con un máximo de 10 arcos, dicho camino es desechado y se procede a buscar el siguiente. Una vez que, con el robot virtual, se obtiene el mejor camino, dicho camino es pasado como referencia al controlador de seguimiento trayectorias de la silla de ruedas robotizada.

El procedimiento de generación de arcos se basa en el modelo cinemático de la silla de ruedas presentado en la Ec. (3). Así, mediante el método de Monte Carlo se obtienen las posibles velocidades lineales y rotacionales $-u_t$ y ω_t . De acuerdo a esto, se genera un camino cinemáticamente compatible con la silla de ruedas motorizada. Aún más, considerando que, debido a las dimensiones de la silla de ruedas, la misma no puede ser considerada como una masa puntual para los efectos de la planificación, entonces la planificación de caminos y las vecindades de colisión se calculan de acuerdo a un rectángulo que circunscribe al vehículo (la Cruz *et al.*, 2010). Si alguna de las vecindades de las esquinas del rectángulo que circunscribe a la silla de ruedas robotizada sobrepasa la frontera del espacio navegable y maniobrable, entonces el arco se interrumpe y desde esa pose del vehículo se genera el próximo arco con dirección contraria a la anterior.

El algoritmo presentado a continuación, muestra la estructura del procedimiento utilizado para la generación de arcos.

```

1: Sea  $DIRECCION = -1$  el signo inicial de la velocidad lineal
2: Sea  $\xi_{x,0}$ ,  $\xi_{y,0}$  y  $\xi_{\theta,0}$  la pose inicial del robot virtual
3: Sea  $MAP$  la porción del mapa correspondiente a la zona navegable y maniobrable del entorno.
4:  $[u_t, \omega_t] = MonteCarlo()$ 
5:  $u_t = DIRECCION * u_t$ 
6: for  $j = 1$  to  $\infty$  do
7:    $\begin{bmatrix} \xi_{x,j} \\ \xi_{y,j} \\ \xi_{\theta,j} \end{bmatrix} = \begin{bmatrix} \xi_{x,j-1} \\ \xi_{y,j-1} \\ \xi_{\theta,j-1} \end{bmatrix} + \Delta_t \begin{bmatrix} \cos(\xi_{\theta,j-1}) & 0 \\ \sin(\xi_{\theta,j-1}) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_t \\ \omega_t \end{bmatrix}$ 
8:    $ARCO = [Arco; [\xi_{x,j} \ \xi_{y,j} \ \xi_{\theta,j}]]$ 
9:   if Cualquier esquina del rectángulo que circunscribe al vehículo se aproxima a la frontera  $MAP$  then
10:     break
11:   end if
12: end for
13:  $DIRECCION = -1 * DIRECCION$ 

```

En el algoritmo anterior, la sentencia (1) establece que la primera dirección de la velocidad lineal es negativa por diseño. Velocidad lineal negativa significa movimiento en reversa, por lo tanto el primer arco generado desde la pose del vehículo implica un movimiento en reversa de la silla de ruedas. En la sentencia (3), MAP es la espacio navegable y maniobrable correspondiente al mapa adquirido por el SLAM, tal como se muestra en el ejemplo de la Fig. 10b. La sentencia (4) indica la generación de los comandos de velocidad lineal y angular en base a una aplicación del método de Monte Carlo. El lazo *for* –sentencias de la (6) a la (12) inclusive– muestra la generación del arco. Como se puede ver, el arco se genera a partir del modelo cinemático del vehículo, luego, la compatibilidad cinemática del arco generado con el robot está probada. La generación del arco se interrumpe si cualquier vértice del rectángulo que circunscribe el vehículo virtual se aproxima a una vecindad de la frontera del espacio navegable y maniobrable (MAP). La sentencia (7) muestra la generación de los puntos del arco basados en el modelo cinemático discreto de la silla de ruedas robotizada (Ec. (3)). Una vez que se encuentra un arco, la variable $DIRECCION$ cambia de signo. Así, el siguiente arco tendrá una velocidad lineal con signo opuesto a su valor previo. Por ejemplo, si el primer arco implica un movimiento en reversa, el segundo arco implicará un movimiento hacia adelante de la silla de ruedas robotizada. Los comandos de velocidad permanecen constantes durante el procedimiento de generación de un arco.

Método de Monte Carlo

El método de Monte Carlo genera los comandos de velocidad lineal y angular del modelo cinemático de la silla de ruedas robotizada en el algoritmo mostrado en el algoritmo de la sección anterior. Estos comandos de velocidad permiten obtener un camino factible para ser transitado por el vehículo.

Como una forma de proteger la integridad del usuario de la silla de ruedas, la velocidad lineal (u_t en la Ec. (3)) es una función de la velocidad angular (ω_t en la Ec. (3)). La Ec. (7) muestra la generación uniforme asociada al método de Monte Carlo, mientras que la Ec. (8) muestra el cálculo de la velocidad lineal como una función de la velocidad angular.

$$\mu \sim U(0, 1), \quad i = 1 \dots N$$

$$\omega = \omega_{max} * \mu \quad (7)$$

$$u = u_{max} \cos\left(\frac{\pi}{2\omega_{max}}\omega\right) \quad (8)$$

En la Ec. (7), U es una distribución uniforme de media cero y varianza uno. De esta manera, cuando $\omega = \omega_{max}$ (entonces $\mu = 1$), $u_t = 0$ y si, por el contrario, $\omega_t = 0$ ($\mu = 0$) entonces $u_t = u_{max}$. Así, cuando el robot virtual usado en la planificación de caminos está girando, su velocidad lineal decrece. Esto implica que se protege al usuario de las fuerzas centrífugas generadas durante el proceso de giro. Las velocidades máximas (u_{max} y ω_{max}) usadas en el método de Monte Carlo descrito en las ecuaciones (7) y (8) son usadas luego en el controlador de seguimiento de trayectorias que gobernará los movimientos de la silla de ruedas robotizada. Más información sobre el método de Monte Carlo se puede encontrar en (la Cruz *et al.*, 2010).

Cabe aclarar, además, que cuando μ (el resultado del experimento de Monte Carlo) es próximo a uno, entonces el robot

virtual gira sobre su punto de control sin desplazarse, es decir, gira sobre sí mismo (la posibilidad de giro sobre sí mismo de los vehículos tipo unicycle es una consecuencia de sus restricciones cinemáticas (Siegwart and Nourbahsh, 2004)). Por el contrario, si $\mu \approx 0$, entonces el vehículo avanza en línea recta. En este trabajo, las velocidades máximas utilizadas fueron: $u_{max} = 0,2 \text{ m/s}$ y $\omega_{max} = 0,15 \text{ rad/s}$.

Generación de caminos con orientación final definida

La estrategia de planificación implementada en este trabajo está destinada a guiar la silla de ruedas motorizada para alcanzar la orientación especificada por el usuario, teniendo en cuenta la disposición del ambiente y el espacio apto para la navegación y maniobrabilidad del vehículo.

Para alcanzar la orientación deseada, mediante el método de Monte Carlo y el algoritmo de generación de arcos presentado anteriormente, se generan sucesivos caminos posibles. Entre todos los caminos que alcancen de forma exitosa la orientación deseada por el usuario, se escoge aquel que tenga asociada la menor energía cinética. El algoritmo general de generación de caminos se muestra a continuación.

```

1: Sea  $\psi$  la orientación final deseada (dada por el usuario)
2: Sea  $Robot = [\xi_{x,0} \ \xi_{y,0} \ \xi_{\theta,0}]^T$  la pose del robot al momento de empezar el proceso de giro
3: Sea  $MCE$  una matriz con los valores de energía cinética asociada a cada camino generado por Monte Carlo y  $\#MCE$ , el número de caminos exitosos obtenidos por Monte Carlo
4: Sea  $L$  el máximo número de arcos que un camino puede tener
5: for  $j = 1$  to  $\#MCE$  do
6:    $bool \ Angulo = false$ 
7:    $EnergiaTotal = 0$ 
8:    $ArcoPrincipal = 0$ 
9:   for  $m = 1$  to  $L$  do
10:     $[EnergiaDeArco, \ Angulo, \ Arco] = ArcoGeneracion(Robot, \psi)$ 
11:     $EnergiaTotal = EnergiaTotal + EnergiaArco$ 
12:     $ArcoPrincipal = [ArcoPrincipal; \ Arco]$ 
13:    if  $Angulo = true$  then
14:      break
15:    end if
16:  end for
17:   $MCE(j) = EnergiaTotal$ 
18: end for

```

En la sentencia (1), ψ es el ángulo que el usuario desea alcanzar dentro del ambiente navegado; MCE en la sentencia (3), es la matriz que almacena el valor de la energía cinética asociada a cada camino exitosamente obtenido mediante el experimento de Monte Carlo; en la sentencia (4), L determina el máximo número de arcos que un camino puede tener. En este trabajo, se adoptó $L = 10$, aunque esto se puede modificar por diseño. Las sentencias (5)–(18) muestran el procedimiento de generación de caminos. La función $ArcoGeneracion$ en la sentencia (10) corresponde al algoritmo mostrado en la sección 6.2 aunque en este caso, la función $ArcoGeneracion$ toma dos parámetros: la pose actual de la silla de ruedas robotizada y el ángulo deseado por el usuario. Esta misma función devuelve tres valores: $EnergiaArco$, $Angulo$ y $Arco$. $EnergiaArco$ es un valor de energía cinética asociada al arco calculado durante la invocación del algoritmo mostrado en la sección 6.2. Si el vehículo virtual alcanza la orientación deseada por el usuario,

la generación del arco se interrumpe y la variable Booleana $Angulo$ cambia su valor a *true*, quebrando de esa manera el bucle interior *for* en el algoritmo para la generación de caminos. $Arco$ es el conjunto de puntos Cartesianos que representan el arco generado por el algoritmo de la sección 6.2.

En la sentencia (5), $\#MCE$ es el número de caminos factibles a ser generados por el planificador. Después de que $\#MCE$ caminos hayan sido encontrados, aquel camino que tenga asociada la menor energía cinética será escogido para ser ejecutado por la silla de ruedas robotizada. La energía cinética instantánea de un robot móvil –y por ende, de cualquier cuerpo– es una función cuadrática de su velocidad lineal y su velocidad angular de la forma: $Energia \ Cinetica = \sum_{j=1}^L \frac{1}{2} (mu_j^2 + I\omega_j^2)$, donde L es el número de arcos asociado al camino, m es la masa del vehículo e I es su momento de inercia; u_j y ω_j son las velocidades lineal y angular respectivamente, generadas por el experimento de Monte Carlo para el j -ésimo arco.

6.3 Controlador de seguimiento de trayectorias

En este trabajo se implementó la ley de control de seguimiento de trayectorias propuesta por (Kanayama *et al.*, 1990). Este controlador es asintóticamente estable y dicha estabilidad fue demostrada mediante la teoría de Lyapunov. Las entradas del controlador son las posturas de referencia $[\xi_{x,r} \ \xi_{y,r} \ \xi_{\theta,r}]^T$ (posición y orientación que el robot tendría sobre los puntos que componen el camino generado mediante arcos y las referencias de velocidad (u_r y ω_r , obtenidas mediante el experimento de Monte Carlo). El error de postura se define como:

$$\begin{bmatrix} \xi_{x,e} \\ \xi_{y,e} \\ \xi_{\theta,e} \end{bmatrix} = \begin{bmatrix} \cos(\xi_{\theta,t}) & \sin(\xi_{\theta,t}) & 0 \\ -\sin(\xi_{\theta,t}) & \cos(\xi_{\theta,t}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_{x,r} - \xi_{x,t} \\ \xi_{y,r} - \xi_{y,t} \\ \xi_{\theta,r} - \xi_{\theta,t} \end{bmatrix} \quad (9)$$

donde $[\xi_{x,t} \ \xi_{y,t} \ \xi_{\theta,t}]^T$ es la pose actual de la silla de ruedas robotizada. La ley de control es:

$$\begin{bmatrix} u_t \\ \omega_t \end{bmatrix} = \begin{bmatrix} u_r \cos(\xi_{\theta,e} + K_x \xi_{x,e}) \\ \omega_r + u_r (K_y \xi_{y,e} + K_\theta \sin(\xi_{\theta,e})) \end{bmatrix} \quad (10)$$

donde K_x , K_y y K_θ son constantes positivas.

Kanayama también propone un parámetro de selección para obtener un amortiguamiento crítico en el control. El amortiguamiento del controlador de seguimiento de trayectorias se puede calcular como:

$$\zeta = \frac{K_\theta}{2\sqrt{K_y}}.$$

Se obtiene un amortiguamiento crítico en el controlador cuando $\zeta = 1$.

En el caso que $\mu \approx 1$ en la Ec. (7), entonces el vehículo giraría sobre su punto de control B en la Fig. 7. En este caso particular se usa un controlador de orientación en vez del controlador de seguimiento de trayectorias de Kanayama presentado en la Ec. (10). El controlador de orientación implementado en este trabajo se muestra en la Ec. (11),

$$\begin{cases} \omega = k_\omega \tilde{\xi}_\theta \\ u = 0 \end{cases} \quad (11)$$

donde k_ω es una constante positiva y $\tilde{\xi}_\theta = \xi_{\theta,r} - \xi_{\theta,t}$. Considerando un control de regulación, $\xi_{\theta,r}$ –la orientación deseada– es constante, entonces,

$$\dot{\xi}_{\theta,t} = -\dot{\tilde{\xi}}_\theta \quad (12)$$

Reemplazando la Ec. (11) y la Ec. (12) en el modelo cinemático de la silla de ruedas robotizada (ver la Ec. (3)), se tiene que:

$$\begin{cases} \dot{\xi}_{x,t} = 0 \\ \dot{\xi}_{y,t} = 0 \\ 0 = k_\omega \tilde{\xi}_\theta + \dot{\xi}_\theta \end{cases} \quad (13)$$

Por lo tanto, $\tilde{\xi}_\theta \rightarrow 0$ con $t \rightarrow \infty$.

La forma en que el algoritmo de SLAM se implementa junto al controlador de seguimiento de trayectorias se puede ver en la Fig. 11.

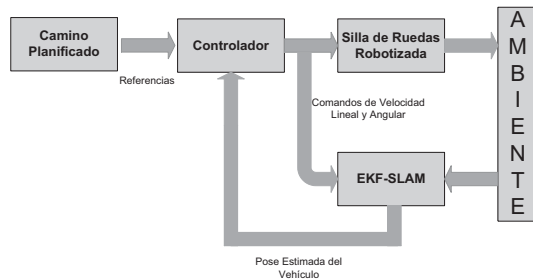


Figura 11. Combinación del EKF-SLAM con el controlador de seguimiento de trayectorias.

La Fig. 11 muestra cómo se relacionan el algoritmo de EKF-SLAM con el controlador de trayectorias. Esta relación se puede resumir como sigue:

- i Las referencias al controlador ($[\xi_{x,r} \ \xi_{y,r} \ \xi_{\theta,r}]^T$) son dadas por el algoritmo de planificación de caminos descrito anteriormente.
- ii La pose del robot es estimada a partir del algoritmo de SLAM implementado en el vehículo. El algoritmo de SLAM utiliza la información del ambiente para estimar la pose del robot dentro del mapa. La pose estimada de la silla de ruedas robotizada se considera como la pose actual ($[\xi_{x,t} \ \xi_{y,t} \ \xi_{\theta,t}]^T$) del vehículo en la ley de control presentada en el Ec. (10).
- iii Con la referencia dada y la información de pose actual del vehículo, el controlador genera los comandos de velocidad impartidos a la silla de ruedas robotizada.
- iv Los comandos de velocidad impartidos a la silla de ruedas robotizada y la información extraída del ambiente, constituyen las variables de entrada del algoritmo de EKF-SLAM.

7. RESULTADOS EXPERIMENTALES

La figura 12 muestra la construcción del mapa del Instituto de Automática de la Universidad Nacional de San Juan, Argentina, usando el algoritmo de EKF-SLAM descrito en este trabajo. Las Figs. 12a, 12b y 12c, muestran reconstrucciones parciales de las instalaciones del Instituto de Automática. El rectángulo de aristas rojas presente en estas figuras representa el rectángulo que circunscribe el espacio navegable según el cual se calcula el

espacio navegable y maniobrable a ser empleado por la silla de ruedas robotizada; los puntos en magenta son mediciones del sensor de rango láser; la línea a trazo azul, es el camino seguido por el vehículo; los segmentos sólidos negros están asociados a las líneas extraídas del ambiente; los círculos verdes son las esquinas del entorno. Con el fin de apreciar la consistencia del mapa reconstruido, en las Figs. 12a a 12c se muestra la reconstrucción sin el truncamiento en el vector de estados del SLAM, de las características no contempladas por el rectángulo rojo (como se explicó en la sección 6.1). La Fig. 12d muestra la reconstrucción total del Instituto de Automática. Como se puede ver, el mapa obtenido es consistente (Thrun *et al.*, 2005). La Fig. 13 muestra la evolución de la varianza asociada a algunas características del mapa mostrado en la Fig. 12d. La tendencia a cero de las varianzas indica que el algoritmo de EKF-SLAM implementado es convergente (Whyte and Bailey, 2006b,a).

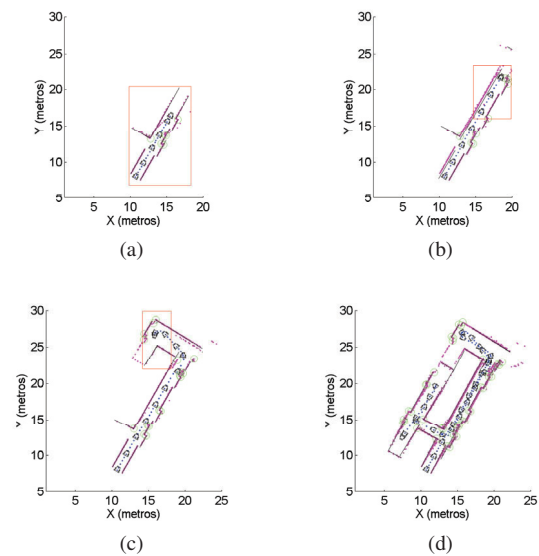


Figura 12. Reconstrucción de las instalaciones del Instituto de Automática en diferentes instantes de la navegación.

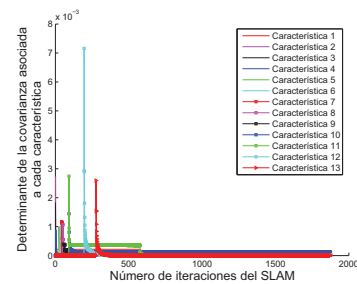


Figura 13. Evolución de la varianza asociada a algunas características extraídas del entorno.

La Fig. 14 muestra un ejemplo del algoritmo de maniobrabilidad propuesto en este trabajo. Como se puede ver, la silla de ruedas robotizada avanza hasta llegar al final de un callejón. Allí, el usuario decide retornar. Este experimento fue llevado a cabo en las instalaciones del *Departamento de Engenharia Elétrica* de la *Universidade Federal do Espírito Santo*, Brasil. La silla de ruedas robotizada utiliza un sensor láser –construido por SICK– cuyo rango máximo de medición fue programado a 6 metros. En el experimento de la Fig. 14 se llevaron a cabo

2500 realizaciones de Monte Carlo –es decir, se obtuvieron 2500 caminos factibles de ser seguidos por la silla de ruedas–. Estas 2500 realizaciones requirieron un tiempo de cómputo de 60 segundos aproximadamente en una computadora *Intel Core Duo 2.6 GHz*, bajo el sistema operativo Windows, con el algoritmo de maniobrabilidad funcionando con prioridad máxima. Este tiempo de procesamiento es proporcional al número de caminos generados. Así, para la generación de 50 posibles caminos, la estrategia de maniobrabilidad requiere aproximadamente de 1 segundo.

En el experimento mostrado en la Fig. 14, el vehículo navega el entorno hasta que llega al callejón, en cuyo caso el usuario escoge girar 180 grados desde su orientación actual para poder retornar por el camino transitado. La pose del vehículo dentro del entorno, en la Fig. 14, al momento de ejecutar el algoritmo de maniobrabilidad es $[\xi_{x,t} \ \xi_{y,t} \ \xi_{\theta,t}]^T = [10,7 \ 8,0 \ \pi/2]^T$ y la orientación deseada por el usuario es $\psi = -\pi/2$. La Fig. 14a muestra, por motivos de visualización, 10 realizaciones de las 2500 realizaciones generadas durante el experimento; mientras que la Fig. 14b muestra el camino escogido –corresponde al camino con menor energía cinética asociada (ver sección 6)–. El camino en azul es el camino seleccionado por el algoritmo de planificación. La línea magenta representa el camino transitado por la silla de ruedas robotizada –estimado por el algoritmo de SLAM–. En el caso del seguimiento del camino planificado, el vehículo es guiado por el controlador de trayectorias presentado en la sección 6.3. Los segmentos grises sólidos corresponden a las líneas –asociadas a paredes– extraídas del ambiente por el algoritmo de SLAM. El rectángulo azul sólido que circunscribe el entorno, determina la zona del ambiente respecto a la cual se determina el espacio navegable y maniobrable. En este experimento, el número máximo de arcos por camino fue establecido en $L = 10$.

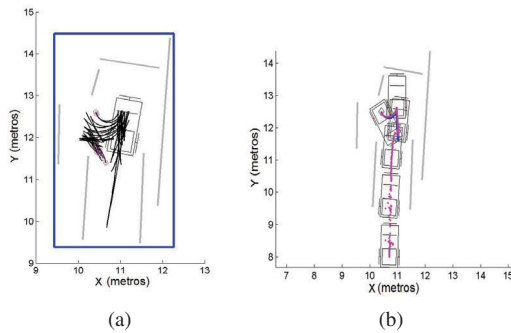


Figura 14. Estrategia de maniobrabilidad. a) Muestra 10 posibles caminos de los 2500 generados por la estrategia de planificación; b) muestra el camino escogido (en azul) mientras que el camino seguido por la silla de ruedas se muestra en magenta.

La Fig 15 muestra dos experimentos llevados a cabo en el entorno mostrado en la Fig. 8. En este experimento, cuando el usuario de la silla de ruedas llega al final del pasillo, debe decidir su ángulo de giro. Como se puede observar, la silla de ruedas se detiene de forma muy próxima a los elementos del ambiente, por lo que no puede girar sobre sí misma. La Fig. 15a muestra el caso en que el usuario decide girar 90 grados respecto a la orientación actual de la silla de ruedas (decide orientarse en dirección al pasillo de su izquierda). En la Fig. 15b se muestra el caso en que el usuario decide girar 90 grados a su derecha (para tomar el pasillo de la derecha). En ambos

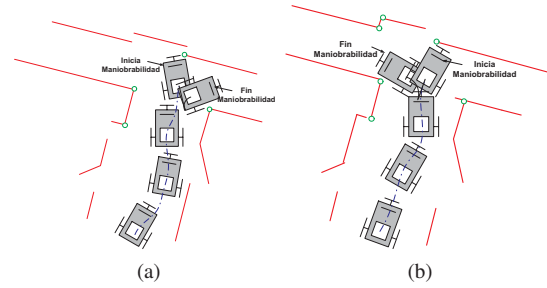


Figura 15. Reconstrucción del ambiente usando la estrategia de maniobrabilidad presentada en este trabajo. a) La figura muestra el camino de giro cuando el usuario escoge una orientación destino de -90 grados respecto a la orientación actual del vehículo (sentido antihorario). b) La figura muestra el camino de giro cuando el usuario elige una orientación destino de 90 grados.

casos, el camino seguido por la silla de ruedas robotizada se muestra en línea azul a trazos y el camino planificado se muestra con línea negra sólida. Las paredes del entorno se representan mediante segmentos rojos sólidos, mientras que las esquinas, por círculos verdes. Tanto en la Fig. 15a como en la Fig. 15b se pueden ver los arcos generados por el planificador de caminos para guiar a la silla de ruedas robotizada a alcanzar la orientación deseada por el usuario (Auat Cheein *et al.*, 2010b). En la Fig. 15 se puede ver también que la estrategia de planificación propuesta en este trabajo genera dos arcos suaves (siendo el primer arco prácticamente un segmento recto) que conducen al vehículo a alcanzar la orientación deseada por el usuario.

La Fig. 16 muestra los resultados estadísticos de una población de 7 usuarios cognitivamente normales que usaron la interfaz propuesta en este trabajo. La edad de los usuarios fue entre 21 y 35 años. La Fig. 16 muestra el tiempo que requirió cada usuario para ejecutar la navegación en el pasillo mostrado en la Fig. 15 –el cual tiene aproximadamente 12 metros de largo– y en escoger el ángulo deseado de giro. El tiempo promedio para escoger el ángulo de 90 grados fue de 35.4 segundos, mientras que para escoger -90 grados fue de 36.5 segundos. Para escoger un giro de 180 grados, el tiempo promedio fue de 34.7 segundos. En los tiempos promedios también se incluye el tiempo demandado para la planificación del camino (no así el tiempo que le lleva a la silla de ruedas robotizada seguir la trayectoria generada).

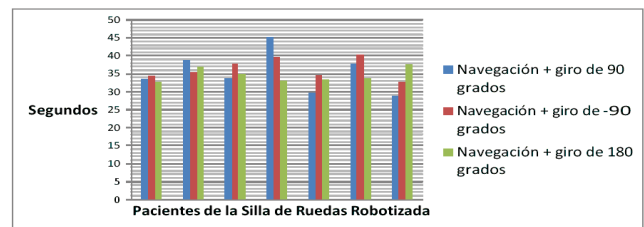


Figura 16. Tiempo requerido por cada usuario para navegar y girar dentro del ambiente (para 90 , -90 y 180 grados de orientación deseada por el usuario).

8. CONCLUSIÓN

En este trabajo se presentó una interfaz que combina una estrategia de maniobrabilidad de una silla de ruedas robotizada

para la navegación en espacios restringidos. El uso del algoritmo de SLAM en el proceso de planificación y control, aportó la información necesaria relacionada a la pose del vehículo dentro del entorno y del modelo del entorno en sí mismo. El uso de la interfaz propuesta disminuye el esfuerzo requerido por el usuario de la silla de ruedas cuando debe maniobrar en ambientes restringidos.

La estrategia de planificación autónoma presentada en este trabajo, permite generar un mapa de probabilidad asociado al mapa geométrico construido por el algoritmo de SLAM. Esto es posible gracias a que el algoritmo de SLAM está implementado en un EKF y, por lo tanto, su vector de estados tiene distribución Gaussiana. Mediante el método de *suma de Gaussianas* y del rejillado del mapa, se obtuvo la probabilidad de ocupación de cada celda asociada al nuevo mapa rejillado. El hecho de usar mapas de probabilidad en vez de mapas geométricos extraídos directamente del vector de estados del algoritmo de SLAM, permitió inferir la información faltante en aquellas zonas vacías del entorno.

Finalmente, la interfaz propuesta fue probada en una población de 7 personas cognitivamente normales, mostrando resultados exitosos en cada uno de los usuarios.

AGRADECIMIENTOS

Los autores agradecen al CONICET-Argentina y a FAPES-Brasil (proceso 39385183/2007) por la financiación parcial de este proyecto.

REFERENCIAS

- Auat Cheein, F., C. De La Cruz, C. Soria, R. Carelli and T.F. Bastos (2010a). Slam-based turning back in restricted environments for non-holonomic assistive vehicles. *ISSNIP Biosignals and Biorobotics Conference* pp. 26–31.
- Auat Cheein, F., C. De la Cruz, R. Carelli and T. Bastos (2009). Solution to a door crossing problem for an autonomous wheelchair. in *Proc. of the IEEE International Conference on Intelligent Robots and Systems* pp. 4931–4936.
- Auat Cheein, F., C. De la Cruz, R. Carelli and T. Bastos (2010b). Autonomous assistance navigation for robotic wheelchairs in confined spaces. In *IEEE International Conference on Engineering in Medicine and Biology Society-EMBC10*.
- Auat Cheein, F., J. M. Toibero, F. di Sciascio, R. Carelli and F. Lobo Pereira (2010c). Monte carlo uncertainty maps-based for mobile robot autonomous slam navigation. pp. 1433–1438.
- Bailey, T., J. Nieto, J. Guivant, M. Stevens and M. Nebot (2006). Consistency of the ekf-slam algorithm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* pp. 3562–3568.
- Bourhis, G., O. Horn, O. Habert and A. Pruski (2001). An autonomous vehicle for people with motor disabilities. *IEEE Robotics and Automation Magazine* **8**(1), 20–28.
- Dario, P., E. Guglielmelli, C. Laschi and G. Teti (1999). Movaid: a personal robot in everyday life of disabled and elderly people. *Journal of Technology and Disability*.
- Dellaert, F., D. Fox and W. Burgard (1999). Monte carlo localization for mobile robots. *Proc. of the IEEE International Conference on Robotics and Automation* pp. 1322–1328.
- Ferreira, A., W.C. Celeste, F.A. Cheein, T.F. Bastos, M. Sarcinelli and R. Carelli (2008). Human-machine interfaces based on emg and eeg applied to robotic systems. *Journal of NeuroEngineering and Rehabilitation* **5**(10), doi:10.1186/1743-0003-5-10.
- Guivant, J. and M. Nebot (2001). Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation* pp. 242–257.
- Kanayama, Y., Y. Kimura, F. Miyazaki and T. Noguchi (1990). A stable tracking control method for an autonomous mobile robot. *IEEE International Conference on Robotics and Automation (ICRA)* pp. 384–389.
- la Cruz, C. De, F. Auat Cheein, R. Carelli and T. Bastos (2010). Slam-based robotic wheelchair navigation system designed for confined spaces. *IEEE-International Symposium on Industrial Electronics*.
- Liu, Y., J. Dong and F. Sun (2008). An efficient navigation strategy for mobile robots with uncertainty estimation. *Proc. of the World Congress on Intelligent Control and Automation* pp. 5174–5179.
- Martins, F., W. Celeste, R. Carelli, M. Sarcinelli and T. Bastos (2008). An adaptive dynamic controller for autonomous mobile robot trajectory tracking. *Control Engineering Practice*.
- Miralles, A. Sanchez and M. A. Sanz Bobi (2004). Global path planning in gaussian probabilistic maps. *Journal of Intelligent and Robotic Systems* **40**, 89–112.
- Montesano, L., M. Diaz, S. Bhaskar and J. Minguez (2010). Towards an intelligent wheelchair system for users with cerebral palsy. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **2**, 193–202.
- Parikh, S.P., V. Grassi, V. Kumar and J. Okamoto (2007). Integrating human inputs with autonomous behaviors on an intelligent wheelchair platform. *IEEE Intelligent Systems* **22**(2), 33–41.
- Park, H. K., H. S. Hong, H. J. Kwon and M. J. Chung (2004). A nursing robot system for the elderly and the disabled. *Proceedings of the International Workshop on Humanfriendly Welfare Robotic Systems*.
- Siegwart, R. and I. Nourbahsh (2004). *Introduction to Autonomous Mobile Robots*. MIT Press. Cambridge.
- Thrun, S., W. Burgard and D. Fox (2005). *Probabilistic Robotics*. MIT Press. Cambridge.
- Whyte, H. and T. Bailey (2006a). Simultaneous localization and mapping (slam): part ii state of the art. *IEEE Robotics and Automation Magazine* **13**, 108–117.
- Whyte, H. D. and T. Bailey (2006b). Simultaneous localization and mapping (slam): part i essential algorithms. *IEEE Robotics and Automation Magazine* **13**, 99–108.
- Zeng, Q., Ch.L. Teo, B. Rebsamen and E. Burdet (2008). A collaborative wheelchair system. *IEEE Trans. on Neural Systems and Rehabilitation Engineering* **16**(2), 161–170.