

COMUNICACIÓN ENTRE VEHÍCULOS AUTÓNOMOS EN TIEMPO REAL, PARA MANIOBRAS DE ALTO RIESGO.

Joshué Pérez

Instituto de Automática Industrial – CSIC

Ctra. Campo Real km 0.200, 28500 La Poveda – Arganda del Rey (Madrid)

jperez@iai.csic.es

Enrique Onieva, Teresa de Pedro, Ricardo García, Javier Alonso, Vicente Milanes, Carlos González

Instituto de Automática Industrial – CSIC

Ctra. Campo Real km 0.200, 28500 La Poveda – Arganda del Rey (Madrid)

{onieva, tere, ricardo, jalonso, vmilanes, gonzalez}@iai.csic.es

Resumen

En la investigación en los Sistemas Inteligentes de Transporte (ITS) se dedica mucho esfuerzo para establecer una comunicación segura entre vehículos y con la infraestructura en tiempo real. En el caso de vehículos autónomos que maniobran conjuntamente en situaciones de alto riesgo, esta comunicación es indispensable para conocer la información relevante con la que tomar decisiones y realizar las acciones adecuadas en cada momento. En este trabajo se estudian dos sistemas de comunicaciones para la realización segura de maniobras críticas.

Palabras Clave: Sistemas de transporte Inteligentes, control borroso, conducción autónoma, tiempo real.

1 INTRODUCCIÓN

El mundo de la conducción autónoma ha venido utilizando controladores borrosos para emular el comportamiento humano. Este es el caso del programa AUTOPIA del Instituto de Automática Industrial (CSIC) en el que se ha automatizado tres vehículos comerciales con los que han llevado a cabo muchos experimentos de maniobras individuales o colectivas en circuitos urbanos.

El sistema de control del vehículo necesita conocer su posición y velocidad en todo momento. La posición se obtiene de un GPS diferencial y la velocidad directamente del tacómetro. Estos datos son utilizados para realimentar el control sobre los accionadores del vehículo: volante, acelerador y freno [1]. Además para poder realizar maniobras conjuntas entre los vehículos es muy importante garantizar la fiabilidad y la llegada de todos los datos de los vehículos involucrados [2]. Este artículo presenta cómo se puede garantizar que llegue a cada vehículo una respuesta en tiempo real que satisfaga los límites de tiempo establecidos, tomando como premisa mayor la seguridad en la conducción.

Una de las maniobras más peligrosas es el adelantamiento en carreteras de doble sentido [3]; muchas veces los conductores no pueden percibir con total claridad si viene un vehículo en sentido contrario, sobretodo si están próximos a una curva o hay problemas de visibilidad. En cualquier caso las cifras son alarmantes, mientras la siniestralidad ha descendido un 27% en las redes principales, ha crecido un 100% en las secundarias. Por otra parte en 2006 hubo un total de 99.797 accidentes con víctimas en España.

Lo anterior motiva que muchas empresas y centros de investigación vuelquen sus esfuerzos para desarrollar sistemas que permitan mejorar el tráfico en zonas urbanas, así como en auto vías [4] y [5]. Muchos estudios de estas maniobras han quedado plasmados en simulaciones [6], hasta ahora no han sido probados en vehículos reales. Los trabajos de adelantamiento desarrollados en el IAI [7] y [8] son de los pocos que han utilizado vehículos autónomos reales, que hasta ahora han involucrado solo dos vehículos. También se han simulado adelantamientos entre tres vehículos con tráfico frontal.

2 ENTORNO DE TRABAJO

Los controladores de vehículos autónomos pueden clasificarse como sistemas en tiempo real estrictos, ya que deben satisfacer las limitaciones temporales severas en la mayoría de las ocasiones. Además, casi siempre los coches están obligados a interactuar con otros vehículos, personas y problemas imprevistos en las carreteras. Por eso cuanto más información se tenga del entorno de trabajo más segura será la conducción realizada.

2.1 VEHÍCULOS Y PROTOCOLO DE COMUNICACIONES

Como se mencionó anteriormente, el grupo AUTOPIA cuenta con tres vehículos automatizados:

dos furgonetas Citroen Berlingo eléctricas y un Citroen C3 impulsado a gasolina. En la Figura 1 se muestra la arquitectura general de las furgonetas, donde se puede observar el GPS diferencial como principal sensor de percepción del entorno.

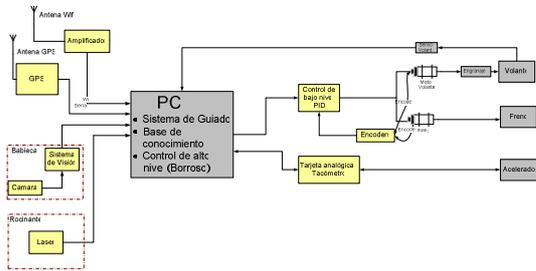


Figura 1. Arquitectura de los coches.

La arquitectura general del C3 se diferencia respecto a la de furgonetas en que el C3 utiliza el bus-CAN para leer la velocidad real del vehículo en lugar de un tacómetro como se mostró en la Figura 1. Además, cuenta con una Unidad de Medida Inercial (IMU), pero para los propósitos de este artículo estos detalles no tienen influencia [9].

En el PC central se encuentra el sistema de guiado: Un controlador borroso da valores de salida normalizadas que luego son interpretados por los controladores de bajo nivel que actúan sobre los motores añadidos para mover el volante y los pedales del coche. La cámara y sensor láser no serán considerados en este trabajo, ya que básicamente se usan para detectar obstáculos.

Para comunicarse, los vehículos utilizan una antena Wifi conectada a una tarjeta de red de área local inalámbrica, específicamente la Orinoco 2b/g PC Card y la Orinoco sylvet 11b/s PC Card, las cuales funcionan con la norma IEEE 802.11b. En líneas generales los protocolos 802.x se aplican a las redes de área local. La 802.11b define el uso de los dos niveles inferiores de una arquitectura tipo OSI, es decir, la capa física y la de enlace de datos. Las principales características de esta norma son: Operan en la banda de 2,4 GHz, utilizan el protocolo de acceso CSMA/CA (el cual permite que múltiples estaciones tengan acceso a un mismo punto evitando las posibles colisiones) y su velocidad teórica es de 11Mbps (pero dado el espacio ocupado por la codificación del protocolo CSMA/CA, en la práctica, la velocidad máxima de transmisión con este estándar es aproximadamente 5.9 Mbit/s sobre TCP y 7.1 Mbit/s sobre UDP).

Luego de hacer varias pruebas utilizando los protocolos TCP y UDP con los vehículos detenidos y en marcha, se observó que no había pérdida de la información transmitida, pero para el caso del TCP los paquetes enviados reciben confirmación, además

muestran el estado de conexión. Para el caso de UDP la información está siempre enviándose, esto hace que los datos se repitan, por otra parte, no hay manera de saber cuando se pierde la conexión. Estas razones determinan la escogencia del protocolo TCP, el cual nos brinda mayor seguridad en el tratamiento de la información.

2.2 DGPS

Los dispositivos de posicionamiento global diferencial por satélite DGPS se utilizan para superar los motivos que impiden medir con una precisión centimétrica, tales como: desviación de la órbita de los satélites, transmisión de las señales a través de la ionosfera y troposfera, ruido de los receptores, entre otros. Su funcionamiento se basa en comparar las señales recibidas por un GPS con las recibidas por una estación base (cuya posición es conocida) separadas pocos centenares de metros o dentro de un rango de una red aérea que pueda los pueda cubrir. Esta estación transmite las correcciones que deben aplicarse a las posiciones medidas por el GPS.

En nuestro caso la estación base se encuentra a unos 300 metros del punto más alejado de las pistas de prueba. Una red Wifi transmite a la corrección diferencial a los coches, así tenemos una precisión de hasta 2 cm, y podemos controlar los vehículos aunque se muevan a velocidades de hasta de 60km/h.

Cada coche lleva un receptor GPS que transmite su posición en tiempo real. Las frecuencias de transmisión son distintas para cada dispositivo de cada vehículo, como muestra la Tabla 1.

Nombre del Vehículo	Marca del DGPS	Velocidad de envío de datos	Frecuencia
Clavileño	Trimble MS750	19200 bps	10 Hz
Rocinante	ASTECC Z-12	9600 bps	5 Hz
Babieca	DGPS Max CSI	19200 bps	5 Hz

Tabla 1

2.3 SISTEMA OPERATIVO Y ENTORNO DE PROGRAMACIÓN.

Actualmente hay muchos sistemas operativos específicos para tiempo real (QNX, Linux, VTRX, por mencionar algunos) pero muchos no tienen facilidad para incorporar los *drivers* que necesitan los componentes de nuestro sistema, sobre todo en lo que a recepción y actuación se refiere (tarjetas de red, puertos serie, tarjetas de control de motores, lecturas de GPS, entre otros).

Sabemos que es controvertida la capacidad del sistema Windows para trabajar en tiempo real, ya

que, entre otras razones, no garantiza un reparto equitativo de los recursos del sistema (lo que podría dar problemas en un sistema crítico como el guiado automático) y puede bloquearse con facilidad. Sin embargo, como veremos más adelante esto no es del todo cierto, ya que muchas veces los problemas dependen del grado de seguridad con que se realicen las operaciones. Lo más importante no es ejecutar las acciones sin demora, sino dentro de un tiempo finito aceptable, que garantice la respuesta del sistema en tiempo real.

Nosotros elegimos Windows porque es totalmente compatible con el MS-DOS, en el que se desarrolló la comunicación con los controladores de bajo nivel, así como en las primeras pruebas del ORBEX (Ordenador Borroso Experimental) [10], en el que se basa el sistema de control principal.

El sistema de control y la comunicación entre los coches se programó C++ Builder, que al ser orientado a objetos tiene las ventajas de ser modular, reutilizable y hereditario y permite realizar distintas tareas en paralelo sin saturar al programa principal.

3 CICLO DE CONTROL.

Como es normal en la mayoría de los sistemas en tiempo real existe un ciclo de control principal que realiza tres tareas secuenciales: Percibir, Planificar y Actuar (ciclo PPA). El tiempo de ejecución de este hilo principal depende de la frecuencia a la cual el DGPS envía los paquetes NMEA (87 bytes) a través del puerto serie asignado. En nuestro caso puede ser 5 o 10 Hz (Tabla 1).

Esto permite que por cada posición enviada por el DGPS se ejecute un ciclo de control. Es importante mencionar que la lectura del DGPS no se realiza en el bucle de control, sino en otro hilo, permitiendo mantener la total independencia del ciclo PPA, al que se asigna la máxima prioridad permitida por Windows (prioridad crítica para tiempo real). Esto garantiza que el bucle de control se ejecuta siempre, aunque exista cualquier tarea pendiente en el sistema. De hecho, todas las tareas importantes del sistema se ejecutan en hilos independientes, sin ocupar tiempo de procesamiento del hilo principal, el cual se limita a hacer las distintas llamadas. A continuación se citan los hilos de trabajo asociados a la comunicación de datos:

- **Principal:** se ejecuta al arrancar el sistema. Antes de entrar en el bucle de control inicia y verifica los periféricos de entrada y de salida del sistema. Siempre debe estar liberado al máximo. Cualquier tarea nueva se debe realizar en un hilo multitarea.
- **Lectura GPS:** lee las posiciones del GPS.

- **Interfaz Grafico:** Muestra las variables de interés en la interfaz grafica.
- **Mensajes de Posición:** Lee todos datos de los coches a través de red (utilizando *socket*).
- **Control Coche:** Manda los valores a los controladores borrosos y a los distintos hilos de acción.

Cada uno de estas tareas se ejecuta de forma independiente. La Figura 2 muestra los tiempos que tarda cada etapa del bucle de control: Percepción, Planificación (Control) y Acción, para un recorrido completo del circuito, realizado en 820 ciclos. Estos resultados se obtuvieron considerando el reloj interno del ordenador (precisión de 1 μ s).

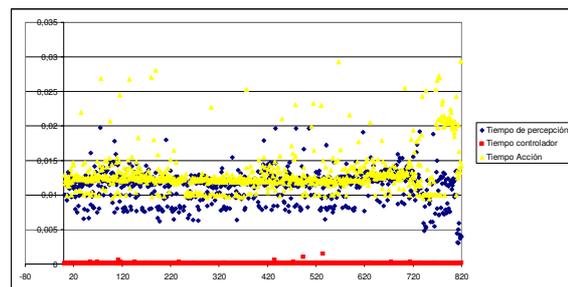


Figura 2. Tiempo de percepción (Azul), Tiempo de control (Rojo) y Tiempo de Acción (Amarillo).

La distribución de los datos no es uniforme, esto hace que se descarte la media. Ya que se deben considerar los valores de mayor repetición, tomamos como valor de referencia la mediana. La Figura 3 muestra el tiempo total del bucle de control PPA, y el valor de la mediana, 23 μ s.

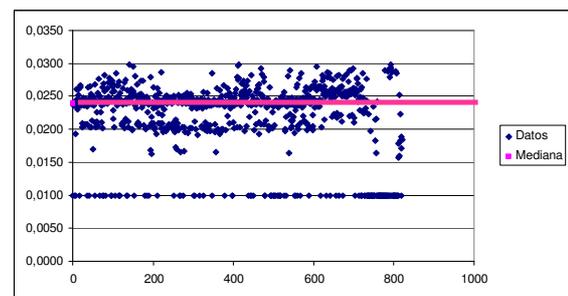


Figura 3. Bucle de control PPA.

Lo ideal es que el bucle de control se ejecute lo más rápido posible, pero garantizando que cada ciclo de control haya terminado antes de que llegue la próxima trama del DGPS.

Es importante señalar que el controlador borroso es mucho más rápido que las etapas de percepción y actuación. Esto es debido a que el ORBEX genera controladores difusos del tipo TSK [12], con consecuentes *singleton*, lo que permite, a su vez, tomar decisiones de control en un corto periodo de tiempo y con muy buena precisión [13].

4 ADELANTAMIENTOS ENTRE TRES VEHÍCULOS.

Utilizaremos los tres vehículos comerciales de los que ya hemos hablado. La Figura 4 muestra el esquema de comunicación entre ellos y con la estación base para la maniobra de adelantamiento. El vehículo que va a adelantar es el rojo, por tanto, desde el punto de vista de las comunicaciones, se va a comportar como cliente de los otros dos, ya que es el que va a recibir los paquetes de datos. Las condiciones del adelantamiento son las siguientes: El vehículo rojo se conduce automáticamente y los otros dos manualmente, pero siempre enviando la trama de datos de su posición. La comunicación es unidireccional, es decir, los datos fluirán hacia el vehículo que va a adelantar. (Sólo fluirá información de ambos sentidos cuando se realice la confirmación en la comunicación TCP).

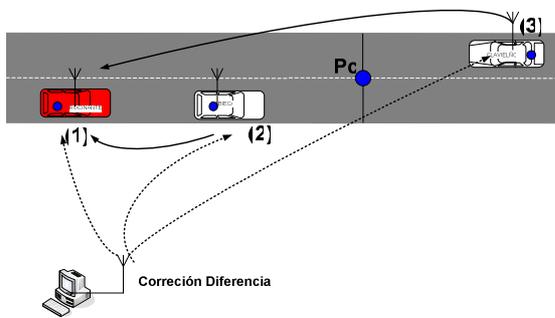


Figura 4. Comunicación entre vehículos y corrección diferencial.

Los datos necesarios que circulan por la red son: un identificador del vehículo, las coordenadas norte y este y la velocidad del vehículo.

En la Figura 5 se observa que la velocidad calculada a partir de la posición del DGPS fluctúa mucho más que la medida por el tacómetro de los coches. Por ello usamos esta velocidad una vez filtrada, para evitar cualquier sobre pico de la misma o error en la lectura.

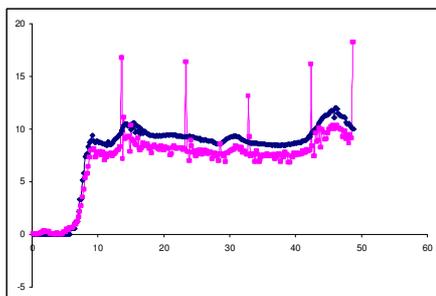


Figura 5. Velocidades de vehículo a adelantar (2).

Con estos datos formamos la trama completa y podemos hacer todas las estimaciones, necesarias para la maniobra de adelantamiento

;Nombre, Norte, Este, Velocidad_real;

En total se envía un máximo de 48 bytes (depende del nombre del coche y del valor instantáneo de la velocidad). Si consideramos el GPS más rápido (clavileño @ 10Hz) mandaríamos la trama:

$$T_{bps} = 48 \times 8 \times 10 = 3840 \text{ bps (bits por segundo)}$$

Para un coche pero la maniobra exige que dos vehículos manden información al adelantador y el segundo la envía a una frecuencia máxima de 5 Hz, con lo que tendremos 5670 bps, utilizando las tarjetas de radio Ethernet en la capa física que transmiten a 11 Mbps (para el caso práctico 7,1Mbps UDP y 5,9Mbps) y garantizando que ninguno de los datos enviados se va a perder.

Por otra parte hay que asegurar que el vehículo adelantador realiza un ciclo de control entre cada medida del GPS. Si su GPS transmite a 5 Hz, el puerto serie esta configurado a 9600 bps y cada trama NMEA (87 bytes) tarda en llegar al PC:

$$T_{envio} = \frac{87 \times 8}{9600} = 7.25 \text{ milisegundos.}$$

El tiempo entre dos tramas sería:

$$T_{entramas} = 0.2 - 0.00725 = 127.5 \text{ milisegundos.}$$

Por lo tanto se cumple holgadamente la condición de que el bucle de control utilice todas las posiciones que recibe del GPS, ya que la mediana del PPA, antes calculada, era de 23 ms para todo el bucle control, y solo 11,4 ms para la etapa de percepción.

El tiempo exacto que tardan en llegar los datos de los otros coches dependerá, entre otras cosas, de la distancia a la que estén y de la calidad de la conexión. Estos datos serán enviados a la misma frecuencia en la transmiten sus respectivos GPS. Tal como se indicó en la tabla 1, la velocidad del puerto serie es más rápida, de 19200 bps, y en promedio el ciclo de control en estos sistemas es más rápido (por seguridad tomamos el mismo valor de bucle antes descrito), ya que su única función es recibir datos del GPS y enviarlos en el formato descrito anteriormente a través de red inalámbrica. Por lo tanto, calculamos el tiempo aproximado en que llegarán:

$$T_{GPS} = \frac{87 \times 8 \text{ bits}}{19200} = 36.25 \text{ miliseg} + 23 \text{ miliseg.}$$

$$+ T_{trans} = \frac{48 \times 8}{2000000} = 0.2 \text{ miliseg}$$

Considerando un factor de seguridad de 10 ms, tenemos que los datos llegarán en menos 70 ms. Y dado que los hilos de recepción de GPS y recepción a través de *sockets* se ejecutan de forma independiente, podemos garantizar que el ordenador central dispone de todos los datos en tiempo real, para ejecutar o abortar el adelantamiento, pues el tiempo entre cada trama del GPS (espacio del ciclo PPA) es de 127.5 milisegundos.

Una vez garantizados todos los datos en el ciclo PPA, se puede determinar que para un bucle a 5 Hz, y considerando que nuestros vehículos circulan a velocidades máximas de 60 km/h (en el circuito) o a 16,6 m/seg, se puede asegurar que se recorren 3.32 m como máximo en cada ciclo de control. Estos resultados pueden considerarse muy aceptables, ya que son mucho más rápidos que lo una persona tarda en reaccionar ante situaciones de alto riesgo como la planteada.

Para determinar si es posible adelantar hay que comparar el tiempo que tardarían los vehículos 1 y 3 en alcanzar la línea de corte (Figura 4). La línea de corte se sitúa a una distancia de seguridad por delante del vehículo 2. Las velocidades relativas $V_{r1} = v_1 - v_2$ y $V_{r2} = v_1 - v_3$ permiten calcular si el vehículo 1 puede llegar a la línea de corte antes que el vehículo 3, en cuyo caso se puede adelantar, o no, en cuyo caso habría que abortar el adelantamiento.

Tomando valores de velocidad entre 2 km/h y 50 km/h para el vehículo 2 (la velocidad máxima que permite el circuito de pruebas) se determinó empíricamente la expresión de la distancia de corte en función de la velocidad v_{r1} :

$$Dists = 0,0018 * v_{r1}^2 + 0,0862 * v_{r1} + 20.943$$

Expresión a partir de la que se puede conocer los tiempos de corte Tc_1 y Tc_3

$$Tc_1 = \frac{dist_1}{V_{rel}_1}$$

Donde $dist_1$ es la suma de la distancia de separación entre los vehículos 1 y 2 más $dists$. La maniobra de adelantamiento no será abortada mientras el tiempo Tc_3 (tiempo del coche 3 a la línea de corte) sea mayor que Tc_1 , garantizando de esta manera una maniobra segura.

5 RESULTADOS EXPERIMENTALES

La (Figura 6) presenta las trayectorias de los vehículos 1 y 2 en el adelantamiento, el vehículo 3 está detenido, pero enviando datos. El eje vertical refleja la distancia cubierta a lo ancho del carril y el horizontal la distancia recorrida a lo largo del carril, de aproximadamente unos 120 metros.

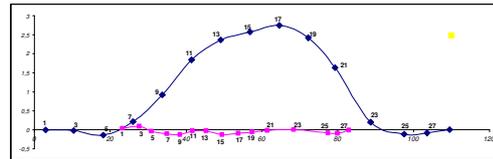


Figura 6. Datos recibidos por el vehículo adelantador.

La Figura 7 presenta las velocidades de cada vehículo en función del tiempo y con una frecuencia de transmisión del GPS de 5Hz:

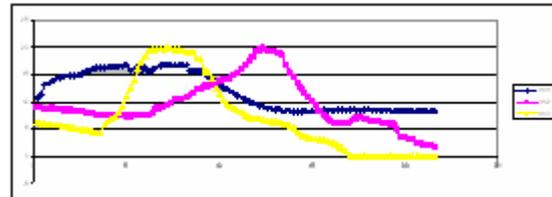


Figura 7. Velocidad de los tres vehículos.

Los valores de velocidad de la Figura 7 no corresponden a los utilizados en la Figura 6. Este experimento se realizó con la finalidad de comprobar si los valores de posición y velocidad llegaban sin problemas y en orden al vehículo adelantador.

En la siguiente grafica se muestran los tiempos para determinar si abortar el adelantamiento:

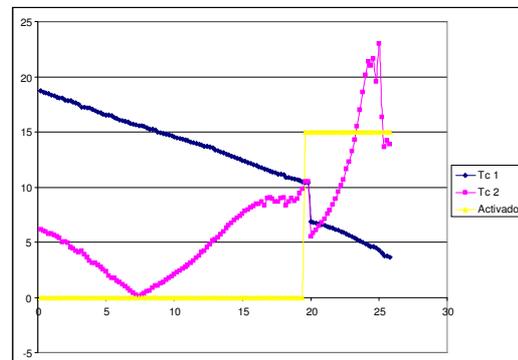


Figura 8. Tiempo de abortar adelantamiento.

Una vez que se active la condición de abortar, el vehículo solo volverá a adelantar cuando haya vuelto al carril derecho. En la Figura 8 también se observa un salto brusco en ambos (grafica azul y

fucsia) tiempos, esto se debe a que, como solo se estaba recibiendo datos y verificando las condiciones, el vehículo 2 fue girado repentinamente, lo que afecta a ambos tiempos pero en igual orden.

6 CONCLUSIONES Y TRABAJOS FUTUROS

Un buen sistema de comunicaciones permite conocer todos los datos necesarios para completar una maniobra de adelantamiento.

La velocidad de transmisión de 2 Mbps de las tarjetas de red aérea que hemos utilizado es más que suficiente para la maniobra de adelantamiento, tenemos tiempo de sobra en la ejecución.

La comunicación punto a punto entre vehículos proporciona buenos resultados, y los datos enviados llegan dentro de los márgenes establecidos. La comunicación entre *socket* TCP garantiza una comunicación segura a nivel de la capa de transporte de datos, estando los únicos inconvenientes posibles en la capa de física. (Éstos pueden ser solventados cuando esté en el mercado otro tipo de plataforma de envío de datos, por ejemplo WiMax, de mayor ancho de banda y mayor cobertura).

Para mejorar la comunicación se puede proponer un control de datos de todos los vehículos que haya en el circuito desde un ordenador principal, que funcione como servidor y envíe a cada vehículo la posición de los demás. Luego, cada coche puede desechar o utilizar lo que necesite en ese momento.

En otros proyectos nacionales y europeos en los que participa el grupo AUTOPIA, se están desarrollando protocolos específicos de comunicación entre vehículos e infraestructuras.

Agradecimientos

Este trabajo ha sido realizado gracias al apoyo de los proyectos CyberCars-2 (UE STREP FP6-028062), SARA (DPI2005 07980 C03-01) ENVITE (Ministerio de Fomento, T7 / 06) y MARTA (CDTI, CENIT 2007).

Referencias

- [1] Naranjo, J.E. et. Al (2007). "Using Fuzzy Logic in Automated Vehicle Control", Intelligent transportation Systems.
- [2] Pérez J. (2008) "Comunicación entre vehículos autónomos en Tiempo Real", trabajo de fin de curso de Programación de sistemas de control en tiempo real, Doctorado en Ingeniería de Sistemas y Automática, UNED-UCM, Madrid, España.

- [3] DGT (Dirección General de Tránsito) Balance anual, Datos de seguridad vial, 2006.
- [4] Proyecto MARTA: Movilidad y Automoción con Redes de Transporte Avanzadas.
- [5] Proyecto Cybercars, 2008 "Real-Time Data Structures and Procedures".
- [6] Ngai, D. et Al (2007) "Automated Vehicle Overtaking based on a Multiple-Goal Reinforcement Learning Framework", Intelligent transportation Systems, Seattle USA.
- [7] Naranjos, J.E. et. Al (2003) "Overtaking Maneuver Experiments with Autonomous Vehicles", International Conference on Advanced Robotics Coimbra, Portugal.
- [8] Alonso J. et Al. (2007) "Cooperative Maneuvers Study Between Autonomous Cars: Overtaking". Computer Aided Systems Theory – EUROCAST, Canarias, España.
- [9] Milanes V. (2008, Revisión) "Autonomous Vehicle based in cooperative GPS and Inertial System", Robotica.
- [10] R. Garcia, T. De Pedro. (1999) "First Application of the ORBEX Coprocessor: Control of Unmanned Vehicles". EUSFLAT-ESTYLF Joint Conference. Mathware and Soft Computing, n. 7, vo12-3, 2000, pp. 265-273.
- [11] Bob Quinn & Dave Chute (1996). "Windows Sockets Network Programming". Charter 5, pages 76-117. May.
- [12] Fuzzy Identification of Systems and its Application to Modelling and Control. IEEE Trans. on Syst. Man and Cybernetics, 15:166-132.
- [13] M. Sugeno, (1999) "On stability of fuzzy systems expressed by fuzzy rules with singleton consequents," IEEE Transactions Fuzzy Systems., vol. 7, pp. 201-224.