TRANSPORTE DE PERSONAS CON VEHÍCULOS AUTÓNOMOS EN ENTORNOS URBANOS

Luis Merino
Universidad Pablo de Olavide, Crta. Utrera, km. 1, 41013 Sevilla, lmercab@upo.es

Francisco Real, Pablo Soriano, Aníbal Ollero
Universidad de Sevilla, Camino de los Descubrimientos, s/n, 41092, Sevilla, {psoriano,aollero}@cartuja.us.es

Resumen

El artículo presenta la arquitectura y algoritmos de navegación de una coche eléctrico autónomo capaz de navegar en zonas peatonales urbanas para las tareas de transporte de personas y objetos. Se describe fundamentalmente la arquitectura de navegación, así como la integración del vehículo en un sistema de robots en red, que provee al vehículo de autonomía operacional y le permite colaborar con otros robots y sensores en el entorno para realizar sus tareas. El artículo presenta resultados experimentales obtenidos en los experimentos finales del proyecto europeo URUS.

Palabras clave: Robots urbanos, navegación, localización.

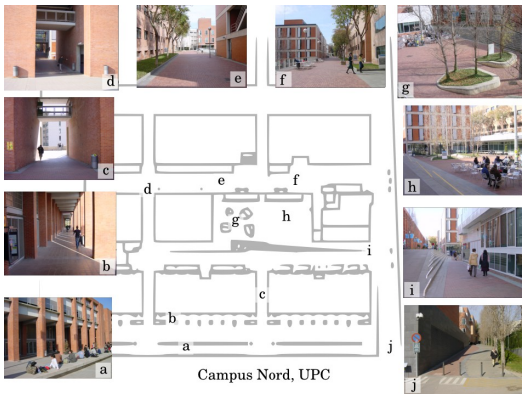

Figura 1: Descripción del escenario. Se trata de un cuadrado de unos 100 metros de lado. Hay complicaciones como rampas (c, d, j) y escaleras (h, i); y por supuesto hay gente habitualmente en el escenario.

1 INTRODUCCIÓN

Muchas de las grandes ciudades europeas buscan reducir el tráfico en ciertas áreas, para así mitigar la polución acústica y ambiental, los atascos y, en definitiva, mejorar la calidad de vida. Para ello, ciertas tareas llevadas a cabo en dichas zonas por automóviles podrían ser sustituidas por sistemas autónomos. El proyecto europeo URUS [12] ha indagado en esta idea: la introducción de *robots de servicio* urbanos.

El presente artículo describe un vehículo autónomo capaz de navegar en zonas peatonales para la tarea de transporte de personas (lo que denominamos misión TAXI). El objetivo es que, bajo la petición de una persona (a través de una interfaz en su teléfono móvil), el vehículo sea capaz de recogerla en un determinado lugar y llevarla a un lugar distinto en la zona peatonal. Esta misión involucra no sólo el desarrollar una navegación autónoma segura, sino también gestionar la interacción hombre-robot (la persona a bordo debe poder no sólo indicar el destino, sino también parar o reiniciar el vehículo en cualquier momento). Además, el sistema desarrollado en URUS permite al robot colaborar con otros robots y usar las redes de cámaras y otros sensores disponibles en el entorno urbano para realizar las tareas correspondientes. Por ejemplo, las redes de cámaras se emplean para localizar a las personas para realizar ciertos servicios; o robots tipo humanoide pueden colaborar para guiar a las personas a las estaciones de taxi.

El artículo se centra en la arquitectura y algoritmos de navegación desarrollados para la tarea de transporte automático. También se describe la integración de un vehículo autónomo en la arquitectura URUS. Se presentan resultados experimentales en los que el vehículo realiza misiones TAXI en entornos peatonales. El escenario de dichos experimentos se muestra en la Fig. 1. Este escenario presenta varios retos desde el punto de vista de la navegación, como es la presencia de peatones y la existencia de obstáculos como pendientes, escaleras, árboles y pasillos estrechos, así como la falta de cobertura GPS.

Tras resumir algunos trabajos relacionados, el artículo describe someramente la arquitectura URUS para robots urbanos en la Sección 2. A continuación, la Sección 3 describe el vehículo considerado en este artículo. La Sección 4 presenta los algoritmos de navegación y, finalmente, la Sección 5 presenta los resultados experimentales previamente a las conclusiones.

1.1 TRABAJOS RELACIONADOS

Hay un interés creciente en el desarrollo de vehículos capaces de navegar autónomamente en zonas urbanas. Es bien conocido el DARPA Urban Challenge [1], cuyo objetivo es el desarrollo de vehículos inteligentes capaces de circular autónomamente en ciudades, cumpliendo al mismo tiempo con las reglas de circulación. La principal característica de las soluciones propuestas es que normalmente se emplea equipamiento muy costoso, incluyendo GPS y sistemas inerciales para la localización; asimismo, no se considera la navegación en zonas peatonales. [13] muestra un ejemplo de los vehículos diseñados.

El presente artículo describe un robot para la navegación en áreas peatonales para el transporte de pasajeros. En [8], los autores muestran la arquitectura de navegación de un robot para el Tsukuba Challenge, competición que requiere la navegación en corredores peatonales durante 1 kilómetro. Algunas conclusiones y decisiones de diseño en dicho artículo son similares a las descritas aquí, aunque otras son muy diferentes, como el uso de cobertura de GPS diferencial. Además, la plataforma empleada es una plataforma diferencial.

Una de las mayores limitaciones en nuestro caso es que empleamos un vehículo no-holónomo con dirección Ackermann. Un ejemplo de coches autónomos es el Cycab [10]. La arquitectura de navegación tiene semejanzas, empleando una combinación probabilística de comandos de movimiento que son generados por distintos modos de navegación (evitación de obstáculos, seguimiento de caminos, etc). El robot emplea marcas artificiales para la localización. Sin embargo, si se busca la introducción de robots en entornos urbanos es necesario desarrollar métodos para la localización que no impongan requerimientos en la infraestructura de la ciudad. En [3], los autores presentan técnicas para el uso de la información en Sistemas de Información Geográfica (SIG) para la localización de robots en entornos urbanos.

El SmarTer [5] es un Smart modificado para la navegación autónoma en exteriores. En el proyecto URUS el robot ha sido adaptado para navegar en zonas peatonales.

2 LA ARQUITECTURA URUS

La Figura 2 muestra los módulos de la arquitectura software desarrollada en URUS para robots urbanos conectados en red con el entorno. La arquitectura fue diseñada para poder integrar un conjunto de robots heterogéneos mediante lo que

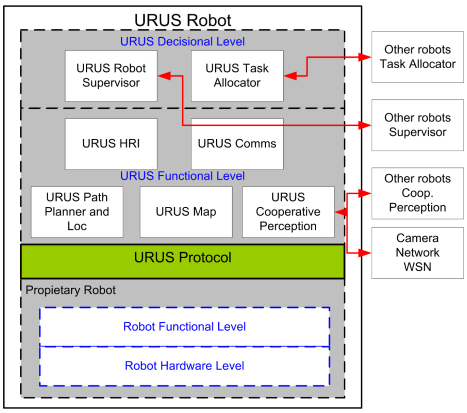

Figura 2: La arquitectura de Romeo como un robot URUS.

se denomina el protocolo URUS. La arquitectura asume que cada robot a ser integrado es capaz de seguir caminos usando sus módulos propietarios. A partir de ahí, la arquitectura aumenta la autonomía del robot aportando una serie de módulos de más alto nivel.

Los diferentes componentes software han sido desarrollados por distintos miembros del proyecto, y se integran en los distintos robots usando una Arquitectura Orientada a Servicios (SOA). Alguno de los módulos son:

- Un servicio de asignación de tareas multi-robot [9].
- Un módulo de supervisión a cargo de controlar las tareas que indica el módulo anterior. Para realizar eso, el supervisor es capaz de controlar el resto de servicios de la arquitectura a través de una serie de interfaces también definidas en el protocolo URUS.
- Un planificación de caminos global basado en A^* que determina los caminos para alcanzar los destinos seleccionados y que es la interfaz principal con los módulos propietarios de navegación de cada robot.
- Una interfaz hombre-robot (HRI en sus siglas inglesas), que informa a las personas del estado del robot y que permite a las personas indicar comandos, expresar sus intenciones y crear una nueva misión si es necesario.
- El servicio de comunicaciones permite al robot comunicarse con otros robots y el resto del sistema. Este servicio monitoriza el nivel de señal de la conexión WiFi y es capaz de cambiar a 3G en cualquier momento en que dicho nivel cae por debajo de un determinado umbral.

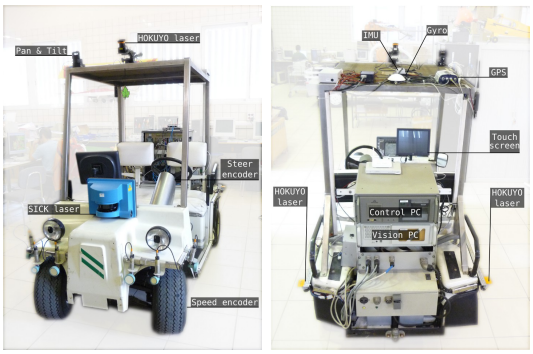

Figura 3: Sensores a bordo de Romeo. Posee sensores para localización (GPS, girósocopos, codificadores, láser Sick), percepción del entorno y navegación (láseres).

- Un módulo de percepción descentralizado [2, 12] que permite al robot mejorar su percepción local combinándola con la información de otros robots y los sensores del entorno.

3 DESCRIPCIÓN DE ROMEO

Romeo es un coche eléctrico modificado con sensores y actuadores para navegar de forma autónoma en escenarios en exteriores, incluyendo entornos urbanos.

3.1 HARDWARE

La Fig. 3 muestra el robot y los principales sensores que incorpora:

- Odometría: Romeo posee codificadores para la estimación de la velocidad y la curvatura, giróscopos de KVH Industries y una unidad de medida inercial (IMU en sus siglas inglesas) para la estimación de las velocidades angulares.
- Láseres: Romeo tiene un láser SICK LMS 220 en la parte frontal, a 95 cm. de altura, para evitación de colisiones y localización. Además, posee dos láseres Hokuyo URG-04LX (con un rango de 4 m.) en los laterales para percibir obstáculos hacia los lados y hacia atrás; y un láser Hokuyo UTM-30LX (con un rango de 30 m.) en el techo de Romeo, inclinado ligeramente para obtener una percepción 3D del entorno, como se verá en la sección siguiente.
- Un receptor GPS diferencial de Novatel.
- Una cámara de color firewire, que puede ser empleada para seguimiento y guiado de personas.

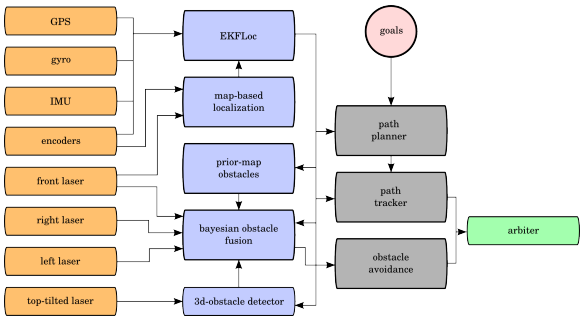

Figura 4: Arquitectura software básica de Romeo.

- Una pantalla táctil, que se emplea para controlar el robot y para interacciones hombre-robot.

Romeo tiene un motor eléctrico que actúa sobre el volante, además del que actúa sobre las ruedas de tracción. Los controladores de más bajo nivel cierran los bucles con los codificadores, aceptando referencias en velocidad linear y curvatura para el vehículo.

3.2 ARQUITECTURA SOFTWARE DE BAJO NIVEL

La Figura 4 muestra la arquitectura software básica para la navegación en entornos urbanos. Para la localización, Romeo fusiona todos sus sensores de odometría (codificadores, giróscopos e IMU) así como la información obtenida con sus láseres mediante un filtro extendido de Kalman (módulo `EKFLoc`) para estimar su posición y orientación en 6D.

Para navegación, Romeo combina la información de la localización con la de sus láseres para construir un modelo del entorno, básicamente un mapa de obstáculos: es decir, una malla en la que para cada celda se indica si puede ser atravesada por Romeo o no. También permite identificar obstáculos móviles. La información de este modelo la emplea el módulo de evitación de obstáculos (`obstacle avoidance`) para controlar el robot. La salida de este módulo es combinada con la salida del seguidor de caminos (`path tracker`) por un árbitro de comportamientos (`arbiter`), que asegura que no se producen colisiones mientras trata de seguir el camino indicado de la forma más precisa posible.

4 NAVEGACIÓN SEGURA DE BAJO NIVEL

Los módulos de navegación de bajo nivel de cualquier robot que quiera integrarse en el sistema

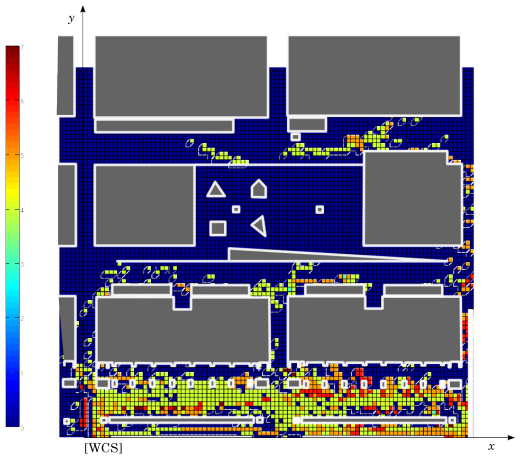

Figura 5: Cobertura GPS en el escenario considerado (mientras más rojo mayor el número de satélites). Nótese que no sólo la cobertura es limitada, sino que además el número de satélites es bajo en la mayor parte del entorno.

URUS deben ser capaces de seguir caminos como capacidad básica. Es decir, el robot debe ser capaz de seguir un camino dado de la forma más precisa posible, mientras evita obstáculos; la seguridad de los peatones es la prioridad principal.

4.1 LOCALIZACIÓN

Para navegar en los escenarios considerados (Fig. Fig. 1) se necesita una localización en 6 grados de libertad (posición en 3 dimensiones y la orientación en el espacio). El algoritmo empleado para esto es un filtro de Kalman extendido (EKF en sus siglas inglesas) que combina toda la información disponible (giróscopos, IMU, codificadores, etc). El filtro emplea un modelo cinemático del vehículo, discretizado y linealizado, en cada paso de predicción.

Si hay medidas GPS, el filtro, de forma automática, es capaz de estimar la orientación relativa del robot con respecto al sistema de referencia del GPS, y emplear sus medidas para cancelar la deriva. El principal problema es que el entorno considerado no dispone de una cobertura GPS fiable. Como muestra la Fig. 5, la cobertura GPS está limitada a un área determinada del escenario. Además, el número de satélites en tal área es pequeño. Esto, unido a efectos multi-trayecto, imposibilita el uso del GPS (pues sus estimaciones están normalmente afectadas por grandes saltos).

Por tanto, un sistema de localización basado en mapas se incorpora en el sistema. El algoritmo, descrito en [7], ha sido desarrollado por colegas de la Universidad Politécnica de Cataluña, e incorporado en Romeo empleando el protocolo URUS.

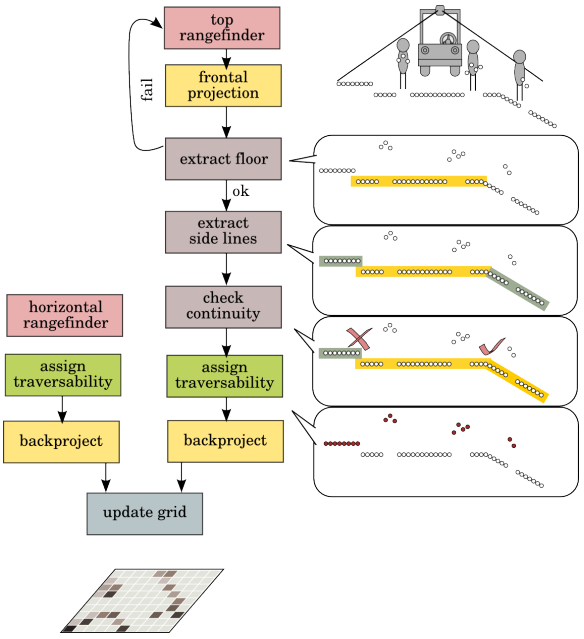

Figura 6: Esquema de percepción de obstáculos. Izquierda: las medidas de los láseres horizontales se relacionan directamente con los obstáculos, y se incorporan directamente al mapa. Derecha: el análisis del láser situado en el techo es hecho línea a línea. La localización del robot se emplea para trasladar los resultados a coordenadas globales.

Se trata de en un filtro de partículas que emplea la información del láser delantero y un mapa para seguimiento de posición, y es capaz de dar estimaciones de la posición con una frecuencia de 1Hz. Por tanto, aunque el algoritmo es suficientemente preciso para la localización a largo plazo, se necesitan estimaciones basadas en odometría para la navegación. La solución final basada en EKF integra la odometría a 40 Hz., e incorpora las correcciones del módulo de localización basado en mapas como un GPS virtual en el filtro. La Sección 5 mostrará los resultados de dicho sistema.

4.2 PERCEPCIÓN 3D

El sistema de percepción debe ser capaz de categorizar como obstáculo o zona navegable el terreno circundante al robot, terreno que contiene elementos como aceras, pendientes y caídas desde diferentes niveles o escaleras. Estos elementos son, en el mejor de los casos, sólo parcialmente observable por los láseres horizontales. Al mismo tiempo, es necesario tener en cuenta los obstáculos dinámicos.

El algoritmo de percepción consta de dos niveles (ver Fig. 6). En primer lugar, cada láser en el vehículo es analizado por separado para determinar los obstáculos de su zona de cobertura. A

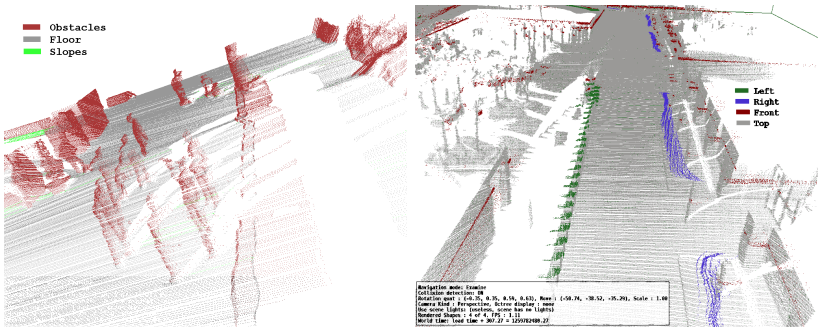

Figura 7: Información 3D obtenida por Romeo a partir de sus láseres. Se puede identificar varios obstáculos. como escaleras (derecha), árboles y pequeños escalones (esta figura puede apreciarse mejor en la versión electrónica del artículo).

continuación, las medidas obtenidas por los distintos láseres se combinan en un mapa global de ocupación de forma dinámica empleando un filtro Bayesiano.

4.2.1 Análisis de los láseres horizontales

Las medidas procedentes de láseres con barrido horizontal se procesan del mismo modo: cada punto del láser dentro del rango máximo del sensor se asume que es un obstáculo, y la línea recta hacia él libre de obstáculos a la altura a la que se encuentra el láser.

4.2.2 Análisis del láser frontal superior

El láser 2D situado en el techo de Romeo (inclinado hacia abajo, ver Fig. 6, derecha) es la fuente principal para la percepción 3D del entorno. Dicho láser, combinado con la localización y el movimiento de Romeo, actúa como un láser de barrido 3D. Determinar si un punto corresponde a un obstáculo o no a partir de la información de este láser no es tan sencillo como en el caso anterior, pues el mismo obtiene en cada instante un corte 2D del entorno 3D.

En una primera aproximación al problema, se trató de reconstruir la nube de puntos 3D combinando los cortes 2D empleando información de la odometría. En ese caso, los obstáculos se calculan empleando técnicas de filtrado sobre la nube 3D (por ejemplo, calculando la pendiente local en cada punto). Esta aproximación da buenos resultados en la mayor parte de los casos. Sin embargo, los experimentos mostraron que errores pequeños en la estimación de la orientación se amplifican con la distancia a los puntos, dando lugar a rugosidades en la nube 3D (sobre todo cuando el robot gira). Dichas rugosidades inducen obstáculos fantasma no existentes. En resumen, el principal problema de esta aproximación es que la coherencia del mapa depende en gran medida de la precisión en la estimación de la posición (sobre todo la orientación).

La solución adoptada finalmente depende en menor medida de la localización. Cada medida del láser (cada corte 2D) se procesa de forma individual (ver Figura 6). La suposición es que el suelo es una línea aproximadamente horizontal situada delante del robot; por tanto, el suelo se extrae de la medida láser mediante un ajuste de mínimos cuadrados y empleando un algoritmo RANSAC modificado. La modificación permite definir una función de probabilidad antes de la fase de muestreo de RANSAC, favoreciendo ciertas regiones espaciales de búsqueda.

Una vez el suelo es identificado, dos líneas rectas se extraen a ambos lados del mismo. La continuidad de dichas líneas con respecto al suelo es analizada, y si es así, se consideran rampas. Los puntos de dichas rampas se consideran navegables si la pendiente de la rampa es menor que un cierto umbral, y obstáculos en otro caso. Una vez determinados los obstáculos, se emplea la localización para actualizar el mapa global.

Adicionalmente, la información de líneas previas del láser se emplea para determinar la fiabilidad de las medidas. En particular, se chequea la altura estimada del suelo relativa al robot, cambiando el estado interno a no confiable cuando el láser casualmente incide en un muro o en otra superficie elevada. La Figura 7 muestra un ejemplo de los elementos clasificados como obstáculos a partir de un conjunto de medidas láser.

4.2.3 Mapa global de obstáculos

La información de todos los láseres se combina empleando la regla de Bayes. Sin embargo, al combinar la información de los distintos láseres en el mapa, se tiene en cuenta la altura de cada medida. Se asume que los obstáculos surgen del suelo. Por tanto, si se recibe una medida de espacio libre en

la celda (i, j) con una láser a una altura z_m, dicha medida no modificará la clasificación de esa celda como obstáculo si un obstáculo con altura $z < z_m$ fue previamente detectado en la misma celda.

4.3 CONTROL DEL ROBOT

El objetivo de control de los módulos de bajo nivel es tratar de seguir el camino comandado de la forma más fiel posible mientras se evitan los obstáculos presentes (estáticos y dinámicos). Para ello, se emplea un módulo de arbitraje, inspirado por el trabajo desarrollado en [11], que recibe consignas de una serie de módulos de comportamiento; en este caso, de los módulos de seguimiento de caminos y evitación de colisiones.

Cada uno de estos módulos genera una determinada intención sobre el espacio de actuación. Dicha intención se codifica como una serie de votos $\{h_c, h_s\}$, cada voto un histograma normalizado sobre una variable de actuación (en este caso, velocidad y curvatura). El módulo de arbitraje mantiene una copia de los histogramas que recibe y los fusiona empleando una regla simple de multiplicación. El máximo del histograma resultante se elige como la actuación a enviar a los motores.

4.3.1 Evitación de colisiones

La evitación de colisiones es especialmente complicada en el escenario, un campus universitario principalmente diseñado para peatones. Los pasillos estrechos y curvas pronunciadas pueden llevar a situaciones de bloqueo para un vehículo no holónomo (ver Fig. 1).

El algoritmo de evitación reactiva tiene en cuenta la cinemática del vehículo, en línea con la propuesta de Mínguez et al. [6]. Un láser virtual, generado a partir del mapa global de obstáculos, da la distribución de obstáculos alrededor de Romeo. Basado en dicha información, se calcula la distancia $d(\tau)$ hasta el obstáculo más cercano para cada curvatura τ (considerando trayectorias de curvatura constante). A cada curvatura se le asocia un peso dado por:

$$w(\tau) = \begin{cases} 0 & d(\tau) < d_{NEAR} \\ 1 & d(\tau) \geq d_{FAR} \\ f(d(\tau)) & otherwise \end{cases}$$

donde $f(.)$ es una función monótona de la distancia entre 0 y 1. En el caso de regiones libres de obstáculos, se añade una campana a los pesos centrada en la curvatura media de dicha región, de forma que se priorizan las curvaturas situadas en el centro de las áreas libres de obstáculos.

Los pesos, una vez normalizados, constituyen la intención de curvatura generada por el módulo de evitación de obstáculos. La intención en velocidad se modela como una distribución normal, con su media proporcional a la cantidad de espacio libre que rodea al robot. Puesto que la seguridad tiene la máxima prioridad, la velocidad se pone a 0 en el momento en que un obstáculo entra dentro de una distancia de seguridad.

4.4 Seguimiento de caminos

Un algoritmo de persecución pura adaptado a la aplicación, descrito en [4], se emplea para seguimiento de caminos. Los comandos dados por dicho algoritmo se convierte en votos para el módulo de arbitraje situando una campana de Gauss con media dichos comandos y una desviación típica de modo que la campana cubra el rango admisible de controles.

5 RESULTADOS EXPERIMENTALES

Romeo ha participado en los experimentos finales del proyecto europeo URUS. Como se ha comentado, el rol principal de Romeo era el realizar misiones TAXI. En cada misión, el robot es requerido a desplazarse hacia una estación desde su posición actual, donde un usuario es recogido. A continuación, el usuario indica su destino y es llevado allí. A lo largo del camino, el usuario puede parar y reanudar la marcha del robot a través de la interfaz HRI.

Aquí se resume uno de los experimentos. Éste consiste en 4 misiones TAXI consecutivas. La longitud total recorrida en el mismo es de 772 metros, y la duración es de 34 minutos, sin ninguna intervención externa. El experimento incluye 8 paradas y reanudaciones por los usuarios que están siendo transportados, así como la ejecución de 25 trayectorias distintas.

En relación con la localización del robot en el experimento, la Fig. 8 muestra los resultados. La trayectoria consta de varios bucles. Puede observarse como la corrección basada en mapa permite mantener acotado el error de localización.

La Fig. 9 muestra algunos detalles del comportamiento del robot durante la navegación. Puede verse el camino realmente ejecutado frente al camino indicado por el planificador. La presencia de desviaciones es previsible, pues el robot tiene que evitar peatones en su camino. La Fig. 10 describe una vista de los obstáculos típica en otro experimento. Un grupo de personas está situado sobre el camino planificado, de forma que el robot evita dicho grupo ejecutando el camino de coste

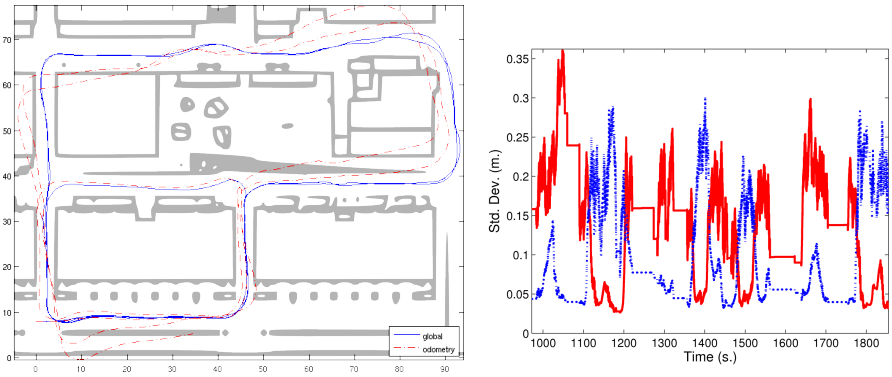

Figura 8: Izquierda: posición 2D estimada considerando sólamente odometría (línea discontínua roja) y con la corrección basada en mapas (línea contínua azul). Derecha: desviación estándar de la estimación. Las líneas rectas son momentos donde el robot está parado por solicitud del usuario o cuando una tarea a finalizado.

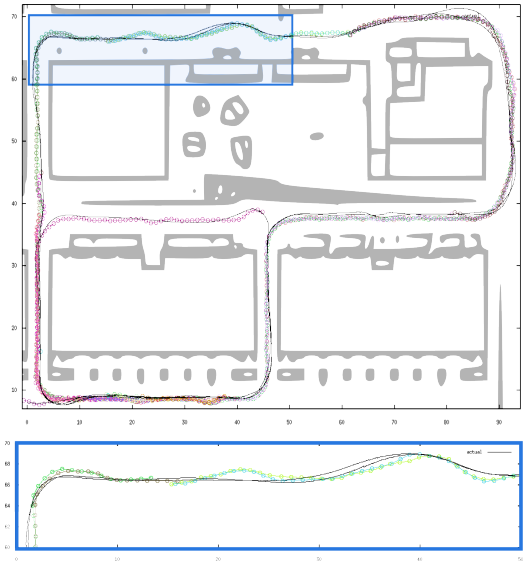

Figura 9: Trayectoria de referencia (círculos) y camino realmente ejecutado. Se incluye una vista de detalle. Las desviaciones se deben sobre todo al comportamiento de evitación. La misión involucró la ejecución de 25 trayectorias distintas.

mínimo indicado por el árbitro.

A lo largo de la semana de experimentos, Romeo se desplazo de modo totalmente autónomo más de 7 kilómetros, con un tiempo de funcionamiento aproximado de 590 minutos. Durante los 2 últimos días ejecutó toda la arquitectura descrita en la Sección 3 y realizó 10 misiones completas. Ocho de las misiones fueron totalmente exitosas, incluyendo la descrita más arriba. Una misión no se completó debido a un fallo hardware en uno de los láser Hokuyo empleados. En la otra, el robot acabó en una situación de bloqueo seguro al evitar un obstáculo de la que no pudo recuperarse debido a las restricciones de maniobrabilidad del robot.

6 CONCLUSIONES

El presente artículo ha presentado la arquitectura de un coche eléctrico para realizar servicios en zonas urbanas. Los algoritmos de bajo nivel han sido descritos. La arquitectura ha sido probada en experimentos reales en los que se realizaron misiones de taxi, en las que un pasajero es trasladado a un destino determinado.

Una de las principales conclusiones es la posibilidad que ofrecen coches eléctricos para las tareas propuestas. La arquitectura desarrollada es flexible para evitar el sobreajuste al escenario del proyecto. Si se busca la introducción de robots en entornos urbanos, éstos deben imponer requerimientos mínimos en la infraestructura disponible. El principal requerimiento de la arquitectura presentada es un mapa para la localización del robot.

Una importante lección aprendida está relacionada con la actuación de robots en entornos compartidos con humanos. Tras algunos días de experimentos, la mayoría de la gente en el campus se acostumbra a la presencia de los robots,

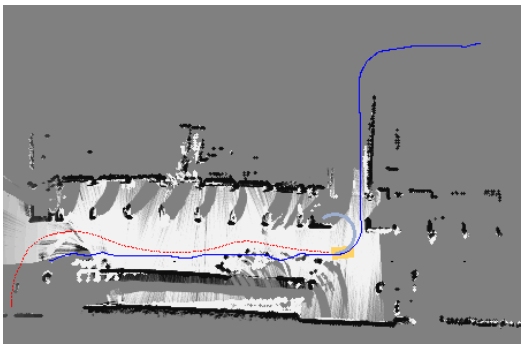

Figura 10: Mapa de obstáculos. Punteada en rojo se muestra la trayectoría ejecutada, frente a la requerida (en azul). El robot debe evitar un grupo de personas situadas en su camino.

sin preocuparse de sus acciones. Esto tiene que ver con el modo en el que los robots se mueven. Romeo ha sido programado para moverse de forma suave, evitando maniobras agresivas. Parece claro que el uso de movimientos aceptables socialmente es necesario para la integración de robots en entornos urbanos, lo que requerirá el desarrollo de nuevos algoritmos de planificación.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto URUS (IST-2006-33579), de la Comisión Europea, y el proyecto Robótica Ubícua en Entornos Urbanos (P09-TIC-5121) financiado por la Junta de Andalucía. Los autores agradecen la colaboración de todos los miembros del proyecto URUS durante los distintos experimentos en Barcelona.

Referencias

[1] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. Editorial: Special Issue on the 2007 DARPA Urban Challenge. *Journal of Field Robotics*, 25(8):423–424, 2008.

[2] J. Capitán, L. Merino, F. Caballero, and A. Ollero. Delayed-state information filter for decentralized tracking. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation, ICRA*, 2009.

[3] C.U. Droguer, A.B. Koku, and M. Dolen. Novel Solutions for Global Urban Localization. *Robotics and Autonomous Systems*, 2009.

[4] F. Gómez-Bravo, D.A. López, F. Real, L. Merino, and J.M. Matamoros. Integrated path planning and tracking for autonomous car-like vehicles maneuvering. In *Proc. of the 6th Intl. Conf. on Informatics in Control, Automation and Robotics, ICINCO*, 2009.

[5] P. Lamon, S. Kolski, and R. Siegwart. The SmartTer - a Vehicle for Fully Autonomous Navigation and Mapping in Outdoor Environments. In *Proceedings of CLAWAR*, 2006.

[6] J. Minguez, L. Montano, and J. Santos-Victor. Reactive navigation for non-holonomic robots using the ego kinematic space, 2002.

[7] J. M. Mirats-Tur, C. Zinggerling, and A. Corominas-Murtra. Geographical information systems for map based navigation in urban environments. *Robotics and Autonomous Systems*, 57(9):922–930, 2009.

[8] Yoichi Morales, Alexander Carballo, Eijiro Takeuchi, Atsushi Aburadani, and Takashi Tsubouchi. Autonomous robot navigation in outdoor cluttered pedestrian walkways. *Journal of Field Robotics*, 26(8):609–635, 2009.

[9] Alejandro R. Mosteo and Luis Montano. Simulated annealing for multi-robot hierarchical task allocation with flexible constraints and objective functions. In *Workshop on Network Robot Systems: Toward intelligent robotic systems integrated with environments, IROS*, 2006.

[10] C. Pradalier, J. Hermosillo, C. Koike, C. Braillon, P. Bressière, and C. Laugier. The Cycab: a car-like robot navigating autonomously and safely among pedestrians. *Robotics and Autonomous Systems*, 50(1):51–68, 2005.

[11] Julio Rosenblatt. Damn: A distributed architecture for mobile navigation - thesis summary. In *Journal of Experimental and Theoretical Artificial Intelligence*, pages 339–360. AAAI Press, 1995.

[12] A. Sanfeliu, J. Andrade-Cetto, M. Barbosa, R. Bowden, J. Capitan, A. Corominas, A. Gilbert, J. Illingworth, L. Merino, J.M. Mirats, P. Moreno, A. Ollero, J. Sequeira, and M.T.J. Spaan. Decentralized Sensor Fusion for Ubiquitous Networking Robotics in Urban Areas. *Sensors*, 10:2274–2314, 2010.

[13] Christopher Urmson et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 25(1):425–466, June 2008.