

Ajuste Genético de Controladores Difusos para Conducción Autónoma

Enrique Onieva Vicente Milanés Javier Alonso Joshué Pérez Teresa de Pedro

Instituto de Automática Industrial. Consejo Superior de Investigaciones Científicas.
Ctra. Campo Real Km.0,200. 28500 La Poveda, Arganda del Rey. Madrid, Espaa.
{onieva, vmilanes, jalonso, jperez, tere}@iai.csic.es

Resumen

Existe una amplia gama de tecnologías creadas con el fin de dar solución a los problemas de transporte, llamados Sistemas Inteligentes de Transporte; el control automático del volante es uno de los retos más importantes al que se enfrentan investigadores de esta área, pero no es el único. Este trabajo presenta un método para el ajuste automático de controladores difusos capaces de manejar automáticamente el volante de un vehículo convencional, reproduciendo actitudes de un conductor humano en diferentes situaciones. Para hacerlo, se ha recopilado información sobre el estado del coche mientras es conducido por un piloto, gracias a la cual, y por medio de algoritmos genéticos se han obtenido controladores difusos apropiados para el manejo del volante del vehículo. Los dos principales requisitos que debe llevar a cabo el controlador son: ser capaces de manejar el vehículo a través de una ruta prefijada y ejecutar acciones de control suaves, para lograr así, una conducción confortable.

1 INTRODUCCIÓN

Los Sistemas Inteligentes de Transporte (ITS, Intelligent Transportation Systems) usan tecnologías de la información y la comunicación con el fin de conseguir una conducción segura y eficiente. En la industria del automóvil, los sensores se usan principalmente para dar información al conductor y, en algunos casos, están conectados a un ordenador, que ejecutará determinadas acciones de control con el fin de minimizar daos y prevenir colisiones [1].

Actualmente, la implementación del control total del volante de un vehículo es una de las disciplinas de la investigación industrial en vehículos inteligentes que está recibiendo menor atención ahora que en años anteriores. La razón quizás se encuentre en que queda un muy largo camino hasta que esto llegue al mercado [5], y los fabricantes de vehículos se centran en etapas más maduras, especialmente en la automatización del control de velocidad; algunos ejemplos pueden encontrarse hoy en día en el mercado. Pero no recibe tan poca atención en el área de la investigación académica, tal y como se demuestra en el DARPA Grand Challenge [6], que consiste en una competición de vehículos autónomos financiada por la Agencia de Proyectos de Investigación Avanzada en Defensa (DARPA), la organización central de investigación del Departamento de Defensa de los Estados Unidos. DARPA ha financiado tres competiciones en el área de vehículos autónomos hasta la fecha.

El trabajo llevado a cabo por el Programa AUTOPIA, del Instituto de Automática Industrial, del Consejo Superior de Investigaciones Científicas está centrado en el área de vehículos autónomos. Concretamente, el objetivo del trabajo aquí presentado se centrará en el control automático del volante, también conocido como control lateral, de un vehículo convencional; otras áreas de investigación abordadas por el Programa AUTOPIA son el control longitudinal, el control de la velocidad, y su adaptación a las condiciones de la carretera, mediante la automatización de los pedales de freno y acelerador [3], [4].

El trabajo pretende modelar el comportamiento de un conductor humano mediante el ajuste de un controlador borroso [15] mediante un algoritmo genético [16]. En la sección 2 se presenta, de una manera muy breve, el Programa AUTOPIA, con el objetivo de dar una idea sobre la instrumentación e infraestructura usada para llevar a cabo este trabajo; la sección 3 se presenta el método usado para generar automáticamente controladores difusos, tomando como entrada, una se-

rie de puntos Entrada/Salida y aplicando un método iterativo, consistente en la optimización de la base de datos y de reglas que definen un controlador por separado por medio de algoritmos genéticos. Finalmente, en la sección 4, el método es aplicado para modelar el comportamiento de un conductor en el uso del volante de un vehículo. Una vez generados una serie de controladores difusos, éstos serán probados en una pista de pruebas, con lo que podremos verificar su correcto funcionamiento.

2 EL PROGRAMA AUTOPIA

AUTOPIA pretende transferir las técnicas desarrolladas para el control de robots autónomos al control de vehículos, modificando en la menor medida posible el entorno en que éstos han de evolucionar. Se han realizado inicialmente experimentos simples que muestren los resultados que van obteniéndose y creado la infraestructura que permita llegar a hacer cada vez unas maniobras más sofisticadas.

El Coche y ZOCO: El coche usado en este trabajo es un Citron C3 Pluriel. ZOCO (ZOna de CONducción) es una pista de pruebas de vehículos automáticos que está dedicada exclusivamente a tareas de investigación, es decir, en ella no hay ningún otro tráfico de vehículos, lo que se ha hecho por seguridad. Tiene una forma reticulada, como las manzanas o cuadras de una ciudad, con algunas irregularidades. También posee ZOCO una estación base de posicionamiento global diferencial basado en información geográfica de información vía satélite, DGPS, que puede ser utilizado por sistemas móviles embarcados en los coches, para obtener su posición con una precisión cercana al centímetro. En la figura 1 podemos ver una fotografía del coche utilizado, así como la vista aérea de ZOCO puede verse en la figura 2.



Figura 1: Vehículo Utilizado

Arquitectura de Control: La arquitectura general usa un esquema donde un conjunto de coches independientes se comunican entre sí mediante una estación central, gracias a lo que comparten información con



Figura 2: Vista aérea de ZOCO

el fin de cooperar para llevar a cabo maniobras tal y como son realizadas por humanos. La arquitectura individual instalada en cada vehículo es una arquitectura jerárquica en tres capas: sensorial, de control y de actuación. La capa de control está dividida en tres etapas secuenciales: planificador, copiloto y piloto. En el presente trabajo, nos centraremos en el piloto, ya que es donde se encuentran y ejecutan los controladores difusos capaces de llevar a cabo maniobras simples. El piloto recibe una serie de parámetros de entrada y una maniobra de bajo nivel a realizar (seguir al coche de delante, tomar una curva, parar...) provenientes del copiloto y es capaz de generar una salida aplicable sobre los actuadores del vehículo (pedales y volante).

ORBEX: ORBEX [7], [8] (ORdenador Borroso EXPERimental) se usa para la construcción y ejecución de los controladores difusos usados por el piloto. ORBEX trabaja con controladores difusos del tipo TSK [10], con consecuentes Sigleton, lo que permite tomar decisiones de control en un corto período de tiempo y con muy buena precisión; cualidades más que deseables para sistemas de tiempo real, donde el tiempo de inferencia es un aspecto vital, tal y como puede ser la conducción de un coche [9],[11].

3 MÉTODO ITERATIVO PARA EL AJUSTE DE CONTROLADORES BORROSOS

La base de reglas y las funciones de pertenencia que definen un controlador borroso serán modificadas con el fin de adaptarse a las acciones y actitudes mostradas por un conductor humano a la hora de conducir un vehículo. El método se divide en tres fases; primero, se aplica una fase de captura y procesamiento de información; tras lo que, de una forma iterativa, ajustamos las funciones de pertenencia y la base de reglas por medio de dos algoritmos genéticos independientes. Las siguientes cuatro subsecciones describen la fase de captura y procesamiento de datos, los dos algoritmos genéticos utilizados y el esquema global seguido por el método con el fin de ajustar los controladores difusos.

3.1 CAPTURA Y PROCESADO DE LA INFORMACIÓN

Un conductor humano controlará el vehículo a través de una ruta prefijada, mientras el ordenador de a bordo tomará datos acerca del estado del vehículo y maniobras realizadas. Los datos que pueden ser recogidos por el ordenador pueden ser, por ejemplo: velocidad, posición, giro del volante, presión aplicada sobre los pedales...

Una vez hecho esto, se obtiene un conjunto de datos que será sometido a un proceso de normalización en el intervalo $[-1,1]$, tras lo que se le aplicará una rejilla de dimensiones elegidas por el usuario y, en cada punto, la salida considerada será la media entre los valores más cercanos del conjunto de datos inicial. Por otra parte, existe la posibilidad de añadir nuevos puntos no incluidos en el conjunto de datos inicial, ya que pueden representar zonas del espacio de las que no es posible tener información. Lo que permitirá al usuario introducir información proveniente de su propio conocimiento, tal como *Si el coche esta desviado lo máximo hacia la derecha, gira el volante lo máximo hacia la izquierda.*

3.2 AJUSTE DE LAS FUNCIONES DE PERTENENCIA

En la primera etapa del método, las funciones de pertenencia (FP's) que definen las variables de entrada del controlador difuso serán ajustadas. Una función de pertenencia se definirá mediante cuatro valores reales (a,b,c,d), lo que permitirá trabajar con etiquetas tanto trapezoidales como triangulares ($b=c$). Con el fin de obtener controladores difusos precisos e interpretables, las funciones de pertenencia deberán cumplir una serie de restricciones semánticas tales como: las funciones de pertenencia deberán estar ordenadas; todos los posibles valores de entrada deben estar cubiertos por una o dos funciones de pertenencia...

Así pues, para representar un conjunto de funciones de pertenencia trapezoidales de forma que pueda ser mejorado por medio de un algoritmo genético, usaremos $4N$ valores, donde N es el número total de funciones de pertenencia de las variables de entrada. Las variables de salida no serán modificadas. El algoritmo genético estacionario usado para ajustar las funciones de pertenencia usará los siguientes parámetros:

- Inicialización: cada valor v_i de un cromosoma será inicializado a un valor uniformemente distribuido en el intervalo $[m_i \pm \xi]$ con una probabilidad ρ , en otro caso, $v_i = m_i$. Donde m representará el mejor cromosoma encontrado a lo largo de la ejecución. Es decir, la población inicial en cada

iteración se creará a partir del mejor individuo encontrados hasta el momento, introduciendo sobre sus valores de las funciones de pertenencia una pequeña variación aleatoria.

- Selección: Torneo Binario.
- Operador de Cruce: BLX - α [13].
- Operador de Mutación: Modificación aleatoria bajo una probabilidad de mutación, p_m [11], [12].
- Reemplazamiento: cada nuevo cromosoma reemplazará al peor cromosoma de la población, siempre que obtenga mejor fitness que él.
- Función Objetivo: Para medir la calidad de un determinado controlador, serán tenidos en cuenta dos aspectos: el ajuste a las acciones tomadas por el piloto y tener una superficie de control suave, para no tener cambios de salida bruscos. Para evaluar el primero de ellos, se usará el error cuadrático medio (MSE) entre la salida del controlador y las salidas dadas por el piloto. Para evaluar el segundo, la superficie de control será evaluada bajo una rejilla de 21×21 , y se tomará D como la mayor diferencia obtenida entre dos puntos adyacentes de la rejilla. Finalmente, la función objetivo consistirá en una ponderación de las dos medidas comentadas (MSE y D), siendo el valor final de la siguiente forma: $F = vMSE + (1-v)D$, donde v es elegido por el usuario con el fin de dar más importancia al aspecto deseado.

3.3 AJUSTE DE LA BASE DE REGLAS

La segunda etapa del método consiste en el ajuste de la base de reglas (BR). Para hacerlo, se modificarán los consecuentes asociados a una serie de conjuntos de antecedentes dados. El proceso comienza con la selección, por parte del usuario del número de posibles singleton de Sugeno que podrán ser seleccionados como consecuentes de una determinada regla. Una vez hecho esto, se crea el conjunto de singletons uniformemente distribuido en el rango de la salida y el objetivo del algoritmo genético consistirá en asociar cada posible antecedente con un consecuente, con el fin de obtener la mejor configuración de reglas para el sistema. Independientemente del número de funciones de pertenencia usado para cada una de las variables de entrada, el programa dispone de tres conjuntos de antecedentes prefijados, llamados *Marginal*, *Central* y *Total* respectivamente. El primero de ellos está formado por reglas con antecedente simple, el segundo por reglas compuestas mediante el operador 'Y' y el tercero por la unión de los dos anteriores.

Por ejemplo, si tuviéramos dos entradas, cada una de ellas con 4 etiquetas diferentes, los casos cubiertos por

cada uno de los conjuntos de antecedentes podemos verlo en la tabla 1:

$In_1 \setminus In_2$	\emptyset	Et_{21}	Et_{22}	Et_{23}	Et_{24}
\emptyset		M	M	M	M
Et_{11}	M	C	C	C	C
Et_{12}	M	C	C	C	C
Et_{13}	M	C	C	C	C
Et_{14}	M	C	C	C	C

Tabla 1: Casos cubiertos por la base de reglas Central (C) y Marginal (M).

De esta forma, el conjunto de antecedentes Marginal almacena todos los antecedentes del tipo:

Si ($In_1 = Et_{11}$) Entonces ...
 Si ($In_1 = Et_{12}$) Entonces ...
 ...
 Si ($In_2 = Et_{23}$) Entonces ...
 Si ($In_2 = Et_{24}$) Entonces ...

Mientras que el conjunto de antecedentes Central almacena todos los del tipo:

Si ($In_1 = Et_{11}$) Y ($In_2 = Et_{21}$) Entonces ...
 Si ($In_1 = Et_{11}$) Y ($In_2 = Et_{22}$) Entonces ...
 ...
 Si ($In_1 = Et_{14}$) Y ($In_2 = Et_{23}$) Entonces ...
 Si ($In_1 = Et_{14}$) Y ($In_2 = Et_{24}$) Entonces ...

Como hemos comentado, el conjunto de antecedentes Total, representará la unión de los conjuntos Marginal y Central. Dicho esto, para representar la base de reglas de forma que pueda ser tratada por un algoritmo genético, usaremos un vector de N enteros, donde N es el número total de reglas de la base seleccionada y en cada posición i del vector se almacenará el índice del consecuente asociado a la regla i. Para el ajuste de la base de reglas, nuevamente usaremos un algoritmo genético estacionario con igual selección, reemplazamiento, y función objetivo al usado para el ajuste de las funciones de pertenencia, pero con distinto operador de cruce; en este caso, usaremos cruce simple [12]. En el caso de la inicialización, nuevamente partiremos del mejor individuo encontrado hasta el momento y dos valores (ξ, ρ); así pues, para generar una población inicial, cada valor m_i del mejor individuo será modificado, con una probabilidad ρ a un valor aleatorio en el intervalo $[m_i \pm \xi]$.

3.4 ESQUEMA DEL MÉTODO

La ejecución del método se hace en dos fases diferenciadas, que serán repetidas iterativamente un cierto número de veces y representarán, respectivamente, los

dos algoritmos genéticos comentados con anterioridad; uno de ellos estará encargado de mejorar las funciones de pertenencia, mientras que el otro, de la base de reglas. Gracias a esto conseguimos que, cada vez que obtengamos un buen conjunto de etiquetas, poder encontrar el conjunto de reglas más apropiado para él, al igual que, si encontramos un buen conjunto de reglas, la otra fase del método buscará el conjunto de etiquetas más apropiado. Así, mediante dos *pequeos* algoritmos genéticos independientes, ejecutados iterativamente, logramos la progresiva mejora conjunta de base de reglas y conjunto de etiquetas. Para evaluar la base de reglas se usará el mejor conjunto de funciones de pertenencia encontrado y para evaluar las funciones de pertenencia se usará la mejor base de reglas encontrada. Así pues siempre existirá una copia del mejor individuo encontrado en la población actual. Una vez finalizado, el proceso devolverá la mejor combinación Base de Reglas - Funciones de Pertenencia encontrada.

Inicialmente, se usará una base de reglas aleatoria, con lo que, la primera fase (mejora de etiquetas) podrá evaluarse sobre ella. Una vez aplicado el algoritmo genético encargado de mejorar las etiquetas, el mejor conjunto encontrado será usado para mejorar la base de reglas, y así sucesivamente. El esquema podemos verlo en la figura 3.

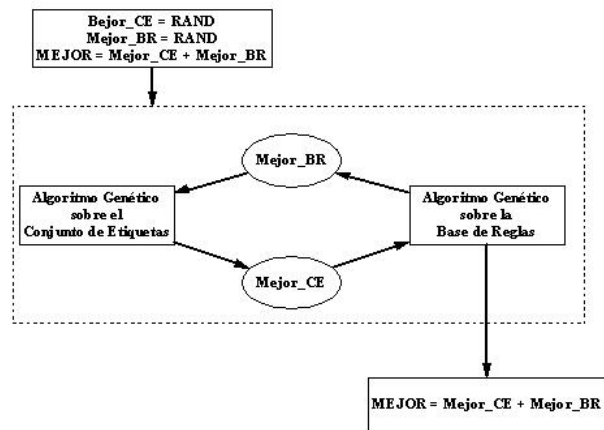


Figura 3: Esquema Usado

Antes de aplicar el método, se realizará la etapa de captura y procesamiento de información, descrita en la sección 3.1. Los algoritmos genéticos ejecutados iterativamente serán los descritos en las secciones 3.2 y 3.3.

4 EXPERIMENTACIÓN: MODELADO DEL CONTROL LATERAL DE UN VEHÍCULO

El objetivo del experimento consistirá en el modelado del control del volante de un coche (también conocido como control lateral del vehículo). El control automático tanto de la velocidad como de la dirección del vehículo son dos pasos básicos para llegar a conseguir el desarrollo de vehículos inteligentes autónomos. En este experimento, el diseño de la automatización del volante del vehículo gracias al método presentado anteriormente es abordado.

El primer punto del proceso de modelado consistirá en la toma de datos, para ello, un piloto ha seguido la ruta mostrada en la figura 7, mientras el computador embarcado en el vehículo ha ido tomando datos acerca del estado del coche y las acciones tomadas. Para el modelado del control lateral, se usarán dos variables de entrada, obtenidas mediante diferencias entre la ruta GPS de referencia y la posición del vehículo en un momento dado. Las variables de entrada definidas son: el error lateral y el error angular. El error lateral representa la distancia actual del vehículo a línea de referencia, los valores pueden tomar cualquier valor dentro del intervalo $(-\infty, \infty)$. Mientras que el error angular es el ángulo existente entre la línea de referencia y el vector de velocidad del vehículo, cuyos valores están restringidos al intervalo $(-180, 180)$. La representación gráfica de ambas variables se muestra en la figura 4:

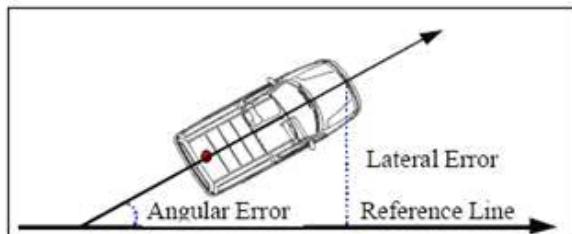


Figura 4: Representación gráfica de las variables de entrada

La salida consistirá en la posición de referencia, que será enviada a un controlador robusto de bajo nivel, capaz de manejar un motor enganchado a la barra de dirección, con el objetivo de llevar el volante a la posición deseada. A la hora de proceder a la normalización de los valores del conjunto de datos inicial al intervalo $[-1,1]$, se ha asumido que el límite para el error lateral es de ± 5 metros y para el error angular (como ya hemos comentado) de ± 180 grados. Tras esto, hemos definido una rejilla 21×21 definida en el plano X-Y, y, en cada punto de la rejilla tomamos como salida la media de las salidas más cercanas para

el conjunto de datos de entrada. Tal y como podemos ver en la figura 4, existen puntos del espacio de entrada donde no existe salida asociada, así pues, acciones tomadas por el usuario en situaciones extremas, tales como de error angular de ± 180 grados,... no son contempladas, ya que no se ha dado el caso durante la toma de datos. Para asegurarnos que casos extremos como estos quedan cubiertos, añadiremos los siguientes puntos al conjunto final:

- $(1,1,1)$: Si el error lateral y angular son máximos a la izquierda, el volante debe girarse lo máximo a la derecha
- $(-1,-1,-1)$: Si el error lateral y angular son máximos a la derecha, el volante debe girarse lo máximo a la izquierda

Una vez terminado el proceso, en la figura 5 podemos ver, tanto el conjunto de datos de partida como el conjunto de datos final, que será el que le pasaremos al programa que realizará el ajuste.

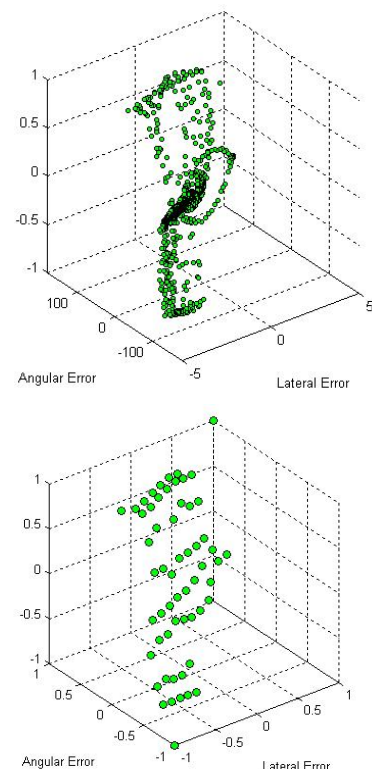


Figura 5: Conjunto de datos inicial (arriba) y final (abajo)

El método se ha ejecutado bajo la siguiente configuración de parámetros. En este caso, aquellos parámetros que son comunes a las fases de ajuste de la

base de reglas y funciones de pertenencia son iguales para ambas fases, excepto ξ . La configuración de parámetros es la siguiente: 100 iteraciones del método, con 15 individuos y 20 generaciones para cada uno de los algoritmos genéticos; probabilidad de mutación de 0.1; α (Cruce BLX) de 0.25; $\xi = 0.25$ (optimización de funciones de pertenencia) y $\xi = 3$ (optimización de base de reglas). Probabilidad de modificación del mejor individuo para la inicialización (ρ) de 0.5. Se usarán 21 posibles consecuentes para las reglas de la base de reglas y 5 funciones de pertenencia para cada una de las variables de entrada. Finalmente, para la función objetivo usaremos $v = 0.75$, por lo que estaremos dando más importancia a la semejanza con las salidas del piloto que a tener una superficie sin cambios bruscos.

Una vez aplicado el proceso, se han obtenido 3 controladores difusos, que denotaremos $5M$ (5 funciones de pertenencia con base de reglas marginal), $5C$ (5 funciones de pertenencia con base de reglas central) y $5T$ (5 funciones de pertenencia con base de reglas Total) respectivamente. Las superficies de control de cada uno de los controladores se muestran en la figura 6. Tras esto, los controladores han sido puestos a prueba en la Zona de Conducción, para comprobar si son capaces de seguir una ruta prefijada. Todas las rutas han sido realizadas a velocidades entre 15 km/h en curvas y 25 km/h en tramos rectos. El comportamiento seguido por cada uno de los controladores puede apreciarse a continuación.

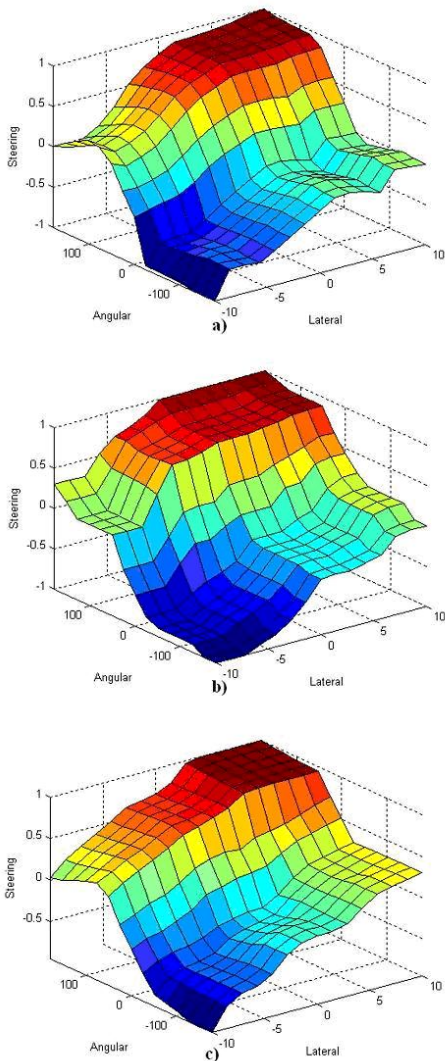


Figura 6: Superficies de control: 5M (a), 5C (b) y 5T (c)

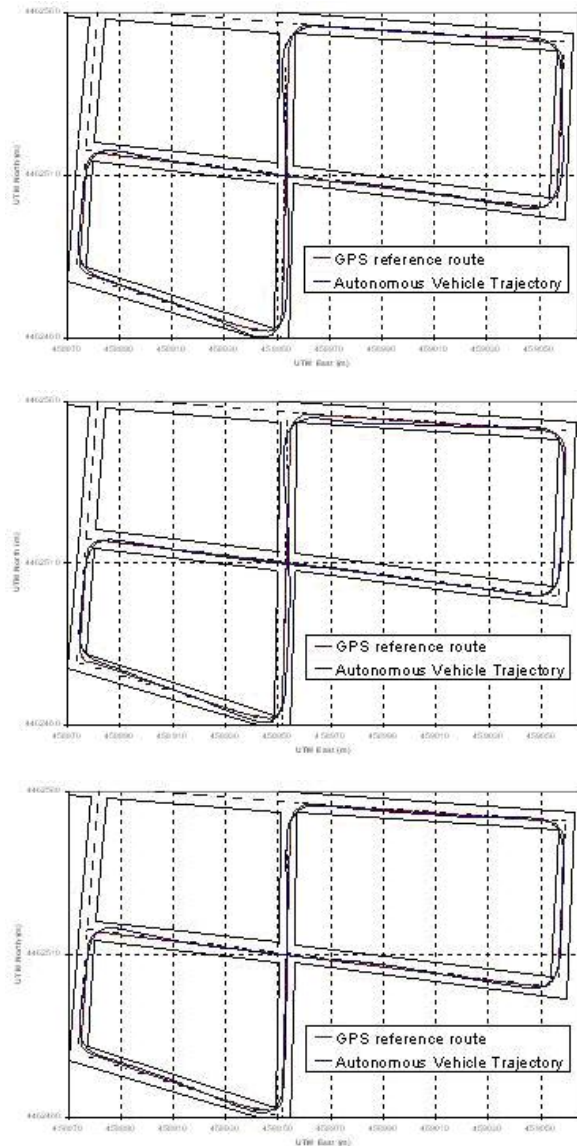


Figura 7: Referencia y Trayectoria seguida por 5M (arriba), 5C (centro) y 5T (abajo).

En las figuras puede apreciarse un relativo buen comportamiento de los controladores; todos ellos son capaces de, con mayor o menor precisión, seguir un tramo de carretera recto pero todos presentan el mismo fallo, tras una curva, se desvían hacia fuera de la línea de referencia; lo que puede ser provocado por el carácter no predictivo del modelo que, unido a la velocidad a la que han sido ejecutadas las pruebas, hacen que, una vez dentro de la curva, no le dé tiempo de reaccionar ante un nuevo punto de referencia GPS. La figura 8 muestra una comparación entre el error lateral y angular normalizado cometido por cada uno de los controladores. El gráfico muestra, debido a que ambas medidas pueden tomar valores positivos y negativos, el valor medio en valor absoluto de cada una de las medidas. El error angular ha sido normalizado a 10.55 grados, que se trata del mayor error angular promedio obtenido, mientras que el error lateral ha sido normalizado a 0.74 metros.

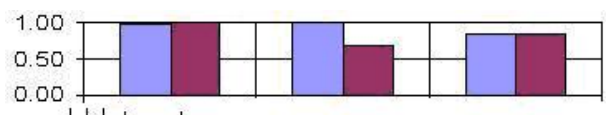


Figura 8: Comparativa del error obtenido. Error Angular (morado) y Error Lateral (magenta)

En la figura puede apreciarse que el peor comportamiento en lo que respecta al error angular ha sido obtenido por el controlador 5C y con respecto al error lateral, por 5M; el error angular obtenido por 5M es muy similar al obtenido con 5C; por tanto, diremos que 5M presenta el peor comportamiento general. Por otra parte, el controlador 5C presenta el menor valor para el error lateral. El valor mínimo para el error angular lo vemos en el controlador 5T, el cual también presenta un buen valor para el error lateral, por lo que diremos que 5T presenta el mejor comportamiento general.

5 CONCLUSIONES

A lo largo del presente trabajo se ha presentado un método para el ajuste automático de controladores difusos para el manejo de un vehículo autónomo, el cual ha sido equipado previamente con la instrumentación necesaria para llevar a cabo una conducción autónoma. Para lograr esto, se ha implementado un algoritmo genético iterativo, el cual es capaz de ajustar separadamente las funciones de pertenencia y la base de reglas del controlador difuso; para lograr un ajuste más preciso e interpretable, se han impuesto una serie de restricciones sobre los controladores, las cuales han ayudado al método, no sólo a ser capaz de obtener controladores apropiados para el manejo del volante del vehículo, sino también de llevar a cabo una conducción

suave y confortable. Los controladores obtenidos han sido probados en una pista de pruebas, mostrando, la mayoría de ellos, un buen comportamiento en el seguimiento de tramos de carretera rectos, aunque un tanto peores en lo que a tramos curvos se refiere; esto se debe al carácter no predictivo del modelo usado para generar los controladores, ya que no tienen en cuenta lo que viene por delante de la posición en la que se encuentran, sino únicamente el punto de la ruta en la que se encuentra. Los controladores han mostrado una conducción suave, incluso circulando a velocidades relativamente altas, lo que es un buen punto de partida para posteriores investigaciones en el área de los Sistemas de Transporte Inteligentes.

6 AGRADECIMIENTOS

Este trabajo cuenta con el apoyo de Ministerio de Educación y Ciencia, a través del proyecto SARA, DPI2005-07980-C03-01 y el proyecto Europeo Cyber-Cars2, STREP FP6-028062.

Referencias

- [1] Willie D. Jones. "Keeping Cars from, Crashing". IEEE SPECTRUM, September 2001, pp 40-45
- [2] J.E. Naranjo, C. González, J. Reviejo, R. García, T. de Pedro, and M.A. Sotelo, "Using Fuzzy Logic in Automated Vehicle Control" IEEE Intelligent Systems. Jan/Feb 2007.
- [3] J.E. Naranjo, C. González, J. Reviejo, R. García, and T. de Pedro, "A throttle & brake fuzzy controller: Towards the automatic car", in Lecture Notes on Computer Science, vol. 2809. Berlin, Germany: Springer-Verlag, Jul. 2003, pp. 291-301.
- [4] J. E. Naranjo, C. González, J. Reviejo, R. García, and T. de Pedro, "Adaptive fuzzy control for inter-vehicle gap keeping," Special Issue on Adaptive Cruise Control, IEEE Trans. Intell. Transp. Syst., vol. 4, no. 3, pp. 132-142, Sep. 2003.
- [5] E. Dickmanns, "The development of machine vision for road vehicles in the last decade," in Proc. IEEE Intelligent Vehicles Symposium., Versailles, France, 2002, pp. 268-281.
- [6] Gibbs, W. W. (2006). Innovations from a robot rally. Scientific American, 294(1), 64-71.
- [7] R. García Rosa, T. De Pedro. Modeling a fuzzy coprocessor and its programming language. Math-

ware and Soft Computing, vol V, n. 2-3, pp 167-174, 1998.

- [8] R. Garcia, T. De Pedro. "First Application of the ORBEX Coprocessor: Control of Unmanned Vehicles". 1999 EUSFLAT-ESTYLF Joint Conference. Mathware and Soft Computing, n. 7, vol2-3, 2000, pp. 265-273.
- [9] Huang, S. and W. Ren (1999). Use of neural fuzzy networks with mixed genetic/gradient algorithm in automated vehicle control. IEEE Transactions on Industrial Electronics, 46, 1090-1102
- [10] T. Takagi & M Sugeno. Fuzzy Identification of Systems and its Application to Modelling and Control. IEEE Trans. on Syst. Man and Cybernetics, 15:166-132
- [11] M. Sugeno, "On stability of fuzzy systems expressed by fuzzy rules with singleton consequents," IEEE Transactions Fuzzy Systems., vol. 7, pp. 201-224, Apr. 1999.
- [12] Holland, J.H. (1975).Adaptation in Natural and Artificial Systems. The University of Michigan Press.
- [13] Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, New York.
- [14] Eshelman L.J. & Schaffer J.D. (1993). Real Coded Genetic Algorithms and Interval Schemata. Foundation of Genetic Algorithms 2,L. Darrell Whitley (Ed.) (Morgan Kaufmann Publishers, San Mateo), 187-202.
- [15] W. Pedrycz, editor. Fuzzy Modelling: Paradigms and Practice. Kluwer Academic Press, 1996.
- [16] D. T. Pham and D. Karaboga. Optimum design of fuzzy logic controllers using genetic algorithms. Journal of Systems Engineering, 1 : 114-1 18, 1991.
- [17] F. Herrera and J. L. Verdegay, editors. Genetic Algorithms and Soft Computing. Number 8 in Studies in Fuzziness and Soft Computing. Physica-Verlag, 1996.