

Arquitectura basada en ROS para el vehículo iCab (Intelligent Campus Automobile)

P. Marin-Plaza, Ahmed Hussein, Carlos Guindel, Fernando García, David Martín, Arturo de la Escalera.
Laboratorio de Sistemas Inteligentes (LSI)

Departamento de Sistemas y Automática. Universidad Carlos III de Madrid. Calle Butarque 15
{pamarinp, ahussein, cguindel, fegarcia dmgomez, escalera}@ing.uc3m.es

Resumen

Este documento presenta la plataforma inteligente de investigación para entornos urbanos llamada iCab (Intelligent Campus Automobile). El principal objetivo es describir los pasos iniciales para alcanzar un vehículo autónomo funcional. La plataforma es un carrito de golf modelo E-Z-GO el cual ha sido modificado para operar de forma autónoma. El núcleo de la arquitectura está basado en ROS (Robot Operating System), y es capaz de recibir información de múltiples sensores con huella temporal utilizando un ordenador embebido. El sistema propuesto muestra las ventajas de la arquitectura basada en ROS para el manejo de datos entre procesos, así como la capacidad de gestionar, procesar y entender la gran cantidad de información recibida por las cámaras o el láser. La información de los sensores se ha integrado en la arquitectura para desarrollar aplicaciones innovadoras capaces de cubrir los requerimientos de las decisiones que tienen que tomar los vehículos autónomos. El ciclo de vida de los proyectos de estas características suele ser corto debido a la evolución de los sistemas de percepción del entorno y la capacidad de procesamiento; por tanto, la utilización de una arquitectura modular, portable e independiente de los sensores es indispensable.

Palabras clave: Vehículos Autónomos, Arquitectura Software, ROS, Sistemas de Percepción, Láser, Cámara Estéreo, Visión por Computador.

1. INTRODUCCIÓN

Los datos procedentes de la Organización Mundial de la Salud (OMS) muestran que las muertes por accidentes se han estabilizado desde el año 2007 gracias a los avances en medidas de seguridad. Siguen siendo un problema para los gobiernos de todo el mundo ya que las muertes globales por accidentes derivados del tráfico superan los 1.25 millones por año [19]. La gran mayoría de estos accidentes son causados por errores humanos que podrían haberse minimizado utilizando vehículos autónomos. Las primeras demostraciones de las capacidades de este tipo de vehículos tuvieron lu-

gar en los Estados Unidos a través de una agencia de proyectos de defensa llamado DARPA (Defense Advanced Research Projects Agency) para desarrollar los primeros coches autónomos. La primera competición del DARPA Grand Challenge se desarrolló entre 2004 y 2005 en el desierto de Mojave y sin obstáculos dinámicos [18]. La segunda competición, DARPA Urban Challenge, fue en el 2007 en un circuito urbano en el cual los vehículos que competían lo hacían a la vez simulando el tráfico normal de la ciudad [3]. Posteriormente, el primer coche autónomo funcional fue el Google Self-Driving Car, diseñado para cumplir las reglas de circulación [9]. Los coches autónomos han seguido siendo un tema de discusión muy importante en los sistemas de transporte inteligente, donde la investigación más reciente muestra que los vehículos sin conductor estarán disponibles de 5 a 10 años [10]. Hoy en día, las aplicaciones que utilizan al vehículo como medio para obtener información del entorno consiguen muy buenos resultados gracias al análisis mediante visión por computador y dispositivos láser. Estos dispositivos tienen limitaciones técnicas de percepción a la hora de hacer frente a cambios bruscos en el entorno como pueden ser variaciones en la iluminación, sombras o visión nocturna entre otros. Por ello, las aplicaciones de percepción deben asegurar la robustez adecuada en todas las condiciones mientras el vehículo esté en la carretera [4]. Uno de los ejemplos de la implementación de sistemas basados en la percepción visual y el procesamiento de imágenes por computador en vehículos autónomos fue la prueba llevada a cabo por el coche Mercedes Benz Clase S que condujo sin conductor desde Mannheim a Pforzheim 103 Km. El coche disponía de un equipo completo de sensores de largo alcance, mapas digitales y un sistema de visión por computador [6].

Los problemas a los que se tiene que enfrentar un coche autónomo a lo largo de su vida útil son los derivados de las siguientes cuestiones: ¿Dónde estoy? ¿Dónde voy? y ¿Cómo voy? Estos problemas se resuelven mediante la localización del vehículo, la planificación y la navegación, por ese orden. Muchos investigadores ya han resuelto estos problemas para entornos interiores estructurados. El

resultado muestra la posibilidad de generar una ruta libre de obstáculos desde un punto inicial al punto final y navegar por la trayectoria generada gracias a la localización del vehículo [16] [5] [8]. Por otra parte, para entornos exteriores, muchos desarrolladores han preferido implementar algoritmos más robustos basados en láser multiplano [7] [11] para evitar los problemas derivados de la visión por computador arriba mencionados. Debido a la gran cantidad de datos provenientes de este tipo de sensores, la carga computacional debe ser muy elevada. Existen otros investigadores que basan su trabajo en la visión estereoscópica que también proporciona información del entorno. Para la parte de navegación, localización y planificación, ROS dispone de unos repositorios oficiales con paquetes dedicados a resolver estos problemas que superan en simplicidad, eficiencia y portabilidad a otros de género similar [1] [17] [12].

Este trabajo presenta los primeros pasos en la implementación de una arquitectura basada en ROS para la plataforma de desarrollo iCab con el objetivo de conseguir un vehículo autónomo funcional. La principal meta es implementar algoritmos de planificación tanto global como local, localización global y local y navegación autónoma con la toma de decisiones basada en la información visual y la interpretación de los datos obtenidos por el láser. La implementación de la arquitectura multifuncional basada en ROS ha sido en un ordenador embebido situado en la parte trasera del carrito de golf.

Las aplicaciones desarrolladas en la plataforma iCab se basan en la fusión sensorial, la cual mejora de forma significativa los resultados obtenidos en los distintos problemas descritos anteriormente. Gracias a la arquitectura software propuesta, la comunicación entre procesos permite extender el entendimiento del entorno mejorando la toma de decisiones. Otra de las ventajas es la modularidad de los procesos, que son independientes entre sí e independientes de los sensores. Esto permite actualizar los sensores de manera rápida y sencilla mientras mantengan la misma estructura de mensajes. Todos los procesos se ejecutan en paralelo compartiendo información en memoria accesible por todos los nodos implicados gracias a los punteros inteligentes. Las aplicaciones generadas de bajo nivel proporcionan la información necesaria a las aplicaciones híbridas que elaboran los mapas para las aplicaciones de toma de decisión.

La estructura de este documento se divide en cinco secciones. La sección 2 describe la plataforma experimental iCab. La sección 3 presenta la arquitectura propuesta basada en tres capas. La sección 4 muestra los resultados experimentales en diferen-

tes escenarios y finalmente la sección 5 contiene las conclusiones y el trabajo futuro.

2. DESCRIPCIÓN DE LA PLATAFORMA

El vehículo elegido para ser la plataforma de investigación ha sido un carrito de golf eléctrico, modelo E-Z-GO, mostrado en la Figura 1. Ha sido modificado mecánicamente para cumplir los objetivos del proyecto, en términos de navegación autónoma y planificación. Además se dispone de otro vehículo de las mismas características para la comunicación entre vehículos.



Figura 1: Plataforma de investigación iCab

Las modificaciones realizadas consisten en: la sustitución del volante por un sistema motor-encoder para controlar la dirección, como se muestra en la Figura 2; la desactivación física del acelerador; y la incorporación de un microcontrolador que gobierna el estátor y el rotor del motor eléctrico de forma independiente. Todos estos cambios sirven para adaptar la potencia a los diferentes entornos como cuestas o terreno de difícil navegación. Las entradas al microcontrolador son la potencia y el ángulo deseado.



Figura 2: Sustitución del volante por sistema motor-encoder

Para la percepción del entorno, los vehículos están equipados con un láser de largo alcance (SICK LMS291) capaz de proporcionar información 2D con una resolución de 0.25° en un abanico de 100° [15]. Se ha posicionado a 30cm del suelo en la parte frontal del vehículo. Se ha limitado a 100 grados de apertura para evitar las lecturas de las ruedas.

La información proporcionada por este láser se actualiza cada 5Hz.

Además del láser, se dispone de una cámara de visión estéreo binocular [13] (Bumblebee 2). Esta cámara se ha configurado a 640x480 píxeles de resolución a 20 imágenes por segundo. Está montada en el parabrisas frontal del vehículo a 1.60m de altura con una orientación de 15 grados de diferencia con el plano del suelo apuntando hacia abajo. La cámara se usa en tres tareas principales: la primera es complementar la información proporcionada por el láser a la hora de construir un mapa de ocupación libre de obstáculos; la segunda es generar una odometría visual basada en características del entorno; y la tercera es detectar la señalización de la carretera para definir un mapa de reglas de la carretera.

Como sensor inercial, se utiliza una brújula electrónica y un GPS integrado en el mismo dispositivo (3DR uBlox GPS) con motivo de mejorar la orientación y navegación creando una odometría más precisa. Está instalada en la parte exterior del techo del vehículo para evitar interferencias con la potencia suministrada al motor.

Todos los dispositivos se conectan a un ordenador embebido Intel Core i7 que trabaja bajo el sistema operativo Ubuntu. La pantalla táctil de 7 pulgadas LCD permite al usuario tener constancia de la ubicación en el mapa del vehículo e interactuar con el sistema por si se produjese algún cambio en la ruta. Está instalada en la parte interior fijada al techo donde los pasajeros tienen muy fácil acceso a la información y a la interacción con la misma.

3. DESCRIPCIÓN DE LA ARQUITECTURA

3.1. ARQUITECTURA PROPUESTA

En este documento, el objetivo es definir una arquitectura completa de varios niveles de abstracción categorizada en tres capas: deliberativa, secuencial y reactiva [14]. La Figura 3 describe estas tres capas. La principal ventaja, entre otras, es la posibilidad de añadir más habilidades de bajo y alto nivel sin modificar el código fuente, sólo ejecutando ciertas habilidades que interesen en ese momento. Debido a que es una plataforma de investigación, cabe la posibilidad de realizar las mismas tareas pero con diferentes algoritmos. La capa baja de la arquitectura, o capa reactiva, contiene los comandos básicos sobre los actuadores y realiza la lectura principal de los sensores que definen el entorno cercano. La capa intermedia, o híbrida, recibe órdenes de la capa de más alto nivel y los transforma en tareas sencillas ordenadas por una

secuencia lógica. La capa de alto nivel contiene las tareas necesarias para seleccionar la ruta óptima entre dos puntos.

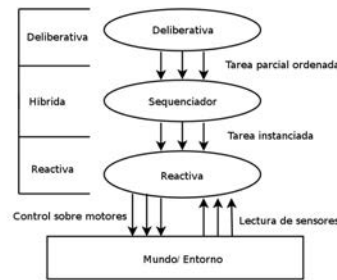


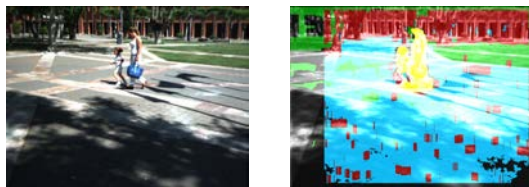
Figura 3: Arquitectura en tres capas

- **Capa Reactiva:** el control básico del vehículo autónomo está implementado en esta capa. Los comandos seleccionables como “Avanza”, “Gira”, “Retrocede” o “Para” se pueden seleccionar para realizar complejos movimientos autónomos. Las salidas de control van directamente al micro controlador PIC para que realice las acciones directas en los motores.
- **Secuenciador:** la tarea de alto nivel se divide en tareas simples en esta capa. La complejidad reside en la elección de tareas sencillas para componer de la mejor forma la tarea de alto nivel propuesta.
- **Capa Deliberativa:** la lógica de esta capa determina las acciones y las elecciones que debe tomar el vehículo en base a la localización, planificación, navegación y mapeado del entorno. La información proporcionada a esta capa proviene directamente del usuario que seleccionará los puntos a los que quiere ir y la propagación de información de los sensores hasta la creación de los mapas determinará la mejor ruta posible.

3.2. DESCRIPCIÓN DE NODOS EN ROS

En concordancia con la descripción anterior de la arquitectura y el trabajo relacionado mencionado, se ha decidido implementar todos los algoritmos en un sistema modular llamado ROS. El driver de la cámara estéreo recibe la información directamente del sensor y publica las imágenes de ambas cámaras rectificadas, tanto en escala de grises como en color, para su posterior uso en algoritmos de detección de obstáculos, como el presentado en [2], y en algoritmos de navegación. Para el sensor láser, se ha utilizado el driver por defecto para ROS *sicklms*, que publica los valores de las distancias en un array para su posterior postprocesado.

Un ejemplo de la gestión de los obstáculos detectados mediante la cámara estéreo y el procesamiento posterior está en la Figura 4 (a), donde aparecen peatones, zonas de césped y obstáculos estáticos como son árboles y edificios. En la Figura 4 (b), se han clasificado estos obstáculos por colores mediante máscaras. En azul, el espacio libre; en verde el espacio detectado como césped o vegetación; y en rojo los obstáculos no pertenecientes al espacio libre. Puede comprobarse la aparición de pequeñas zonas rojas en el plano transitable de la calzada debido al ruido procedente del mapa de disparidad del procesamiento estéreo. Son debidos a las sombras y no se tienen en cuenta los menores de un determinado área a la hora de generar los avisos de posibles amenazas de obstáculos en el trayecto. También puede comprobarse cómo la zona libre de obstáculos, en azul, no cubre del todo la imagen, ya que sólo contiene información de la zona común a las imágenes izquierda y derecha.



(a) Imagen izquierda a color de la cámara estéreo (b) Máscaras de los obstáculos sobre la imagen izquierda

Figura 4: Detección de Obstáculos

Para la interfaz gráfica de usuario, se ha diseñado un nodo llamado *icab_reconfigure*, el cual envía mensajes de control, supervisión y estado al nodo encargado del movimiento del vehículo. En la Figura 5 se puede observar la interfaz creada con Qt, que muestra el estado del vehículo, su localización y los botones de marcha y paro.

En la Figura 6 se describen los nodos involucrados en la conducción autónoma de la plataforma iCab. Para la localización se han desarrollado los nodos de *wheel_odometry*, *gps_odometry*,



Figura 5: Interfaz gráfica

visual_odometry y *odometry_manager*. Para la detección de obstáculos y generación del mapa local, se han desarrollado *obstacle_detection*, *hog_pedestrian*, *grass_detection*, *labeller* y *mapping_server*. Por último, la interfaz de usuario mediante *icab_reconfigure* y el control de sonido mediante *sound_manager* cierran la arquitectura.

4. EJEMPLIFICACIÓN DE COMPORTAMIENTO

En esta sección, se expone una ejemplificación de la arquitectura propuesta, donde la estructura de bajo nivel, la gestión de la información de los distintos dispositivos de adquisición de datos y la sincronización de procesos son evaluados en base a las maniobras generadas por el vehículo. Los escenarios evaluados se han sometido a diferentes experimentos y en esta sección se muestra un escenario representativo acorde a todas las pruebas realizadas.

Los resultados han sido obtenidos usando el ordenador embebido del vehículo con una versión en proceso de desarrollo de la arquitectura expuesta. Para gobernar los movimientos de la plataforma de investigación iCab, se han establecido una serie de condiciones iniciales que se exponen a continuación: el comando de control del vehículo se realiza mediante la adición o sustracción del 5% de la potencia máxima del rotor poniendo fija la potencia al 70% del estátor para que el balance entre el consumo y la potencia sea óptima. El límite máximo de la potencia del rotor será del 40% y el mínimo de 0%. El ángulo de giro del vehículo se calculará en base a la distancia del muro entre +/- 5°. El ángulo máximo en cualquier dirección será de 30°. Estos límites se han seleccionado por motivos de seguridad como pasos iniciales hacia el vehículo completamente autónomo.

El primer experimento consiste en seguir una pared paralela al vehículo a una 5 metros. Los sensores determinan esta distancia y el vehículo actuará en consecuencia girando las ruedas para acercarse o alejarse del muro. En la Figura 7 (a) se muestra en verde la distancia del vehículo en paralelo a la pared. En azul, se muestra los comandos de control generados por la capa de bajo nivel.

El segundo experimento consiste en la reacción del vehículo ante un obstáculo situado en su camino. La detección del mismo mediante el láser situado en la parte central (con ángulo 0, indica justo en el frontal del vehículo) como referencia. Se ha establecido que, en el momento de la detección del vehículo, se envíe de forma inmediata la señal de desactivación de la aceleración. La Figura 7 (b)

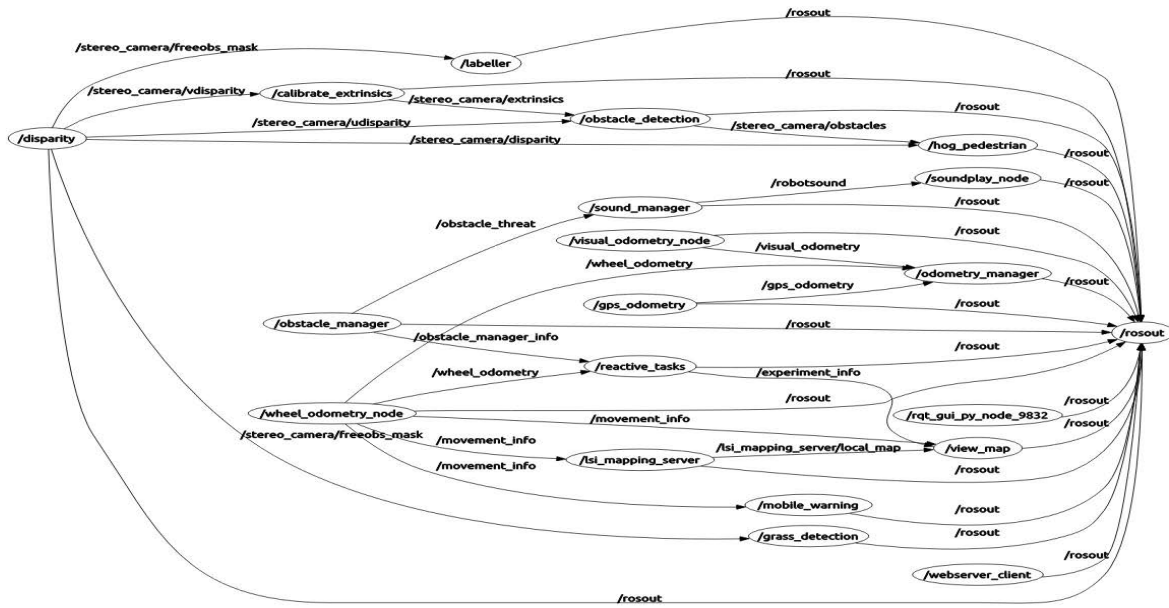


Figura 6: Arquitectura general con nodos de ROS

muestra en verde la distancia del obstáculo más cercano enfrente del vehículo. En azul se muestran los comandos de control de aceleración. En torno al segundo 64, un peatón se cruza en la trayectoria del iCab. Debido a que se detecta primero por un lateral, el vehículo para antes de que el peatón pase por el centro. Posteriormente, sobre el segundo 84, otro peatón se cruza y el vehículo para.

5. CONCLUSIONES Y TRABAJO FUTURO

Este trabajo presenta el diseño, desarrollo y puesta a prueba de una arquitectura basada en ROS para la plataforma de desarrollo iCab sobre vehículos autónomos en entornos urbanos. El principal objetivo es proporcionar al iCab las capacidades necesarias para usarse como un método inteligente de transporte funcional. Los procesos de bajo nivel propuestos en la arquitectura usando los sensores láser y cámara estéreo para generar comandos de control han quedado demostrados. Por lo tanto la utilidad básica de este sistema en cuanto a adquisición de datos, procesamiento de imágenes y cohesión de procesos trabajando de forma paralela es funcional.

Como trabajos futuros queda la implementación de una planificación autónoma, donde sea el usuario quien introduzca el punto de inicio y final y el vehículo elija la ruta óptima, el secuenciador para elegir la mejor división de puntos de camino con su correspondiente comportamiento de tareas sencillas, la localización del vehículo en el campus de forma autónoma utilizando el GPS y los distintos métodos de odometría disponibles, y la generación

del mapa de ocupación basado en la fusión de los distintos sensores.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Gobierno de España a través de los proyectos Cycit (TRA2013-48314-C3-1-R y TRA2015-63708-R), y por la Comunidad de Madrid a través del proyecto SEGAUTO-TRIES (S2013/MIT-2713).

Referencias

- [1] A. Llamazares, E. Molinos, M. Ocana, and F. Herranz, “Integrating ABSYNTH autonomous navigation system into ROS,” International Conference on Robotics and Automation (ICRA’2014). IEEE., 2014.
- [2] B. Musleh, D. Martín, A. de la Escalera, and J. M. Armingol, “Visual ego motion estimation in urban environments based on uv disparity,” in Intelligent Vehicles Symposium (IV), 2012 IEEE. IEEE, 2012, pp. 444–449.
- [3] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al., “Autonomous driving in urban environments: Boss and the Urban Challenge,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.
- [4] D. Martín, F. García, B. Musleh, D. Olmeda, G. Peláez, P. Marín, A. Ponz, C. Rodríguez, A. Al-Kaff, A. De La Escalera, et al., “IVVI 2.0: An intelligent vehicle based on computational perception,” *Expert Systems with*

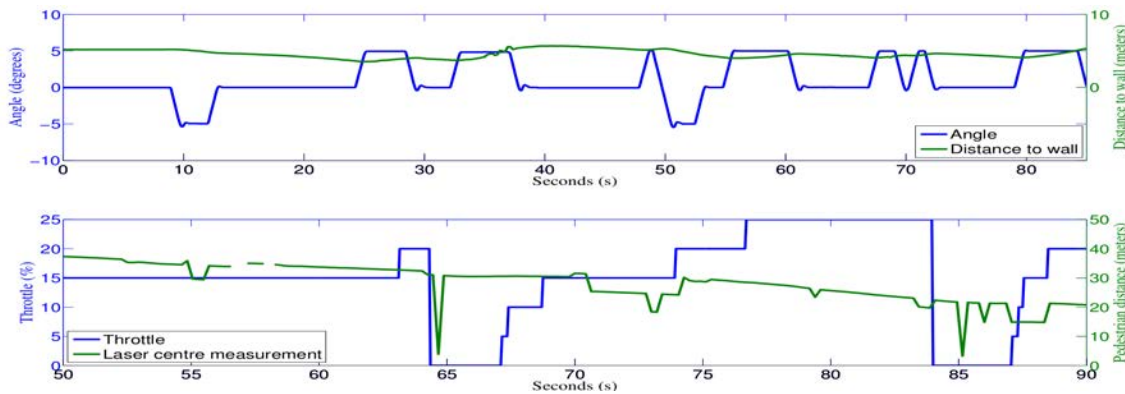


Figura 7: Tareas básicas de bajo nivel usando los sensores láser y la cámara estéreo, comandos de control de la velocidad y de la dirección: (a) Seguimiento de una pared, (b) detección y freno ante un obstáculo

- Applications, vol. 41, no. 17, pp. 7927–7944, 2014.
- [5] I. Wieser, A. V. Ruiz, M. Frassl, M. Angermann, J. Mueller, and M. Lichtenstern, “Autonomous robotic slam-based indoor navigation for high resolution sampling with complete coverage,” Position, Location and Navigation Symposium-PLANS’2014. IEEE., pp. 945–951, 2014.
- [6] J. Ziegler, P. Bender, M. Schreiber, H. Lagetahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. Keller, et al., “Making Bertha drive? an autonomous journey on a historic route,” Intelligent Transportation Systems Magazine, IEEE, vol. 6, no. 2, pp. 8–20, 2014.
- [7] Li, Y., Olson, E. B. (2010, May). Extracting general-purpose features from LIDAR data. In ICRA (pp. 1388-1393).
- [8] M. A. Hossain and I. Ferdous, “Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique,” Robotics and Autonomous Systems, Elsevier., vol. 64, pp. 137–141, 2015.
- [9] M. Harris. (2015) Google’s self-driving cars revealed. IEEE Spectrum. [Online]. Available: <http://spectrum.ieee.org/cars-that-think/transportation/self-driving/googles-selfdriving-car-pals-revealed>
- [10] M. Mitchell, “Autonomous vehicles: No drivers required,” Nature, vol. 518, pp. 20–23, 2015.
- [11] Moosmann, F., Pink, O., Stiller, C. (2009, June). Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion. In Intelligent Vehicles Symposium, 2009 IEEE (pp. 215-220). IEEE.
- [12] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source robot operating system,” in ICRA workshop on open source software, vol. 3, no. 3.2, 2009, p. 5.
- [13] P. Grey, “Bumblebee: stereo vision camera systems” Technical Description, 2012.
- [14] R. P. Bonasso, R. J. Firby, E. Gat, D. Kortenkamp, D. P. Miller, and M. G. Slack, “Experiences with an architecture for intelligent, reactive agents,” Journal of Experimental and Theoretical Artificial Intelligence, vol. 9, no. 2-3, pp. 237–256, 1997.
- [15] S. S. Intelligence, “Lms200/211/221/291 laser measurement systems”. Technical Description, 2006.
- [16] S. Shen, N. Michael, and V. Kumar, “Autonomous multi-floor indoor navigation with a computationally constrained mav,” International Conference on Robotics and Automation (ICRA’2011). IEEE., pp. 20–25, 2011.
- [17] S. Zaman, W. Slany, and G. Steinbauer, “Ros-based mapping, localization and autonomous navigation using a pioneer 3-dx robot and their relevant issues,” Saudi International Electronics, Communications and Photonics Conference (SIEPC’2011). IEEE., pp. 1–5, 2011.
- [18] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al., “Stanley: The robot that won the DARPA Grand Challenge,” Journal of field Robotics, vol. 23, no. 9, pp. 661–692, 2006.
- [19] W. H. Organization. Global status report on road safety 2015