



Ingeniería e Investigación

ISSN: 0120-5609

revii_bog@unal.edu.co

Universidad Nacional de Colombia
Colombia

Rodríguez Garavito, Cesar Hernán; Delgado Rivera, Alberto
Plataforma autónoma inteligente para fútbol robot
Ingeniería e Investigación, vol. 28, núm. 1, abril, 2008, pp. 81-98
Universidad Nacional de Colombia
Bogotá, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=64328110>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Plataforma autónoma inteligente para fútbol robot

An intelligent autonomous platform for robot football

Cesar Hernán Rodríguez Garavito¹ y Alberto Delgado Rivera²

RESUMEN

Este artículo presenta el diseño, la implementación y los resultados obtenidos, confrontados con simulación, de una plataforma hardware-software para fútbol robot en un entorno artificial que consta de un campo o arena de juego, un agente móvil controlado por la aplicación (AMC), un agente móvil controlado por usuario (MCU), una bola de golf, un sensor para capturar el estado del entorno y un controlador de trayectoria. El problema que enfrenta el agente AMC en su entorno es la búsqueda de una posición con línea de vista libre para golpear la bola hacia la portería del equipo contrario, mientras esquivando adversarios en su camino. El campo de juego es un área confinada de 1,5 m x 2 m, en la cual el agente móvil se desplaza según la cinemática de un robot diferencial a una velocidad máxima de 25,7 cm/s, el móvil posee la capacidad de pateo de una bola de golf, la cual se puede impulsar a una velocidad máxima de 1,43 m/s. Por otro lado, se implementa y optimiza un sensor de posición. El proceso parte con la captura de imágenes de la arena de juego, continúa con la conversión al formato HSI y la aplicación de filtros digitales sobre las imágenes, y termina con el cálculo de las componentes cartesianas a partir de los patrones en color, que identifican a los móviles y bola de juego, la velocidad que alcanza el sensor es de 30 cuadros por segundo. Finalmente, se realiza el control de navegación inteligente a través de una estrategia híbrida, en la cual se combina linealmente el efecto de dos controladores basados en lógica difusa, uno para llevar el agente a la posición de tiro al arco y el otro para esquivar obstáculos mientras navega a la posición de tiro.

Palabras clave: lógica difusa, planeación de trayectorias, fútbol robot, robot móvil, visión por computador.

ABSTRACT

This article presents the design, implementation, virtual simulation and test results for a hardware-software platform for robot soccer. The platform consists of a soccer field, a software-controlled mobile agent (AMC), a user-controlled mobile agent (MCU), a golf ball, an image sensor system and a path-planning controller. The problem for agent AMC is to sense the environment and move to a kicking position to score while avoiding opponents in its way. The soccer field is a bounded area of 1.5m x 2m in which mobile agents move following differential robot kinematics at 25.7cm/s speed; each robot has a kicking system which can speed up a golf ball at 1.43m/s velocity. A position sensor was implemented and optimised in three stages: capturing the soccer field image, converting to HSI format and applying digital filters and calculating the Cartesian components from colour patterns for identifying the mobile robots and the golf ball. Image sensor system processing speed is 30 frames per second. Robot path planning is controlled by a hybrid strategy based on two linearly combined fuzzy logic controllers; one takes agent AMC to the goal position and the other avoids obstacles while it moves to the goal position.

Keywords: fuzzy logic, path planning, RoboSoccer, mobile robot, artificial vision, image processing.

Recibido: septiembre 25 de 2007

Aceptado: febrero 28 de 2008

Introducción

Las competiciones robóticas a nivel mundial han sido desde hace 20 años un escenario de intercambio académico. Con el paso de los años los robots se han convertido en agentes cada vez más rápidos e inteligentes, llegando al punto de tener cotejos independientes del control supervisado humano.

Algunos certámenes internacionales de competencia robótica que se efectúan periódicamente en la actualidad, en orden de aparición, son: The Micromouse Contest (Yong, 1964), llevado a cabo en la Universidad de Queensland, allí se planteó originalmente el desafío de un robot móvil que explora un laberinto para encontrar la salida en el menor tiempo posible; años más tarde aparece The AAAI Mobile Robot Competition (Kortenkamp), allí los robots en cada versión del certamen tienen que cumplir diferentes ta-

¹ Ingeniero electrónico, Universidad Distrital "Francisco José de Caldas", Colombia. M.Sc., en Automatización Industrial, Universidad Nacional de Colombia, Bogotá. Profesor, dedicación exclusiva, Universidad de la Salle, Facultad de Diseño y Automatización electrónica, Colombia. chrodriguez@gmail.com, chrodriguez@unal.edu.co, cesarrodriguez@la.salle.edu.co.

² Ingeniero electrónico. M.Sc., en Ingeniería eléctrica. Ph.D., en Cibernética. Profesor titular, Facultad de Ingeniería, Universidad Nacional de Colombia, Bogotá. jadelgador@unal.edu.co, adelgado@ieee.org.

reas con un aumento gradual en su complejidad año tras año. Por ejemplo, en Obregon (1996), se propuso resolver las tareas de navegación en un ambiente de oficina actuando como un mensajero.

Finalmente surge *RoboCup Robot Soccer*, un espacio para promover la investigación en robótica e inteligencia artificial fijando un objetivo para que los participantes lo cumplan a través de la integración de diferentes tecnologías. Su principal campo de trabajo es la línea The Robot World Cup Soccer Games and Conference, que escoge el juego de fútbol como iniciativa de competición, Robocopa es una tarea de varios robots moviéndose rápidamente sobre un ambiente cambiante (Robocup.org). Incluye tópicos como la navegación autónoma de un agente, colaboración multiagente, adquisición y reconocimiento de imágenes, toma de decisiones en tiempo real, aprendizaje en equipo, entre otros. Esta iniciativa fue propuesta por (Kitano, 1997), actualmente The Robot World Cup Soccer se divide en cinco ligas, donde la línea central es la liga de humanoides, que aunque aún incipiente, encarna la visión a futuro de RoboCup (Burkhard, 2002), y consiste en que para el año 2050 sea el escenario para un juego entre los campeones robóticos y el equipo de humanos campeón del mundo.

El presente trabajo se enmarca en *Robot World Cup Soccer Games and Conference Small League*, donde se establecen 5 jugadores por campo de juego, la utilización de sensores externos es permitida (el estándar es una cámara ubicada encima de la cancha para posicionamiento global de los móviles), el tamaño del espacio de juego es de 1,7 m por 3,4 m, y cada equipo se identifica con una marca de color, al igual que la bola. Normalmente los equipos envían la información de movimiento a los robots desde una estación de trabajo que procesa las imágenes, genera las directivas de juego y las realimenta al campo. En pocas ocasiones se tiene el concepto de *robot* autónomo con sensores locales.

Hoy por hoy, en muchas universidades del mundo se trabaja en el desarrollo de equipos de juego para competir en la siguiente versión RoboCup. En Colombia se encuentran los grupos de investigación Uncolteam, de la Universidad Nacional, y el del Laboratorio de Automatización, Microelectrónica e Inteligencia Computacional, Lamic de La Universidad Distrital. En La Universidad Nacional a nivel de pregrado se ha realizado un trabajo presentado por Molina y Cárdenas, (2005), donde se presenta una solución mecánica capaz de ubicar y dirigir una bola hacia un objetivo. A nivel de posgrado, en Sotomonte (2007) se realiza un estudio de las distintas estrategias cooperativas, aplicado al caso de estudio, fútbol de robots. Estas propuestas abren el camino hacia un trabajo continuado que retome el conocimiento adquirido y lo potencie teniendo en cuenta el estado del arte de los algoritmos en imágenes, IA, y sistemas mecánicos.

El problema que se pretende resolver con el presente trabajo es la implementación de dos unidades móviles con capacidad de movimiento diferencial y posibilidad de impacto sobre una bola, diseñadas específicamente para jugar fútbol.

Estas unidades son controladas inalábricamente en un espacio confinado, provisto de la ambientación propia de una cancha de juego proporcional al tamaño de los móviles.

La ubicación de los robots, así como la de la bola y los marcos de anotación, se efectúa a través de una cámara de video que toma varias imágenes por segundo de la arena de juego en formato digital y las transmite a una estación de trabajo, donde se abren y editan por medio de un software de programación de alto nivel (C# Csharp). A las matrices obtenidas según el formato escogido, en este caso RGB, se les aplica algoritmos de procesamiento de imágenes para extraer las ubicaciones de los móviles, la bola, los límites exteriores del campo de juego y los objetivos para anotación de gol. Los procedimientos aplicados a los fotogramas capturados son: filtrado, acondicionamiento, umbralización y, finalmente, segmentación de la imagen. Con esta información se alimenta el algoritmo de planeación de trayectorias, basado en lógica difusa para determinar el movimiento de los robots, en orden a evadir obstáculos dinámicos o a intersecarlos según sea el caso y así cumplir el objetivo de llevar la bola a la portería contraria evadiendo al oponente.

Para verificar el comportamiento del jugador robótico se hará uso de un simulador donde se pruebe la efectividad teórica de los algoritmos de juego, un paso importante para el desarrollo futuro de un equipo que fije su objetivo en las contiendas internacionales, sumando fuerzas con otras universidades de Bogotá.

El diseño y la implementación de la plataforma para fútbol robot se aborda de la siguiente manera: primero se describe en forma general la aplicación, a partir de allí se presenta el diseño del agente robot AMC, que consta de una componente de hardware y otra de software, en donde se resalta el diseño e implementación del controlador de velocidad CCRB (controlador difuso basado en relaciones booleanas) (Soriano, 2001); en la siguiente sección se desarrolla la optimización del sensor de posición basado en imágenes, partiendo del algoritmo del equipo RoboRoos (Wyeth, 2000); posteriormente se presenta el diseño e implementación del algoritmo de planeación de trayectorias o *path planning*, para el agente AMC; luego se muestran los resultados de las pruebas realizadas al sensor de velocidad del móvil AMC, al sensor de posición por captura de imágenes de la plataforma general, y se hace una comparación entre trayectorias obtenidas por simulación y trayectorias capturadas en línea desde la plataforma, bajo las mismas condiciones iniciales, posición de los elementos de la plataforma. Finalmente, se concluye sobre los resultados obtenidos, y se enuncian las fuentes bibliográficas que sustentan el trabajo desarrollado.

Descripción de la aplicación

La arquitectura general de la plataforma consta de diferentes elementos tales como la cancha de juego, el sistema de visión por computador, los robots jugadores de fútbol, el sistema de control automático para planeación de trayectorias y,

finalmente, la pelota de juego. Una vista de la plataforma en conjunto se observa en la Figura 1.

El *hardware* del robot móvil y su funcionamiento, se explican a continuación. El sensor de video y el sistema de control para planeación de trayectorias son tema de secciones siguientes.

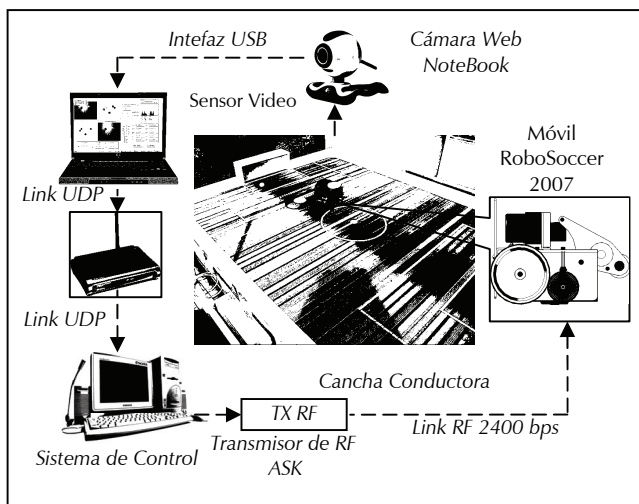


Figura 1. Imagen de la plataforma RoboSoccer completa. Detalle del sistema de visión por computador. Detalle de los móviles. Detalle del sistema de control.

Un esquema de los bloques funcionales de los móviles, tanto de su componente de *hardware* como de *software*, se muestra en la Figura 2.

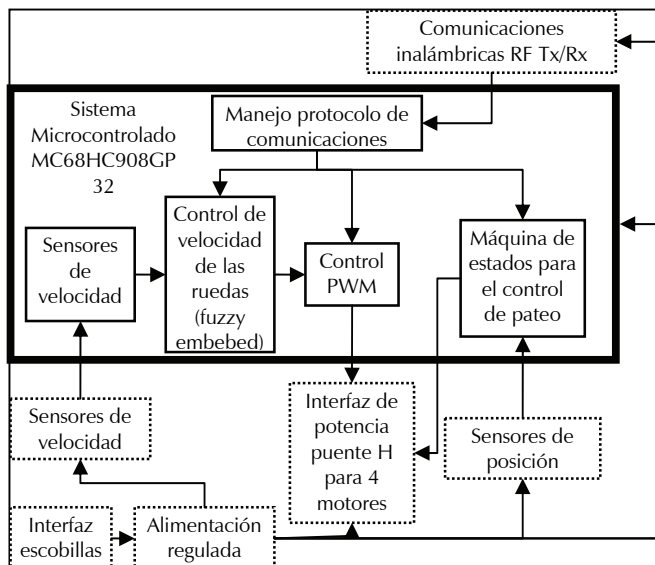


Figura 2. Esquema de la distribución de funciones en el robot móvil. En línea punteada se muestran los circuitos electrónicos, y en línea gruesa continua se observa la componente de software que contiene el microcontrolador MC68HC908GP32.

Hardware del robot móvil

La componente mecánica del *hardware*, mostrada en la Figura 3, cuenta con dos bloques principales: el sistema de lo-

comoción y el de pateo.

El sistema de locomoción se implementa a través de dos ruedas fijadas a un chasis, junto con un tercer apoyo de movimiento omnidireccional. Los actuadores utilizados son motores DC de 12V referencia HN-GH35GMB. La velocidad máxima del conjunto es de 25 cm/s.

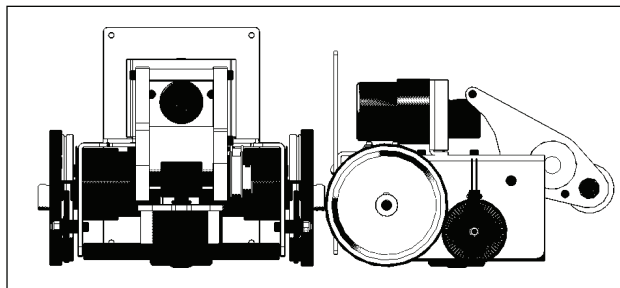


Figura 3. Perspectivas del móvil jugador de fútbol. Vistas frontal y lateral.

El sistema de pateo es un mecanismo capaz de impulsar una bola a partir de la transmisión de impulso por contacto directo. El principio consiste en presionar la pelota de juego con un rodillo giratorio de material adherente contra el suelo; el efecto de giro del motor imprime una fuerza tangencial en el sentido de la superficie contra la cual se aprisiona la bola produciendo el efecto de movimiento.

En cuanto a la componente electrónica de los móviles, los sistemas más relevantes de la aplicación son el sensor de velocidad y el enlace de comunicaciones.

Para medir la velocidad promedio de las ruedas del móvil *robot* se construye un *encoder* de disco ranurado de 68 cavidades maquinadas que gira por acción de un acople con p Polea, con relación de 6.25:1, con cada una de las ruedas del móvil, y que intercepta una barrera de luz establecida entre un diodo emisor de luz infrarroja y un foto detector ensamblados sobre un dispositivo mecánico llamado herradura, el cual se sujeta a la estructura del móvil, como se muestra en la Figura 4. La luz incidente proveniente del led penetra las ranuras del disco y genera una señal oscilante sobre el fototransistor conforme el disco se revoluciona; la frecuencia del tren de pulsos sentido es proporcional a la velocidad de giro de las llantas donde el arreglo es fijado.

La comunicación se efectúa haciendo uso de los circuitos integrados TLP y RLP 434A. Estos dispositivos funcionan a una velocidad de 2.400 bits/s y exhiben un alcance real de transmisión de hasta 30 m. Estos CI sólo requieren alimentación, antena o dipolo de $\lambda/2$ construido con cable, y la señal a transmitir en formato RS232.

Software del robot móvil

Se implementa un controlador de velocidad difuso CCRB (Soriano, 2001) en cada una de las ruedas. Para abordar el diseño primero se realiza una conversión de dominios desde la variable velocidad, a indicadores relacionados con el punto donde se pretende fijar la variable (*set point*) como se ob-

serva en la Figura 5, error e integral de error, a través de la Ecuación 1. Posteriormente, se plantean los conjuntos involucrados en los universos de discurso: integral de error positivo $\mu_p(e)$, integral de error negativo $\mu_n(e)$, error positivo $\mu_p(e)$ y error negativo $\mu_n(e)$, como se muestra en la Figura 6. Ahora bien, en función de los conjuntos se define la base de reglas representada en la Tabla 1, según la información obtenida del conocimiento práctico del hardware.

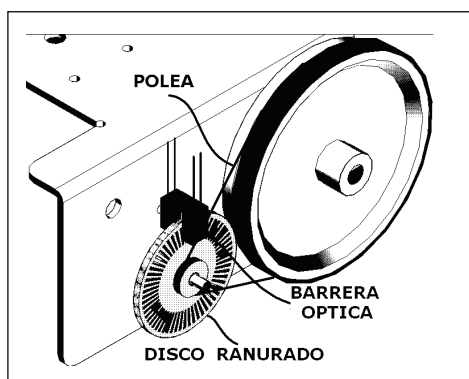


Figura 4. Disco ranurado, plasmado sobre un acetato transparente.

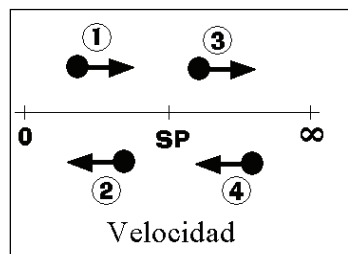


Figura 5. Representación de los cuatro posibles estados de la variable velocidad, producto de combinar los diferentes conjuntos de los dominios en los universos de discurso. Velocidad baja con tendencia a subir, velocidad alta con tendencia a subir, velocidad baja con tendencia a bajar y velocidad alta con tendencia a bajar.

$$\text{Error}(k) = \text{SP} - \text{Velocidad actual}(k)$$

$$\text{Integral de error}(k) = \text{Integral de error}(k-1) + \text{error}(k) \cdot \Delta t \quad (1)$$

Definición de las variables dependientes de la velocidad que configuran los universos de discurso del controlador CCRB. K indica el tiempo de muestreo discreto.

Ahora bien, la metodología para el diseño de un CCRB indica que se debe realizar una transformación de la información contenida en la base de reglas a una estructura bivaluada de forma simbólica como aparece en la Tabla 2. En esta tabulación se ubica una columna por cada conjunto de cada universo de discurso, más un conjunto de columnas que representan la codificación de la variable de salida. Energía de actuación para este caso, conformada por la suma de las variables auxiliares UPg, ponderación grande, y UPp, ponderación pequeña. Adicionalmente, se disponen tantas filas como posibles combinaciones de conjuntos de diferentes universos pueda hacerse; en el control de velocidad con dos universos y en cada uno de ellos, dos conjun-

tos, se pueden establecer cuatro relaciones distintas.

Con la representación simbólica o tabla de verdad de las funciones UPg y UPp, se obtiene una representación booleana de las funciones. Por simple inspección es posible dar una solución para la salida virtual UPp. $UPp = Ep$, mientras que para UPg es necesario simplificar por algún método de simplificación booleana.

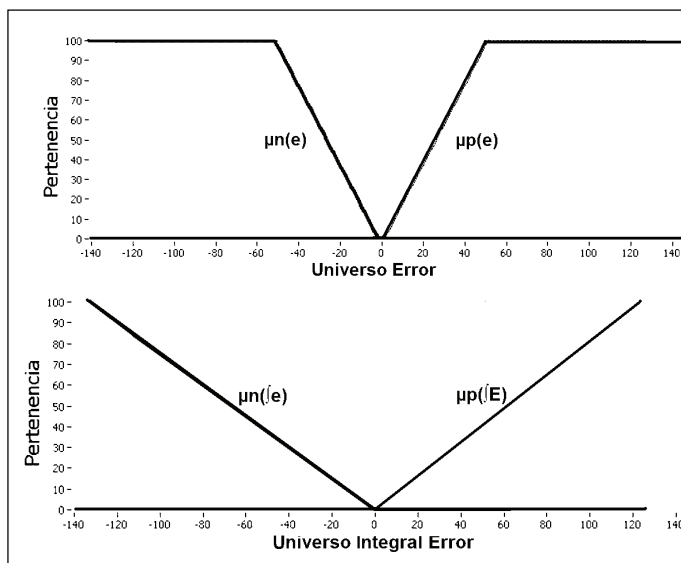


Figura 6. Universos Error, conjuntos error negativo, error positivo e Integral de error, conjuntos integral de error positiva e integral de error negativa.

Tabla 1. Base de reglas para control CCRB de velocidad del móvil agente jugador de fútbol

1	Si Error positivo e integral de error positivo	→	energía alta
2	Si Error positivo e integral de error negativo	→	energía pequeña
3	Si Error negativo e integral de error positivo	→	energía nula
4	Si Error negativo e integral de error negativo	→	energía nula

Tabla 2. Representación simbólica de la base de reglas para un control difuso de velocidad tipo CCRB, el recuadro señala las variables virtuales que codifican la energía asignada a los motores.

Ep	En	fEp	fEn	UPg	UPp
0	1	0	1	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	1	1

Usando el método gráfico de mapas de Karnaugh (Wakerly, 1999), se obtienen varias funciones lógicas que dan solución a la variable UPg. La escogencia de una asociación no es inmediata, la selección de la mejor solución se realiza en base a los resultados obtenidos en simulación para cada función resultante.

Finalmente, para obtener un valor continuo que sea la salida del controlador difuso, una vez halladas las funciones boo-

leanas para las salidas virtuales del sistema, es necesario realizar una generalización de dichas ecuaciones reemplazando el operador lógico OR por la s-norma máximo, el operador lógico AND por la t-norma mínimo, y el complemento Digital por el operador complemento difuso $1 - \mu_X(x)$ (Driankov, 1999), obteniendo las funciones continuas de la Ecuación 2.

$$UP_p = E_p \quad y,$$

$$UP_g = \min[\mu_p(e), (1 - \mu_n(e))] \quad (2)$$

Funciones continuas que constituyen la salida de actuación del controlador de velocidad.

La salida total del controlador difuso CCRB de velocidad, ψ , es la suma ponderada de las variables generalizadas UP_p y UP_g según se muestra en la ecuación 3. β es una constante que define el peso de cada componente, varía entre 0 y 1.

$$\psi = \beta * UP_p + (1 - \beta) * UP_g \quad (3)$$

La respuesta del controlador aplicado a un sistema de primer orden de la forma $2/(s+2)$ y a una entrada escalón unitario de magnitud 100 se muestra en la Figura 7. El controlador exhibe un tiempo de estabilización de 50 unidades sin error de estado estacionario.

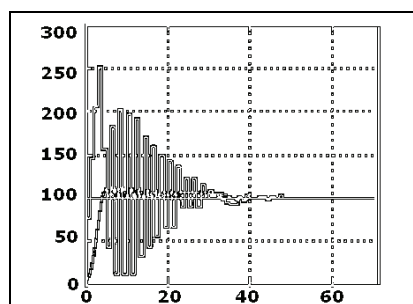


Figura 7. La señal oscilante de baja amplitud describe la velocidad controlada en ciclo cerrado del móvil, mientras que la señal amortiguada de mayor amplitud, describe el aporte energético del controlador a los motores del robot móvil.

Por otro lado, el sistema de comunicaciones que enlaza el sistema de control y el robot jugador utiliza un protocolo implementado a nivel físico sobre RS232, la descripción de la trama de comunicaciones se observa en la Figura 8.

Finalmente, el algoritmo que implementa el robot móvil para desarrollar la tarea de pateo utiliza una máquina de 5 estados, como se observa en la Figura 9, el primero se denomina estado ask o de pregunta, en este punto del flujo de datos se evalúa las condiciones iniciales que se deben cumplir para posibilitar la entrada de la máquina a un estado distinto, por ejemplo, se revisan las banderas de final de conteo para el tiempo de impulso de rodillo, o la bandera de petición de activación de un pateo por parte del usuario. Este es el estado más costoso computacionalmente y se puede considerar como sub estado de otros procesos. El segundo estado

de la tarea general es ready, donde el móvil se encuentra con el brazo de pateo levantado, tiene los motores tanto de la leva como del rodillo apagados, y se encuentra listo para patear la bola. El tercer estado es adj, allí el control ubica el brazo en una posición contraria a la cual se detecta, así por ejemplo, si el brazo está en posición arriba, el estado enciende el motor de la leva para llevarlo a la posición abajo, la operación termina cuando se detecta el cambio en la posición del brazo. El cuarto estado en que se puede encontrar el proceso de patear la bola, es el denominado accel, es donde el rodillo se enciende y permanece en dicha condición hasta que un tiempo fijo transcurre, delay de impulso, tiempo en el cual se garantiza que la velocidad del rodillo ha alcanzado una velocidad máxima. Por último, se tiene el estado de pateo, que es aquel en el cual, se colocan nuevamente a cero las condiciones iniciales porque se asume que para el momento cuando el proceso general haya alcanzado dicho estado, las tareas efectuadas con anterioridad habrán dado como resultado la transmisión de impulso a la bola.

Cabecera			
AA	Dirección móvil(4 bits)	Sin función(3 bits)	Tipo trama(1 bit)
Byte1	Byte 2		
Carga útil			
Si es una trama de configuración (tipo de trama =0)			
Pend e(4bits)	Pend int e(4bits)	mot a diagnt(1bit)	sinf(3bits)
Byte 3		Byte 4	
Si es una trama de operación (tipo de trama =1)			
set point1(4bits)	set point2(4bits)	sin_f(4bits)	pateo(1bit)
Byte 3		Byte 4	

Figura 8. Esquema de la estructura de las tramas de control enviadas desde la estación control de juego hacia cada uno de los móviles en la cancha.

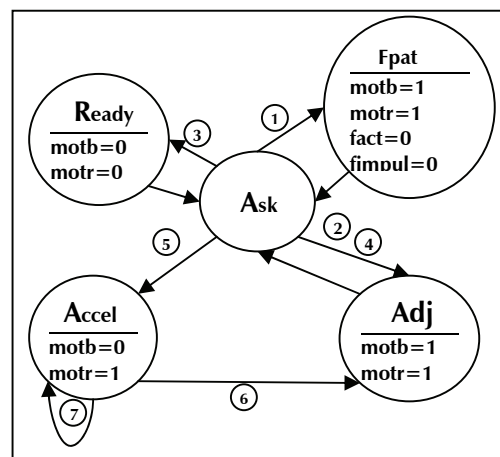


Figura 9. Máquina de estados finitos que controla el sistema de pateo en el robot móvil.

Las funciones que determinan las transiciones entre estados de la máquina que controla el pateo de la bola de los agentes móviles se tabulan en la Tabla 3.

Diseño del sensor de posición

La etapa de detección global por video de cada uno de los

elementos de juego, es el primer paso en la solución del problema, generar trayectorias que controlen el movimiento de los robots móviles en una competición robo-futbolística. El objetivo de cada uno de los jugadores robot es alcanzar una pelota en movimiento y llevarla a la portería contraria dentro de su campo de referencia.

Tabla 3. Funciones de transición de estados del diagrama de la Figura 9.

1	<i>fact and !arm_up</i>
2	<i>!fact and !arm_up</i>
3	<i>!fact and arm_up</i>
4	<i>fact and arm_up and !impul</i>
5	<i>fact and arm_up and !impul</i>
6	<i>delayimpul</i>
7	<i>!delayimpul</i>

Soluciones orientadas a obtener el mínimo costo computacional se encuentran en los trabajos presentados por los equipos que participan año tras año en las competiciones "The Robot World Cup Soccer Games and Conference". Por ejemplo, dentro de los enfoques que se abordaron en la versión 2001 de la liga F180 (Birk, 2001), se encuentra el trabajo desarrollado en (Costa, 2001) de la universidad de Porto, un sistema de visión que realiza la segmentación a través de la clasificación difusa del color para cada píxel, asociándolos en grupos contiguos para alimentar un conjunto de filtros Kalman sintonizados a la dinámica de cada clase de objeto. El equipo CM-Dragons por su parte, en (Browning, 2001) describe un sistema de posición global basado en segmentación de píxeles que usa tablas de búsqueda en 3 dimensiones en el espacio de color YUV. El equipo FU-Fighters (Rojas, 2001) describe como enfrentar el problema de percibir el efecto de una acción de control 150ms después de ser ejecutada, a través de una estrategia de predicción del movimiento de los robots para unos pocos cuadros en el futuro, entrenando una red neuronal de propagación hacia delante para tal efecto. Otro ejemplo eficiente de un sistema de alta velocidad de reconocimiento de posición, orientación e identificación de móviles, lo presenta el equipo Owaribito (Hibino, 2001), el algoritmo trabaja sobre imágenes en blanco y negro, cada agente es identificado por un patrón con forma específica, estos patrones son buscados a lo largo de toda la imagen en todas las combinaciones de desplazamiento y rotación posibles, aplicando una máscara cuadrícula de forma circular, la cual, al ser decodificada asocia un identificador y una orientación, a partir de la plantilla reconocida se calcula la posición del grupo de píxeles analizado, si la plantilla es reconocida. Finalmente, la Universidad de Girona, España (De la Rosa, 2001), presenta en su equipo Rogi, la implementación de un sensor en hardware desarrollado en lógica programable sobre una FPGA.

Estos trabajos previos junto a la propuesta del equipo RoboRoos (Wyeth, 2000), y a su implementación adaptada en (Molina y Cardenas, 2005), donde se presenta un procesamiento desarrollado en Matlab con un costo computacional de 200ms y segmentación de formas por umbralización de color en el espacio RGB, dan un punto de partida

para contemplar los principales desafíos que se enfrentará en el desarrollo de un sensor eficiente y adecuado para la tarea de identificar tres elementos de juego, dos robots móviles y una pelota.

El reto planteado en este trabajo fue superar el tiempo de captura de la cámara web de 30 cuadros por segundo en resolución de 320 X 240 píxeles a 24 bits, 1,6 millones de colores, lo que quiere decir que cada 33ms se puede disponer de una nueva imagen proveniente del sensor CCD de la cámara. Este lapso es el tiempo máximo que se dispone para ejecutar el control de juego, incluido los algoritmos de planeación de trayectorias y conformación de roles de juego.

El algoritmo que se propone en la Figura 10, se implementa en la plataforma de desarrollo nativa de Windows, .Net C#.

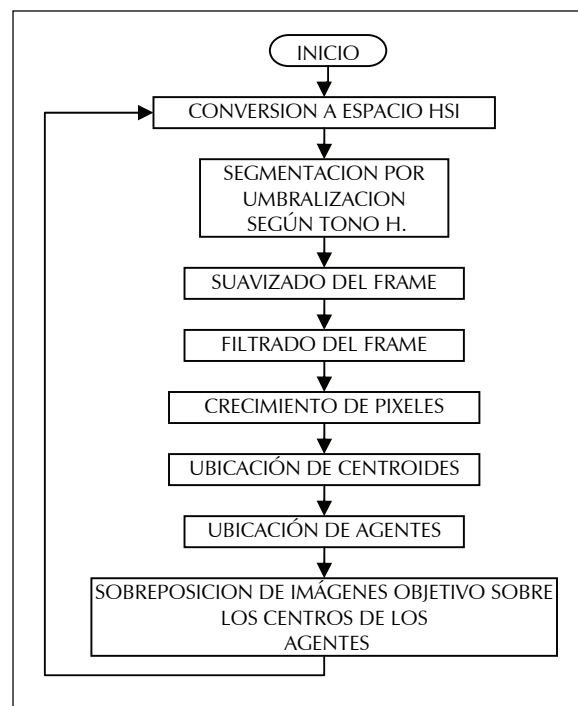


Figura 10. Diagrama ASM del algoritmo de posicionamiento de agentes sobre el campo de juego.

El primer paso en el proceso de sensar la posición de objetos sobre el campo de juego es la captura de un cuadro estático dentro de una corriente de video, dada las restricciones de la cámara Web se logra capturar imágenes de hasta 320 X 240 píxeles a una tasa de 15 a 30 cuadros por segundo.

Con relación al cambio de espacio de color, se decidió utilizar HSI porque en él se separan las componentes de tonalidad, saturación de color e intensidad, logrando con ello independizar los colores de la intensidad de luz. En las Figuras 11 y 12, se puede apreciar como los histogramas de las componentes RGB del campo de juego cambian radicalmente en presencia del offset sumado por la luz día, mientras que la componente tono, huge, se comporta estable, ya que como se sabe en un ambiente no controlado sujeto a las variaciones normales de la luz día, el flujo luminoso puede lle-

gar a cambiar en hasta 5000 Luz con un valor promedio de 10000 Luz (Molina y Cardenas, 2005).

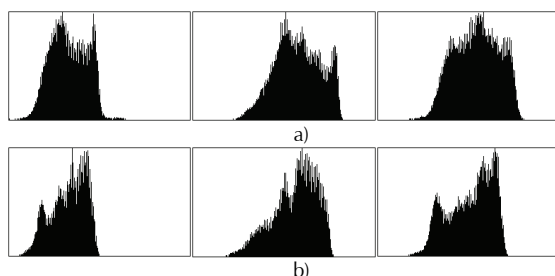


Figura 11 a) Histogramas RGB del campo de juego con objetos de color con luz artificial, b) Histogramas RGB del campo de juego con objetos de color con luz artificial + luz día.

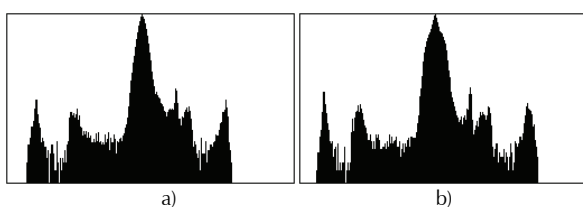


Figura 12. a) Histograma Hugu del campo de juego con objetos de color, con luz artificial, b) Histograma Hugu del campo de juego con objetos de color, con luz artificial + luz día.

La imagen que se muestra en la Figura 13, muestra el resultado de convertir la imagen inicial³, a la componente Hugu vista en escala de grises.

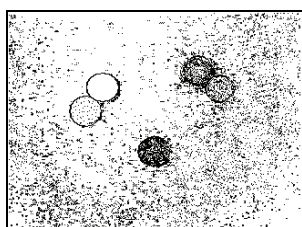


Figura 13. Imagen convertida al espacio de color HIS, componente Hugu vista en escala de grises.

Es posible plantear en forma algorítmica la conversión del espacio RGB para cada píxel de la imagen al espacio HSI en procura de disminuir el costo computacional como se observa en la Figura 14.

El tiempo gastado en el proceso de convertir la imagen a su componente de tonos es de 31ms en un procesador Centri-no de 1.5GHz, tiempo pequeño si se lo compara con los 240 ms que se toma realizar el mismo proceso en LabView.

A este respecto una optimización al proceso fue realizar una selección de píxeles antes de iniciar su tratamiento en HSI,

³ Dos círculos de color amarillo y azul respectivamente, unidos tangencialmente para identificar al móvil AMC, ubicados en la parte superior izquierda de la Figura 13; dos círculos de color rojo y morado unidos tangencialmente para identificar al móvil MCU, ubicados en la parte superior derecha de la Figura 13; y un círculo de color naranja de menor tamaño para identificar la bola de juego, en la parte central inferior de la Figura 13.

se trata de no operar la conversión sobre los píxeles que estén en el rango RGB del fondo del campo de juego, ya que la mayoría de píxeles pertenecen a esta clase. Si se comparan los histogramas RGB del campo de juego vacío y del campo de juego con objetos de color en la Figura 15, se observa que los rangos se superponen y por lo tanto se requiere hallar los histogramas de cada color por separado, extrayendo porciones de la imagen donde se encuentre cada objeto y realizando el respectivo histograma. Este análisis previo fue realizado en LabView y se presenta en la Figura 16.

```

min = Math.Min(Math.Min(r, g), b);
max = Math.Max(Math.Max(r, g), b);
delta = max - min;
Luminance = (max + min) / 2;
if (delta == 0) {
    Hue = 0;
    Saturation = 0.0;
}
else {
    Saturation = (Luminance < 0.5) ?
        (delta / (max + min)) :
        (delta / (2 - max - min));
    del_r = (((max - r) / 6) + (delta / 2)) / delta;
    del_g = (((max - g) / 6) + (delta / 2)) / delta;
    del_b = (((max - b) / 6) + (delta / 2)) / delta;
    if (r == max)
        hue = del_b - del_g;
    else if (g == max)
        hue = (1.0 / 3) + del_r - del_b;
    else
        hue = (2.0 / 3) + del_g - del_r;
    if (hue < 0)
        hue + = 1;
    if (hue > 1)
        hue - = 1;
    Hue = hue * 360;
}

```

Figura 14. Conversión algorítmica al espacio de color HSI.

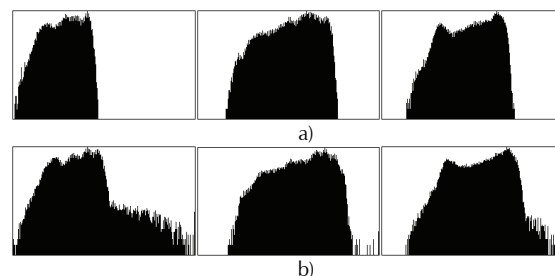


Figura 15. a) Histogramas RGB del campo de juego sin objetos en escala logarítmica, b) Histogramas RGB del campo de juego con objetos de color en escala logarítmica para resaltar detalles pequeños de la imagen.

De los datos obtenidos en el análisis de color para cada objeto y el campo de juego, se evidencia que existe una superposición para las tres componentes R, G y B en los objetos rojo y morado. Con respecto al fondo verde se puede ver que si se restringe los rangos de la componente R y G a 62-103 y 126-181, ya no existe cruce con los objetos rojo y morado.

El problema de una segmentación RGB estriba en la variabi-

lidad de los rangos con respecto a pequeños cambios en la intensidad de luz que cubre a los objetos. Sin embargo, limitando razonablemente los rangos en los cuales se puede encontrar los pixeles pertenecientes al fondo del campo de juego y excluyéndolos en un prefiltrado, se logra eliminar hasta un 50% el coste computacional, logrando aumentos de la velocidad de conversión a HSI de 15ms.

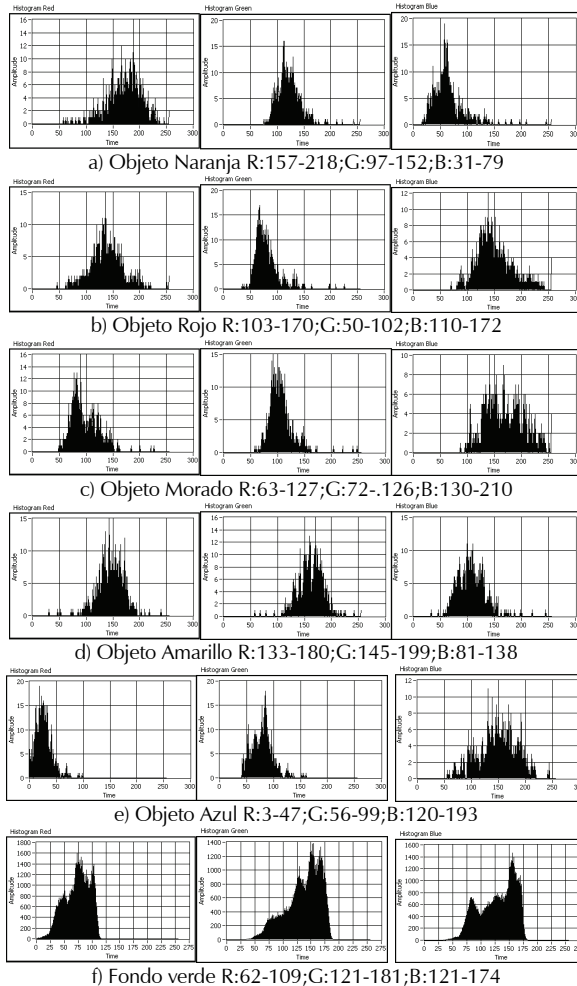


Figura 16. Análisis de las componentes RGB de cada objeto de color

Una vez optimizado el proceso de conversión a HSI, se efectúa un proceso de suavizado sobre todos los vecinos de cada pixel, la operación es un promedio de los valores de cada vecino y la asignación de este resultado al píxel bajo prueba como se ve en la Figura 17.

A continuación se realiza la segmentación por multiumbralización basada en los rangos que posee cada tonalidad correspondiente a cada objeto en juego. En la Figura 18, se presenta cada uno de los rangos sobre los respectivos histogramas Hugué en el espacio de llegada HSI y el resultado de la segmentación en esta fase del algoritmo.

Posteriormente, como consecuencia de tener presente una componente remanente de ruido de fondo después del proceso de suavizado, es preciso aplicar un filtro donde si un píxel

no es vecino de un número de pixeles mínimo, se elimina. La Figura 19 muestra el efecto del filtrado.

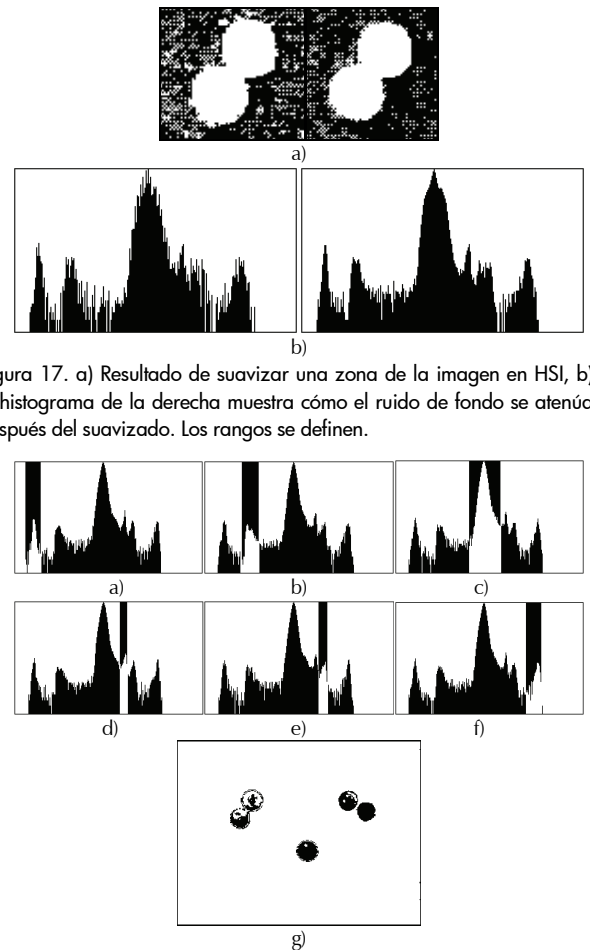


Figura 18. Rangos diferenciados de cada tonalidad presente en el campo de juego a) objeto naranja, b) objeto amarillo, c) fondo, d) objeto azul, e) objeto rojo, f) objeto morado, g) resultado de la segmentación por tonalidad

Como última etapa antes de calcular la ubicación de los móviles, se realiza el proceso de crecimiento de pixeles que consiste en fijar el color de un píxel a todos sus vecinos, si este es distinto de blanco. El objetivo del filtro es adicionar estabilidad a la medición del sensor, el resultado se muestra en la Figura 20.

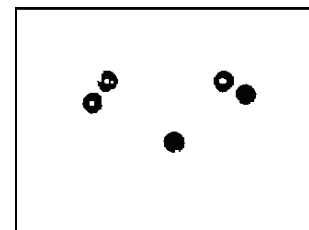


Figura 19. Resultado del filtro para eliminación de ruido

La posición de los móviles se obtiene a partir de los centroides de cada clase de color, mediante la siguiente fórmula:

$$\begin{aligned}
 PosX_{clasek} &= \frac{\sum_{j=1}^{Alto\ Imagen} \sum_{i=1}^{Ancho\ Imagen} i; \forall pixel_{ij} \in clasek}{n} \\
 PosY_{clasek} &= \frac{\sum_{j=1}^{Alto\ Imagen} \sum_{i=1}^{Ancho\ Imagen} j; \forall pixel_{ij} \in clasek}{n} \\
 n &= \text{número_pixeles} \in clasek
 \end{aligned} \quad (3)$$

Posición de un centroide de clase k , el $pixel_{ij}$ es el pixel ubicado en la fila i columna j .

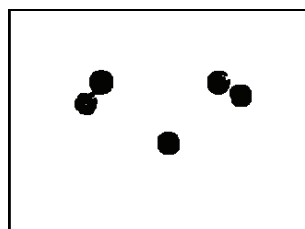


Figura 20. Resultado del crecimiento.

Finalmente se calcula un vector de posicionamiento para cada móvil, el móvil 1 es la unión de una bola de color rojo y una bola de color morado, mientras que el móvil 2 es la unión de una bola de color amarillo y una bola de color azul, la pelota de juego es la bola de color naranja. El vector se establece desde un centroide a otro con una dirección fija, la posición final del móvil es el centro del vector extendido como se observa en la Figura 21.

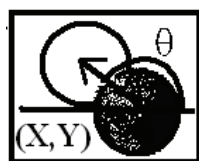


Figura 21. Posición final del móvil 1 (X,Y,Θ) .

El tiempo total que tarda la aplicación en procesar una imagen de la arena de juego en un procesador Centrino de 1.6 GHz, es de 66ms, la versión final de la aplicación se implementó junto con el control de juego en un procesador doble núcleo de 3GHz, donde el tiempo de computo de la posición de los agentes móviles y de la bola tarda 33ms. Tiempo que permite trabajar óptimamente con una cámara CCD de 30 cuadros por segundo.

Diseño del algoritmo de *path planning*

La sintonía del controlador para evasión de obstáculos y direccionamiento del móvil a una posición objetivo, es una tarea compleja porque los procesos se interfieren mutuamente. Para simplificar el ajuste de las formas de los conjuntos de cada controlador, así como su conjunto de reglas de inferencia difusa óptimo, se realiza el siguiente diseño. El enfoque consiste en tener un controlador actuando como principal, y otro controlador actuando como una perturbación del pri-

mero. En este caso, un controlador que esquiva obstáculos y lleva a un agente a una posición objetivo, se implementa como una combinación lineal de dos controladores, uno para la evasión de obstáculos CD1, el principal, y otro para direccionar el móvil hasta la posición de disparo de la bola, CD2, a través de la perturbación de los niveles energéticos que aporta el primer controlador para la evasión de obstáculos.

La idea central del diseño es lograr un controlador para la evasión de obstáculos que reaccione significativamente en las cercanías a uno de ellos, creando una diferencia de velocidad grande para las ruedas del móvil. En la medida que el agente se encuentre más lejos de un obstáculo, se genera niveles de actuación iguales y proporcionales a la distancia al obstáculo, de tal manera que el Agente Móvil Controlado (AMC), se acerque al obstáculo lo antes posible. En otras palabras, el controlador para la evasión de obstáculos, si fuera la única componente que afectara al agente móvil, lo llevaría siempre hasta el obstáculo que tenga en frente y al llegar a él, lo esquivaría buscando una línea de vista libre, para luego buscar un nuevo obstáculo y dirigirse a él. Se puede entender el objetivo de este controlador como un buscador de obstáculos lejanos y un evasor de obstáculos cercanos.

Por otro lado, la componente aportada por el controlador que direcciona el móvil al objetivo, afectará su trayectoria como una perturbación en la zona denominada lejana, como se muestra en la Figura 22, creando un desequilibrio en los valores de actuación de la combinación de ambos controladores, llevando al AMC a la posición de disparo, siempre que, como se ha explicado anteriormente, el controlador de evasión de obstáculos, tome valores de actuación iguales para ambas ruedas como resultado de encontrarse lejos de un obstáculo.

En la zona cercana al objetivo, la ponderación de los controladores se invierte y la componente que prima en la cinemática del móvil será la aportada por el controlador de dirección al objetivo, dado que el controlador de evasión, en ese mismo escenario, atenuará dramáticamente sus aportes de actuación.

La flexibilidad de poder atenuar o intensificar cualquiera de las dos componentes de actuación, evadir obstáculos o alcanzar un objetivo, se manipula a través de una función alfa que depende de la distancia al objetivo, como se muestra en la Figura 23. En este enfoque para el controlador de planeación de trayectoria, la salida de actuación total queda determinada por la ecuación (4).

$$V_{\text{final}} = \alpha(\text{Dist_objt})V_{CD1} + (1 - \alpha(\text{Dist_objt}))V_{CD2} \quad (4)$$

Aporte final del controlador de planeación de trayectorias con α variable.

El resultado entonces, es una estrategia híbrida de inteligencia artificial basada en controladores difusos independientes, que trabajan en paralelo, como consecuencia de ser implementa-

dos en un procesador de doble núcleo, y que resuelven un comportamiento multiobjetivo, llevar a un agente móvil hasta un objetivo mientras esquiva obstáculos en su trayectoria.

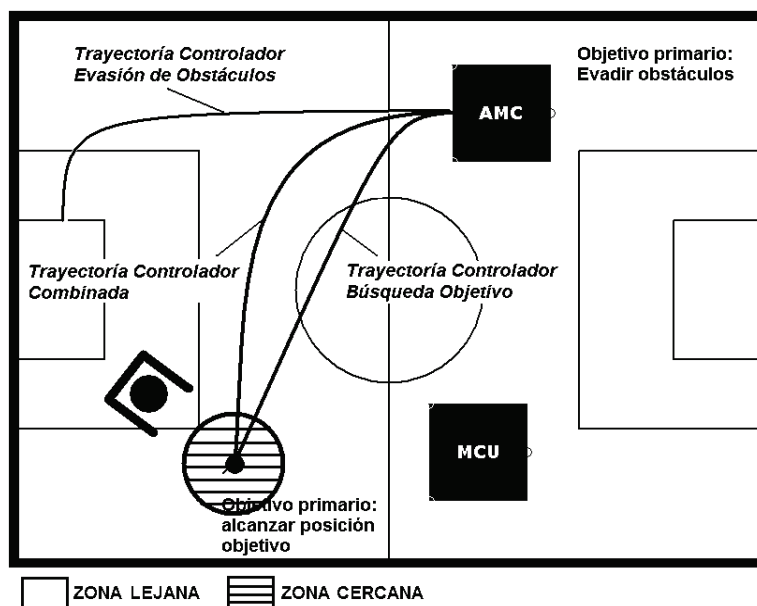


Figura 22. Definición de zonas de preponderancia de los Controladores Difusos CD1 y CD2, zona cercana y zona lejana.

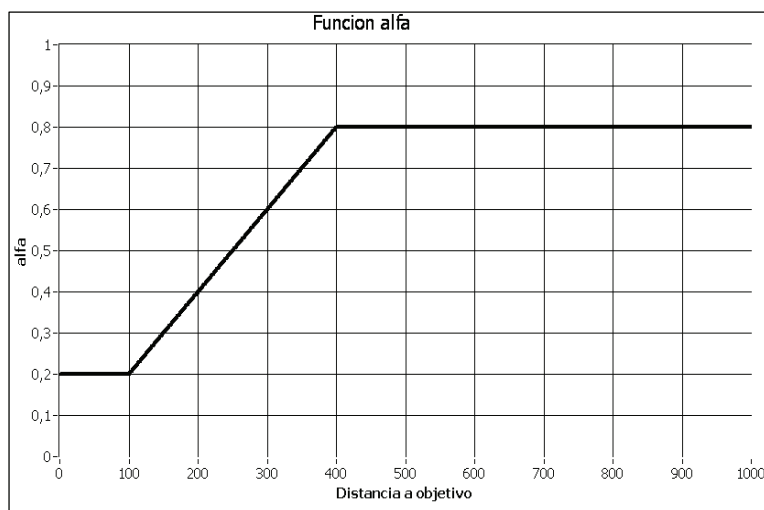


Figura 23. La magnitud alfa, en función de la distancia medida desde el móvil controlado al objetivo o posición de disparo, representa la ponderación de las componentes de actuación de los controladores de evasión de obstáculos y alcance de objetivo.

Controlador difuso que busca y esquivo obstáculos

A continuación se presenta en la Tabla 4, la base de reglas para la propuesta optimizada del controlador para exploración y evasión de obstáculos. Los conjuntos de entrada y salida con que se formulan las reglas se muestran en la Figura 24, también, en la Figura 25, se presentan las correspondientes superficies de control.

Controlador de dirección hacia posición objetivo

Este controlador actúa como una perturbación sobre el controlador de exploración y evasión de obstáculos. Cuando el agente se encuentra lejos del objetivo, esta componente es atenuada por la función alfa de ponderación, su aporte en el controlador está diseñado para desequilibrar la dirección del móvil y llevarlo al objetivo de tiro al arco. En esta condición, la componente dominante, evasión de obstáculos, si está lejos de un obstáculo, se encuentra llevando al móvil en línea recta, condición de fácil desequilibrio. En las cercanías a la posición de disparo, la componente que direcciona el móvil hacia el objetivo se magnifica por efecto de la función alfa, logrando llegar con precisión a una coordenada (x,y) específica, también como consecuencia de la atenuación en la componente de evasión de obstáculos.

A continuación se presenta la especificación de las reglas del controlador en la Tabla 5, basadas en los conjuntos de entrada y salida mostrados en la Figura 26. Las superficies de actuación para la velocidad de la rueda derecha e izquierda se presentan en la Figura 27.

Resultados experimentales

En la presente sesión se consolida y analiza los resultados de las distintas pruebas efectuadas a la plataforma Robosoccer para evaluar sus componentes más importantes, ellos son: el controlador de velocidad CCRB embebido en los agentes móviles, el sensor de posición por captura de video, y el controlador de trayectorias sobre la arena de juego.

Pruebas realizadas al sensor de velocidad

En la gráfica de la Figura 28, se observa el rendimiento del controlador difuso de velocidad, CCRB (Controlador con concreción basada en relaciones booleanas), implementado para cada rueda, en cada uno de los agentes móviles sobre su microcontrolador on board.

La captura de información del microcontrolador donde corre el algoritmo CCRB de velocidad, se efectúa cada 25ms desde la plataforma de control en el ambiente de diagnóstico. Las señales capturadas son: el error producido por la diferencia entre la velocidad fijada por la plataforma y la velocidad medida en el sensor de velocidad del móvil, el acumulado de error que integra en forma discreta la componente de error, la velocidad medida en tiempo de ejecución sobre la rueda a la cual se aplica el control, y la energía de actuación con que se excita el motor correspondiente.

En la Figura 28, entre la muestra 1400 y 1680, se aprecia la

acción del controlador, la velocidad de la rueda se mantiene constante, alrededor de 25 unidades, ante perturbaciones externas, detención forzosa de la rueda, incrementando sú-

bitamente los valores con que es excitado el motor generador del movimiento de la rueda.

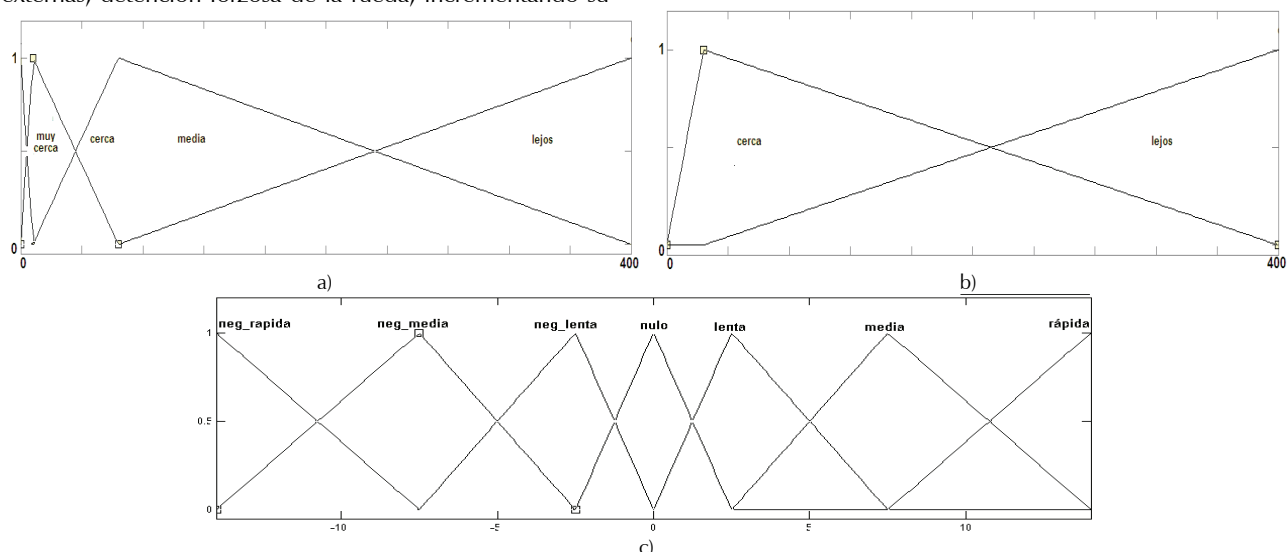


Figura 24. Universos y conjuntos de entrada y salida para el controlador difuso de evasión de obstáculos: a) Universo de discurso, distancia sensor derecho, izquierdo. b) Universo de discurso, distancia a objetivo. c) Universo de discurso de salida, distancia a sensor derecho, distancia a sensor izquierdo.

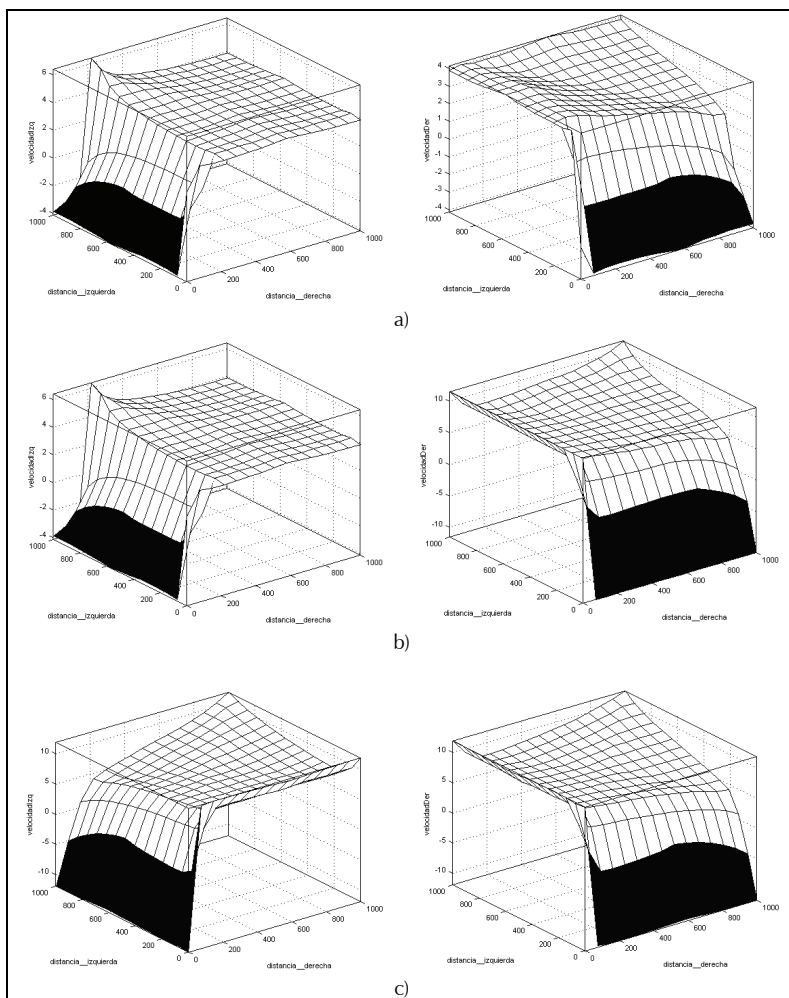


Figura 25. Superficies de control para el controlador de evasión de obstáculos optimizado. a) Superficies de actuación derecha e izquierda para una distancia a objetivo de 100 píxeles. b) Superficies para una distancia a objetivo de 500 píxeles. c) Superficies para una distancia a objetivo de 900 píxeles. El método de defusificación utilizado es centro de área.

Si no se presentan perturbaciones externas, la energía de excitación es una amplificación de la componente de acumulado de error. Ahora bien, como la integral de error puede establecerse en cualquier valor mientras el error fluctúa entre valores pequeños tanto positivos como negativos, el controlador puede llevar al sistema a cualquier velocidad con un error muy cercano a cero, tal y como se observa entre las muestras 770 y 980.

En resumen, la condición de error estacionario cero se cumple, tal y como se espera para un controlador PI, proporcional e integrativo, y el tiempo de establecimiento como se ve a partir de la muestra 1360, es 2 muestras, 50ms.

Pruebas realizadas al sensor de posición por video

Para determinar la precisión de las coordenadas generadas en el sensor de posición por video, se realizó un registro de las magnitudes calculadas para 1450 ciclos de control de la plataforma Robosoccer. Las ocho señales se muestran en la Figura 29, y corresponden a la coordenada X, Y y orientación del móvil AMC,

coordenada X, Y y orientación del móvil MCU (Móvil controlado por usuario), y coordenada X e Y de la bola de

juego.

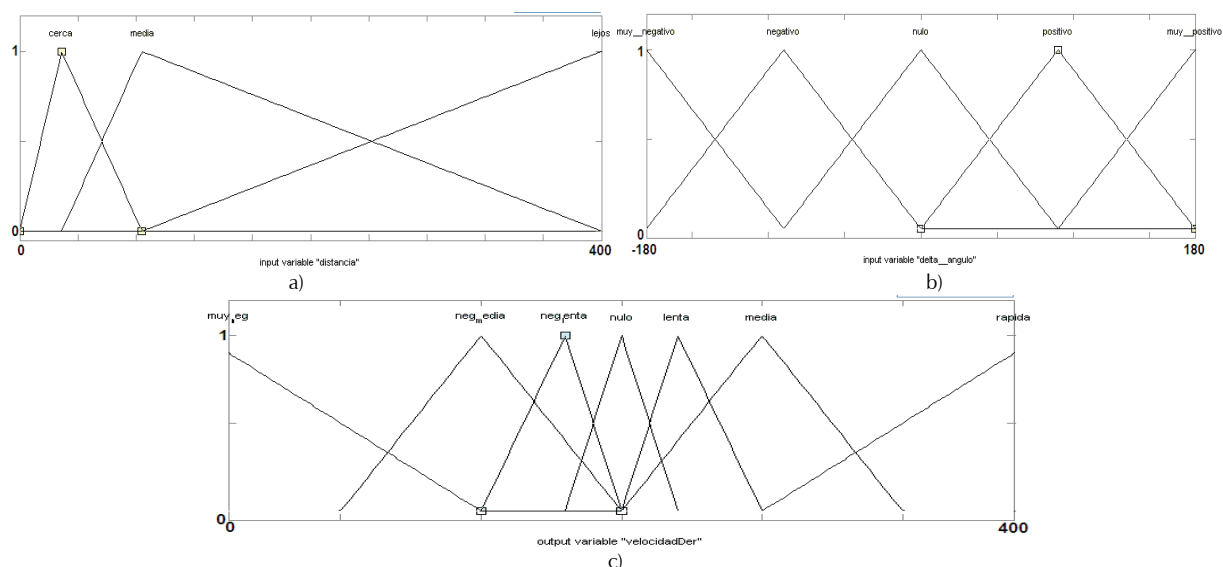


Figura 26. a) Conjuntos universo de discurso distancia a objetivo para el controlador difuso que lleva el agente hasta el objetivo, b) Conjuntos universo de discurso delta de ángulo entre la orientación del agente móvil y el vector formado entre su posición y la posición del objetivo para el controlador difuso que lleva el agente hasta el objetivo. c) Conjuntos, universo velocidad de salida de cada una de las ruedas del agente móvil, velocidad izquierda, velocidad derecha.

Tabla 4. Tabulación de la base de reglas tipo mamdani para un controlador difuso que busca y esquiva obstáculos.

N	Entrada										Salida	
	Sensor Izquierdo				Sensor Derecho				Distancia Objetivo		Velocidad	
	M_cer	cer	med	lej	m_Cer	cer	med	lej	Cer	lej	Izq	Der
1	X				X					X	R	N R
2	X					X				X	R	N R
3	X						X			X	R	N R
4	X							X		X	R	N R
5		X			X					X	N R	R
6		X				X				X	R	N R
7		X					X			X	R	N R
8		X						X		X	R	N R
9			X		X					X	N R	R
10			X			X				X	N R	R
11			X				X			X	M	M
12			X					X		X	M	L
13				X	X					X	N R	R
14				X		X				X	N R	R
15				X			X			X	L	M
16				X				X		X	R	R
17	X				X				X		L	N L
18	X					X			X		L	N L
19	X						X		X		L	N L
20	X							X	X		L	N L
21		X			X				X		N L	L
22		X				X			X		L	N L
23		X					X		X		L	N L
24		X						X	X		L	N L
25			X		X				X		N L	L
26			X			X			X		N L	L
27			X				X		X		L	L
28			X					X	X		L	N
29				X	X				X		N L	L
30				X		X			X		N L	L
31				X			X		X		N	L
32				X				X	X		L	L

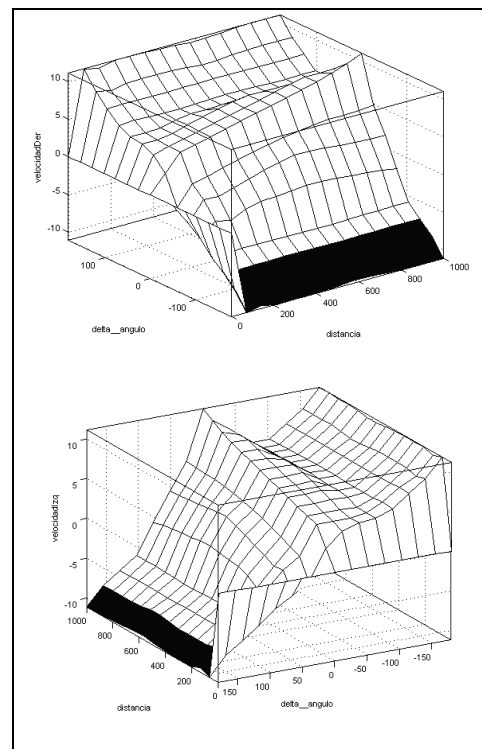


Figura 27. Superficie de control velocidad derecha e izquierda, respectivamente, del controlador difuso optimizado para alcanzar el objetivo. El método de defusificación utilizado es centro de área.

Tabla 5. Representación simbólica de la base de reglas del controlador difuso para alcanzar objetivo y perturbar la evasión de obstáculos, optimizado.

No	Entradas								Salidas	
	Distancia			Delta de ángulo					Velocidad	
	cer	med	lej	m_neg	neg	nulo	pos	M_Pos	Izq	Der
1	X			X					R	M_N
2	X				X				M	N_M
3	X					X			L	L
4	X						X		N_M	M
5	X							X	M_N	R
6		X		X					R	M_N
7		X			X				M	N_M
8		X				X			M	M
9		X					X		N_M	M
10		X						X	M_N	R
11			X	X					R	M_N
12			X		X				M	N_M
13			X			X			R	R
14			X				X		N_M	M
15			X					X	M_N	R

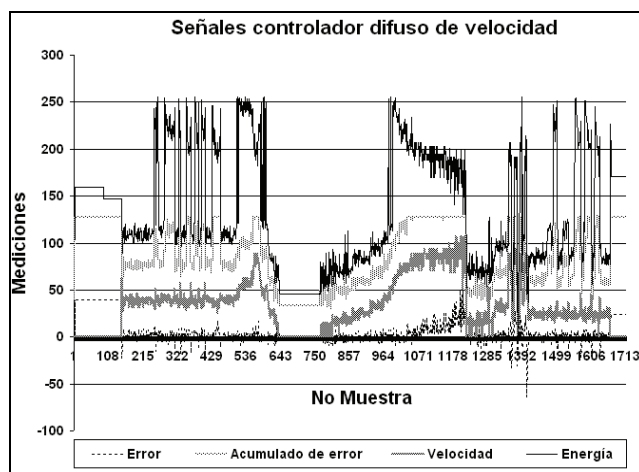


Figura 28. Señales internas del controlador de velocidad CCRB implementado en el microcontrolador de los agentes móviles para fijar la velocidad de cada una de sus ruedas.

La desviación estándar de los valores para cada coordenada espacial, en el intervalo de la muestra 70 a la muestra 310, donde los elementos de juego se encuentran estáticos, se presenta en la Tabla 6.

Tabla 6. Desviaciones estándar para las coordenadas espaciales generadas en el sensor de posición por video para las muestras tomadas en el rango 70 a 310.

Desviaciones estándar							
Móvil 1 X	Móvil 1 Y	Móvil 2 X	Móvil 2 Y	Bola X	Bola Y	Móvil 1 áng	Móvil 2 áng
0,491	0,504	0,602	0,189	0	0	1,305	1,535

Los valores tabulados dan una medida de precisión de las mediciones efectuadas, un valor de dispersión con respecto al valor promedio o valor real. Como se observa en la tabla 6, los valores están muy cercanos al valor promedio, en ningún caso, por encima de 2 unidades. Así mismo, la variación máxima en todas las señales para el intervalo de análisis es también de 2 unidades, ± 2 píxeles para las coordenadas

cartesianas y ± 2 grados, para la orientación de los móviles.

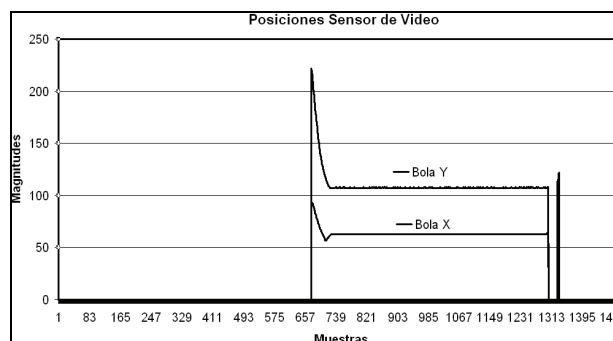
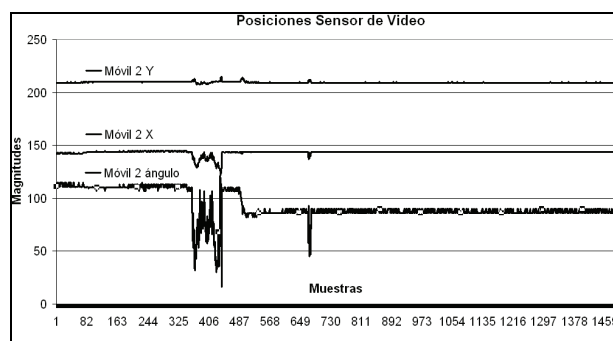
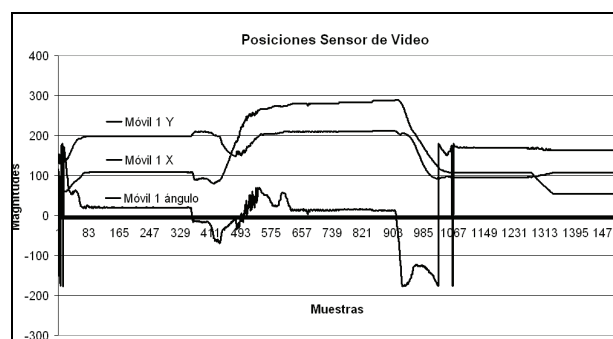


Figura 29. Señales generadas en el sensor de posición por video: coordenadas x e y de los móviles, medidas en píxeles, y orientación, medida en grados; Coordenadas x e y de la bola de juego medidas en las mismas unidades que los móviles.

Estos valores indican que el sensor de posición por video posee una precisión adecuada para los algoritmos de control, también, describen la magnitud de ruido que hay que introducir al sistema de control de juego para realizar simulaciones lo suficientemente cercanas a la realidad. En cuanto a la velocidad del sensor, las muestras se tomaron cada 47ms, un procesamiento cercano a 20 cuadros por segundo.

Pruebas realizadas al control de trayectoria

Con el ánimo de validar el simulador como una herramienta útil en la sintonización del controlador para planeación de trayectorias y verificar el cumplimiento del objetivo del AMC, seguir trayectorias que lleven a solucionar ambientes diversos donde encuentre obstáculos en su camino hacia la posición de tiro al arco contrario, se realiza una serie de corridas del

simulador y de la plataforma real, bajo los mismos parámetros de configuración en distintos escenarios tanto estáticos como dinámicos.

La comparación entre las trayectorias simuladas con respecto a las generadas en la plataforma, se puede realizar tanto cualitativamente como cuantitativamente, teniendo en cuenta la forma de las gráficas capturadas, ver Figura 30, y el tiempo en el cual son trazadas, respectivamente. Es importante aclarar que si bien la magnitud de la imagen emulada en el simulador es de 800×600 píxeles, y la tomada en la plataforma por el sensor de video es de 320×240 píxeles, todos los elementos de juego están escalizados y conservan una relación de 1:2,5.

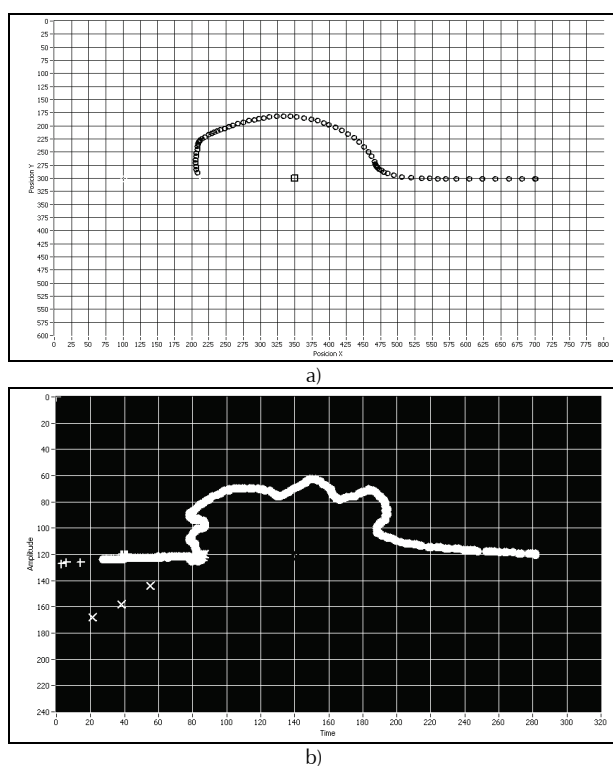


Figura 30. a) Se observa una gráfica de la trayectoria descrita por el agente simulado. b) Se observa una trayectoria real bajo las mismas condiciones iniciales del simulador.

Escenario Estático

Análisis del comportamiento del agente móvil en diferentes escenarios estáticos, en presencia de obstáculos fijos (móvil obstáculo, barreras virtuales superpuestas alrededor de la bola de juego para evitar colisionar con ella, y límites de la arena de juego), variando los niveles alto y bajo de la función alfa (componente de ponderación de los controladores).

Escenario 1

Para realizar esta prueba se escogió una configuración característica en la disposición de los elementos de juego: AMC, MCU y bola. Las trayectorias descritas por el AMC son sensibles a las componentes de perturbación discutidas en la sección anterior, el escenario mostrado en la Figura 31,

presenta la particularidad de posibilitar cambios considerables en la ruta seguida por el móvil, girar a derecha o a izquierda al llegar al MCU, a pesar de poseer una base de reglas fija que fuerza una misma tendencia bajo condiciones iguales del ambiente de juego.

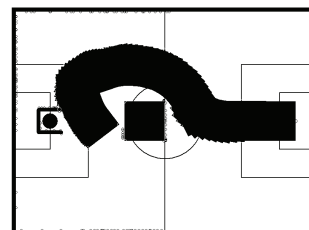


Figura 31. Escenario de prueba 1. Posición móvil controlado: simulador (700,300) píxeles, arena de juego (280,120). Posición móvil obstáculo: simulador (350,300), arena de juego (140,120). Posición objetivo: simulador (100,300), arena de juego (40,120).

A continuación en la Tabla 7, se presenta el registro respectivo del tiempo de la trayectoria planeada por el controlador difuso, variando la componente de ponderación que determina como se combinan los controladores para evasión de obstáculos y búsqueda de objetivo.

Tabla 7. Tiempo promedio de las corridas del simulador y de la plataforma de control para el escenario 1.

Corrida	Arena de juego	Simulador
1	tiempo = 12656	tiempo = 5456
2	tiempo = 15089	tiempo = 4293
3	tiempo = 20949	tiempo = 5211
Tiempo promedio	16231	4986
Alfa=0.48		
1	tiempo = 11577	tiempo = 5806
2	tiempo = 11176	tiempo = 4752
3	tiempo = 9907	tiempo = 5865
Tiempo promedio	10886	5474
Alfa=0.58		
1	tiempo = 9922	tiempo = 5052
2	tiempo = 10197	tiempo = 4966
3	tiempo = 11910	tiempo = 7026
Tiempo promedio	10676	5681
Alfa=0.68		
1	tiempo = Colisión	tiempo = 4372
2	tiempo = Colisión	tiempo = 7493
3	tiempo = Colisión	tiempo = 7623
Tiempo promedio	NA	6496
Alfa=0.78		
1	tiempo = Colisión	tiempo = Infinito
2	tiempo = Colisión	tiempo = Infinito
Tiempo promedio	NA	NA
Alfa=0.88		

Escenario 2

A continuación en la Tabla 8, se presenta el tiempo promedio de las trayectorias descritas por los agentes móviles y la bola, en distintas corridas del simulador y de la plataforma de control, variando el parámetro alfa para el escenario de prueba 2.

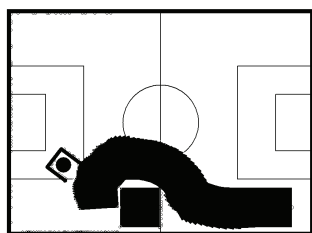


Figura 32. Escenario de prueba 2. Posición móvil controlado: simulador (700,525), arena de juego (280,210). Posición móvil obstáculo: simulador (350,525), arena de juego (140,210). Posición objetivo: simulador (147.5,412.5), arena de juego (59,165).

Tabla 8. Tiempo promedio de las corridas del simulador y de la plataforma de control para el escenario 2.

Corrida	Arena de juego	Simulador
1	tiempo = 10434	tiempo = 5120
2	tiempo = 11110	tiempo = 5259
3	tiempo = 10964	tiempo = 5202
Tiempo promedio	10836	5193
Alfa=0.48		
1	tiempo = 8761	tiempo = 6752
2	tiempo = 14167	tiempo = 7134
3	tiempo = 13635	tiempo = 9065
4	tiempo = 13063	
Tiempo promedio	12406	7650
Alfa=0.58		
1	tiempo =	tiempo = Infinito
Tiempo promedio	NA	NA
Alfa=0.68		

Escenario Dinámico

Análisis del comportamiento del agente móvil en un escenario dinámico. El ambiente esta formado por un obstáculo móvil MCU, (en dirección 0 grados, con una velocidad de 3.6cm/s), obstáculos fijos (barreras virtuales sobrepuestas alrededor de la bola de juego para evitar colisionar con ella, y límites de la arena de juego), variando los niveles alto y bajo de la función alfa (componente de ponderación de los controladores), ver Figura 33.

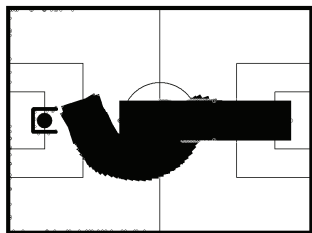


Figura 33. Escenario dinámico. Posición móvil controlado: simulador (700,300), arena de juego (280,120). Posición inicial móvil obstáculo: simulador (350,300), arena de juego (140,120), dirección avance: 0 grados, velocidad: 2 (unidades del sensor de velocidad del móvil, rango 1-15, velocidad min-max), Posición objetivo: simulador (100,300), arena de juego (40,120).

A continuación en la Tabla 9, se presenta el tiempo promedio de las trayectorias descritas por los agentes móviles y la bola, en distintas corridas del simulador y de la plataforma de

control, variando el parámetro alfa para el escenario de prueba dinámico.

Tabla 9. Tiempo promedio de las corridas del simulador y de la plataforma de control para el escenario dinámico, variando el valor de alfa.

Corrida	Arena de juego	Simulador
1 Alfa=0.48	tiempo = Colisión	tiempo = 4749
2 Alfa=0.58	tiempo = Colisión	tiempo = 5966
3 Alfa=0.68	tiempo = 9625	tiempo = 6538
4 Alfa=0.78	tiempo = 10000	tiempo = 8092
5 Alfa=0.88	tiempo = 22595	tiempo = Colisión

De los experimentos anteriores, a partir del análisis de la información cualitativa contenida en la forma de las trayectorias capturadas, y cuantitativa, observando el comportamiento del tiempo promedio de ejecución de trayectoria en función del parámetro alfa, se puede inferir que:

Para condiciones iniciales iguales en la posición de los elementos de juego y en los parámetros de configuración, diferentes corridas del simulador y de la plataforma de control generan trayectorias distintas como se puede verificar en los tres experimentos realizados: escenario de prueba estático 1, escenario de prueba estático 2 y escenario de prueba dinámico.

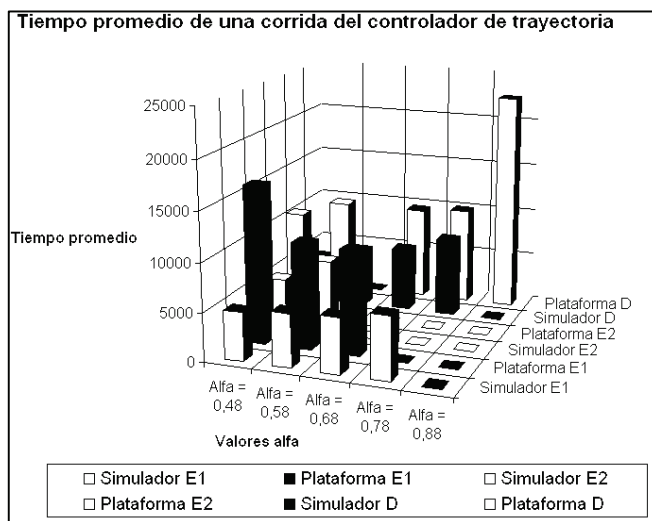


Figura 34. Gráfica de los tiempos promedio de ejecución del controlador de trayectoria para distintos escenarios y para diferentes valores de ponderación alfa.

Con relación al tiempo promedio de las corridas para el simulador y la plataforma de control, según se visualiza en la Figura 34, se identifican dos tendencias distintas en la medida que en un experimento se incrementa el valor del parámetro alfa. Para el caso de las corridas ejecutadas en el simulador, se puede observar una relación directa entre el parámetro alfa y el tiempo promedio de trayectoria, con un va-

lor óptimo de sintonía para alfa de 0.48 en escenarios estáticos y de 0.68 para un escenario dinámico, por lo tanto, el óptimo global se encontraría en algún punto intermedio de este rango. Un comportamiento más complejo se observa en la plataforma de control, debido a que el modelo no contempla todas las no linealidades involucradas. En las corridas ejecutadas en la plataforma de juego para el escenario estático 1, se presenta una relación inversa entre el parámetro alfa y el tiempo promedio de trayectoria, encontrándose un valor óptimo de sintonía de 0.68, para los demás escenarios, el comportamiento creciente en el tiempo promedio se produce por el incremento del parámetro alfa, con un óptimo para el escenario estático 2, de 0.48, y para el escenario dinámico de 0.68. Con base en estos resultados se puede concluir que el valor óptimo global de la función de ponderación se encuentra entre 0.58 y 0.68, rango cercano al valor óptimo de 0.68 para el escenario 1, rango en el cual las trayectorias capturadas para el escenario estático 2, pasan de cumplir con el objetivo de anotación de gol a colisión, y rango en el cual, para el escenario dinámico, se pasa de estado de colisión a óptimo de sintonía.

Por otro lado, revisando la morfología de las gráficas, cuando la plataforma se encuentra sintonizada, las formas de las trayectorias que genera, coinciden con las simuladas en un mismo escenario, con lo cual, la herramienta de simulación permite analizar el comportamiento de los controladores offline.

En cada una de las figuras tomadas, producto de corridas reales en la plataforma de control Robosoccer, se puede apreciar el cambio en el comportamiento del controlador de planeación de trayectorias, se distingue 2 tramos en cada trayectoria: uno continuo o con oscilaciones de gran amplitud, y otro en la zona cercana al objetivo o punto de disparo, donde ocurren micro oscilaciones. En la propuesta inicial del controlador de planeación de trayectoria, cuando no se invertía el valor de ponderación de los sub-controladores CD1 Y CD2 a través de la función alfa, las trayectorias terminaban en comportamientos no estables. Si bien las micro-oscilaciones incrementan el tiempo de estabilización del controlador, garantizan que este siempre llegue a condiciones estables para que los estados de juego puedan evolucionar y así cumplir el objetivo general, anotar un gol en la portería contraria.

En las corridas sobre la plataforma real, se aprecia cómo en todo escenario y configuración factible, el controlador de trayectoria lleva al AMC hasta el punto de disparo, de ahí en adelante el control de estados de juego alinea el AMC con la bola y el arco, lo aproxima a la bola y ejecuta el accionamiento del algoritmo de pateo para efectuar el disparo al arco. Esto se evidencia en el movimiento detectado sobre la trayectoria de la bola, que se dibuja a partir del momento en el que el AMC llega a la posición donde ésta se encuentra, ver Figura 30 parte b).

Finalmente, la precisión promedio en los disparos, coordenada Y de la bola en su entrada al arco adversario, es de 5 u-

nidades alrededor del centro de la portería. Con un rango de la imagen en Y de 240 unidades, se tiene un error porcentual de 2%.

Conclusiones

En la implementación del sensor de video se enfrentaron dos dificultades principales: la variación en la iluminación ambiente, y la velocidad con que se dispone de cada uno de los cuadros generados por la cámara de video, además del tiempo limitado para procesar dichos cuadros, que para el caso de la cámara usada Logitech QuickCam Pro 4000, es de 33 ms.

La solución al problema de iluminación se encontró en la utilización de una cámara CCD con ajuste automático de color y blanco, ya que los primeros ensayos se realizaron con una cámara basada en un chip CMOS no automática, obteniendo una imagen más definida, sobre la cual los filtros generaron una correcta segmentación con una precisión en las mediciones de posición y orientación de ± 2 píxeles y de ± 2 grados respectivamente. Otro punto de ajuste para optimizar el sensor se efectuó buscando un conjunto de colores que se encontrara uniformemente espaciados sobre el espectro de coloridad o tonalidad, dado que el espacio de color donde se aplica la segmentación por umbrales fue el HSI. Finalmente, fue necesario aislar el ambiente de juego y dotarlo de luz artificial que no marcara focos de reflejo sobre el sustrato de la arena de juego, con el ánimo de mantener controlada la iluminación con una intensidad uniforme.

En relación al retardo encontrado entre el instante en que ocurre una acción en el ambiente captado y el momento en el cual se detecta dicha acción, se realizaron experimentos de captura de video en LabView y C#. En LabView se observó un desfase entre acción y evento registrado de casi 1 segundo, mientras que sobre la plataforma .Net, era imperceptible para un observador que mirara las 2 escenas, mundo real e imagen tomada por la cámara. En cuanto al tiempo de procesamiento, el mismo algoritmo "RoboRoos Adaptado" se implementó en las dos plataformas de desarrollo, en LabView se procesaron 5 cuadros por segundo, mientras que en C#, 15 cuadros por segundo en un procesador Intel Centrino de 1.6GHz. En un procesador Intel Core Duo de 2.8 GHz, se procesaron 30 cuadros por segundo. Diferencia debida en gran medida al acceso directo a memoria usando código inseguro, procedimiento con el cual se trataron las matrices que representaban las imágenes captadas.

El comportamiento del sensor de posición por video cumplió con su objetivo, logrando identificar los dos agentes móviles AMC y MCU, además de la bola de juego, con un tiempo razonable y con una precisión adecuada para que el algoritmo de planeación de trayectorias pueda ejecutarse en tiempo real de aplicación (movimiento continuo de los robots).

En cuanto a la plataforma mecánica utilizada, el sistema de pateo que portan los móviles cumplió a cabalidad su objetivo, lograr impulsar una bola de golf de extremo a extremo

de la cancha a una velocidad de hasta 1.43m/s. Su implementación es un trabajo conjunto del controlador embebido en el móvil y la plataforma de control de juego, que en conjunto, constituye una solución simple y funcional a un requisito que debe cumplir todo agente en una competición robo futbolística.

Así mismo, el protocolo de comunicaciones establecido entre la plataforma y los móviles, mostró un comportamiento estable en presencia de interferencia electromagnética producida por la banda en que trabajan las redes inalámbricas domésticas, Wi-Fi, canales de voz de los teléfonos inalámbricos y enlaces de RF que comunican a un PC con periféricos como el teclado y/o el mouse.

En conclusión, las plataformas mecánicas funcionaron de acuerdo a la especificación planteada en los objetivos ESPECÍFICOS.

El siguiente eslabón en la cadena de control lo constituye el algoritmo de planeación de trayectorias. Para esta tarea se utilizó un controlador basado en lógica difusa, ya que se propuso inicialmente usar la computación flexible como fundamento teórico para la propuesta de algoritmo de control de juego. Es por ello que soluciones mas livianas computacionalmente, como los campos vectoriales unipotenciales, no fueron abordadas.

Como algoritmo de *path planning*, se implementó un diseño que cumple con el interés primario de llevar al AMC hasta una posición de disparo, evadiendo obstáculos en su camino, evitando chocar la bola en la condición de no encontrarse en posición de tiro al arco adversario, y permaneciendo siempre en el campo de visión del sensor. Todo ello en un tiempo de aproximadamente 30ms.

La plataforma de control a través del AMC, implementa el rol de anotación de gol o de ataque, a partir de la ejecución de distintos estados de juego, donde uno de ellos es, búsqueda de pelota, y es allí donde se habilita el control de planeación de trayectorias. La consecución de estados determina el rol de juego, por lo cual la construcción de nuevos roles es una tarea simple. La plataforma de control, entonces, es una herramienta flexible, donde como trabajo futuro, se puede implementar estrategias de juego cooperativas en una capa de nivel superior.

Como perspectivas futuras, se aporta el diseño de una plataforma de hardware y software para que trabajos futuros sobre algoritmos de visión artificial y planeación de trayectorias puedan ser validados. Este trabajo es un engranaje mas de la maquinaria que se esta gestando a nivel de la universidad Nacional, a nivel Bogota, a nivel Colombia y finalmente, a nivel Latinoamericano, donde cada vez con mas fuerza se están mostrando resultados competitivos traducidos en equipos robóticos para aplicar a competencias internacionales como RoboSoccer de Robocup y MiroSot de la FIRA (Federation of International Robot-soccer Association).

Agradecimientos

A la Universidad Nacional de Colombia, especialmente al cuerpo docente de la maestría de Automatización Industrial, por su valioso aporte. Al director del proyecto de tesis, profesor Alberto Delgado. A ROCATEC Ltda. A su gerente administrativo Javier Alexander Ballen Salamanca y a su gerente financiero, Henry Roncancio, por estrechar la brecha existente en Colombia entre la Empresa privada y la Academia. A la ingeniera Hasbleidy del Pilar Pardo Acero, por haber sido un soporte fundamental en la terminación del proyecto. A los ingenieros Hugo Velasco, Leandro Perez y Gerardo Muñoz, por su amistad y orientación.

Bibliografía

- Birk, A., Coradeschi, S., Tadokoro, S., RoboCup 2001: Robot Soccer World Cup V., Springer, 2001.
- Browning, B., Bowling, M., Bruce, J., Balasubramanian, R., Veloso, M., CM-Dragons'01 – Vision-Based Motion Tracking and Heterogeneous Robots., RoboCup 2001: Robot Soccer World Cup V., Springer, 2001, pp. 667-570.
- Burkhard, H., Duhant, D., Fujita, M., Lima, P., Murphy, R. y Rojas R., The Road to RoboCup 2050., IEEE Robotics and Automation Magazine., Junio 2002, pp. 31-38.
- Costa, Pa., Sousa, A., Marques, P., Costa, Pe., Gaio, S., Moreira, A., 5dpo Team Description., RoboCup 2001: Robot Soccer World Cup V. Springer, 2001, pp. 653-566.
- De la Rosa, J., Innocenti, B., Montaner, M., Figueras, A., Muñoz, I., Ramon, J. A., Rogi Team Description, RoboCup 2001: Robot Soccer World Cup V. Springer, 2001, pp. 687-590.
- Driankov, D., Hellendoorn, H., Reinfrank, M., An introduction to fuzzy control., Springer-Verlag, 1996.
- Hibino, S. Kodama, Y., Nagasaka, Y., Takahashi, T., Murakami, K., Naruse, T., Owarabito – A Team Description., RoboCup 2001: Robot Soccer World Cup V. Springer, 2001, pp. 679-582.
- Kitano, H., Tambe, M., Stone, P., Veloso, M., Coradeschi, S., Osawa, E., Matsubara, H., Noda, I., Asada, M., The RoboCup Synthetic Agent Challenge 97., Proc. 15., Joint conf. on Artificial Intelligence., San Francisco CA, 1997, pp.24-29.
- Kortenkamp, D., Huber, M., Cohen, C., Raschke, U., Bidlack, C., Congdon, C., Koss, F., Weymouth, T., Winning the AAAI Robot Competition., Proc. AAAI'93, National Conference on Artificial Intelligence. San Jose, California, pp. 858-859.
- Molina, F., Cardenas, J., Tesis: Control de un equipo de fútbol Robótico Según Especificaciones de la Categoría Small Size Robocup., Universidad Nacional de Colombia, Ingeniería Electrónica, Bogotá, 2005.
- Rojas, R., Behnke, S., Liers, A., Knipping, L., FU-Fighters 2001 (Global vision), RoboCup 2001: Robot Soccer World Cup V. Springer, 2001, pp. 671-574.
- Soriano, J., Gonzalez, O., Munar, F., Ramos, A., Propuesta de 'defuzzifier' basado en relaciones booleanas., Revista

- Ingeniería, Universidad Distrital Francisco José de Caldas, Vol 6, No 2, 2001.
- Sotomonte, W., Tesis: Comparación de estrategias Multi-Agente, caso de estudio: Fútbol Robot., Universidad Nacional de Colombia, Maestría en automatización Industrial, Bogotá, 2006.
- Wakerly, J.F., Digital Desing, Principles and practices., Third Edition, Prentice Hall., 1999.
- Wyeth, G., Brown, B., Robust Adaptive Vision for Robot Soccer., Computer Science and Electrical Engineering., University of Queensland. Australia, 2000.
- Young, G. O., Synthetic structure of industrial plastics., 2nd ed. vol. 3, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
- Robocup., Definición de Robocup. Disponible: <http://www.robocup.org>.