

CLEAR. Un Módulo para la Robotización de Máquinas Ackermann

Saúl Cova-Rocamora, Iván del Pino, Miguel Á. Muñoz-Bañon, Miguel Á. Contreras,
Francisco A. Candelas y Fernando Torres
rscr1@alu.ua.es

Resumen

En este artículo se describe el trabajo realizado para crear un módulo que facilite la robotización de un vehículo con geometría Ackermann genérico. Este sistema permite el control y la monitorización del robot gracias a una interfaz hardware de bajo nivel con los actuadores y sensores de la plataforma, y a una interfaz software de alto nivel totalmente integrada en ROS. Por último, se ha realizado un test del módulo integrándolo en una máquina real analizando los resultados obtenidos. Dicho módulo recibe el nombre de CLEAR (Control Logic for Easy Ackermann Robotization) y ha sido desarrollado para ser genérico y de código y hardware abierto. Se pretende, por tanto, que este módulo sea una herramienta de ayuda a la investigación en robótica móvil terrestre, para todo aquel que quiera transformar una plataforma móvil Ackermann en un robot.

Palabras clave: Genérico, Robotización, Automatización, Vehículos Ackermann, Código abierto, Hardware libre, Robótica Móvil, ROS, Bajo Coste.

1. INTRODUCCIÓN

La robótica móvil terrestre es un campo que cada vez está cobrando más importancia y que va a tener una gran repercusión social y económica en la sociedad [6]. Esto es debido, en parte, a la investigación que se está llevando a cabo para desarrollar vehículos autónomos. En 2004 se celebró la competición DARPA Grand Challenge, que tuvo una segunda edición en 2005 y una tercera (Urban Challenge) en 2007, donde participaron pioneros como Thrun y su equipo con el robot Stanley[10], Mason y su equipo y el robot Golem[7], o Urmson y su equipo y el robot BOSS[11], entre otros. Aunque estos robots fueron desarrollados a partir de vehículos convencionales, se pueden encontrar robots móviles comerciales para investigación, como por ejemplo el RBCAR[8] o el robot RB-SHERPA[9].

En el momento en el que se va a investigar en

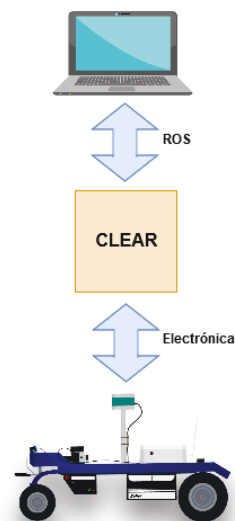


Figura 1: Descripción del módulo CLEAR a alto nivel.

robótica móvil o vehículos autónomos, lo principal es tener un robot sobre el cual desarrollar los algoritmos. En este sentido se plantean dos opciones: comprar un robot ya existente que se adecue bien a las necesidades y características del proyecto (precio, autonomía, velocidad máxima, geometría, etcétera), o robotizar una máquina genérica[12]. Ambas elecciones tienen sus ventajas y sus desventajas. Se podría señalar principalmente que al realizar uno mismo la transformación de una plataforma a un robot se obtiene una mayor flexibilidad al adecuarla perfectamente a las necesidades, pero por contra se han de invertir más horas de ingeniería que si se comprase un robot que ya realizase el control a bajo nivel de los sensores y actuadores, y que proporcione una interfaz de alto nivel.

Debido a que el objetivo del proyecto era la investigación en robótica móvil, una vez estudiadas las soluciones comerciales, se optó por transformar una plataforma Ackermann industrial en un robot. En la decisión se tuvo en cuenta la configuración geométrica de la máquina. Las diferentes configuraciones de los vehículos móviles pueden

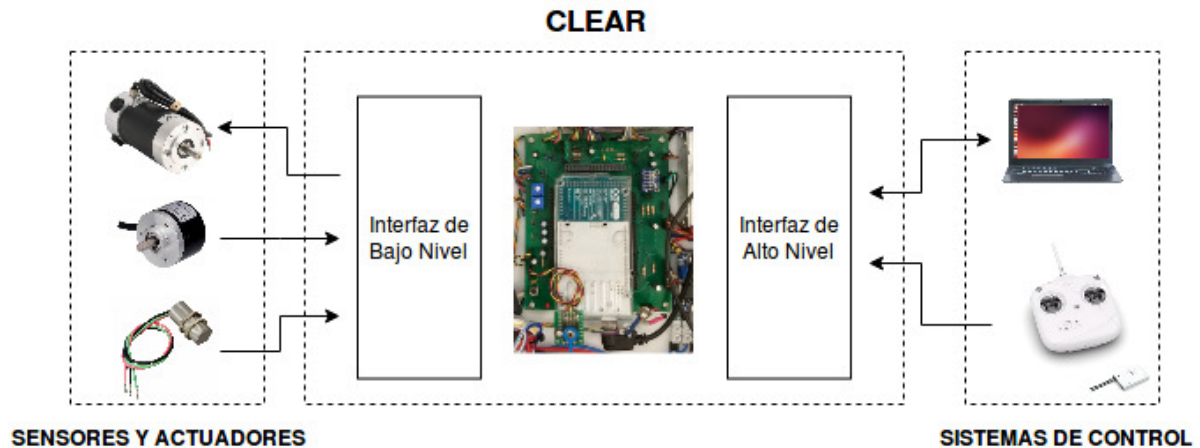


Figura 2: Arquitectura de CLEAR. Representando el Hardware y las interfaces Software con los sensores y actuadores del vehículo de bajo nivel y los sistemas de control de alto nivel.

ser, por ejemplo: diferencial, triciclo, Ackermann, omnidireccional, etcétera. Se decidió por la geometría Ackermann, debido a que suele proporcionar una mayor capacidad de carga que sus homólogos y permite realizar un control de dirección más sencillo[2]. Además, el hecho de que esta geometría sea la más usada en el sector automovilístico, posibilita trasladar la investigación realizada en un robot Ackermann a un vehículo convencional o industrial.

Durante el diseño del robot para adaptarlo a la plataforma escogida, una carretilla eléctrica Zallys Jespi Z105, se vio que los componentes y funcionalidades necesarias para integrarlo en ROS serían comunes a prácticamente cualquier máquina Ackermann, así que se decidió crear un módulo genérico llamado CLEAR (Control Logic for Easy Ackermann Robotization). CLEAR fue diseñado como una caja transparente en la cual los investigadores pudieran ver todo lo que sucedía en el interior del robot, a través de una interfaz de alto nivel que ha sido implementada en ROS (Robot Operating System) debido a las ventajas que ofrece en la arquitectura de las comunicaciones.

Por tanto, en este trabajo se describe el diseño y la implementación del módulo CLEAR (ver Fig. 1) concebido para facilitar la robotización de máquinas Ackermann. CLEAR ha sido diseñado para estar integrado como un nodo en el interior de ROS. Además, implementa diferentes niveles de abstracción para poder integrarse tanto en el hardware y la electrónica de la plataforma como en el software de alto nivel.

A continuación, se presenta en la sección 2 la arquitectura de CLEAR, en el apartado 3 el software, incluyendo la interfaz de alto y de bajo nivel, ya en la sección 4 se presenta el hardware, incluyendo la descripción de la electrónica empleada, a conti-

nuación, con un apartado dedicado a las medidas de seguridad y posteriormente la integración en un vehículo desarrollado por los autores, BLUE. El artículo finaliza con la sección de conclusiones y trabajos futuros.

2. ARQUITECTURA

CLEAR ha sido concebido para ser un **middleware** entre la plataforma y ROS, transformándola así en un robot. Como se puede ver en la Fig. 2, el sistema cuenta con dos niveles de abstracción y está compuesto por una electrónica concreta que conecta la capa de bajo nivel (actuadores y sensores) y un software que hace de interfaz con un sistema de alto nivel. CLEAR ofrece una conexión transparente entre ambos sistemas (vehículo y software) debido a que cuenta con el software y el hardware para controlar la velocidad, la dirección y el freno. Y con la interfaz de alto nivel permite monitorizar, configurar y controlar el vehículo. Finalmente, incluye una interfaz hombre-máquina, instalada en la plataforma para visualizar el estado del sistema y que corta la corriente en caso de emergencia.

Una característica esencial de CLEAR es que cualquier componente que incorpora, sea software o hardware, debe ser lo más fácilmente configurable o incluso intercambiable. Ya que se proporcionan los esquemas electrónicos básicos y los códigos en un repositorio [1] de forma abierta, a partir de los cuales, los investigadores pueden hacer modificaciones para adecuarlo a un hardware específico.

Otro aspecto importante en el diseño de CLEAR ha sido la seguridad. Ya que, si se implementa en un vehículo, sea industrial o no con una cierta robustez, es muy importante tener medidas de seguridad que eviten que el robot pueda causarse

daño a sí mismo, a infraestructuras o a seres vivos que se encuentren a su alrededor. Por eso se ha dotado a CLEAR de medidas redundantes de seguridad en todo el sistema, tanto en la interfaz con el bajo y alto nivel.

3. SOFTWARE

Para conseguir que CLEAR sea fácilmente integrable en diversas plataformas se ha tenido que desarrollar un código lo más genérico posible y que además se pueda modificar de forma sencilla. Para ello CLEAR incorpora varias formas de modificar sus parámetros ya que cada robot tendrá unos parámetros internos diferentes. En este sentido, aparte de poder ajustar parámetros de forma dinámica, es posible reorganizar las entradas y salidas del módulo CLEAR modificando un único fichero. Además, un archivo de descripción del robot permite establecer los valores constantes como las dimensiones del robot, la distancia entre ejes, o la velocidad máxima y el ángulo máximo de giro. Por otro lado, las ganancias del PID de cada controlador y los parámetros de calibración del robot se pueden ajustar de forma dinámica a través de *topics* de ROS. Esto facilita que CLEAR pueda adaptarse a distintas plataformas.

El software de CLEAR ha sido desarrollado en C++, encapsulando los componentes del robot de bajo nivel como objetos de diferentes clases. Estos componentes están dispuestos en una estructura de árbol, con la clase *Vehicle* como raíz principal, que contiene un objeto del sistema de velocidad, que a su vez está compuesto de diferentes objetos (motor, un codificador y un freno), y así sucesivamente. Gracias a esta estructura, resulta muy fácil propagar el valor de las variables desde las clases de bajo nivel, que operan con el hardware del vehículo, a los sistemas de alto nivel facilitando así el proceso de depuración. Esto minimiza los posibles errores que puedan surgir al modificar el código, ya que los errores en una clase están encapsulados en ella, lo cual dificulta que se propaguen a otros sistemas.

CLEAR implementa una máquina de estados[5] (ver Fig. 3) que cambia el comportamiento del robot dependiendo de los diferentes modos de operación. Un LED RGB de alta intensidad (ver Sección 4) permite conocer el estado actual de CLEAR en todo momento. Los posibles modos de funcionamiento son los siguientes:

- *Estado Calibración*: El módulo realiza una calibración del sistema de dirección utilizando los interruptores de fin de carrera, para arrancar el vehículo con un ángulo cercano a cero.

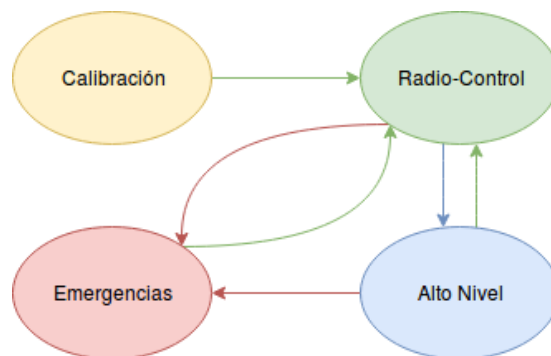


Figura 3: Representación de los estados posibles de CLEAR y sus transiciones.

Este proceso se realiza cada vez que se inicia CLEAR, y cuando finaliza, cambia por defecto al estado radio-controlado. Durante este proceso, el LED de estado emite una luz ámbar.

- *Estado Radio-Control*: En este modo, CLEAR mapea la posición de los joysticks del mando radiocontrol a las velocidades deseadas (metros por segundo) y ángulos de dirección (grados). Estos valores se utilizan como referencias para los controladores PID, permitiendo el control manual de la plataforma, mientras que el resto del sistema no se ve afectado. Además de esto, el RC permite cambiar manualmente el modo de operación a Control de alto nivel o Emergencia. En este modo, se emite una luz verde a través del LED de estado, y la interfaz ROS permite monitorizar el sistema y ajustar los parámetros dinámicos.
- *Estado Control de Alto Nivel*: Modo similar al estado radiocontrol. Las ordenes de dirección y velocidad vienen desde la interfaz de alto nivel. En este modo, se emite una luz azul.
- *Estado Emergencia*: En caso de que haya alguna señal de emergencia activa, la máquina de estados entrará directamente en este estado deteniendo los motores y accionando el freno de estacionamiento, para detener la plataforma. En este estado, CLEAR no ejecuta ningún comando de velocidad o dirección. Para salir de este estado, el operador debe seleccionar el modo de control remoto y reactivar el sistema explícitamente.

3.1. INTERFAZ DE ALTO NIVEL

ROS proporciona una forma sencilla y robusta de comunicaciones entre distintos sistemas (nodos). Teniendo en cuenta esto, se ha encapsulado CLEAR en un nodo que utiliza *rosserial* para comunicarse con ROS. La monitorización y control

del vehículo a través de ROS se realiza utilizando los *topics*. Se han diseñado las comunicaciones en las cuales se encuentran tres arquetipos distintos:

- *Topics de Control*: Estos *topics* permiten mandarle al vehículo los estados Ackermann deseados. También permite mandar los valores a las variables que pueden ser dinámicas en la plataforma como pueden ser ganancias de los controladores, valores para los filtros, etcétera. A través de un *topic* también se puede configurar el nivel de información que envía el vehículo para el depurado.
- *Topics de Depurado*: Estos *topics* permiten visualizar los valores de los comandos que se le han enviado a través de los *topics de control* para poder verificar que internamente está funcionando correctamente.
- *Topics de Monitorización*: Estos *topics* permiten visualizar las variables relevantes, como las medidas de los encoders, las estimaciones y covarianzas EKF, las entradas y salidas de los controladores PID y la configuración actual del sistema.

3.2. INTERFAZ DE BAJO NIVEL

La capa de control de bajo nivel de CLEAR está preparada para utilizar codificadores incrementales para estimar la velocidad y el ángulo de giro de la plataforma Ackermann en la que está instalada y como acción de control usa un controlador PID. Esto suele ser suficiente cuando se usan codificadores de alta resolución. Pero al robotizar una plataforma es posible que no se tenga acceso al motor de impulsión. Y estimar la velocidad o dirección con un codificador de baja resolución puede ser un problema.

Por ello, se han implementado dos variantes del Filtro de Kalman. Un Filtro del Kalman Unidimensional (SDKF)[4] que utiliza la señal de control para alimentar la etapa de predicción del filtro, este enfoque tiene la ventaja de ser extremadamente liviano, ya que las operaciones matriciales involucradas en las ecuaciones del Filtro Kalman se reducen a escalares, debido a la dimensión única del filtro. Esta implementación de SDKF es válida siempre y cuando haya un buen posicionamiento de la dirección. Si no es posible, se emplea el Filtro de Kalman Extendido (EKF) de 3 dimensiones para estimar la velocidad, la variación del ángulo de giro, y el ángulo inicial ya que este es desconocido (ver la Fig. 4). Para el modelo de la estimación de la velocidad en este caso se ha asumido que hay una relación lineal entre la tensión del motor y la velocidad[4],

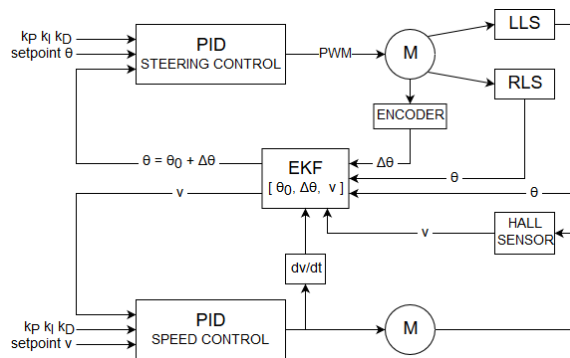


Figura 4: Esquema de control basado en el uso de controladores PID con un estimador de Filtro Kalman Extendido.

lo cual dependerá de las características de cada plataforma.

4. HARDWARE Y ELECTRÓNICA

El núcleo del módulo CLEAR es un Arduino Mega (debido a su gran capacidad de entradas y salidas) que interactúa con los sensores y actuadores de bajo nivel del vehículo. Todos los circuitos principales de CLEAR que trabajan a 5V, Arduino incluido, han sido organizados en la misma PCB principal (CLEAR PCB), mientras que los acondicionadores de señal para los sensores, así como los *drivers* y relés para los actuadores, están en varias PCBs diferentes, para seguridad y facilidad de mantenimiento y reemplazo.

Para la interfaz de sensores se han desarrollado otras placas de circuito impreso que incluyen acondicionamiento aislado para señales digitales de detectores y codificadores NPN o PNP, mediante optoacopladores.

Por último, cabe mencionar que CLEAR incluye un LED RGB de alta luminosidad para mostrar el estado del vehículo desde lejos, monitorizar variables como los niveles de tensión de la fuente de alimentación y las corrientes del motor, y también está conectado a un receptor DR16 RC del DJI que permite el pilotaje remoto desde un controlador RC.

5. MEDIDAS DE SEGURIDAD

Al estar embebido en un microprocesador es muy importante implementar diversas medidas de tolerancia a fallos. En este sentido CLEAR incorpora un *watchdog* que reinicia el microcontrolador si el tiempo de ciclo es mayor a cierto valor. Así se evita

que el procesador se quede atrapado en un bucle infinito y deje de responder.

También es importante tener en cuenta los rangos de las variables. Por eso antes de modificar los valores de cualquier variable se tienen en cuenta los valores máximos y mínimos, no permitiendo que se sobrepasen (configurables a través de ficheros, ver Sección 3). Y en caso de que la entrada supere el máximo o mínimo permitido, la entrada se satura a dichos valores. Esto se tiene en cuenta con las órdenes del mando radiocontrol y desde ROS, pero también con las entradas y salidas de los controladores PID.

Otra medida de seguridad ha sido la de llevar al sistema a emergencias si se pierde la comunicación entre el vehículo y el mando radiocontrol. Por otro lado, se puede configurar el mando de dos formas: usar un pulsador de “hombre muerto”, que cuando se suelta hace que entre en estado de emergencia; o mandarlo a emergencias pulsando un botón. En todo caso para salir de emergencia todas las setas de seguridad deberán estar rearmadas y se deberá rearmar también desde el mando siguiendo una secuencia de botones predefinida.

6. INTEGRACIÓN EN EL VEHÍCULO BLUE

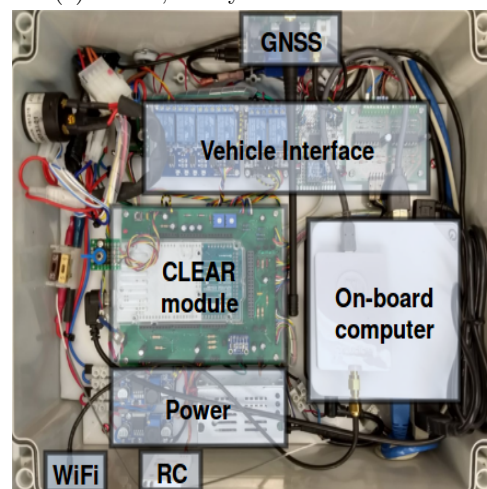
Se ha realizado la experimentación del módulo integrándolo en una plataforma industrial, la máquina eléctrica Zallys Jespi Z105. A la cual se le realizaron una serie de modificaciones para poder automatizarla. Las modificaciones básicas para poder interactuar con el módulo CLEAR han sido: en primer lugar, se le añadió un motor para el timón de dirección y un codificador incremental de alta resolución para medir el ángulo de giro en la dirección, además de dos finales de carrera para las posiciones de giro máximo en ambos sentidos. Como no se tenía acceso al eje del motor de impulsión, se creó un codificador de baja resolución usando un sensor de efecto Hall y piezas impresas en 3D para poder medir la velocidad del vehículo. Con estas pequeñas modificaciones la plataforma estaba lista para ser utilizado con CLEAR (ver Fig. 5). Al integrarse con este módulo la plataforma Jespi y los demás sensores se convirtió en BLUE (roBot for Localization in Unstructured Environments)[3].

Para adaptar CLEAR a BLUE, usamos los ficheros de configuración adaptando los rangos de voltajes de los motores, las ganancias de los controladores PID y del filtro de Kalman, las velocidades máximas angulares de los motores, etcétera. Usando las entradas y salidas digitales y analógicas de

CLEAR controlamos los actuadores y leemos los pulsos de los codificadores. Las interfaces con el vehículo están predefinidas en CLEAR, pero se pueden cambiar y añadir nuevos módulos, como pueden ser relés para dar corriente a los motores. En este caso, se han usado tres señales para la velocidad: dos digitales que definen el sentido y una analógica que define la velocidad. Para la dirección se han usado dos señales digitales y un PWM para controlarlo. Además, de otra señal digital para el claxon de la plataforma.



(a) BLUE, incluyendo sensores extra.



(b) Unidad de Abordo con el módulo CLEAR.

Figura 5: Representación del robot BLUE al integrar el módulo CLEAR.

Se ha incluido un convertidor DC-DC de voltaje para que alimente a CLEAR sin necesidad de fuentes externas, usando las baterías con 24V de la propia plataforma. También se ha añadido otro convertidor para alimentar algunos sensores, como los finales de carrera de la dirección o los codificadores, ya que estos utilizan una tensión de 12V.

Usando el ordenador de a bordo, se ha conectado el nodo de CLEAR a través de interfaz serie al

ordenador donde corre ROS y con la antena Wi-Fi podemos monitorizar y controlar el robot desde un ordenador externo.

El resultado ha sido que en pocos meses se ha conseguido transformar una plataforma industrial en un robot totalmente operativo que permite ser controlado por radiocontrol o tele-operado desde un PC externo de forma suave y precisa.

7. CONCLUSIONES Y TRABAJOS FUTUROS

Las conclusiones de este trabajo son que se ha conseguido desarrollar de forma efectiva el sistema que se pretendía y ha servido para robotizar una plataforma industrial desde cero. Además, se ha comprobado de forma experimental que el robot es controlable y está totalmente integrado en ROS, lo cual ha permitido operar con él y comenzar a recabar datos en el proceso de navegación. Por tanto, ya se pueden realizar investigaciones de algoritmos de alto nivel con la plataforma móvil.

Como trabajos futuros, queda probar el módulo CLEAR en diversas máquinas Ackermann y hacer un estudio cuantitativo del tiempo invertido en adaptar CLEAR a dichas máquinas hasta conseguir robotizarlas y que funcionen correctamente. Así, se verifica que efectivamente reduce el tiempo de ingeniería al transformar una máquina en un robot móvil totalmente operativo.

Agradecimientos

Este trabajo ha sido apoyado por el Gobierno de España a través de la beca FPU15/04446 y del proyecto de investigación DPI2015-68087-R.

English summary

CLEAR. A Module for Ackermann Machines Robotization

Abstract

This article describes the work done to create a module that facilitates the generic Ackermann machines robotization. This allows control and monitoring of the robot thanks to a low-level hardware interface with the platform's actuators and sensors, and a high-level software interface fully integrated in ROS. Finally, the module has been tested by integrating it into

a real machine and obtaining the expected results. This module called CLEAR (Control Logic for Easy Ackermann Robotization) has been developed to be generic and open source and open hardware. It is therefore expect to help research in ground mobile robotics for anyone who wants to transform an Ackermann mobile platform into a robot.

Keywords: Generic, Robotization, Automation, Ackermann Vehicles, Open Source, Open Hardware, Mobile Robotics, ROS, Low-cost.

Referencias

- [1] AUROVA, (2018) Repositorio de CLEAR. <https://www.github.com/AUROVA-LAB/CLEAR>. Consultado el 20 de Junio de 2018.
- [2] Böttcher, s., (2006) "Principles of robot locomotion", *Proceedings of human robot interaction seminar*.
- [3] del Pino, I., et al., (2018) "Presenting BLUE: a roBot for Localization in Unstructured Environments", *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*.
- [4] del Pino, I., et al., (2018) "Speed Estimation for Control of an Unmanned Ground Vehicle Using Extremely Low Resolution Sensors", *To appear in 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*.
- [5] Cebolla, R., De Leon, J., Barrientos, A., (2017) "Estructura de control en ROS y modos de marcha basados en máquinas de estados de un robot hexápodo", *Jornadas de Automática*.
- [6] Fagnant, D. J., Kockelman K., (2015) "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations", *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167-181.
- [7] Mason, R., et al., (2006) "The golem group/university of california at los angeles autonomous ground vehicle in the darpa grand challenge", *Journal of field Robotics*, vol. 23, no. 8, pp. 527-553.

- [8] Robotnik, (2016) Robotnik car. <http://wiki.ros.org/Robots/RBCAR>. Consultado el 15 de Junio de 2018.
- [9] Robotnik, (2016) Robotnik sherpa. <https://www.robotnik.es/robots-moviles/rb-sherpa>. Consultado el 15 de Junio de 2018.
- [10] Thrun, S., et al., (2006) “Stanley: The robot that won the darpa grand challenge,” *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692.
- [11] Urmson, C., et al., (2008) “Autonomous driving in urban environments: Boss and the urban challenge”, *Journal of field Robotics*, vol. 25, no. 8, pp. 425–466.
- [12] Yukumoto, O., Matsuo, Y., Noguchi, N., (2000) “Robotization of agricultural vehicles (Part 2). Description of the tilling robot”, *Japan Agricultural Research Quarterly*, vol. 34, no. 2, pp. 107-114.



© 2018 by the authors.
Submitted for possible
open access publication
under the terms and conditions of the Creative Commons Attribution CC-BY-NC 3.0 license (<http://creativecommons.org/licenses/by-nc/3.0/>).