

CONTROLADOR DIFUSO PARA PROBLEMAS DE NAVEGACIÓN EN PRESENCIA DE OBSTÁCULOS FIJOS

M. Brox A. Gersnoviez S. Sánchez-Solano I. Baturone

Instituto de Microelectrónica de Sevilla, CNM-CSIC, Sevilla, España.

maria@imse.cnm.es andres@imse.cnm.es santiago@imse.cnm.es lumi@imse.cnm.es

Resumen

En esta comunicación se describe un sistema de control difuso para aplicaciones de navegación de robots móviles autónomos en presencia de obstáculos fijos. Las herramientas de CAD del entorno *Xfuzzy 3*, desarrollado en el IMSE, han facilitado el diseño del controlador. En la comunicación se procede a la verificación del controlador diseñado operando en un lazo cerrado con el modelo del robot móvil autónomo eléctrico Romeo 4R, diseñado y construido en la Escuela Superior de Ingenieros de la Universidad de Sevilla. Las simulaciones realizadas demuestran la eficiencia del controlador desarrollado.

Palabras Clave: Robots móviles, Control difuso, Obstáculos fijos.

reactivas para aprovechar las ventajas y reducir los inconvenientes de ambas opciones.

En esta comunicación se describe un sistema de control que emplea una estrategia de navegación reactiva basada en lógica difusa, que combina reglas de actuación fundamentadas en el conocimiento heurístico del problema con datos numéricos obtenidos del análisis geométrico del mismo. El sistema a controlar es el robot móvil autónomo Romeo 4R, dotado de una serie de sensores que facilitan el cálculo del estado del vehículo, así como la detección de los posibles obstáculos [8]. Los valores de posición, orientación y velocidad se obtienen a partir de los datos proporcionados por un encoder asociado al motor de tracción y por un giróscopo. A su vez, los datos del encoder del motor de dirección permiten obtener los valores de la curvatura del vehículo. Finalmente, un láser permite realizar un barrido angular de 0° a 180°, proporcionando, para cada ángulo, la distancia comprendida entre el robot y el posible obstáculo.

1 INTRODUCCIÓN

En la literatura se han reportado diversas alternativas para resolver el problema de evitación de obstáculos fijos con robots móviles autónomos. Estas soluciones se engloban en tres grandes grupos que corresponden a estrategias de navegación deliberativas, reactivas e híbridas [3]. Las deliberativas se basan en el conocimiento a priori del entorno [7]. Se caracterizan por conseguir una navegación en un tiempo mínimo o con un menor consumo energético aunque, como desventaja, presentan problemas en la adaptación a entornos dinámicos parcialmente conocidos. Las estrategias reactivas consisten en provocar una reacción frente a la información sensorial que se recibe. Entre éstas se encuentran las basadas en el seguimiento del contorno del obstáculo, en campos potenciales [2], en optimización local [6] y en el uso de la lógica difusa [4]. Se caracterizan por su flexibilidad y, como contrapartida, presentan la falta de capacidad de navegar de forma óptima. Finalmente, las estrategias híbridas [5] usan esquemas mixtos que combinan técnicas deliberativas y

Las fases de descripción y verificación se han llevado a cabo con las diferentes herramientas de CAD proporcionadas por el entorno de desarrollo de sistemas difusos *Xfuzzy 3* [10]. Las herramientas de descripción de dicho entorno facilitan la especificación del sistema de control mediante una estructura jerárquica que combina bloques difusos con bloques de procesamiento convencional. Por otro lado, las herramientas de verificación permiten ajustar los parámetros del sistema y comprobar su comportamiento mediante simulación.

Esta comunicación está estructurada como a continuación se describe. La sección 2 muestra un estudio teórico del problema de navegación de robots móviles autónomos con evitación de obstáculos. En ella se explica cómo se afronta el problema de detección de obstáculos y cómo modifica el vehículo su trayectoria cuando el obstáculo es detectado. La sección 3 presenta el diseño y la implementación del controlador difuso, usando la metodología del entorno *Xfuzzy 3*; en ella se muestra cómo las herramientas de CAD del entorno han facilitado su desarrollo. En la sección 4 se procede a la verificación

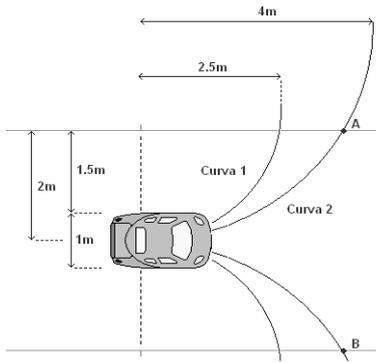


Figura 1. Construcción del pasillo recto

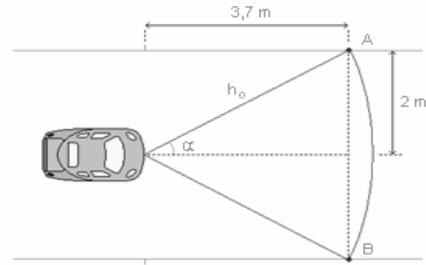


Figura 2. Sector de visión del pasillo

del controlador difuso operando en un lazo cerrado con el modelo del robot Romeo 4R; en ella se muestran los resultados obtenidos tras la simulación realizada con la herramienta *xfsim* del entorno *Xfuzzy 3*. Por último, la sección 5 recoge las conclusiones y las líneas de estudio hacia las que esta investigación se va a encaminar en el futuro.

2 EL PROBLEMA DE NAVEGACIÓN EN PRESENCIA DE OBSTÁCULOS

Supongamos que en un instante dado t_0 el vehículo tiene la configuración $(x_0, y_0, \phi_0, v_0, \gamma_0)$, donde x e y hacen referencia a las coordenadas de posición del vehículo, ϕ a la orientación del mismo con respecto al eje $x = 0$, v a la velocidad y γ a la curvatura. Las relaciones entre las variables implicadas en esta configuración pueden obtenerse utilizando el modelo cinemático de la bicicleta para describir el comportamiento del robot [9]:

$$\begin{cases} \dot{x} = v \cdot \sin(\phi) \\ \dot{y} = v \cdot \cos(\phi) \\ \dot{\phi} = \gamma \cdot v \end{cases}$$

En función de la configuración inicial y de la posición final fijada como objetivo, el sistema de control de conducción debe proporcionar los valores de velocidad y curvatura que serán aplicados en un instante posterior $t_0 + \Delta t$. Para alcanzar la configuración final evitando colisiones con los posibles obstáculos se considerarán dos situaciones. Si no existe obstáculo se activará un controlador difuso cuya finalidad será la de hacer navegar al vehículo hasta la configuración final. Sin embargo, si existe algún obstáculo, se activará un segundo controlador difuso cuya finalidad será evitarlo. De esta forma, es necesario contemplar dos problemas: por un lado la navegación sin obstáculos y, por otro, la modificación de la trayectoria del vehículo para evitar los obstáculos que se detecten.

2.1. DETECCIÓN DE OBSTÁCULOS

Distinguiremos dos situaciones en función de que el vehículo esté describiendo una trayectoria recta ($\gamma = 0$) o circular ($\gamma = \gamma_0$).

Si el robot está describiendo una trayectoria recta, se construye un pasillo recto en torno al coche y se va estudiando si está libre de obstáculos. Se considera que el pasillo tiene una anchura de 2m respecto al eje del vehículo. Teniendo en cuenta que Romeo tiene una anchura de 1m, esto significa que la separación entre los laterales del vehículo y la pared del pasillo será de 1,5m. La curvatura máxima que puede tomar Romeo es de $0,4m^{-1}$, lo cual supone que puede describir una trayectoria con un radio mínimo de 2,5m. Esta trayectoria, representada como curva 1 en la Figura 1, podrá llevarse a cabo si el robot ya tenía una curvatura de $0,4m^{-1}$ o no la tenía pero estaba parado. En otro caso, puesto que las transiciones en la curvatura no pueden ser instantáneas, una trayectoria como la representada por la curva 1 no será realista, sino que lo serán trayectorias como la representada por la curva 2, cuyo radio de curvatura es superior a 4m. Si aproximamos la curva 2 de la Figura 1 por su secante se obtiene el sector circular de la Figura 2. Los valores aproximados de los parámetros h_0 y α de la Figura 2 son 4,2m y 30° respectivamente. Por tanto, la amplitud angular del sector circular (2α) es aproximadamente 60° .

En este caso, el obstáculo queda caracterizado por los parámetros h , Φ_R y Φ_L . Se define h como la distancia mínima existente entre el vehículo y el obstáculo. Se define Φ_R como el ángulo comprendido entre el borde derecho del obstáculo y la perpendicular al eje del vehículo. De forma análoga se define Φ_L como el ángulo comprendido entre el borde izquierdo del obstáculo y la perpendicular al eje del vehículo. Consideraremos que el controlador difuso para evitación de obstáculos entrará en juego si el valor de h es menor que el valor de h_0 y Φ_R o Φ_L están comprendidos entre $(\pi/2 + \alpha)$ y $(\pi/2 - \alpha)$. Todo

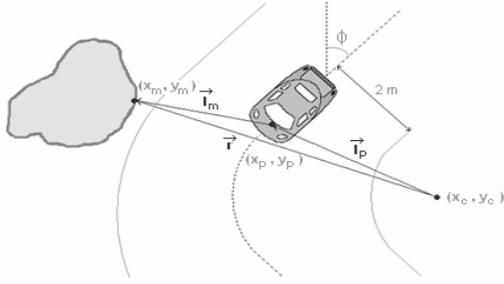


Figura 3. Construcción del pasillo curvo

obstáculo cuyos parámetros característicos estén comprendidos en el sector circular, producirá una modificación en la trayectoria que llevaría el robot si no hubiera obstáculos. Por tanto, los valores de h_0 y α son los valores límites que se han establecido para que el obstáculo provoque un cambio en la trayectoria.

Por otro lado, si el robot está describiendo una trayectoria circular, se construye un pasillo curvo de anchura 2m y se estudia si está libre de obstáculos. Las coordenadas (x_c, y_c) de la Figura 3 hacen referencia al centro de la circunferencia que describe el vehículo en su trayectoria; (x_p, y_p) son las coordenadas de posición del láser del vehículo situado en la parte delantera del mismo; por último, (x_m, y_m) hacen referencia a las coordenadas del punto del obstáculo que se encuentra más próximo al robot. Como se observa en la figura, el vector \vec{r} se puede expresar como suma de los vectores \vec{l}_p y \vec{l}_m :

$$\vec{r} = \vec{l}_p + \vec{l}_m = (x_p - x_c, y_p - y_c) + (x_m - x_p, y_m - y_p) = (x_m - x_c, y_m - y_c) \quad (1)$$

La ausencia de obstáculos que hagan preciso modificar la trayectoria que llevaba el robot viene, en este caso, determinada por la siguiente condición:

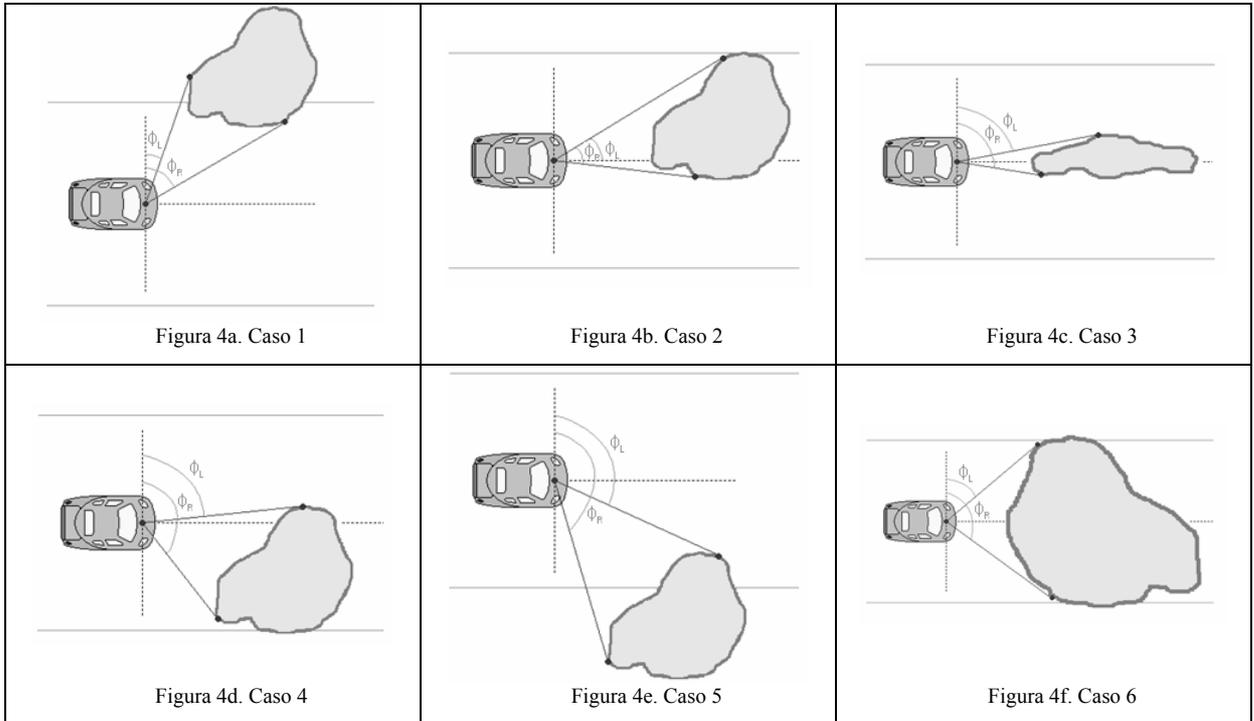
$$(x_m - x_c)^2 + (y_m - y_c)^2 \geq \left(\frac{1}{|\gamma_d|} + 2\right)^2 \quad \text{ó} \quad (x_m - x_c)^2 + (y_m - y_c)^2 \leq \left(\frac{1}{|\gamma_d|} - 2\right)^2 \quad (2)$$

En los casos en que la ecuación (2) no se verifique, el robot deberá modificar su trayectoria.

2.2. MODIFICACIÓN DE LA TRAYECTORIA DEL VEHÍCULO PARA EVITAR OBSTÁCULOS

Para determinar la magnitud y el signo de la curvatura que debe adoptar el vehículo para esquivar el obstáculo, se han de cuantificar criterios como la cantidad y situación espacial de pasillo que está siendo ocupado por el obstáculo. Dependiendo de que el pasillo sea ocupado en mayor o menor medida por el obstáculo, el vehículo deberá aplicar una curvatura mayor o menor para poder esquivarlo. También es preciso considerar la zona de pasillo que está siendo ocupada para saber qué signo debe tomar la curvatura. Para poder cuantificar todo lo anterior han de evaluarse la magnitud y el signo de los ángulos Φ_R y Φ_L . En la Figura 4 se muestran seis casos en los que se analizan las distintas situaciones en que puede encontrarse el obstáculo y la curvatura a adoptar por el vehículo.

En el primer caso el obstáculo se encuentra poco metido en el pasillo invadiendo la zona izquierda, con lo que con una curvatura no muy elevada y positiva sería suficiente para esquivarlo. Se considera como curvatura positiva



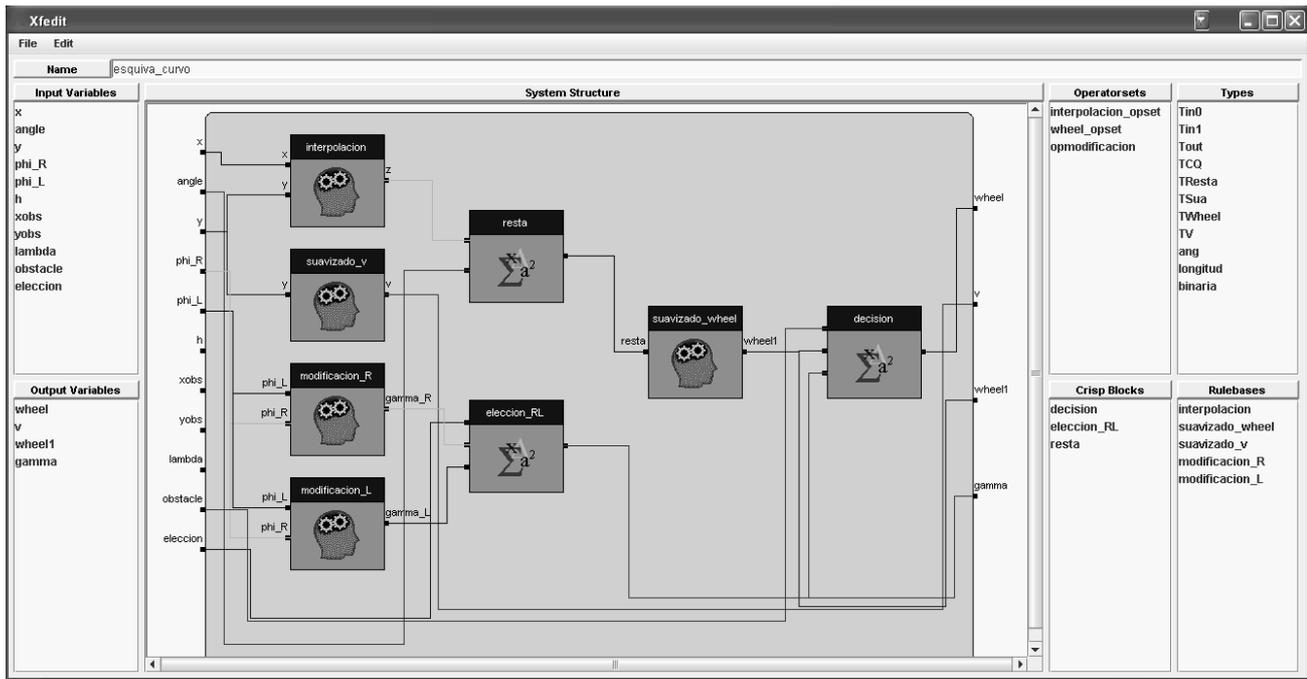


Figura 5. Visualización de la estructura del controlador difuso mediante la ventana principal de *xfedit*

aquella cuyo sentido es el de las agujas del reloj y negativa la que toma el sentido contrario. La Figura 4b representa el caso de un obstáculo bastante metido en el pasillo por la zona izquierda, con lo que se necesita una curvatura elevada y positiva para esquivarlo. El caso mostrado en la Figura 4c representa un obstáculo muy pequeño que se encuentra en medio del pasillo y enfrente del vehículo, con lo que no se necesita una curvatura elevada para esquivarlo. Como el obstáculo se encuentra en medio del pasillo, será (en principio) indiferente el signo que adopte la curvatura. En las Figuras 4d y 4e podemos observar dos situaciones simétricas a las de los casos 1 y 2, con lo cual, la curvatura resultante para evitar el obstáculo será igual en módulo, pero de signo contrario. Por último, en la Figura 4f, se observa un obstáculo muy grande que se encuentra en medio del pasillo y enfrente del vehículo, con lo que se necesita una curvatura elevada para esquivarlo. El signo que debe tomar la curvatura del vehículo se discutirá más adelante.

3 DISEÑO DEL SISTEMA DE CONTROL

Las ideas anteriores sobre detección y evitación de obstáculos han sido aplicadas en el desarrollo de un sistema de navegación para robots móviles que combina dos controladores difusos: uno para guiar al vehículo desde una configuración dada hasta la configuración objetivo ($x = y = \gamma = \phi = v = 0$), y otro para modificar su trayectoria en presencia de obstáculos. Mientras que para el diseño del primer controlador se han analizado las restricciones geométricas del problema con objeto de optimizar la trayectoria seguida por el vehículo [1], en el

diseño del segundo se emplea un esquema de razonamiento similar al que seguiría un conductor humano. Tanto en un caso como en el otro la lógica difusa permite expresar el comportamiento del sistema de forma lingüística mediante reglas del tipo si - entonces.

El sistema de control diseñado sigue un esquema jerárquico cuya estructura se puede describir editando directamente un archivo ".xfl" o gráficamente con la herramienta *xfedit* del entorno *Xfuzzy 3*, tal y como se muestra en la Figura 5. En la estructura del controlador se observan los bloques "interpolación", "resta", "suavizado_wheel" y "suavizado_v" que se encargan del cálculo de la trayectoria del robot hasta la configuración objetivo. Las trayectorias conseguidas con estos bloques son rectas tangentes que unen circunferencias de curvatura máxima trazadas en el punto de inicio y de llegada tal y como muestra la Figura 6.

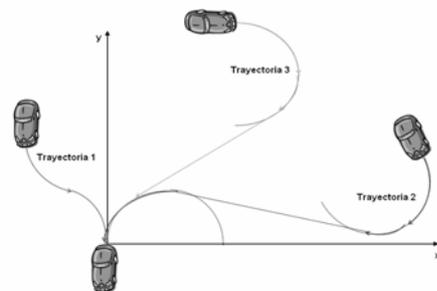


Figura 6. Trayectorias descritas por el vehículo



Figura 7. Base de reglas del bloque modificacion_R mediante *xfedit*

Los bloques “modificacion_R” y “modificacion_L” se encargan de modificar la trayectoria del vehículo en el caso de que se detecte la presencia de obstáculos. Como se indicó en la sección 2, la modificación de las trayectorias se hace en base a la cantidad y zona de pasillo ocupado por el obstáculo; para ver esto se estudian los valores de ϕ_R y ϕ_L . Según los valores de estos ángulos, se distinguen los seis casos comentados en la sección 2 y para cada uno de ellos se propone una curvatura diferente; esto se puede formular a partir de reglas del tipo si-entonces. La herramienta *xfedit* permite editar la base de reglas de estos bloques como se observa en la figura 7. La diferencia entre la base de reglas de ambos bloques radica en que para los casos dudosos de la sección 2 en los que sea indiferente el signo de la curvatura para esquivar el obstáculo, “modificacion_R” toma curvatura positiva mientras que “modificacion_L” toma negativa. El conjunto de operadores implicados en un bloque (figura 8b), así como el tipo de las variables se puede definir también a partir de la herramienta *xfedit*, como se observa para el caso del tipo *ang* de las variables ϕ_R y ϕ_L en la figura 8a. Mediante esta herramienta se puede especificar el número de funciones de pertenencia, los rangos de las mismas, así como el tipo (triangulares, trapezoidales, gaussianas, singleton, etc.).

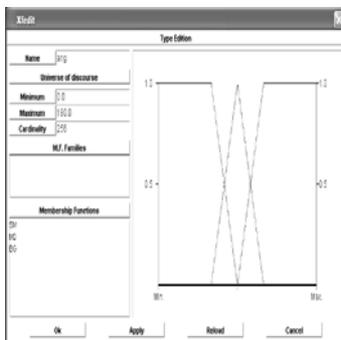


Figura 8a. Especificación de los tipos de variables

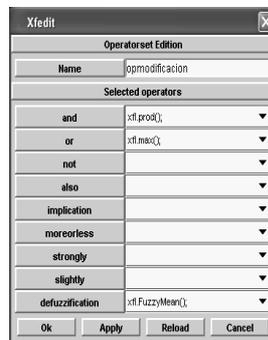


Figura 8b. Definición del conjunto de operadores

En la estructura también aparece “Eleccion_RL” que es un bloque no difuso que se encarga de tomar curvatura positiva o negativa en los casos dudosos dependiendo de con cuál de ellas se consigue una trayectoria mínima. Para ello se estudia cuál es la dirección que llevaba el robot antes de detectar el obstáculo, evaluando el signo de la curvatura proporcionada por “suavizado_wheel”, *wheel*, y se decide así el signo de curvatura más conveniente. Si su valor es positivo, la variable binaria *eleccion* tomará valor cero, mientras que si es negativo, tomará valor uno. El valor tomado por la variable *eleccion* se guarda y no sufre cambios hasta ver un nuevo obstáculo. Para detectar cuándo se dan los casos dudosos se define una variable llamada *indecision*, siendo su valor uno cuando se encuentra en uno de estos casos y cero en caso contrario.

Finalmente, en la estructura del controlador aparece “decision” que nuevamente es un bloque no difuso que toma como salida la del bloque “suavizado_wheel” o la de “eleccion_RL” según el valor de la variable *obstacle*; esta variable toma valor cero cuando el controlador no detecta ningún obstáculo y toma valor uno cuando la detección es positiva. Para calcular el valor de la variable *obstacle* se utilizan las condiciones de obstáculo de pasillo recto y curvo de la sección 2.

4 VERIFICACIÓN DEL CONTROLADOR

Para estudiar si el comportamiento del controlador difuso es correcto, hemos empleado las herramientas de verificación del entorno *Xfuzzy 3*, principalmente la herramienta de simulación *xfsim*. Es importante verificar el comportamiento de un controlador con herramientas de simulación antes de experimentar con la planta real para así detectar posibles errores. Para realizar las simulaciones se utiliza un modelo de la planta cuyo código está escrito en Java. En el código también se incluye el preprocesado de las señales proporcionadas por los sensores así como el cálculo de variables como *obstacle*, *eleccion* e *indecision*.

Los resultados de algunas de las simulaciones realizadas se pueden visualizar en la Figura 9. Las figuras 9a y 9b corresponden a casos con un único obstáculo. En ellas se observa cómo el robot detecta perfectamente el obstáculo representado por un tramo de línea continua de grosor mayor. Desde el momento en el que el obstáculo es detectado, se ve cómo el vehículo comienza a modificar su trayectoria separándose de la línea discontinua, que representa la seguida por el vehículo cuando no hay obstáculo. Las figuras 9c y 9d corresponden a casos más complejos con dos obstáculos. En ellas se observa cómo el vehículo detecta en primer lugar el obstáculo más cercano modificando su trayectoria para poder evitarlo. Seguidamente cuando ve el siguiente obstáculo vuelve a modificarla para evitar la colisión con el segundo.

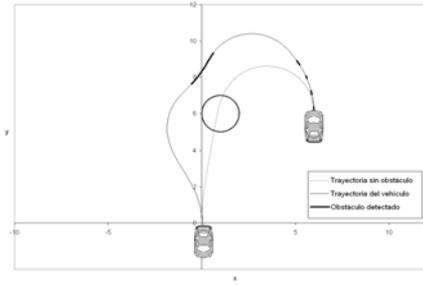


Figura 9a

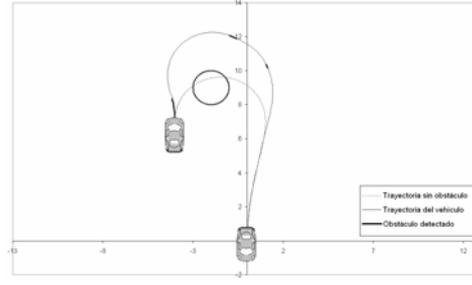


Figura 9b

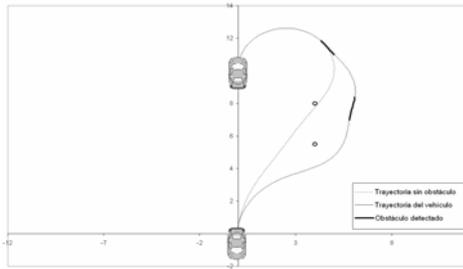


Figura 9c

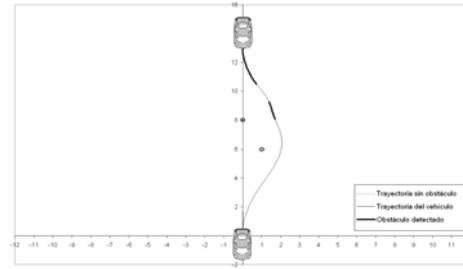


Figura 9d

5 CONCLUSIONES

En esta comunicaci3n se ha presentado una soluci3n al problema de navegaci3n de robots m3viles aut3nomos evitando obst3culos est3ticos, basada en la combinaci3n del an3lisis geom3trico del problema con conocimiento heur3stico modelado mediante l3gica difusa. Para la verificaci3n experimental se ha utilizado el veh3culo Romeo 4R, teniendo en cuenta sus restricciones cinem3ticas y utilizando varios de los sensores y actuadores de los que est3 provisto. En el desarrollo del sistema se han utilizado diversas herramientas de CAD del entorno *Xfuzzy 3* para el dise1o e implementaci3n del controlador. En particular, se ha hecho amplio uso de la herramienta *xfsim*, que permite simular el controlador operando en lazo cerrado con un modelo del robot. Las l3neas de trabajo futuro de esta investigaci3n se centrar3n en la implementaci3n del controlador siguiendo una estrategia h3brida Hardware-Software.

Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos TEC2005-04359/MIC (CICYT), TIC2006-635 (Junta de Andaluc3a) y DPI2005-02293.

Referencias

[1] I. Baturone, F. J. Moreno Velo, S. S3nchez Solano, A. Ollero. Automatic design of fuzzy controllers for car-like autonomous robots, *IEEE Transactions on Fuzzy Systems*, Vol. 12, No. 4, 2004.

[2] C. I. Connolly, J. B. Burns, R. Weiss. Path planning using Laplace's equation, *IEEE Conference on Robotics and Automation*, Cincinnati, 1990.

[3] R. Fern3ndez Ramos. *Navegaci3n aut3noma de robots m3viles en entornos parcialmente conocidos*, Tesis Doctoral, M3laga 2001.

[4] D. Fox, W. Burgard, S. Thurn. The dynamic window approach to collision avoidance, *IEEE Robotics & Automation Magazine*, 1997.

[5] K. Kant, S. Zucker. Toward efficient trajectory planning: The path-velocity decomposition, *The International Journal of Robotics Research*, 1986.

[6] T. Kubota, H. Hashimoto. A strategy for collision avoidance among mobile obstacle for a mobile robot, *11th International World Congress*, Fallin, Estonia, 1990.

[7] N. J. Nilsson. *Principles of artificial intelligence*, Palo Alto: Tioga, 1980.

[8] A. Ollero, B. C. Arrue, J. Ferruz, G. Heredia, F. Cuesta, F. L3pez-Pichaco y C. Nogales. Control and perception components for autonomous vehicles guidance. Application to the Romeo Vehicles, *Control Engineering Practice*, Vol. 7, No. 10, pp 1291-1299, October 1999.

[9] A. Ollero. *Rob3tica, manipuladores y robots m3viles*, Ed: Marcombo, 2001.

[10] *Xfuzzy: Herramientas de CAD para L3gica Difusa* (<http://www.imse.cnm.es/Xfuzzy/>)