

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“DISEÑO E IMPLEMENTACIÓN DE UN EQUIPO DE ROBOTS
AUTÓNOMOS QUE TOMAN DECISIONES EN TIEMPO REAL: FUTBOL
ROBÓTICO – COMPONENTE INTELIGENTE”

TOPICO DE GRADUACIÓN

Previa a la obtención del Título de:

INGENIERO EN COMPUTACIÓN

ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS

Presentado por:

CÉSAR ALBERTO VILLARROEL SAMANIEGO

CARLOS AMÍN CALDERON GARZOZI

ROMMEL PATRICIO CARRILLO

GUAYAQUIL – ECUADOR

AÑO 2003

AGRADECIMIENTO

A nuestro director de tesis, Phd. Boris Vintimilla por la atención prestada, confianza, y apoyo constante.

Al Ing. Carlos Monsalve, por su apoyo y ayuda en cada etapa de este proyecto de fin de carrera.

Al Centro de Investigación de Ciencia y Tecnología (CICYT) por haber creído en este proyecto.

A nuestros compañeros de proyecto ya que sin ellos no se hubiera podido participar en las Competiciones de la FIRA 2002.

A nuestros padres, hermano y amigos que estuvieron siempre apoyándonos.

A la FIRA, YUJIN y todas las personas que de manera directa o indirecta colaboraron en la realización de ésta proyecto de fin de carrera.

DEDICATORIA

QUEREMOS DEDICAR ESTA TESIS A NUESTROS FAMILIAS, QUE
CONFIARON EN NOSOTROS Y NOS APOYARON EN TODO MOMENTO,
ASI COMO A TODAS LAS PERSONAS QUE SUPIERON DAR SU
RESPALDO INCONDICIONAL PARA SU FELIZ CULMINACION.

TRIBUNAL DE GRADUACION

Ing. Norman Chootong
SUB-DECANO DE LA FIEC

Phd. Boris Vintimilla
DIRECTOR DE TOPICO

Phd. Enrique Peláez
VOCAL

Ing. Carlos Jordán
VOCAL

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Proyecto de Tópico, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)

César Alberto Villarroel Samaniego

Carlos Amín Calderón Garzozzi

Rommel Patricio Carrillo Chagcha



RESUMEN

El presente grupo de estudiantes del t3pico de graduaci3n “Visi3n por Computador y Procesamiento Digital de Im3genes” propuso como base para su graduaci3n formar un equipo de robots aut3nomos con decisiones en tiempo real, aplicados al f3tbol rob3tico. El reglamento al cual nos acogimos para desarrollar el proyecto estuvo basado en la Federaci3n Internacional de F3tbol Rob3tico Asociado (FIRA), cuya liga es la Small MIROSOT, la misma que consta de 3 robots por equipo, los cuales ejecutan tareas de movimiento y desplazamiento de un bal3n de forma similar a un encuentro de f3tbol normal, realizado por humanos. Los robots se montan sobre un campo de f3tbol semejante a una mesa de ping-pong que cumple con un est3ndar internacional en sus medidas. Cuando un partido de f3tbol entre robots se desarrolla la posici3n de los robots es censada y controlada a trav3s de un sistema de visi3n. La informaci3n capturada por el sistema de visi3n es usada luego por un computador para generar las estrategias de juego que deber3n seguir los robots. El computador posteriormente env3a las3rdenes a cada uno de los robots para que estos ejecuten sus respectivos movimientos. El proyecto en su forma global es complejo como para tratar de enfrentarlo en un solo problema. Por esa raz3n, se ha dividido el proyecto en tres

componentes: inteligente, visión y electromecánico para facilitar su implementación.

El presente trabajo trata el desarrollo del componente inteligente, que tiene como objetivo principal controlar las posiciones de los robots, implementar los algoritmos que generen estrategias de juego para los robots, haciendo uso de la información del entorno de trabajo de los robots, mismo que es dado por el componente de visión tal como las posiciones de cada objeto, su orientación, etc. Además se realiza la importancia de estudiar el fútbol robótico como una plataforma de desarrollo para sistemas multiagente.

INDICE

RESUMEN.....	V
INDICE GENERAL.....	VII
INDICE DE FIGURAS.....	XII
INDICE DE TABLAS.....	XVI
I. INTRODUCCION.....	1
1.1 Descripción del proyecto.....	1
1.2 Justificación del proyecto.....	4
1.3 Objetivos del proyecto	6
1.4 Análisis de costos	7
1.5 Organización del documento.....	9
1.6 Conclusiones	10
II. ESTUDIO BIBLIOGRÁFICO SOBRE SISTEMAS MULTIAGENTE APLICADOS AL FÚTBOL ROBÓTICO	11
2.1 Introducción.	12
2.2 Sistemas multiagente: fútbol robótico.....	14
2.3 Clasificación de los sistemas multiagente.....	16
2.3.1 Sistemas multiagente homogéneos comunicativos	16

2.3.2	Sistemas multiagente homogéneos no comunicativos	18
2.3.3	Sistemas multiagente heterogéneos comunicativos	19
2.3.4	Sistemas multiagente heterogéneos no comunicativos	22
2.4	Clasificación de sistemas de fútbol robótico.	23
2.4.1	Sistemas de robots con inteligencia parcial dentro del robot, basados en visión.	24
2.4.2	Sistemas de robots sin inteligencia dentro del robot, basados en visión.	26
2.4.3	Sistemas de robots con inteligencia dentro del robot, basados totalmente en sensores incorporados.	28
2.5	Comparación de los sistemas de robots.	29
2.6	Antecedentes de las competiciones de fútbol robótico a nivel mundial.	32
2.7	Conclusiones	34

III. CONTROL DE MOVIMIENTO DE LOS MICRO-ROBOTS DE LA MIROSOT. 36

3.1	Introducción.	37
3.2	Determinación de la cinemática del robot.	39
3.3	Método de campos potenciales para posicionamiento y orientación del robot.	45
3.4	Implementación del campo potencial de atracción para patear la pelota.	49

3.5	Implementación del campo potencial de repulsión para evitar obstáculos.....	55
3.6	Función de predicción para estimar la trayectoria de la pelota.....	62
3.7	Análisis de resultados.	64
3.8	Conclusiones.	66
4.	APRENDIZAJE POR REFUERZO APLICADO AL FÚTBOL ROBÓTICO	68
4.1	Introducción	68
4.2	Conceptos generales de aprendizaje por refuerzo y aprendizaje Q.	70
4.3	Aplicación del aprendizaje Q modular para desarrollar la coordinación y cooperación entre los jugadores.	80
	4.3.1 Arquitectura del aprendizaje Q modular.	80
	4.3.2 Descripción del algoritmo implementado.	82
4.4	Análisis de resultados.	90
4.5	Conclusiones.	93
5.	TÉCNICAS DE MOVIMIENTO UTILIZADAS POR EL EQUIPO FUROEC EN LAS COMPETICIONES DE LA FIRA 2002 WORLD CUP.....	95
5.1.	Introducción.	95
5.2.	Descripción de los algoritmos utilizados en las competencias.	97
	5.2.1 Algoritmo básico para patear la pelota (pelota quieta).	98

5.2.2	Algoritmo para evitar obstáculos móviles mediante la generación de objetivos intermedios.....	101
5.2.3	Algoritmo para salir de los bordes del campo de juego.	105
5.2.4	Algoritmo para mover el arquero de acuerdo a la posición de la pelota.....	108
5.2.5	Descripción del movimiento de giro y rechazo cuando la pelota se encuentra bien cerca del jugador.....	113
5.2.6	Primera implementación del comportamiento para el atacante.....	114
5.2.7	Implementación de la estrategia general del equipo para un equipo de cinco robots usando estructuras de decisión.....	118
5.3.	Análisis de resultados en las competiciones.....	124
5.4.	Conclusiones	126
6.	SUMARIO, CONTRIBUCIONES Y FUTUROS TRABAJOS	128
6.1	Introducción	128
6.2	Sumario.....	129
6.2.1	Implementación del algoritmo para patear la pelota	129
6.2.2	Algoritmo para evitar obstáculos mediante la generación de objetivos intermedios.	130
6.2.3	Algoritmo para salir de los bordes del campo de juego.	131
6.2.4	Algoritmo para mover el arquero de acuerdo a la posición de la pelota.....	131

6.2.5 Implementación de la estrategia de juego para un equipo de cinco robots usando estructuras de decisión. 132

6.2.6 Implementación de la estrategia de juego para un equipo de tres robots usando aprendizaje Q modular. 133

6.2.7 Campos Potenciales univectoriales usados para evitar obstáculos y para ir hacia la pelota. 135

6.3 Contribuciones 136

6.3.1 Movimiento rápido del robot para patear la pelota 136

6.3.2 Habilidad mejorada para salir de los bordes de la cancha 137

6.3.3 Algoritmo optimizado para definir el comportamiento del arquero usando estimación de trayectorias. 138

6.3.4 Especificación de las condiciones para aplicar campos potenciales univectoriales modificados..... 138

6.3.5 Aplicación del algoritmo de aprendizaje Q modular. 139

6.3.6 Logros adicionales obtenidos con la ejecución de este proyecto. 140

6.4 Futuros Trabajos..... 144

6.5 Conclusiones 146

BIBLIOGRAFIA 148

INDICE DE FIGURAS

Figura 1.1.- Descripción gráfica de los componentes de fútbol robótico.....	2
Figura 2.1.- Escenario multiagente	15
Figura 2.2.- Sistemas multiagente homogéneos comunicativos	17
Figura 2.3.- Sistemas multiagente homogéneo no comunicativos	18
Figura 2.4.- Sistemas multiagente heterogéneos comunicativos	20
Figura 2.5.- Sistemas multiagente heterogéneo no comunicativos	22
Figura 2.6.- Ubicación de la inteligencia artificial.....	23
Figura 2.7.- Sistemas de robots con inteligencia dentro del robot, basados en visión.	25
Figura 2.8.- Sistemas de robots sin inteligencia dentro del robot, basados en visión.....	27
Figura 2.9.- Sistemas de robot son inteligencia dentro del robot, basados totalmente en sensores incorporados	28
Figura 3.1.- Robot usado en los torneos de la MIROSOT.....	39
Figura 3.2.- Tipo de rueda del robot móvil	40
Figura 3.3.- Velocidad lineal y angular de una rueda del robot.....	40
Figura 3.4.- a) centro instantáneo de rotación c_r del robot, b) modelo de la cinemática del robot móvil.....	41

Figura 3.5.- Tipos de planificación de trayectorias: a) fuera de línea, b) planificación en línea.	46
Figura 3.6.- a) Identificación del problema de planificación de trayectorias en el fútbol robótico. b) Generación de campos potenciales para la posición final y evasión de obstáculos. b) Trayectoria generada en línea mediante el algoritmo de campos potenciales.	47
Figura 3.7.- Campo univectorial para una posición g deseada.	49
Figura 3.8.- Variables usadas en el cálculo del campo potencial de atracción $N(p)$	52
Figura 3.9.- Generación de la región expandida	53
Figura 3.10.- Nuevo campo univectorial para la posición desea g	54
Figura 3.11.- Generación del campo univectorial de repulsión para evitar un obstáculo.....	56
Figura 3.12.- Zona del campo univectorial $R(p)$ que permite evitar un obstáculo.....	57
Figura 3.13.- Variables usadas en el algoritmo de generación del campo $R(p)$	58
Figura 3.14.- Algoritmo de definición del campo vectorial para evitar obstáculos	60
Figura 3.15.- Distancia entre el obstáculo y la recta formada por el robot y el objetivo (pelota).	61
Figura 3.16.- Predicción de la pelota.....	62
Figura 3.17.- Comportamiento del algoritmo de evitar obstáculos.....	64

Figura 3.18.- Resultado de la aplicación del algoritmo de evitar obstáculos sobre el sistema MIROSOT de 3 vs. 3.....	65
Figura 4.1.- Función de aprendizaje Q para un agente: un aprendiz activo que actualiza el valor de $Q(s,a)$ por cada acción y en un mundo no determinista.	79
Figura 4.2.- Arquitectura de Aprendizaje – Q modular.	81
Figura 4.3.- La Región 1 muestra la zona de conflicto para los robots atacante y defensa de un mismo equipo.	84
Figura 4.4.- (a) Agentes individuales y (b) Agentes acoplados.	84
Figura 4.5.- Componentes de estados en la Región 1	88
Figura 4.6.- Trayectorias realizadas por los robots atacante y defensa para alcanzar a la pelota, después de ejecutar el algoritmo de aprendizaje Q modular implementado.....	91
Figura 4.7.- Número de iteraciones tomadas para que cada robot alcance la pelota.	91
Figura 4.8.- Acciones ejecutadas por el agente acoplado durante las trayectorias de la Figura 4.6.....	92
Figura 5.1.- Análisis vectorial del movimiento de un robot.	98
Figura 5.2.- Algoritmo para que un robot patee la pelota hacia el arco.....	99
Figura 5.3.- Algoritmo implementado por FUROEC para evitar obstáculos	103
Figura 5.4.- Tipos de bordes	106
Figura 5.5.- Algoritmo para evitar un borde cancha y borde arco.	108

Figura 5.6.- Algoritmo del arquero de acuerdo a la posición de la pelota	108
Figura 5.7.- Algoritmo para bloquear la pelota.....	110
Figura 5.8.- Cambio automático en la dirección del robot.....	112
Figura 5.9.- Descripción del movimiento de rechazo	113
Figura 5.10.- Definición del TARGET_DINAMICO.....	114
Figura 5.11.- Diagrama de bloques del componente inteligente	119
Figura 5.12.- Límites de las zonas para cada modo de juego.....	121
Figura 5.13.- Posiciones pasivas	121
Figura 5.14.- Zonas de juego de cada robot.....	122
Figura 5.15.- Árbol de decisión para el arquero.....	122
Figura 5.16.- Árbol de decisión para los defensas	123
Figura 5.17.- Árbol de decisión para los atacantes	124

INDICE DE TABLAS

Tabla 1.1.- Costos necesarios para la implementación de un sistema de fútbol robótico: equipo FUROEC.....	8
Tabla 2.1. - Cuadro comparativo de los sistemas de fútbol robótico.....	29
Tabla 4.1. - Acciones del agente acoplado para la situación de no bloqueo.	89
Tabla 5.1. - Estrategias para cada modo de juego	120
Tabla 5.2.- Marcadores de los partidos jugados en la FIRA 2002 World Cup.....	126

CAPITULO I

INTRODUCCION

Este capítulo presenta una visión general de la problemática existente en la implementación de un equipo de robots autónomos que simulan jugar un partido de fútbol similar al realizado por humanos. Para esto, el capítulo inicialmente describe en forma resumida todo el proyecto de graduación propuesto, destacando el papel que cumple el componente inteligente dentro del mismo. Posteriormente, se mencionan los principales impactos obtenidos con el desarrollo de este proyecto y sus aplicaciones en el mundo real derivadas de estos estudios. Un análisis de los costos necesarios para la implementación del proyecto, así como la estructura de la organización de todo el documento es presentado al final del capítulo.

1.1 DESCRIPCIÓN DEL PROYECTO.

Las competiciones de fútbol con robots fueron creadas básicamente para estimular la investigación y el desarrollo en los campos de la robótica y la inteligencia artificial, además de las áreas de visión por computador, procesamiento de imágenes, electrónica e informática por mencionar unas cuantas.

Los competidores, estudiantes y científicos de universidades, deben diseñar robots equipados con sistemas de visión que detectan la pelota y que distinguen los jugadores de su equipo y los del equipo contrario. Cada robot posee inteligencia y actúa individual o colectivamente de acuerdo a la situación del juego, siempre con el objetivo de meter gol al equipo oponente.

Básicamente la plataforma de trabajo sobre la cual se implementan las técnicas para cada equipo de fútbol de robots consta de: una cámara de video, una tarjeta de digitalización de imágenes, un computador, un transmisor-receptor de radiofrecuencia (RF), 3 micro-robots por equipo, un sistema de iluminación y un tablero de 1.3 mts. x 1.5 mts. que hace las veces de campo de juego, tal como se muestra en la figura 1.1:

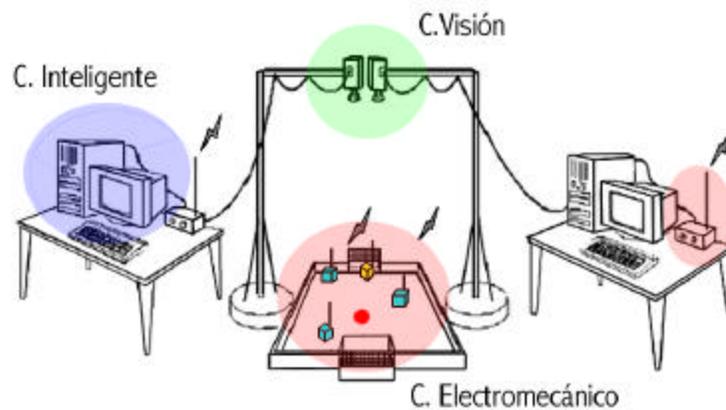


figura 1.1.- Descripción gráfica de los componentes de fútbol robótico

Cuando un partido de fútbol entre robots es desarrollado, el campo de juego consta de 6 micro-robots, de los cuales cada equipo está constituido de 3 de estos. Las posiciones y orientaciones de los robots y de la pelota son capturadas como imágenes a través de una cámara situada a una altura aproximada de 2 mts sobre el campo de juego. Las imágenes adquiridas son luego procesadas y usadas para reconocer y localizar cada objeto dentro de la cancha de juego, la información obtenida es usada por un computador para definir las estrategias de juego que deberá usar el equipo. El computador posteriormente envía órdenes a cada uno de los robots, para lo cual cada robot está equipado con un receptor de radio a través del cual recibe las órdenes y un micro controlador que las interpreta y las traduce en movimientos.

El proyecto en su forma global es complejo como para tratar de enfrentarlo en un solo problema. Por tal razón, y para facilitar la implementación de un sistema de fútbol robótico, éste ha sido dividido en tres componentes que cumplen funciones específicas dentro del sistema. Estos tres componentes son: componente de visión, componente electromecánico y componente inteligente. La característica principal de estos componentes es que ellos forman un sistema de comunicación cerrado. Esto se debe a que la información captada por el componente de visión es procesada y luego transmitida al componente inteligente, para que este defina las estrategias de juego y controle a los robots, y finalmente este último, envía las órdenes al componente electromecánico para que se encargue de transformar las órdenes

recibidas en movimientos. Este documento se enfocará únicamente en el desarrollo del componente inteligente.

De esta forma, el componente inteligente comprende el diseño e implementación de un conjunto de algoritmos y técnicas de control e inteligencia artificial que permiten a los robots cumplir satisfactoriamente el objetivo principal de anotar suficientes goles para ganar el partido de fútbol.

1.2 JUSTIFICACIÓN DEL PROYECTO.

El fútbol robótico es una idea que nació desde 1995 catalogado como un campo intrigante en la comunidad de inteligencia artificial a través del mundo. Este campo probó ser una plataforma excelente para el estudio y desarrollo de los *Sistemas Multiagente*, subcampo de la *Inteligencia Artificial Distribuida*, la cual tiene por propósito construir sistemas que involucran varias entidades (agentes) que guardan un mismo objetivo y se desenvuelven en un ambiente complejo y dinámico.

Peter Stone en [22] menciona que las técnicas y métodos implementados en un sistema de fútbol robótico pueden ser aprovechados como base para ser aplicados en otros dominios. De esta forma, dominios tales como: misiones espaciales, búsqueda y rescate (de información, personas, etc.), mantenimiento de hospitales y fábricas, combates en el campo de batalla, exploración

submarina y enrutamiento de redes, pueden beneficiarse de las técnicas usadas en los sistemas de fútbol robótico.

Adicionalmente, tomando ventaja que un sistema de fútbol robótico es una disciplina estrictamente multidisciplinaria, los estudios y conocimientos adquiridos en el desarrollo de estos sistemas pueden servir como base para en el futuro formular nuevos proyectos afines. Así mismo, el desarrollo de este proyecto es una gran oportunidad para motivar la formación de jóvenes investigadores en el campo de la robótica, visión e inteligencia artificial. Estudiantes de ingenierías, tales como: electrónica, computación, telecomunicaciones, mecánica, y otras ramas relacionadas pueden formar parte de estos propósitos.

Esperamos que otros estudiantes y profesionales cuyas especialidades afines con esta disciplina se motiven a realizar nuevos experimentos, estudios e investigaciones; y de esta forma encuentren nuevos tipos de aplicaciones que permitan brindar soluciones tecnológicas a la sociedad.

1.3 OBJETIVOS DEL PROYECTO

El objetivo general de este proyecto consiste en diseñar e implementar un equipo de robots autónomos que cooperen en tiempo real y simulen jugar un partido de fútbol realizado por humanos. Para conseguir este objetivo y para

hacer el proyecto menos complejo, éste ha sido dividido en tres componentes principales que son los siguientes: componente inteligente, componente de visión y componente electromecánico. En este trabajo se desarrollará únicamente el componente inteligente.

El componente inteligente será el encargado de controlar a todos los robots de un equipo, para que estos cumplan con el objetivo principal que consiste en anotar el mayor número de goles al equipo oponente. Estas metas se pretenden alcanzar con la consecución de los siguientes objetivos específicos:

- Desarrollar la parte del componente inteligente encargado de controlar los movimientos de los robots, utilizando la información proveniente del componente de visión.
- Aprovechar las técnicas y métodos existentes en la literatura para ayudar a facilitar la implementación del componente inteligente del equipo de robots autónomos.
- Implementar los algoritmos de orientación y definición de trayectorias que permitan a los robots desenvolverse dentro de la cancha de fútbol de forma dinámica y autónoma.

- Reforzar las bases de este proyecto para dar la continuidad requerida, esto es, involucrar a jóvenes emprendedores interesados en formar parte de las comunidades científicas que investigan las temáticas de inteligencia y vida artificial.
- Aprovechar los conocimientos adquiridos de la investigación realizada sobre sistemas multiagente para abrir nuevos campos de aplicación en diversas áreas del sector productivo.

1.4 ANÁLISIS DE COSTOS

Tomando como referencia los equipos adquiridos por el equipo de *Fútbol Robótico Ecuatoriano (FUROEC)* el costo total necesario para la implementación global de un sistema de fútbol robótico es el que se desglosa en la tabla 1.1.

Análisis de costos

<u>Descripción</u>	<u>valor</u>
Cancha	250.00
Sistema de iluminación: 4 luminarias de encendido electrónico Metallight	430.00
Aparato medidor de intensidad de luz: luxó metro	80.00
Cámara CCD Samsung 410 ND *	600.00
Estructura de soporte para la cámara	100.00
Pelotas de golf	10.00
Regletas, extensiones eléctricas	10.00
Cable de video *	20.00
Tarjeta de digitalización de video MyVision *	250.00
6 micro robots tipo YSR A *	5,100.00
2 transmisores rf 430/410 mhz *	1,200.00
1 cargador de baterías *	100.00
7 baterías de Lithium *	210.00
Software de visión *	250.00
Un computador con características:	
Pentium iv 1.6 Ghz	
Memoria ram 256 Mb	
Disco duro de 40 Gb	
Tarjeta de video AGP de 32 Mb	
CD ROM, disquetera	
Monitor de 15 pulgadas	
Ratón, teclado ps2	
	1.500.00
Software de desarrollo	
Microsoft visual C++ (licencia estudiante)	200.00
TOTAL	10.310.00
* este precio es en el país de origen (Corea).	

tabla 1.1.- Costos necesarios para la implementación de un sistema de fútbol robótico: equipo FUROEC

1.5 ORGANIZACIÓN DE LA TESIS.

Con la intención de proporcionar un documento de referencia, seis capítulos han sido reservados para detallar la implementación del componente

inteligente en el proyecto de fútbol robótico. Cada capítulo describe las diferentes técnicas y métodos implementados por el equipo FUROEC. Estos capítulos son resumidos a continuación:

- El capítulo I presenta una descripción general del proyecto desarrollado, destacando el papel que cumple el componente inteligente dentro del mismo, los principales impactos y un análisis de costos de la implementación del proyecto [31].
- El capítulo II realiza un estudio bibliográfico de los diferentes enfoques que se le ha dado al fútbol robótico por parte de investigadores en sistemas multiagente, una introducción a los sistemas multiagente, su aplicación en el fútbol robótico y los tipos de sistemas de micro robots futbolistas [32].
- La implementación y descripción del método de control de movimiento de robots, mediante campos potenciales que se aplica a nuestro sistema de fútbol robótico se explica en el capítulo III [40].
- El capítulo IV realiza la implementación de técnicas de aprendizaje de máquinas por refuerzo como aprendizaje Q, que se aplican al fútbol robótico.

- El capítulo V presenta un resumen acerca de las técnicas de movimiento que se utilizaron en las competencias de la *FIRA World CUP* realizadas en mayo de 2002 en Corea del Sur [30].
- Sumarios, contribuciones y futuros trabajos son presentados en el capítulo VI.

1.6 CONCLUSIONES

Esta sección presentó una descripción de los diferentes componentes que integran nuestro sistema de fútbol robótico, que debido a su relativo bajo costo pueden ser utilizados en la enseñanza y la investigación de áreas como: Robótica, Inteligencia Artificial, Visión por Computador y Electrónica.

Además, se describió de forma general el contenido del trabajo de tesis desarrollado, para que sirva como punto de partida a estudiantes que deseen revisar el documento y sirva como documento de referencia para realizar futuros estudios en fútbol robótico.

CAPITULO II

Estudio bibliográfico sobre sistemas multiagente aplicados al fútbol robótico

Los ambientes de robots con un alto grado de dinamismo son escenarios apropiados para desarrollar técnicas de inteligencia artificial que ayudan a resolver problemas del mundo real. En este sentido, un juego de fútbol robótico es un ejemplo de un escenario dinámico donde múltiples agentes independientes interactúan en un mismo dominio. Estos tipos de escenarios ofrecen un dominio confiable para investigar temas relacionados con el desarrollo de sistemas multiagente.

En este capítulo presentamos diferentes enfoques, un estudio no exhaustivo acerca de los sistemas multiagente aplicados al fútbol robótico que han sido realizados por varios investigadores. El sistema de fútbol robótico es discutido como un apropiado test para los sistemas multiagente.

2.1 INTRODUCCIÓN.

Diversas definiciones sobre agente han sido propuestas en el campo de la inteligencia artificial, pero ninguno ha sido totalmente aceptado. Russell y

Norvig en [21] definen a un *agente* como algo que puede percibir el ambiente a través de sensores y actuar por medio de sus efectores. Así, un agente autónomo es capaz de interactuar independiente y efectivamente con sus sensores y efectores a modo de lograr la consecución de metas específicas o compartidas.

Peter Stone en [22] define a un agente como una entidad con percepciones, metas, cogniciones, acciones y dominios de conocimiento, el cual está situado dentro de un ambiente. La manera de actuar o mapear desde percepciones a acciones son llamadas sus *comportamientos*. De esta forma, el comportamiento de un agente es el que le permite interactuar con el mundo real.

Cuando un grupo de agentes en un sistema multiagente comparte una meta común ellos forman un equipo. Los miembros del equipo coordinan sus comportamientos adoptando procesos cognitivos compatibles y afectando directamente a cada una de las otras entradas (percepciones) incluyendo acciones a través de la comunicación [22].

Tradicionalmente, los sistemas multiagente se ocupan del manejo del comportamiento de varios agentes, además, nos proveen temas de investigación como protocolos de cooperación por control distribuido,

comunicación efectiva y tolerancia de fallas, mientras muestran eficiencia en temas como cooperación, adaptación, robustez en tiempo-real [10].

Desde el punto de vista de los sistemas multiagente, el fútbol robótico está recibiendo mucha atención por varios investigadores para profundizar el desarrollo en tópicos tales como control de robots móviles, cooperación entre múltiples agentes, técnicas de aprendizaje de máquinas y otros. Esto es debido a que el fútbol es un típico deporte de equipo, el cual es un buen ejemplo de los problemas del mundo real abstraídos moderadamente.

En otro ámbito los sistemas multiagente han sido investigados más a fondo dentro del fútbol robótico donde han surgido múltiples campos relacionados con este. Existen diferentes tipos de esquemas disponibles para el control de fútbol robótico [16]. Así por ejemplo, existen *sistemas de fútbol robótico basados en visión sin inteligencia*, donde la velocidad de los robots es controlada por el computador a través de una tarjeta de transmisión, y existen también *sistemas de fútbol robótico basados en visión con inteligencia* en el que parte de ella esta incluida en el robot. Otro esquema que señalamos más adelante son los sistemas de fútbol robótico basado en robots con sensores incorporados. Se describen como *robots autónomos*, ya que las decisiones tomadas se realizan en base a una colección de información proporcionada por sus sensores, y en ciertos casos con la capacidad de comunicarse con sus compañeros.

2.2 SISTEMAS MULTIAGENTE: FÚTBOL ROBÓTICO

La *inteligencia artificial distribuida* (DAI, por sus siglas en inglés) es subcampo de la inteligencia artificial. DAI esta dedicada al estudio de técnicas y conocimiento necesario para la coordinación y distribución de acciones y conocimientos en un ambiente con múltiples entidades independientes.

Dentro de la inteligencia artificial distribuida, varios problemas están siendo enfrentados por investigadores [8,19]. Algunos de estos problemas son listados a continuación:

- Como formular, describir, descomponer, localizar problemas y sintetizar resultados entre un grupo de agentes inteligentes.
- Como los agentes se comunican e interactúan, que lenguaje y protocolos de comunicación utilizan.
- Como se aseguran los agentes para actuar coherentemente para realizar y tomar decisiones, dentro de una determinada área del campo de juego.
- Como reconocer y corregir acciones de agentes cuando ellos se encuentren en conflictos, tal como coordinar alguna acción.

Un subcampo de la inteligencia artificial distribuida son los sistemas multiagente. Estos sistemas proporcionan los principios para la construcción de sistemas complejos involucrando múltiples agentes y los mecanismos para la coordinación de los comportamientos de estos agentes.

En el escenario de los sistemas multiagente, los agentes interactúan directamente con varios agentes a través de una vía de comunicación. Esta interacción puede ser vista como un estímulo ambiental y puede ser afectada si otros agentes intervienen indirectamente a través de vías no predecibles. De este modo, el escenario de un sistema multiagente puede ser considerado como un ambiente dinámico, tal como se muestra en la figura 2.1, donde se considera la interacción de un agente con otro.

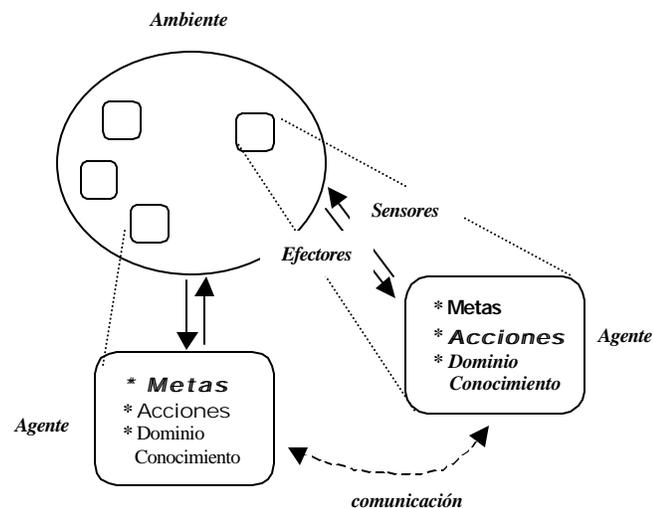


figura 2.1.- Escenario multiagente

De la gráfica anterior se observa que los agentes pueden tener diferentes metas, acciones y dominio de conocimientos, a lo que llamaremos *bloque de*

actividades. El tipo de letra dentro del bloque de actividades de cada agente varía para señalar diferencia o a su vez igualdad de actividades. Del mismo modo, se observa que entre ellos puede o no existir comunicación

2.3 CLASIFICACIÓN DE LOS SISTEMAS MULTIAGENTE.

Peter Stone y Veloso [26] consideran que los aspectos más importantes de agentes son: grado de heterogeneidad y grado de comunicación. Así ellos presentan cuatro combinaciones de heterogeneidad y comunicación entre agentes, estas combinaciones son: agentes homogéneos comunicativos y no comunicativos, agentes heterogéneos comunicativos y no comunicativos.

Dentro de los sistemas homogéneos y heterogéneos, una característica principal de los homogéneos es que todos los agentes tienen definido un mismo bloque de actividades. Mientras que, en los sistemas heterogéneos el bloque de actividades es definido diferente para cada agente.

2.3.1 Sistemas multiagente homogéneos comunicativos

Los sistemas multiagente homogéneos comunicativos se caracterizan por que la comunicación entre los agentes es realizada en forma directa. Adicionalmente, esta comunicación puede ser efectuada punto a punto o a todos los agentes. Una representación de los sistemas multiagente

comunicativos es mostrada en la figura 2.2. Nótese que en los sistemas multiagente, comunicativos o no comunicativos, siempre va existir una comunicación entre los agentes y su ambiente. Esta comunicación difiere para cada agente, en las entradas sensoriales y las acciones que ellos toman, debido a que los agentes están situados en forma diferente dentro del ambiente.

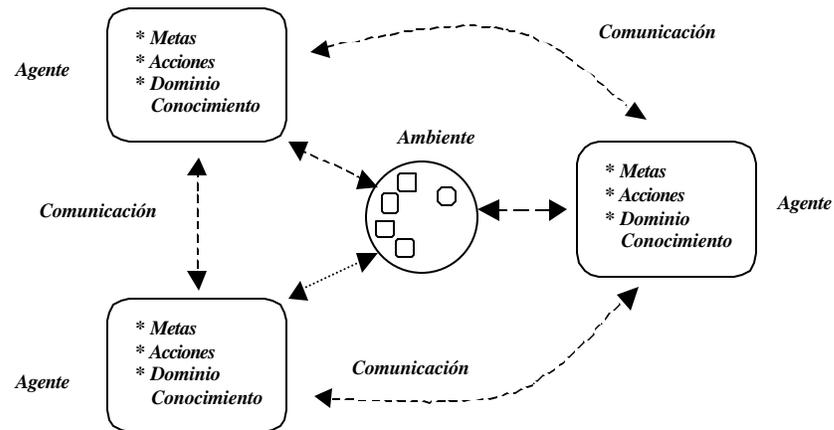


figura 2.2.- Sistemas multiagente homogéneos comunicativos figura 9.4 en [22]

Desde el punto de vista de fútbol robótico, los sistemas multiagente homogéneos comunicativos pueden ser usados fijando el comportamiento del equipo contrario y únicamente permitiendo comunicación entre los agentes de su equipo. Esto último es importante, debido a que en algún momento dado puede existir la posibilidad de que un agente en particular pueda ver solamente una pequeña porción del ambiente, y solo por comunicación con

los compañeros de su equipo pueda conseguir una vista más amplia del ambiente.

2.3.2 Sistemas multiagente homogéneos no comunicativos

Estos sistemas difieren de los sistemas homogéneos comunicativos debido a que en ellos no existe una comunicación entre los agentes que forman el sistema multiagente. Es decir, únicamente existe comunicación entre el ambiente y los agentes. La figura 2.3 muestra una representación de estos sistemas no comunicativos. En estos sistemas, las diferencias entre varios agentes son sus entradas sensoriales y las acciones actuales que ellos toman; además, los agentes tienen el mismo procedimiento para la selección de sus acciones.

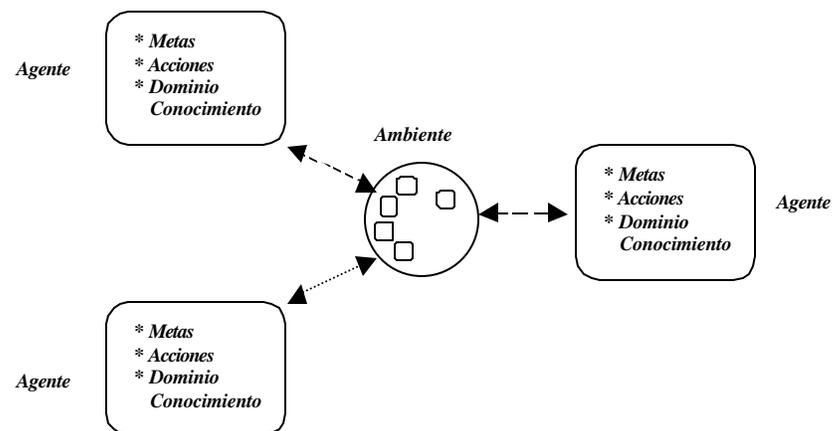


figura 2.3.- Sistemas multiagente homogéneo no comunicativos figura 9.2 en [22]

Desde el punto de vista del fútbol robótico, en el escenario homogéneo no comunicativo los oponentes no deben ser modelados como agentes, en este contexto los jugadores actúan como agentes reactivos o deliberativos en algún grado. Esto es, son reactivos cuando el agente actúa hacia el objetivo sin conocimiento previo, y es deliberativo cuando el agente actúa prediciendo cuál será la situación final del objetivo. Un ejemplo de agente en extremo reactivo puede ver el balón y moverse hacia él para patearlo como sea posible, además los jugadores pueden tener o no conocimiento de que ellos son parte del equipo. Por otro lado, los jugadores pueden ser provistos con un razonamiento deliberativo dependiendo de que si la acción de dirigirse hacia la pelota, o moverse en diferentes partes del campo de juego, es para defenderse o receptar pases [22].

2.3.3 Sistemas multiagente heterogéneos comunicativos

Anteriormente se definió que la principal diferencia entre los sistemas homogéneos y heterogéneos es que el bloque de actividades en los heterogéneos es diferente para cada agente, mientras que, en los homogéneos todos los agentes tienen definido un mismo bloque de actividades. De este modo, los sistemas heterogéneos comunicativos, de la figura 2.4, se caracterizan por que transmiten la información directamente entre varios agentes y, además esta comunicación puede ser transmitida punto a punto o a todos los agentes. Estos sistemas pueden ser heterogéneos a cualquier grado.

Dos de los problemas más estudiados en estos sistemas multiagente son protocolos de comunicación y teorías de compromiso.

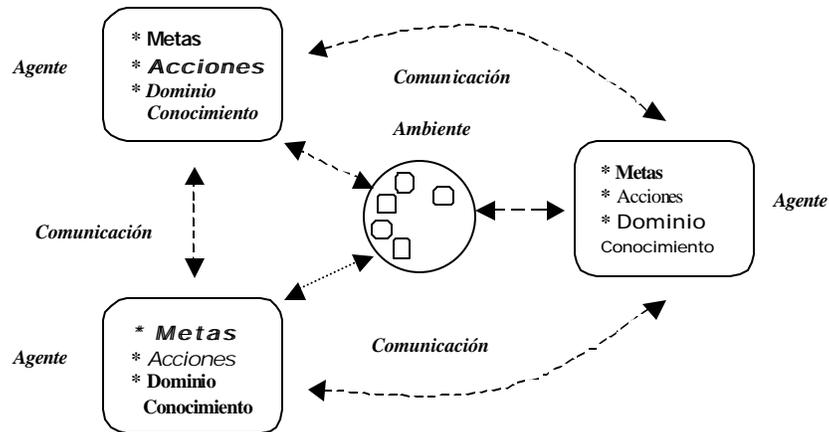


figura 2.4.- *Sistemas multiagente heterogéneos comunicativos* figura 9.5 en [22]

Dentro de los sistemas multiagente comunicativos, un aspecto importante que debe ser considerado es lo referente al uso de los lenguajes y protocolos de comunicación. Estos deben ser los mismos para que exista una interacción entre los agentes. En este sentido, algunos aspectos independientes de protocolos son: el contenido de la información, el formato del mensaje, y las convenciones de coordinación. Existen protocolos para estos tres niveles: *KIF* (*Knowledge Interchange Format*) [9] para contenido, *KQML* (*Knowledge Query and Manipulation Language*) [7] para formato del mensaje y *COOL* [3] para coordinación

El fútbol robótico es quizás el mejor escenario apropiado para el estudio de protocolos de comunicación. En algunos sistemas los jugadores envían mensajes a sus compañeros de equipo, para lo cual deben tener definido un lenguaje que les permita comprenderse, y de esta forma efectuar alguna acción en conjunto. Si los agentes son diferentes, la negociación de protocolos debe ser útil en el dominio de fútbol robótico, basado sobre sus perspectivas sensoriales diferentes, y sus diferentes opiniones acerca de que cursos de acción deben ser los mejores para el equipo. En un ambiente real, el tiempo es muy importante para el juego del equipo. Así la administración de recursos en términos de tiempo o coordinación de acciones es vital. Los protocolos son también necesarios para el compromiso de realizar diferentes roles del juego en equipo. Por ejemplo, en el juego puede haber un acuerdo para la recepción y ejecución de pases entre compañeros.

Cuando los agentes se comunican estos pueden decidir cooperar sobre un tarea dada o asignar una cantidad de tiempo. De esta manera ellos hacen compromisos con los otros agentes, estos compromisos involucran acuerdos para seguir una meta determinada, prescindiendo de cuanto esto le puede servir a sus propios intereses. Los compromisos pueden definir sistemas que se ejecuten más fácilmente proveyendo una forma en la que los agentes confíen mutuamente. Por tanto esta teoría de compromiso tiende a conseguir una considerable atención en su estudio [22].

2.3.4 Sistemas multiagente heterogéneos no comunicativos

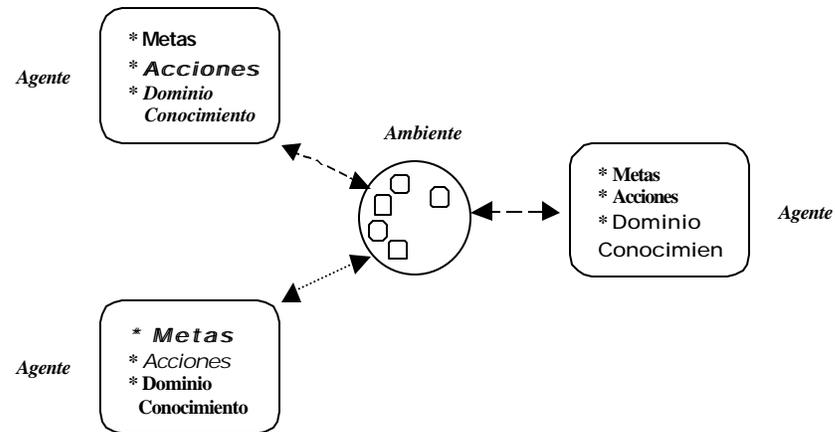


figura 2.5.- *Sistemas multiagente heterogéneo no comunicativos* figura 9.3 en [22]

En los sistemas multiagente heterogéneos no comunicativos, figura 2.5, los agentes son situados en ambientes diferentes, lo cual los obliga a tener diferentes entradas sensoriales y a su vez sus necesidades toman diferentes acciones. Sin embargo en este escenario los agentes poseen diferencias mucho más significativas en lo que respecta a sus metas, acciones y dominios de conocimiento.

Los agentes pueden ser heterogéneos considerando diferentes vías (tipos de sensores y efectores), desde tener diferentes metas hasta tener diferentes modelos de dominio y acciones.

Desde el punto de vista de fútbol robótico, como cada jugador tiene varios compañeros de equipo con los mismos propósitos globales y varios oponentes con metas opuestas, cada jugador es a la vez cooperativo y competitivo al mismo tiempo.

2.4 CLASIFICACIÓN DE SISTEMAS DE FÚTBOL ROBÓTICO.

Un importante problema para los diseñadores de robots consiste en determinar donde será ubicada la inteligencia del robot. En este sentido, existen robots cuya inteligencia está ubicada en el CPU de un computador, a los cuales llamaremos *robots sin inteligencia dentro del robot*. Mientras que en otros casos, la inteligencia está ubicada en el propio robot, a estos llamaremos *robots con inteligencia dentro del robot*.

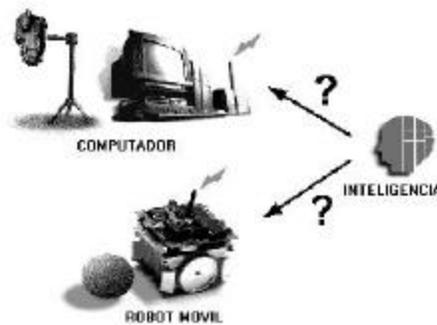


figura 2.6.- Ubicación de la inteligencia artificial [39]

Existen casos especiales donde parte de la inteligencia del robot está en el CPU y otra parte está en el robot mismo, cuando este sea el caso hablaremos

de *robots con inteligencia parcial*. Una gráfica que muestra una representación de lo anteriormente mencionado es dada en la figura 2.6.

En el fútbol robótico dos métodos que operan y están disponibles son: los sistemas de fútbol robótico basados en sensores de visión artificial, y sistemas de fútbol robótico basados totalmente en sensores incorporados en el robot [33] [35].

2.4.1 Sistemas de robots con inteligencia parcial dentro del robot, basados en visión.

En un sistema de fútbol robótico con inteligencia parcial, figura 2.7, los robots deben tener incorporado diferentes tipos de sensores, para el control de velocidad, control de posición, evitar obstáculos, etc. Estos sensores son útiles dado que ellos suministran información a la inteligencia del robot. Un importante sensor en un sistema de fútbol robótico es el sistema de visión.

Dado que estos sensores permiten proveer de ojos a los robots. De este modo, la información capturada por un sistema de visión es usada por un computador para definir a partir de esta las estrategias de juego que usarán los robots.

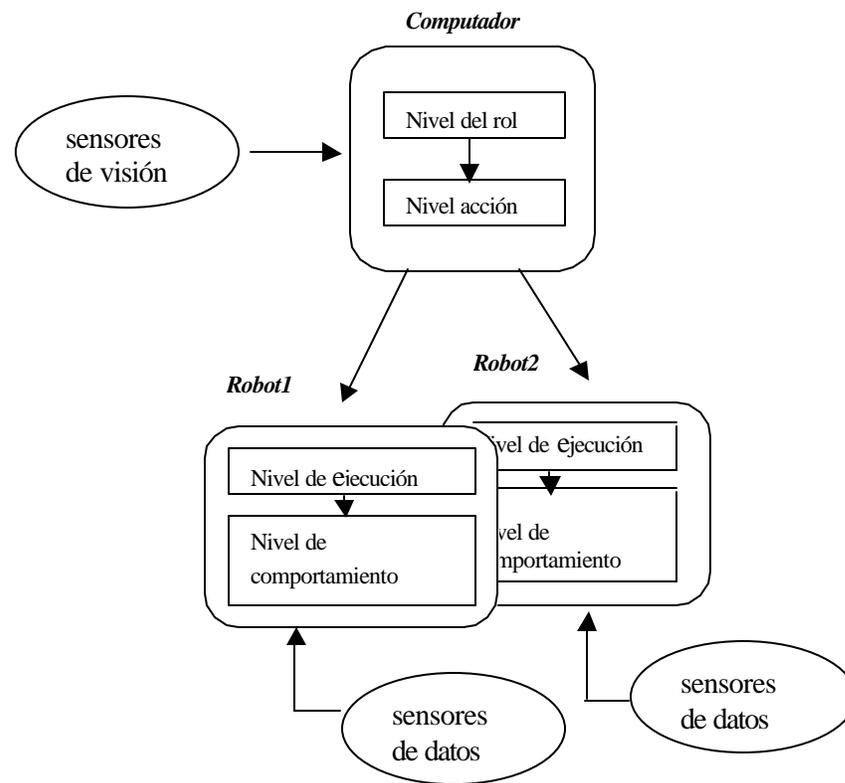


figura 2.7.- *Sistemas de robots con inteligencia dentro del robot, basados en visión [39].*

Los sistemas que posean robots con sensores incorporados, para controlar posiciones y evitar obstáculos (sensores infrarrojos), reducirán significativamente la carga de procesamiento para el computador.

De la figura 2.7 la estructura de control del robot es:

Nivel de rol: determina el rol de cada robot (defensor, atacante o arquero).

Nivel de acción: selecciona las acciones de cada robot (patear, rechazar, bloquear).

Nivel de ejecución: control del motor.

Nivel de comportamiento: seguir pelota o evitar obstáculos.

2.4.2 Sistemas de robots sin inteligencia dentro del robot, basados en visión.

En los sistemas de robots sin inteligencia dentro del robot, la inteligencia es controlada totalmente por un computador externo. Para esto, un sistema de visión es usado para capturar toda la información referente a cada robot y el balón. La información obtenida es procesada para elaborar las estrategias de juego del equipo que luego serán transmitidas a los robots por diferentes medios de transmisión. La figura 2.8 muestra un ejemplo de un sistema de este tipo. Es necesario recalcar que en un sistema remoto sin inteligencia cada robot tiene su propio mecanismo de manejo, una unidad de procesamiento y un módulo de comunicación.

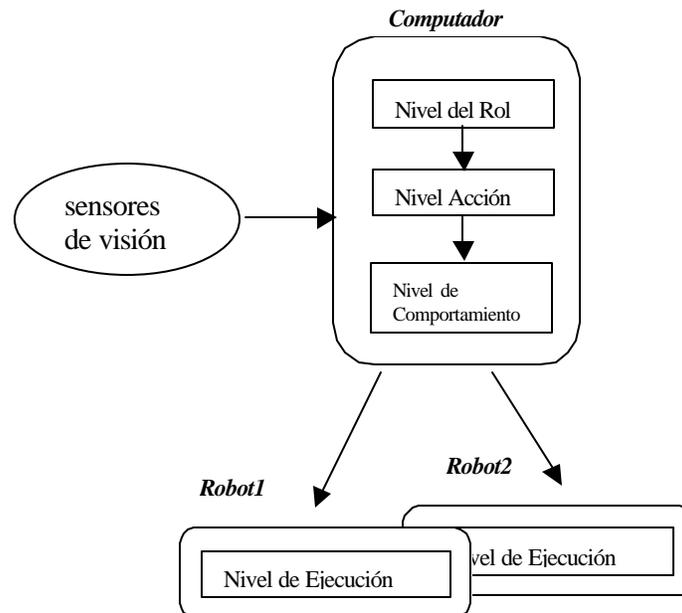


figura 2.8.- Sistemas de robots sin inteligencia dentro del robot, basados en visión[39].

Para controlar el posicionamiento exacto de los robots es necesario que el tiempo de adquisición de las imágenes tomadas por el sistema de visión sea muy pequeño para mejorar la precisión y exactitud de los algoritmos de control de movimiento de los robots. Del procesamiento de visión depende mucho para que el sistema tenga un buen rendimiento puesto que el robot recibe la información del ambiente de este componente. Los protocolos de comunicación para un sistema cooperativo de múltiples robots son completamente sencillos, el problema radica en que el costo computacional se incrementa cuando aumenta el número de robots debido a que el computador

debe atender y controlar simultáneamente a cada robot, y procesar la información del ambiente al mismo tiempo.

2.4.3 Sistemas de robots con inteligencia dentro del robot, basados totalmente en sensores incorporados.

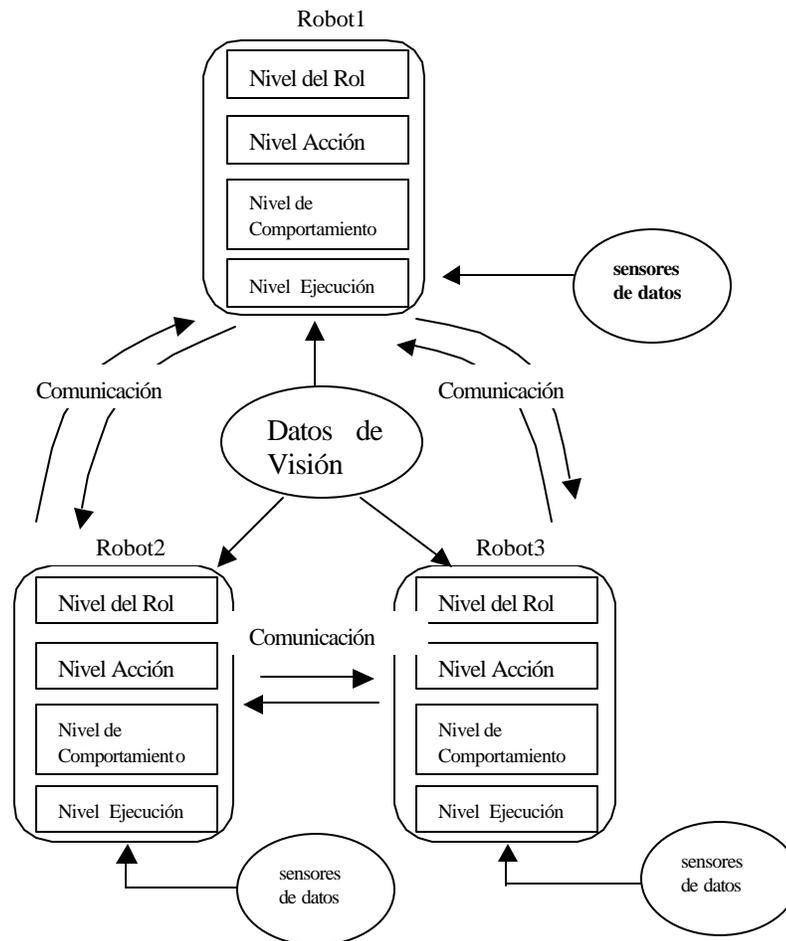


figura 2.9.- Sistemas de robot son inteligencia dentro del robot, basados totalmente en sensores incorporados [39].

En los sistemas de fútbol robótico con sensores incorporados en el robot, figura 2.9, cada robot va a tener alguna función de comportamiento autónomo. Todos los cálculos son hechos localmente en el robot. El computador realiza el proceso de visión sobre el posicionamiento del balón y robots. Cada robot decide sobre su propio comportamiento usando los datos recibidos por la parte de visión y sus sensores. Este puede ser visto como un sistema de control distribuido. Donde cada robot tiene su propia inteligencia, por tanto resulta más compleja la tarea de construir robots con funciones de control de velocidad, posicionamiento, evitar obstáculos, comunicación y control de decisión [16]. En estos sistemas el computador solamente procesa los datos de visión, y por tanto puede considerarse como un tipo de sensor [35].

2.5 COMPARACIÓN DE LOS SISTEMAS DE ROBOTS

	Ventajas	Desventajas	Propósito de investigación
Sistema de Fútbol Robótico sin Inteligencia.	Relativo bajo costo. Fácil para desarrollar.	No pueden usar sensores locales. Alto costo computacional.	Sistemas de Visión. Teoría de sistemas multiagente.
Sistema de Fútbol Robótico con Inteligencia parcial.	Conveniente para modularización.	Riesgo de incoherencia entre el computador y el robot	Sistemas basados en visión. Sistemas basados en Robot.
Sistemas de Fútbol Robótico con sensores incorporados en el Robot con inteligencia	Conveniente cuando son múltiples agentes. Pueden usar información local.	Alto costo de los robots y complejidad. Complicado al momento de construirlo.	Sistemas de Robot. Desarrollo de sistemas multiagente.

tabla 2.1.- Cuadro comparativo de los sistemas de fútbol robótico [39].

Los resultados obtenidos en la tabla 2.1 se basan en costos aproximados en el desarrollo y construcción de los distintos sistemas de fútbol robótico.

En las competiciones de fútbol robótico existen varias categorías, en las cuales se pretende demostrar los avances y aplicaciones en cada uno de estos sistemas. Así por ejemplo:

- Para sistemas de robots sin inteligencia basados en visión tenemos a los robots MIROSOT (*Micro Robot World Cup Soccer Tournament*), NAROSOT (*Nano Robot World Cup Soccer Tournament*), que son robots con dos ruedas cuyo costo oscila entre los \$ 800 US, el sistema de visión no está incorporado, por lo cual se pueden desarrollar los campos de visión por computador y sistemas multiagente, los mismos que son llevados a ejecución por el CPU externo.

- Para sistemas de robots con inteligencia parcial dentro del robot basados en visión, como SOTY II su precio oscila en \$1000 US, estos robots con dos ruedas poseen sensores para evitar obstáculos y para la detectar la presencia de la pelota. La ventaja que presentan estos sistemas es liberar carga computacional al CPU externo, permitiendo que éste último se encargue de procesar únicamente la información de visión y ejecute estrategias [39].

- Para sistemas de robots con sensores e inteligencia dentro del robot tenemos a los tipos KEPHERASOT (*Khepera World Cup Soccer Tournament*), ROBOSOT (*Robot World Cup Soccer Tournament*), SONY AIBO que son totalmente autónomos (visión, CPU, sensores incorporados), los costos de estos robots sobrepasan los \$1.500 US [37][38]. Por esto la principal desventaja es su costo y complejidad al momento de construirlo, versus la ventaja de tener información disponible proveniente de cada robot, con el objetivo de construir equipos de robots con la función de realizar trabajos complejos como son tareas de servicio, exploración y entretenimiento.

Según Kim en [36] un robot de la categoría HUROSOT (*Human Robot World Cup Soccer Tournament*), catalogado como un sistema con sensores e inteligencia incorporados, dentro del robot, de alta complejidad en sus movimientos, cuesta 10 veces más que un robot de la categoría MIROSOT (catalogado como un sistema sin inteligencia dentro del robot móvil con ruedas). Debido a que la tecnología en nuestro medio se encuentra aún en desarrollo, por los costos y el tipo de investigación, se decidió trabajar sobre los sistemas de robots móviles sin inteligencia incorporada basados en visión, los cuales permiten abordar la teoría de sistemas multiagente, visión por computador y mecatrónica.

2.6 ANTECEDENTES DE LAS COMPETICIONES DE FÚTBOL ROBÓTICO A NIVEL MUNDIAL

En Junio de 1993, un grupo de investigadores en Japón, entre ellos Minoru Asada, Yasuo Kuniyoshi, y Hiroaki Kitano, decidió lanzar una competición de fútbol robótico, tentativamente con el nombre de *Liga-J de Robots* (la Liga-J es el nombre japonés recientemente establecido en la liga de fútbol profesional). Luego de esto hubo una gran aceptación por parte de otras universidades e investigadores de otros países que en poco tiempo pidieron participar e integrarse en este evento científico. A partir de este punto se decidió realizar un campeonato mundial denominado *Robot World Cup*, o en su forma abreviada RoboCup. La primera competición en el torneo RoboCup se sostuvo en 1997 [20].

En los torneos de la RoboCup se determinan ciertas reglas y especificaciones que deben cumplir los participantes, entre ellas las siguientes:

- **Liga de robots reales:** Usando robots físicos para jugar fútbol robótico.
- **Liga de fútbol simulado:** Usando un software para simular a los robots jugando como si fuera un partido real por medio de un servidor llamado *Soccer Server*.

- **Desarrollo de campos de la inteligencia artificial:** desarrollo de sistemas multiagente.

En paralelo, en septiembre 1995, el profesor Jong-Hwan Kim formó un Comité Internacional llamado *Micro-Robot World Cup Soccer Tournament (MIROSOT)* y propuso el desarrollo de un campeonato. Se realizó un pre-campeonato MIROSOT en el *KAIST (Instituto Avanzado Coreano de Ciencia y Tecnología)* durante el 29 de julio al 4 de agosto de 1996, en el que 30 equipos de 13 países asistieron para conocer detalles técnicos y reglas para participar en el torneo. Posteriormente, la primera competición de *fútbol robótico MIROSOT' 96* tuvo lugar en el KAIST entre el 9 a 12 de noviembre de 1996, donde 23 equipos de 10 los países participaron. Luego, a partir de 1998 la liga MIROSOT toma el nombre de *FIRA (Federación Internacional de Fútbol Robótico Asociado)*, la cual consiste de robots con dimensiones de 7.5cm x 7.5cm x 7.5cm, donde cada equipo consta de 3 jugadores, aunque posteriormente se crearon otras categorías que utilizan otros tipos de robots [6] [20].

Hay muchas diferencias en el hardware y reglas que guardan la RoboCup y la FIRA como dos competiciones separadas. A pesar de los problemas polémicos que existen entre estas dos comunidades, ellas son un gran aporte para la investigación científica en el campo de la inteligencia artificial, mecatrónica, visión artificial, control automático y muchos otros campos más.

2.7 CONCLUSIONES

Este capítulo presentó al lector los diferentes enfoques por parte de varios investigadores en la construcción de sistemas de fútbol robótico, vistos como una buena plataforma de desarrollo para sistemas multiagente.

En base a la experiencia adquirida durante el transcurso del proyecto, se comprobó que nuestro sistema de fútbol robótico es sin inteligencia dentro del robot basado en visión, el mismo que se sitúa como un sistema multiagente homogéneo no comunicativo. Por tanto se ha aprovechado las ventajas que presentan estos sistemas, puesto que no requiere mucho tiempo en su construcción (resuelto por el componente electromecánico satisfactoriamente), además los costos de sus componentes presentados en la tabla 1.1 resultan accesibles para cualquier universidad, esto permite la construcción de una plataforma de investigación que brinde a estudiantes la oportunidad de incursionar en esta áreas.

El objetivo principal de quienes estamos inmersos en estas áreas, es que en un futuro no muy lejano construir robots con la suficiente inteligencia y habilidad para poder enfrentar a un equipo de fútbol conformado por humanos. Cabe señalar que según resultados de investigadores experimentados a nivel mundial, resulta muy complejo y costoso construir sistemas de robots basados

en sensores incorporados con inteligencia dentro del robot, debido a que su desarrollo se encuentra en sus etapas de prueba.

CAPITULO III

Control de movimiento de los micro-robots de la MIROSOT.

Puesto que la categoría de MIROSOT (*Micro-Robots World Cup Soccer Tournament*) usa robots reales móviles con ruedas que deben trasladarse desde un punto a otro en el campo de juego de la forma más eficiente y rápida, es necesario aplicar diversas técnicas de control que tomen en consideración las restricciones cinemáticas existentes en los robots cuando estos ejecutan sus movimientos.

En el siguiente trabajo se describen las técnicas aplicadas para controlar en tiempo real el movimiento de los robots usados en los torneos de la MIROSOT. Las técnicas implementadas se basan en un estudio bibliográfico sobre el control de robots móviles mediante métodos reactivos, se estudian las leyes de la cinemática que gobiernan a los robots de la categoría MIROSOT, los parámetros de entrada y las salidas de los algoritmos de control. Finalmente, se implementa un planificador de trayectorias usando campos potenciales de vectores unitarios, con el propósito de resolver dos importantes problemas que son: primero hacer que el robot llegue a un punto dentro del campo de juego, garantizando su posición y orientación final; y segundo para evitar obstáculos que pueden ser robots compañeros y robots oponentes.

3.1 INTRODUCCIÓN.

El fútbol robótico es un excelente banco de pruebas para la investigación en el campo de los sistemas multiagente, el juego de fútbol con robots se diferencia de los otros sistemas, puesto que los jugadores de un equipo deben cooperar entre sí, considerando además, que su ambiente de trabajo es continuamente alterado tanto por las acciones de los robots de su equipo como por las del equipo oponente. Por estas razones, los algoritmos de control e inteligencia artificial de este tipo de sistemas multiagente deben solucionar problemas que van desde niveles bajos relacionados con las leyes de la cinemática y la dinámica que gobiernan los movimientos de los robots, hasta la implementación de estrategias de alto nivel para competir con otros robots [15].

El construir sistemas de múltiples robots que deben desempeñarse como un equipo de fútbol, requiere por tanto, además de aplicar técnicas que mejoren el comportamiento del equipo, controlar correctamente los movimientos del robot. Esto último se debe a que puede ocurrir que la estrategia del equipo y la inteligencia del jugador sean las adecuadas en el campo de juego, pero si los robots muestran en sus movimientos un bajo rendimiento, el equipo presentará una gran desventaja frente a su equipo oponente.

Debemos considerar que el tipo de robots que utilizamos en los torneos de la MIROSOT no usan sensores incorporados en su hardware, además no presentan comunicación entre sí, y en su lugar usan un sistema de visión a través de una cámara que les permite sentir los eventos ocurridos en el campo de juego. La cámara está ubicada a 2 metros aproximadamente por encima del tablero de juego. Varias investigaciones acerca de técnicas de planificación y navegación de robots basados en sensores pueden encontrarse en [1],[5],[2] y [4].

Teniendo presente que nuestro sistema multiagente está compuesto por *agentes homogéneos reactivos* (puesto que todos reciben las mismas entradas, son capaces de realizar las mismas acciones y actúan dependiendo de las posiciones de los compañeros y de la pelota) *no comunicativos* (puesto que no existe comunicación entre ellos) resulta más conveniente aplicar un método de navegación autónomo reactivo para controlar los movimientos del robot.

Una vez que se ha resuelto el problema de controlar los movimientos del robot, se implementan las habilidades básicas como patear, seguir la trayectoria de la pelota, bloquear al oponente, etc., para luego aplicar una técnica de aprendizaje de máquina que defina un proceso de selección de la mejor acción en un determinado estado del juego.

Los métodos de control para robots móviles con ruedas pueden ser aplicados a otros campos donde se utilicen estos vehículos tales como operaciones de salvamento, misiones espaciales, exploración minera, etc.

3.2 DETERMINACIÓN DE LA CINEMÁTICA DEL ROBOT.

Para implementar las técnicas de control de movimiento de estos robots, debemos entender las leyes de la cinemática que gobiernan a este tipo de vehículos.

Los robots usados en los torneos de la MIROSOT se mueven utilizando dos ruedas, tal como se muestra en la figura 3.1. Cada rueda es controlada por un motor, analizaremos como primer punto este aspecto.

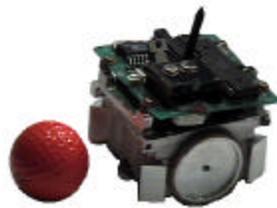


Figura 3.1.- Robot usado en los torneos de la MIROSOT.

Las ruedas de los robots mantienen un único *punto de contacto* con la superficie, como lo muestra la figura 3.2. Según esta figura, los radios de la rueda son perpendiculares a la superficie, esto hace que el vehículo no resbale en la dirección ortogonal a la del movimiento, y tenga deslizamiento de

traslación entre la rueda y el piso (rodamiento puro) en la dirección del movimiento. Esto es conveniente para los sistemas de fútbol robótico ya que los movimientos tienden a ser suaves.

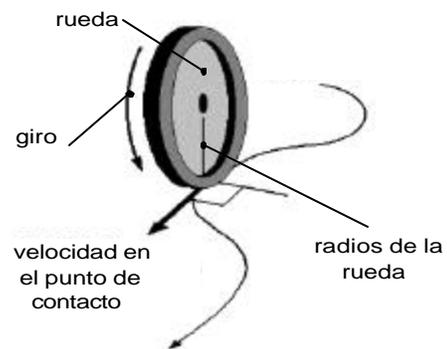


Figura 3.2.- Tipo de rueda del robot móvil

La figura 3.3 muestra la *dirección de la velocidad lineal* de una rueda con radio r y con velocidad angular w en el punto de contacto p .

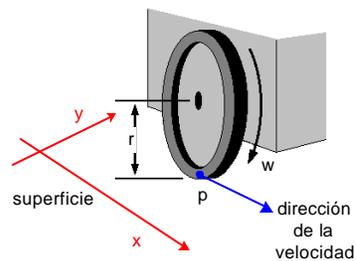


Figura 3.3.- Velocidad lineal y angular de una rueda del robot [33]

La velocidad en el punto p , punto de contacto con la superficie, es calculada como se muestra en la ecuación(3.1):

$$\text{Velocidad} = (r \times \omega) a_x \quad \text{donde } a_x \text{ es un vector unitario al eje X.} \quad (3.1)$$

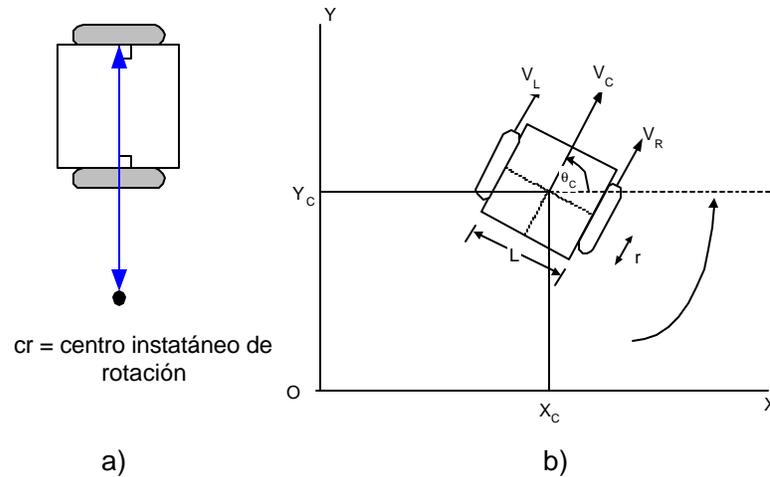


Figura 3.4.- a) centro instantáneo de rotación cr del robot, b) modelo de la cinemática del robot móvil [13]

En la figura 3.4.a se muestra el centro instantáneo de rotación cr (punto que cruza a todos los ejes de la rueda) del robot, el cual está definido por el centro donde gira el robot, mientras que la figura 3.4.b muestra el modelo de la cinemática del robot, donde r es el radio de las ruedas, L es el ancho del robot, V_R es la velocidad del motor derecho, V_L es la velocidad del motor izquierdo, V_C es la velocidad promedio de los dos motores, θ_c es el ángulo del robot respecto a las coordenadas absolutas x - y , y (X_c, Y_c) es el centro de masa del robot.

Típicamente los robots móviles con ruedas tienen tres grados de libertad, dos en la posición x - y y uno correspondiente a su dirección. Sin embargo, los micro-robots de la categoría MIROSOT sólo poseen dos grados de libertad controlables, ya que podemos desplazarlo en la dirección en la que mira el robot, o hacerlo girar, pero nunca moverlo lateralmente. Por tanto, como la cantidad de grados de libertad (tres) es mayor que la cantidad de grados de libertad controlables (dos), en la teoría se dice que este tipo de robots móviles con ruedas son *no holonómicos*.

La posición p y la orientación del robot puede ser representada por un vector p_s como se define en la siguiente ecuación:

$$p_s = \begin{bmatrix} x_c \\ y_c \\ \dot{\theta}_c \end{bmatrix}, \quad p = \begin{bmatrix} x_c \\ y_c \end{bmatrix} \quad (3.2)$$

Anteriormente se mencionó que (X_c, Y_c) es la posición del centro de masa del robot, y q_c es el ángulo del robot respecto a los ejes x - y . De esta forma, el vector de velocidad s del robot puede ser definido como:

$$s = \begin{bmatrix} v \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \frac{V_R + V_L}{2} \\ \frac{V_R - V_L}{L} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} V_L \\ V_R \end{bmatrix} \quad (3.3)$$

donde v es la velocidad de translación desde el centro del robot, w es la velocidad angular del robot con respecto a su centro. La ecuación (3.3) nos da la relación entre el vector velocidad s y las velocidades de las dos ruedas, V_L y V_R . Por tanto, los valores del vector s son aquellas que pueden controlarse en el robot.

Considerando la relación existente entre estas entradas de control, las velocidades V_L y V_R pueden calcularse como (3.4), y w y v como (3.5):

$$V_L = r \dot{\theta}_L \quad V_R = r \dot{\theta}_R \quad (3.4)$$

$$\dot{\theta} = \frac{V_R - V_L}{L} \quad v = \frac{V_R + V_L}{2} \quad (3.5)$$

La cinemática asociada con la *matriz Jacobiana* $J(\mathbf{q}_c)$ y el vector de velocidad s

$$\dot{p}_s = \begin{bmatrix} \cos \hat{\theta}_c & 0 \\ \sin \hat{\theta}_c & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = J(\hat{\theta}_c) s \quad (3.6)$$

Para conseguir la posición y la orientación (3.2) se debe satisfacer la siguiente restricción no-holónica:

$$H \bullet \dot{p} = (\sin \theta \quad -\cos \theta) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (3.7)$$

donde H es un vector ortogonal al plano formado por las ruedas del robot. Esta ecuación es igual a la expresión $\tan(\theta) = dx/dy$ y significa que la dirección de la velocidad en cada instante es la misma que la del ángulo frontal de los robots o que la dirección del movimiento instantáneo del robot es la misma que la dirección del lado frontal del robot, lo cual implica un rodamiento puro y se asume que no existirá deslizamiento.

Para hacer que el robot se mueva rápidamente además de la velocidad máxima se debe considerar la región donde el robot no derrapará, o deslizará. Una de las principales causas para el derrape es la gran fuerza centrífuga ejercida en el límite de fricción de las ruedas [13].

Dos ecuaciones que definen las restricciones de los robots móviles con ruedas son las siguientes:

$$|V_L| \leq V_m \quad , \quad |V_R| \leq V_m \quad (3.8)$$

$$|\nu \omega| \leq R_m \quad (3.9)$$

donde V_m es la velocidad máxima de cada rueda (150 cm/seg) y R_m es la máxima velocidad de giro (500 rad/seg). Debido a la restricción (3.9) el robot puede ser controlado considerando la máxima fuerza centrífuga.

3.3 MÉTODO DE CAMPOS POTENCIALES PARA POSICIONAMIENTO Y ORIENTACIÓN DEL ROBOT.

Un importante indicador de la eficiencia de un equipo de fútbol con robots es la habilidad que poseen sus jugadores para dirigir la pelota hacia la portería contraria. Esta tarea no es fácil tomando en cuenta que el entorno de trabajo de un sistema de fútbol robótico sufre continuos cambios, ocasionados principalmente por los robots y la pelota presentes en el campo de juego. Esto último se puede complicar aún más si el equipo contrario presenta un buen plan defensivo. Por esta razón es conveniente incluir movimientos en el robot para que eviten colisionar con robots oponentes e inclusive con los robots compañeros, con la finalidad de conducir la pelota hacia el arco contrario [22].

Entre los robots móviles con rueda existen dos tipos de planificación de trayectorias que pueden ser implementados: *planificación fuera de línea* figura 3.5.a y *planificación en línea* figura 3.5.b. La planificación fuera de línea tiende a producir una trayectoria óptima pero con un alto costo computacional. Este tipo de planificador calcula con anticipación una trayectoria a seguir permitiendo al robot controlar la velocidad a seguir por ese camino. Por otro lado, en la planificación en línea las trayectorias pueden

ser sub-óptimas pero son adaptables al medio en que se las calcula con un costo computacional menor. Debido a que los entornos de trabajo en un sistema de fútbol robótico cambian continuamente es recomendable usar por tanto algoritmos de planificación en línea.

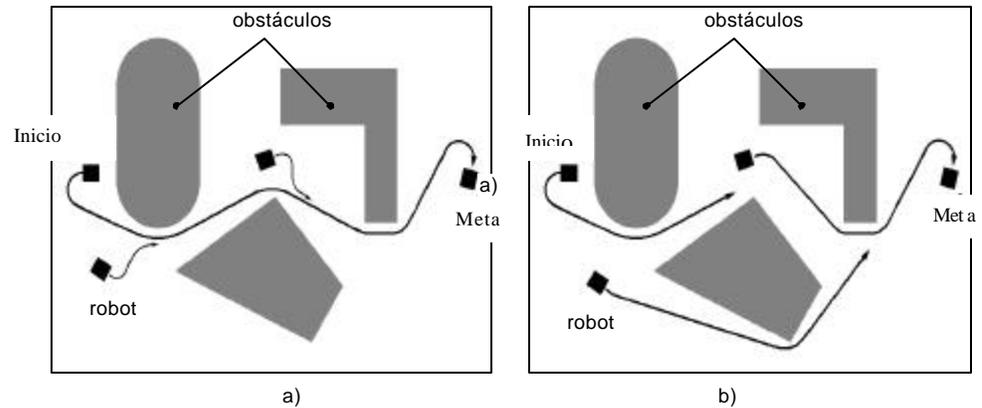


Figura 3.5.- Tipos de planificación de trayectorias: a) fuera de línea, b) planificación en línea.

Entre los algoritmos de planificación en línea, la efectividad de los métodos convencionales de navegación reactiva para llegar al objetivo no está garantizado puesto que los objetos del entorno de trabajo se encuentran en continuo movimiento [14].

Un conocido algoritmo de planificación en línea es el algoritmo de funciones potenciales [1] [34], el cual se caracteriza por que genera diferentes tipos de vectores de tal forma que el robot es orientado en la dirección de esos vectores. Estos vectores son generados continuamente permitiendo controlar a los robots en tiempo real. Los vectores generados por el algoritmo de

funciones potenciales pueden ser de al menos dos tipos [1] [15]: uno de atracción hacia el objetivo y otro de repulsión hacia los obstáculos sean estos adversarios o compañeros de equipo, e inclusive en algunos casos la pelota. Por tanto, la planificación consistirá en una adecuada combinación de ambos tipos de vectores.

La figura 3.6.a muestra una situación real de un partido, donde el robot requiere aplicar algún algoritmo de planificación para dirigirse a la pelota atravesando obstáculos, como son los oponentes y compañeros de equipo. La figura 3.6.b muestra la aplicación del algoritmo de funciones potenciales para esta situación, y la figura 3.6.c la trayectoria generada en línea.

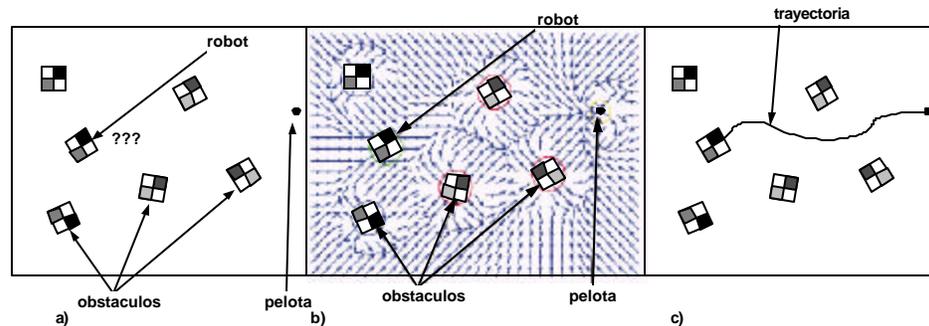


Figura 3.6.- a) Identificación del problema de planificación de trayectorias en el fútbol robótico. b) Generación de campos potenciales para la posición final y evasión de obstáculos. b) Trayectoria generada en línea mediante el algoritmo de campos potenciales.

Un trabajo sobre campos vectoriales para navegación considerando las restricciones (3.8) y (3.9) es presentado por Kim en [13], [15]. Este algoritmo

permite al robot ir a una posición deseada a gran velocidad sin oscilaciones o movimientos ineficientes.

Para facilitar la implementación del algoritmo, se asume que la magnitud de los vectores en cada momento tiene el valor de la unidad [13], es por ello que estos campos son llamados *campos univectoriales*.

Para cada tarea del robot sólo se calculan los vectores de los campos que se generan en el centro de masa del robot. En este capítulo se presentarán dos tipos de campos univectoriales: uno resolverá la tarea de guiar al robot a una posición final con la orientación adecuada a un objetivo, y otro resolverá la tarea de evitar obstáculos. La planificación de trayectorias en línea consistirá en la combinación adecuada de estos dos campos.

El campo univectorial N , se define como sigue:

$$N : F \rightarrow I \tag{3.10}$$

donde F es el espacio de trabajo en R^2 e I es un conjunto de vectores unitarios con ángulos arbitrarios. Se asume que la magnitud de los vectores es la unidad. La función de campos potenciales debe ser diferenciable en los límites de los obstáculos, debe tener un ángulo ortogonal a la superficie de los obstáculos en la dirección hacia fuera del obstáculo. Como se utilizan vectores

normalizados el campo vectorial N puede ser representado en términos de ángulos como lo siguiente:

$$\mathbf{f}: F \rightarrow [-p, p] \quad (3.11)$$

Mientras se controla el robot, estos vectores unitarios corresponden a la dirección frontal del robot deseada.

3.4 IMPLEMENTACIÓN DEL CAMPO POTENCIAL DE ATRACCIÓN PARA PATEAR LA PELOTA.

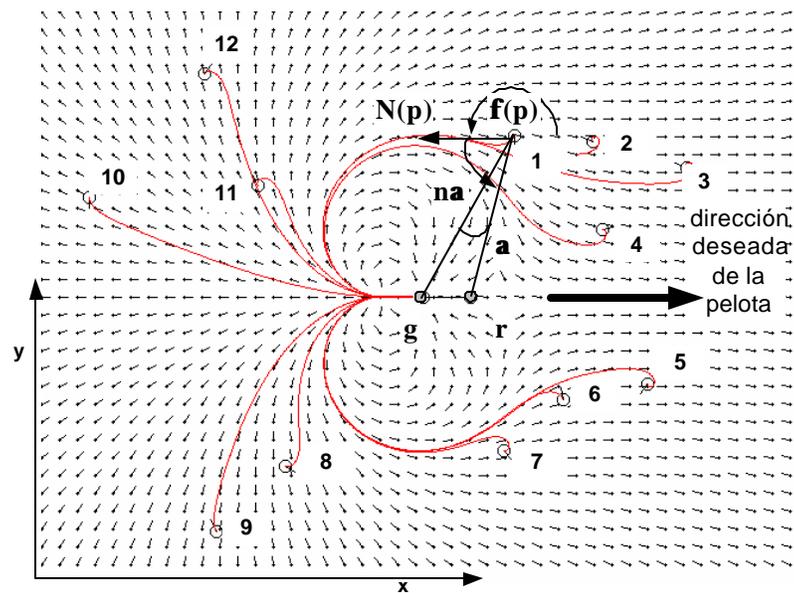


Figura 3.7.- Campo univectorial para una posición g deseada.

La figura 3.7 muestra el campo univectorial para que un robot ubicado en la posición inicial 1 pueda llegar al punto g y patear la pelota hacia una dirección deseada. Las pequeñas flechas representan la dirección del campo potencial en un determinado punto. Las líneas curvas rojas representan las posibles trayectorias que puede seguir el robot desde doce posiciones iniciales.

De forma general, el campo univectorial para una posición p del robot se define como $N(p)$ de magnitud igual a 1, y su ángulo $f(p)$ puede ser generado por:

$$\mathbf{f}(p) = \angle pg - n\mathbf{a} \quad (3.12)$$

$$\mathbf{a} = \angle pr - \angle pg \quad (3.13)$$

donde n es una constante positiva, g es el punto final destino del robot, y r determina la orientación final deseada. El símbolo \angle denota el ángulo del vector. El campo vectorial y el movimiento de giro del robot varían de acuerdo con el parámetro n y la distancia entre los dos puntos g y r . Usando (3.12) podemos obtener un *vector univectorial* para la posición g deseada.

Conforme se aplica esta técnica en el robot para el punto destino g a lo largo del campo vectorial calculado, el ángulo tiende a ser cero. Como consecuencia de este movimiento el robot se ubica en el punto destino g . La distancia entre los puntos g y r es ajustada para generar el campo univectorial. Los valores calculados como (3.14) son usados para reducir el error del ángulo entre el robot y los campos vectoriales:

$$\dot{u} = K_p \dot{e}_e + K_D \ddot{e}_e \quad (3.14)$$

$$\dot{e}_e = \angle N(p) - \dot{e}_e \quad (3.15)$$

$$\ddot{e}_e = \frac{d\dot{e}_e}{dt} \quad (3.16)$$

donde $N(p)$ es el vector del campo potencial en la posición p de coordenadas (x,y) con magnitud unitaria, q_e es el error en el ángulo entre la dirección frontal del robot y la dirección del campo, \dot{e}_e es la derivada de q_e , K_p es la ganancia de realimentación proporcional, K_D es la ganancia de realimentación derivada en el tiempo, como se muestra en la figura 3.8.

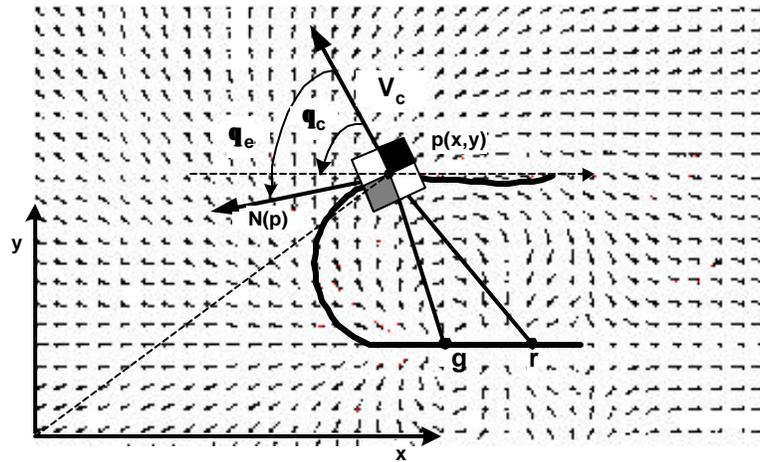


Figura 3.8.- Variables usadas en el cálculo del campo potencial de atracción $N(p)$

Se asume que la velocidad de traslación V_c del robot es constante. Si $V_c=0$ el ángulo frontal del robot tomará la dirección de $N(p)$ sin variar su posición. Con las ecuaciones (3.17) y (3.18) es posible controlar el movimiento del robot a través de las velocidades de la rueda izquierda y derecha como se mencionó al principio del capítulo, estas ecuaciones están en función del tiempo y se definen como:

$$V_R = V_c + K_p \dot{e}_c + K_D \ddot{e}_c \quad (3.17)$$

$$V_L = V_c - K_p \dot{e}_c - K_D \ddot{e}_c \quad (3.18)$$

donde V_C es la velocidad del centro del robot, V_R y V_L son las velocidades de la rueda derecha e izquierda respectivamente.

Algo importante de destacar sobre la gráfica del campo univectorial es que en el área cercana al punto destino g se crea una región horizontal estrecha, figura 3.9 en esta región el robot puede presentar oscilaciones en su movimiento. Para evitar este problema en [13] se recomienda que esta región deba ser expandida con un ancho igual al ancho del robot, como se muestra en la figura 3.10.

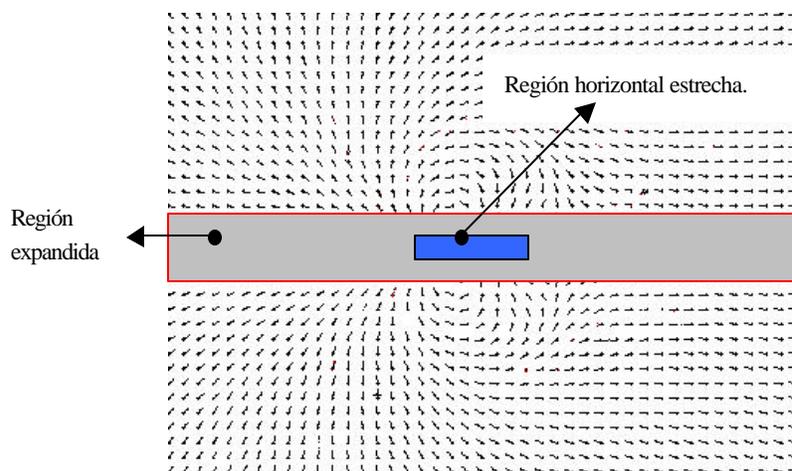


Figura 3.9.- Generación de la región expandida

Considerando lo anterior, el algoritmo se modifica tal como sigue. Los puntos g y r se proyectan en dirección vertical hacia el borde superior de la región expandida. De esta forma, se obtienen los puntos g' y r' como se muestra en la figura 3.10, y un nuevo campo univectorial es generado para la posición g a

partir de g' y r' . Este nuevo campo resultó de la integración del campo generado para las zonas fuera de la región expandida y el campo generado para la región expandida. Las direcciones del campo univectorial para las zonas fuera de la región expandida son generadas usando el algoritmo anterior, mientras que las direcciones del campo dentro de la región expandida se orientan de acuerdo a la dirección deseada de la pelota. La figura 3.10 muestra el nuevo campo univectorial para la posición deseada g .

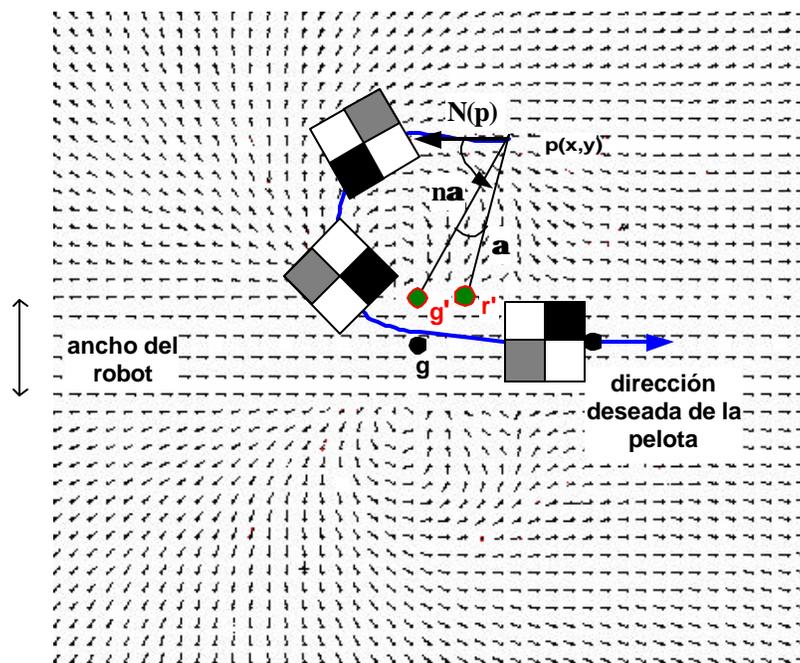


Figura 3.10.- Nuevo campo univectorial para la posición deseada g .

3.5 IMPLEMENTACIÓN DEL CAMPO POTENCIAL DE REPULSIÓN PARA EVITAR OBSTÁCULOS.

Inicialmente la dirección del robot en una posición p sigue la dirección del *campo univectorial de atracción* $N(p)$ hacia el objetivo, explicado en la sección 3.4. Sin embargo, cuando el robot se encuentra en presencia de un obstáculo, la dirección del campo univectorial debe ser modificada para evitar chocar con el obstáculo, definiendo un *nuevo campo potencial de repulsión* $R(p)$ alrededor del obstáculo [13][14].

Para generar $R(p)$ en la posición p del robot se define un radio R_o desde el centro del obstáculo hacia un poco mas allá del borde del círculo englobante del obstáculo (círculo de línea roja), figura 3.11. De esta forma, la dimensión de R_o tiene una holgura suficiente para que el robot no colisione con el obstáculo. El *nuevo campo univectorial* $R(p)$ se orienta en dirección tangencial al borde del círculo de radio R_o , y una línea L desde la posición p del robot hacia la posición del centro del obstáculo forma un triángulo rectángulo.

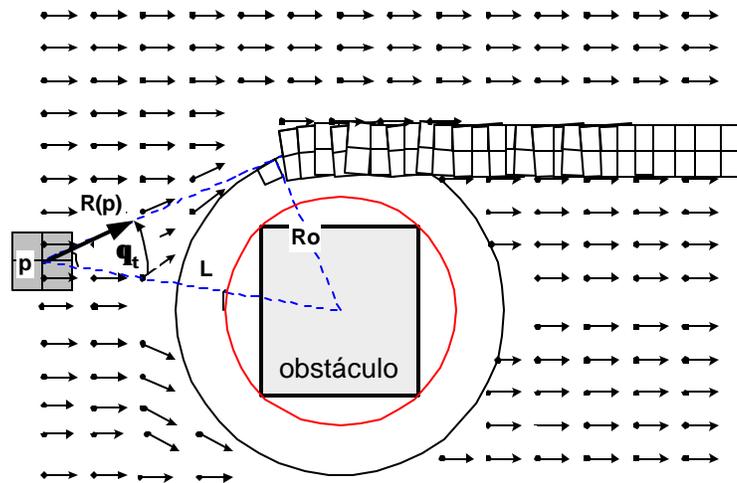


Figura 3.11.- Generación del campo univectorial de repulsión para evitar un obstáculo

Para propósitos de la implementación del algoritmo de generación de $R(p)$ se define un anillo de grosor M_o , el cual es aproximadamente igual al ancho del robot, alrededor de R_o , tal como se muestra en la figura 3.12 (anillo sombreado de rojo). Este anillo define la zona del nuevo campo univectorial $R(p)$ que permite evitar el obstáculo, una vez fuera de este anillo el robot usa el campo univectorial de atracción $N(p)$ para ir hacia el objetivo.

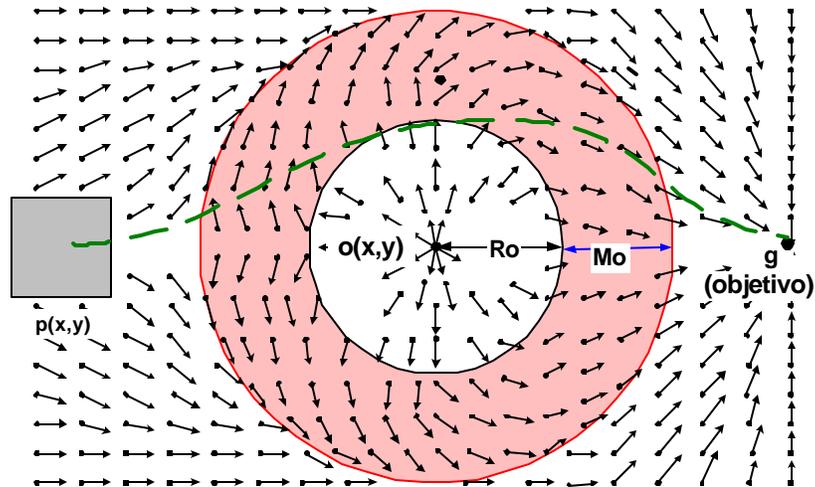


Figura 3-12.- Zona del campo univectorial $R(p)$ que permite evitar un obstáculo

Adicionalmente según las restricciones dadas por (3.8), los valores de velocidad deben estar en un rango para que el robot no resbale o se vuelque. En nuestra implementación, establecimos que el rango de la velocidad lineal esté definido entre $[-50, 50]$ cuando el robot está dentro del anillo formado por Mo . En esta zona definida por Mo el valor del campo modifica los valores de las velocidades V_L y V_R correspondientes a los motores izquierdo y derecho del robot respectivamente.

Para la descripción del algoritmo de generación de $R(p)$ usaremos (ox, oy) para referirnos a las coordenadas del obstáculo; (px, py) para referirnos a las coordenadas del robot; y (gy, gx) para las coordenadas del objetivo (pelota) en los ejes x-y respectivamente, tal como se muestra en la figura 3.13.

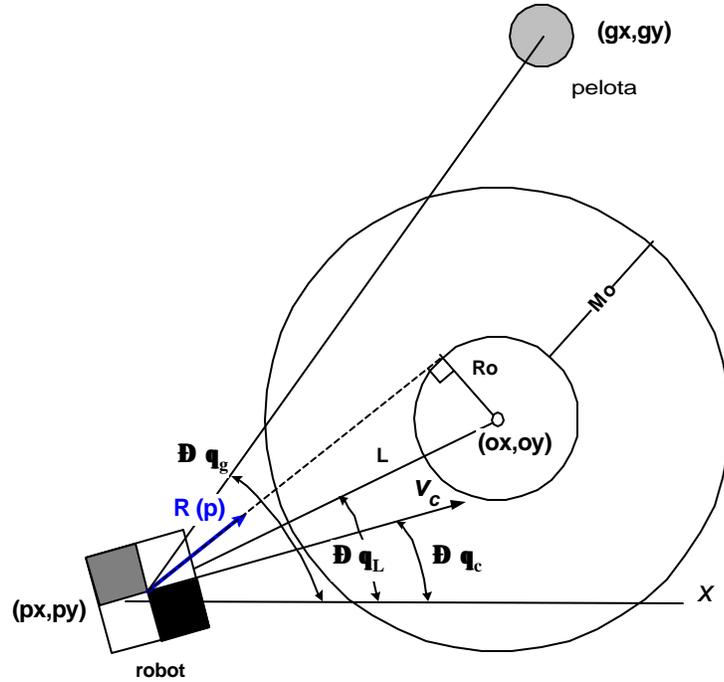


Figura 3.13.- Variables usadas en el algoritmo de generación del campo $R(p)$

El cálculo de L , el ángulo q_L de L y el ángulo q_g de la pelota con respecto al robot lo calculamos de la siguiente manera:

$$L = \sqrt{(px - ox)^2 + (py - oy)^2} \quad (3.19)$$

$$\angle q_L = \tan^{-1}((py - oy)/(px - ox)) \quad (3.20)$$

$$\angle q_g = \tan^{-1}((py - gy)/(px - gx)) \quad (3.21)$$

Para conocer cual será el ángulo del campo univectorial $R(p)$, desarrollamos el siguiente algoritmo descrito en la figura 3.14.

```

// Las líneas encerradas por // son comentarios

// ángulo 1 =  $\angle q_L$ 

// ángulo 3 =  $\angle q_g$  (ángulo del objetivo con respecto a la posición del robot)

// ángulo 2 = ángulo tangencial al obstáculo

// a_campo =  $\angle R(p)$  ángulo deseado (campo potencial)

// dentro del sector circular

if (Ro <= L && L <= Mo + Ro) {

    //ángulo tangencial al obstáculo

    angulo2=(180.0/PI * asin (Ro / L));

    //cálculo de la orientación del campo potencial
    if (angulo3 > 0 && angulo1 > 0)
        if ((angulo3 - angulo1) <= 0)
            a_campo= angulo1 - angulo2;
        else
            a_campo= angulo1 + angulo2;
    if (angulo3 < 0 && angulo1 > 0)
        if ((angulo3 - angulo1) <= 0)
            a_campo= angulo1 + angulo2;
        else
            a_campo= angulo1 - angulo2;
    if (angulo3 < 0 && angulo1 < 0)
        if ((angulo3 - angulo1) <= 0)
            a_campo= angulo1 - angulo2;
        else
            a_campo= angulo1 + angulo2;
    if (angulo3 > 0 && angulo1 < 0)
        if ((angulo3-angulo1) <= 0)
            a_campo= angulo1 + angulo2;
        else
            a_campo= angulo1 - angulo2;
}

if (L < Ro) { //el robot está muy cerca del obstáculo

    angulo2=(180.0/PI * atan2(Ro + Mo,L));
}

```

```

//calculo de la orientación del campo potencial
if (angulo3 > 0 && angulo1 > 0)
    if ((angulo3-angulo1) <= 0)
        a_campo= angulo1 - angulo2;
    else
        a_campo= angulo1 + angulo2;
if (angulo3 < 0 && angulo1 > 0)
    if ((angulo3 - angulo1) <= 0)
        a_campo= angulo1 + angulo2;
    else
        a_campo= angulo1 - angulo2;
if (angulo3 < 0 && angulo1 < 0)
    if ((angulo3 - angulo1) <= 0)
        a_campo= angulo1 - angulo2;
    else
        a_campo= angulo1 + angulo2;
if (angulo3 > 0 && angulo1 < 0)
    if ((angulo3 - angulo1) <= 0)
        a_campo= angulo1 + angulo2;
    else
        a_campo = angulo1 - angulo2;
}

//  $q_e$  entre el campo univectorial resultante y ángulo del robot

theta_err= angulo_robot - a_campo;

```

Figura 3.14.- Algoritmo de definición del campo vectorial para evitar obstáculos

Para el cálculo de las velocidades en la zona definida por M_o los valores de K_d y K_p de la ecuación (3.17) y (3.18) fueron 0.5 y 0.7 respectivamente, los mismos que se determinaron mediante ajustes experimentales. Investigaciones sobre métodos *heurísticos* y *programación evolutiva* que permiten calcular campos univectoriales más eficientes dependiendo de la situación en que se encuentra el ambiente de trabajo del robot pueden ser encontrados en [5].

Para determinar cuando se debe aplicar *el campo univectorial de repulsión* $R(p)$ para evitar obstáculos se ha propuesto considerar lo siguiente:

Inicialmente se calcula la distancia d desde el punto del obstáculo (ox, oy) hacia la recta formada por los puntos (gx, gy) y (px, py) , como muestra en la figura 3.15.

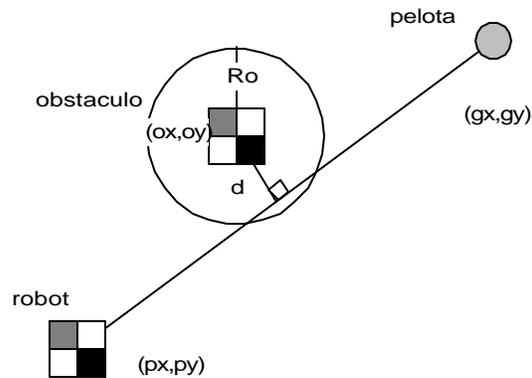


Figura 3.15.- Distancia entre el obstáculo y la recta formada por el robot y el objetivo (pelota).

Luego, el criterio propuesto para saber si el robot está próximo a colisionar es el siguiente: la distancia d debe ser menor a Ro cuando el robot está dentro de la zona definida por Mo ; y la distancia entre (gx, gy) y (px, py) es mayor que la distancia entre (ox, oy) y (gx, gy) menos la longitud del robot. Para nuestro caso, los robots tienen una longitud de 7.5 cm. Si esta condición se cumple se estima que habrá colisión, y se procede a aplicar el campo de repulsión y el campo de atracción descritos anteriormente, con la finalidad de orientar el robot móvil hacia el objetivo. Los resultados de la aplicación del algoritmo se describen en la sección 3.7.

3.6 FUNCIÓN DE PREDICCIÓN PARA ESTIMAR LA TRAYECTORIA DE LA PELOTA.

Esta función es usada por el arquero y los defensas con la finalidad de permitirles bloquear la trayectoria de la pelota cuando esta se dirige hacia su portería. Esta función es la encargada de simular cual será la siguiente posición que tendrá la pelota, cuando esta está en movimiento y en dirección hacia la portería. Las siguientes posiciones son estimadas en base a posiciones anteriores de la pelota.

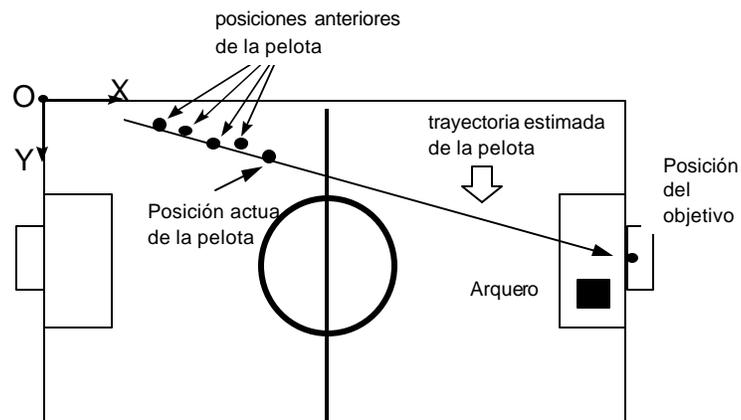


Figura 3.16.- Predicción de la pelota

Para esto, la trayectoria inicial de la pelota es aproximada a una línea recta usando las posiciones anteriores de la pelota, como se muestra en la figura 3.16.

La estimación de la siguiente posición de la pelota se calcula con la aproximación de *suma de mínimos cuadrados* (3.22), esta función se

desempeña adecuadamente cuando la pelota no modifica mucho la trayectoria y cuando no se consideran puntos espurios. A continuación se muestra como se calculan la recta aproximada en base a un conjunto de posiciones de la pelota.

$$Y = \hat{a}_0 + \hat{a}_1 x + \hat{\epsilon} \quad (3.22)$$

donde Y es al variable dependiente, x variable independiente $\hat{\epsilon}$ un error

\hat{a}_0 coordenada y al origen de la recta y \hat{a}_1 pendiente de la recta

$$\text{Pendiente} = \hat{a}_1 = \frac{SS_{xy}}{SS_{xx}} \quad (3.23)$$

$$\text{Coordenada al origen} = \hat{a}_0 = \bar{y} - \hat{a}_1 \bar{x} \quad (3.24)$$

donde :

\bar{y} es el promedio en y, \bar{x} es el promedio en x

$$SS_{xy} = \sum_{i=1}^n x_i y_i - \frac{\left(\sum_{i=1}^n x_i \right) \left(\sum_{i=1}^n y_i \right)}{n} \quad (3.25)$$

$$SS_{xx} = \sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i \right)^2}{n} \quad (3.26)$$

donde n es el número de posiciones observadas (tamaño de la muestra)

3.7 ANÁLISIS DE RESULTADOS.

Para probar los algoritmos de control basados en campos univectoriales (potenciales) se utilizaron robots de 7.5 cm x 7.5 cm de la categoría MIROSOT.

Como se mencionó anteriormente las velocidades de los dos motores se deben acotar en un rango $[-50,50]$ para mantener un mejor control cuando el robot esta en presencia de obstáculos. Debido a que, a grandes velocidades el algoritmo no funcionaba correctamente.

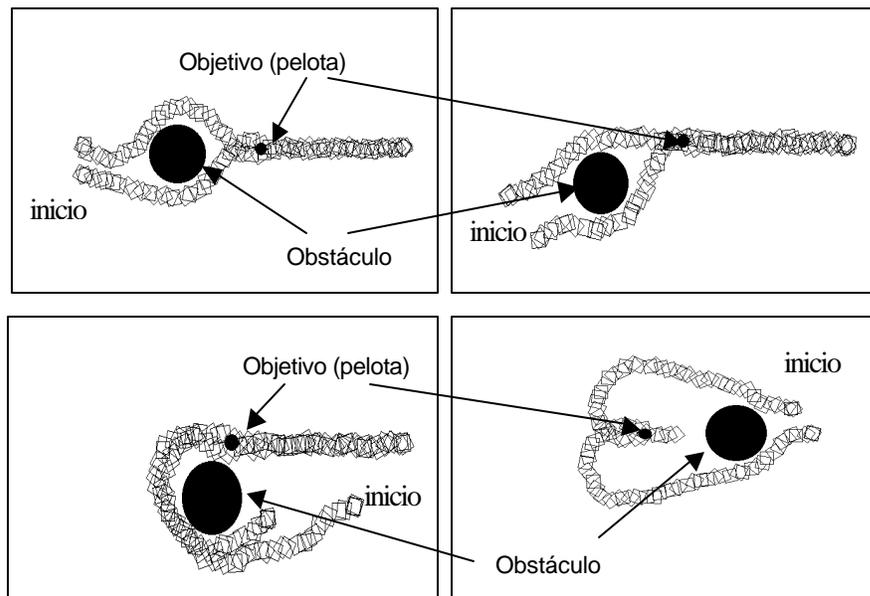


Figura 3.17.- Comportamiento del algoritmo de evitar obstáculos.

Los valores de restricción utilizados para la velocidad máxima lineal y velocidad máxima de giro fueron $V_m = 150 \text{ cm/s}$, $R_m = 500 \text{ rad cm/s}$ respectivamente. La figura 3.17 muestra varias trayectorias ideales que podría tomar el robot con la finalidad de evitar chocar con los obstáculos.

La figura 3.18 presenta el resultado de aplicar el algoritmo de evitar obstáculos sobre una situación de juego en un sistema MIROSOT de 3 vs. 3. En esta figura se aprecia que la trayectoria que sigue el robot para llegar a la pelota sigue hacia la parte superior del tablero de juego en lugar de la parte inferior. Esto se debió a que, el robot estaba inicialmente orientado hacia esa dirección.

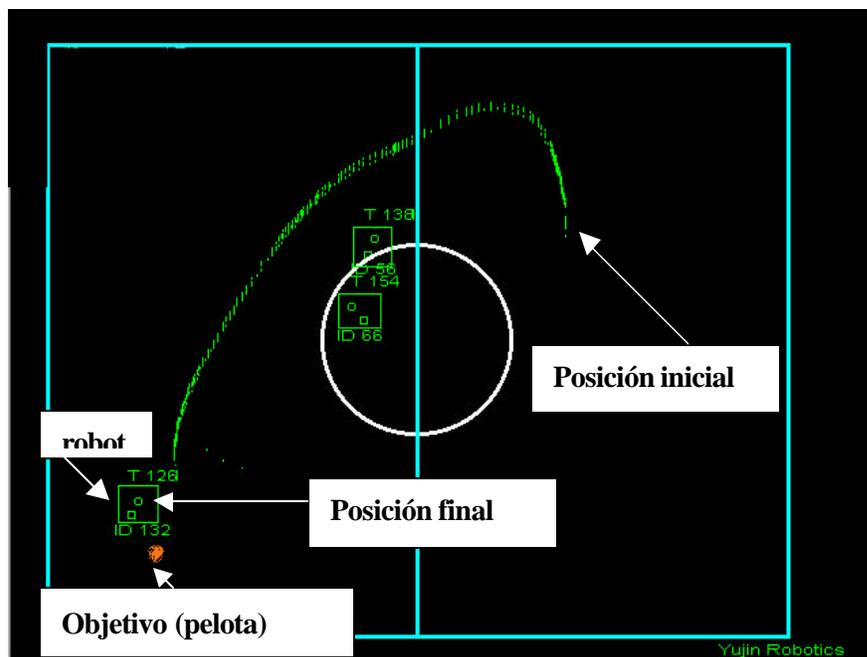


Figura 3.18.- Resultado de la aplicación del algoritmo de evitar obstáculos sobre el sistema MIROSOT de 3 vs. 3

Con relación a la aplicación de la función de predicción de la posición futura de la pelota, es importante configurar adecuadamente los colores de los robots y de la pelota en el sistema de visión, tal que no se detecten posiciones falsas (inexistentes) de la pelota.

3.8 CONCLUSIONES.

Este capítulo presenta un algoritmo para evitar obstáculos mediante la aplicación de la técnica de campos potenciales. Adicionalmente, aprovecha esta técnica para la implementación de un algoritmo de atracción hacia un objetivo (en nuestro caso la pelota). Y finalmente implementa una función para predecir la próxima posición de la pelota cuando éste sigue una trayectoria definida.

Según los resultados obtenidos en los partidos, la utilización de los algoritmos de campos potenciales, mejoró significativamente el control de movimiento del robot móvil con ruedas en un ambiente de trabajo tan dinámico y complejo como es el fútbol robótico.

Además hay que indicar que cuando existe más de un obstáculo frente al robot o un obstáculo de mayor tamaño, el robot puede quedarse oscilando, debido a que varios campos actúan en el robot modificando sus valores de velocidad. En esos casos se pueden definir objetivos temporales fuera de la presencia del

obstáculo, donde el robot deberá ir, y retomar luego el camino hacia su objetivo inicial.

CAPITULO IV

Aprendizaje por refuerzo aplicado al fútbol robótico

En este capítulo se presenta la aplicación del aprendizaje por refuerzo para la coordinación y cooperación entre agentes, en un entorno tan dinámico y complejo como es el fútbol robótico. En particular se implementa una técnica de aprendizaje de máquina bajo un esquema de aprendizaje Q modular, que realiza el proceso de selección de la mejor acción, basados en los valores-Q generados por cada agente. Resultados experimentales y conclusiones de la evaluación de esta técnica son presentados al final del capítulo.

4.1 INTRODUCCIÓN

La construcción de una buena estrategia de juego para un equipo de fútbol robótico puede resultar muy compleja, ya que existen muchas situaciones que se presentan de una manera diferente en cada partido, y por tanto, se deben analizar cada una de ellas para decidir que acción deben ejecutar los robots de un equipo.

Una manera óptima de realizar este trabajo es aplicar una técnica de aprendizaje de máquina, de tal forma que los robots decidan, de manera autónoma, que acción ejecutar en cualquier estado del juego.

Muchas técnicas de aprendizaje de máquina ya han sido aplicados a sistemas de fútbol robótico, los cuales son también tratados como sistemas multiagente. Técnicas tales como aprendizaje por refuerzo [27], co-evolución para comportamientos cooperativos usando algoritmos genéticos [29] son algunas de ellas. Peter Stone y Manuela Veloso en la Universidad de Carnegie Mellon usan múltiples tipos de aprendizaje aplicados directamente al fútbol robótico, como aprendizaje activo (entrenamiento por prueba) [25], aprendizaje basado en memoria [23], aprendizaje colaborativo y del adversario [24], implementación de aprendizaje por capas [22]. Sin embargo, a pesar de que estas técnicas resuelven el problema de la construcción de estrategias de juego, ellas conllevan tiempo en su implementación. Es por esto, que conviene definir adecuadamente la técnica que se pretende usar en nuestro sistema de fútbol robótico, que entregue resultados óptimos en el menor tiempo posible.

En este capítulo se propone utilizar el *aprendizaje por refuerzo* [46][28], debido a que en este tipo de aprendizaje no se requiere proveer al agente de ejemplos de pares entrada-salida, tal como en el caso del *aprendizaje supervisado* descrito en la siguiente sección. De esta forma la *función de*

agente puede ser aprendida en ambientes complejos como el fútbol robótico, con menor tiempo de implementación que cuando se usa aprendizaje supervisado.

El aprendizaje por refuerzo puede ser aplicado a varios problemas del mundo real debido a su generalidad. En el aprendizaje por refuerzo el agente aprende a partir de aciertos y errores, de recompensas y castigos, con el objetivo de utilizar los aciertos o recompensas en el aprendizaje de una satisfactoria *función de agente*. En particular veremos también el concepto de *aprendizaje Q* y su aplicación en este proyecto, para finalmente mostrar los resultados obtenidos de esta aplicación.

4.2 CONCEPTOS GENERALES DE APRENDIZAJE POR REFUERZO Y APRENDIZAJE Q.

Se pueden clasificar a las técnicas de aprendizaje de máquina en dos tipos: *aprendizaje supervisado* y *no supervisado*. El *aprendizaje supervisado* es implementado en situaciones donde es posible percibir las entradas y salidas, mientras en el *aprendizaje no supervisado* es implementado en ambientes donde no se tiene certeza de cuales serán las salidas.

Un agente puede, por ejemplo, aprender a jugar ajedrez mediante un aprendizaje supervisado, en donde el agente recibe un conjunto de ejemplos

de situaciones del juego indicándole cuál es la mejor jugada en esas situaciones. Si el agente no cuenta con un buen maestro que le proporcione esos ejemplos, ¿qué puede hacer el agente?, bien puede probar jugadas al azar, y finalmente logrará construir un *modelo* predictivo de su ambiente (por ejemplo, como queda el tablero después de hacer una jugada o, incluso, cómo responderá el oponente en una situación determinada). Sin al menos algo de retroalimentación acerca de lo bueno y lo malo, el agente no tendrá bases para decidir lo que debe hacer.

Por fortuna, el agente que juega ajedrez si recibe cierta retroalimentación, que se lo conoce como *recompensa* o *refuerzo*. En el caso de juegos como el ajedrez, el refuerzo se recibe sólo hasta que termina el juego. A la situación anterior se la conoce como *estado terminal* en la secuencia de historia de los estados. En otros entornos o aplicaciones, las recompensas son recibidas con mayor frecuencia: en el ping-pong, por ejemplo, puede considerarse que cada punto anotado es una recompensa. A veces las recompensas provienen de las expresiones de un maestro, tal como “buena jugada” o “no, no”, aunque sin aclarar en qué consiste lo que se considera como buena jugada [21].

El objetivo del *aprendizaje por refuerzo*, considerado como un tipo de *aprendizaje no supervisado*, usa las recompensas en el aprendizaje de una satisfactoria *función de agente*. Lo anterior es difícil, ya que al agente nunca se le indica en qué consisten las acciones correctas, ni tampoco qué acciones

dan lugar a determinadas recompensas. Puede suceder que el desempeño de un agente en un juego sea impecable, excepto por un error cometido, y que al final del juego el único refuerzo recibido sea “perdió el juego”. El agente deberá averiguar de alguna manera cuál fue la mala jugada.

En el marco de trabajo del fútbol robótico, en el que consideramos a los agentes (robots jugadores) como funciones que relacionan percepciones con acciones, la recompensa puede expresarse a través de una percepción, y el agente deberá contar con elementos necesarios para darse cuenta cual percepción es una recompensa y cual es simplemente otra entrada sensorial. Al parecer, los animales cuentan con esos elementos, de modo que el dolor y el hambre funcionan en ellos como recompensas negativas, en tanto que el alimento y el placer como recompensas positivas.

Por tanto, un agente puede aprender cuando debe realizar una acción a través del aprendizaje por refuerzo [27] [28]. Se define *aprendizaje por refuerzo* como el problema que enfrenta un agente para aprender una conducta a través de iteraciones de pruebas y errores en un ambiente dinámico [27] [11].

En muchos dominios complejos, como el fútbol robótico, el aprendizaje por refuerzo puede ser una alternativa válida para capacitar a un agente, de modo que su desempeño alcance un alto nivel. En general en los juegos, es muy difícil que un humano proporcione evaluaciones precisas y congruentes de

muchas posiciones, condición que es necesaria para capacitar y evaluar una función directamente a partir de ejemplos. En vez de ello, se informa al programa cuándo gana o cuándo pierde, y utiliza esta información para aprender una función de evaluación que le aporte estimaciones razonablemente exactas de la probabilidad de ganar cuando está en una determinada posición.

En cierta forma, el aprendizaje por refuerzo es otra manera de plantear el problema de la *Inteligencia Artificial (IA)*. Un agente en un ambiente obtiene percepciones, las correlaciona con *utilidades positivas o negativas* y luego decide qué acción emprender [21].

Para evitar repetir el problema global de IA mencionado en el capítulo II, y citar los principios del aprendizaje por refuerzo, resumimos a continuación las consideraciones que hay que tomar en cuenta cuando se aplica aprendizaje por refuerzo:

El ambiente puede ser *accesible* o *inaccesible*. En un *ambiente accesible* los estados se identifican con determinadas percepciones; mientras que en el ambiente *inaccesible* el agente debe mantener cierto estado interno para tratar de llevar un registro de lo que pasa en el ambiente.

El agente puede empezar con cierto *conocimiento del ambiente* y de los efectos de sus acciones; o bien, deberá aprender este *modelo* así como la información de *utilidad*. Las recompensas pueden recibirse ya sea en estados terminales o en cualquier estado.

Las recompensas pueden ser parte de la *utilidad real* (ejemplos: puntos para un agente que juegue ping-pong, o dólares para un agente de apuestas) que el agente se esfuerza por maximizar, o también pueden ser *sugerencias de la utilidad real* (ejemplos: “buena jugada ” o “perro malo”).

El agente puede ser un *aprendiz pasivo* o un *aprendiz activo*. El *aprendiz pasivo* se limita a observar cómo evoluciona el mundo y se esfuerza por aprender la utilidad que implica un estado determinado; el *aprendiz activo* también debe actuar con base en la información aprendida y puede recurrir a su generador de problemas para que le sugiera la exploración de áreas desconocidas del ambiente.

Existen varios diseños básicos de agentes, como en nuestro caso el agente recibirá recompensas, los cuales establecen un vínculo con las utilidades, los diseños básicos que se tomarán en cuenta serán los siguientes:

El agente puede aprender una *función de utilidad* basada en estados (o historias de estados) para escoger aquellas acciones que permitan obtener la utilidad esperada de sus resultados.

El agente puede aprender una *función valor de la acción* (esta función retorna un valor por cada acción ejecutada) que expresa la utilidad esperada al emprender una acción en un estado determinado. A lo anterior se le conoce como *aprendizaje Q*.

El agente que desea aprender funciones de utilidad debe contar con el modelo del ambiente, lo que le permitirá tomar decisiones, ya que debe saber los estados a los que le conducirán sus acciones [21].

Por ejemplo, para poder utilizar una función de evaluación para el ajedrez, el agente deberá estar en condiciones de saber cuáles son las jugadas permitidas y de qué manera éstas se reflejan en las respectivas posiciones del tablero. Sólo de esta manera podrá aplicar la *función de utilidad* en los estados donde se muestren los resultados de las acciones. Por otra parte, el agente puede aprender una *función del valor de una acción*, con la que no tiene necesidad de contar con un modelo, en tanto sepa cuales son las jugadas permitidas, puede comparar sus valores directamente sin necesidad de considerar sus respectivos resultados. Por tanto, puede ser más fácil diseñar aprendices del valor de las acciones que aprendices de utilidad.

Desde el punto de vista de *sistemas multiagente*, el fútbol robótico es un buen ejemplo de un problema en el mundo real, el cual puede ser moderadamente modelado. El juego de fútbol es diferente de otros sistemas multiagente en el que los robots de un equipo deben presentar cooperación mientras encaran una competición con el equipo oponente. La cooperación y las estrategias usadas juegan un papel mayor en un sistema de fútbol robótico [17].

La acción del robot es usualmente seleccionada considerando algunas condiciones en el ambiente del fútbol robótico [12]. Sin embargo, no es posible describir todas las situaciones de un juego de fútbol robótico con algunos enunciados de condición, puesto que el entorno de trabajo del robot es altamente dinámico y complejo, el aprendizaje por refuerzo sería una técnica apropiada para ser empleada en la selección de una acción apropiada.

El *aprendizaje Q* consiste en que el agente aprenda una *función acción-valor* que asigne una utilidad esperada al emprender una determinada acción en un respectivo estado. Como se mencionó anteriormente, estos valores se conocen también como *valores Q*. La notación $Q(s,a)$ representa el valor que implica ejecutar la acción a en el estado s . Los valores Q guardan una relación directa con los valores de utilidad U de la siguiente ecuación:

$$U(s) = \max_a Q(s,a) \quad (4.1)$$

Los valores Q desempeñan un papel primordial en el aprendizaje por refuerzo. Primero, como en el caso de las reglas condición-acción, bastan para tomar una decisión sin necesidad de un modelo; segundo, a diferencia de las reglas de condición-acción, pueden aprenderse directamente a partir de la retroalimentación de recompensas.

Una ecuación para actualizar los valores Q que se van obteniendo de las transiciones de un estado a otro al realizar una acción es la siguiente:

$$Q(s,a) = R(s) + \sum_w M_{sw}^a \max_{a'} Q(w, a') \quad (4.2)$$

Donde $Q(s,a)$ es el valor Q para el estado s y la acción a , $R(s)$ es la recompensa recibida por el agente en el estado s , M_{sw}^a representa una tabla mediante la que se expresa las probabilidades de transición del estado s al estado w ejecutando una acción a , y la última parte corresponde a la ecuación de utilidad (4.1) para el estado w y la acción a' .

Podemos emplear directamente esta ecuación como una ecuación de actualización para un proceso de iteración que sirve para calcular valores exactos de Q partiendo de un modelo estimado. Para ello, sin embargo, es necesario aprender un modelo M_{sw}^a .

Para casos en que el mundo es *determinista*, aquel en el que se conoce la transición de un estado a otro, la regla de actualización de la tabla Q de un estado s a un estado s' es:

$$Q(s,a) \leftarrow r + \mathbf{g} \max_{a'} Q[s', a'] \quad (4.3)$$

Para casos en que el mundo es *no determinista* la regla de actualización de la tabla Q es:

$$Q'_n[s,a] \leftarrow (1 - \mathbf{a}_n)Q'_{n-1}[s,a] + \mathbf{a}_n(r + \mathbf{g} \max_{a'} Q'_{n-1}[s', a']) \quad (4.4)$$

$$\text{donde } \mathbf{a}_n = \frac{1}{(1 + \text{visitas}_n(s,a))} \quad (4.5)$$

Y Q' representa la aproximación a Q , n es el número de la iteración, r es la recompensa, μ corresponde a un parámetro de *velocidad de aprendizaje*, además \mathbf{g} es un *factor de discontinuidad* definido como un número en el rango de [0..1], usado para ponderar términos de refuerzo cercanos más pesados que futuros refuerzos distantes. Esta actualización de los valores Q se calcula al término de cada transición de un estado s al estado s' .

En el caso de un partido de fútbol robótico, el agente solamente conoce los posibles estados y acciones, no las transiciones probables o la estructura de

recompensa [15], por lo tanto consideraremos al entorno de trabajo del robot como *no determinista*, con lo que resulta conveniente aplicar la ecuación (4.4 y. 4.5). De esta forma, el diseño completo del aprendizaje Q para un agente es el que se muestra en la figura 4.1.

función AGENTE-APRENDIZAJE-Q(e) responde con una acción
(e es el estado del ambiente de trabajo)

VARIABLES ESTÁTICAS:

Q , tabla de valores de las acciones
 N , tabla de frecuencias de las acciones por los estados
 a , última acción realizada
 s , último estado que se visitó
 r , recompensa recibida en el estado s
 s' , estado actual
 n , número de iteraciones

$s' \leftarrow ESTADO[e]$

si s no es nula **entonces**

$N[s, a] \leftarrow N[s, a] + 1$

$Q'_n[s, a] \leftarrow (1 - \alpha_n)Q'_{n-1}[s, a] + \alpha_n(r + \mathbf{g} \max_{a'} Q'_{n-1}[s', a'])$

si terminal [e] es terminal **entonces**

$s \leftarrow \text{nula}$

sino (alternativa)

$s \leftarrow s'$

$r \leftarrow RECOMPENSA[e]$

$a \leftarrow \arg \max_{a'} f(Q[s', a'], N[s', a'])$

responde con a

figura 4.1.- Función de aprendizaje Q para un agente: un aprendiz activo que actualiza el valor de $Q(s,a)$ por cada acción en un mundo no determinista.

4.3 APLICACIÓN DEL APRENDIZAJE Q MODULAR PARA DESARROLLAR LA COORDINACIÓN Y COOPERACIÓN ENTRE LOS JUGADORES.

Los algoritmos de aprendizaje Q almacenan el valor de refuerzo esperado, el cual está asociado con cada par estado-acción. Este valor está ubicado usualmente en una tabla de búsqueda.

Sin embargo, al aplicar directamente aprendizaje Q al sistema multi-agente, hay algunos problemas difíciles de resolver debido a que el número de estados para cada agente crece exponencialmente, y el exponente es proporcional al número de agentes. Por ejemplo, dos agentes comprometidos en una tarea conjunta, trabajando juntos para encontrar un método óptimo para patear el balón. Si se asume que para un solo agente son necesarios 10^3 estados para aprender esta habilidad, para el caso de una tarea en conjunto entre dos agentes, el número total de estados del agente podría crecer a 10^6 y como es necesario realizar acciones en cada estado, se necesita más espacio de memoria para el aprendizaje en sistemas multiagente. Para cubrir este problema, se propone aplicar aprendizaje Q modular [17].

4.3.1 Arquitectura del aprendizaje Q modular.

La figura 4.2 muestra un esquema de la arquitectura de aprendizaje Q modular.

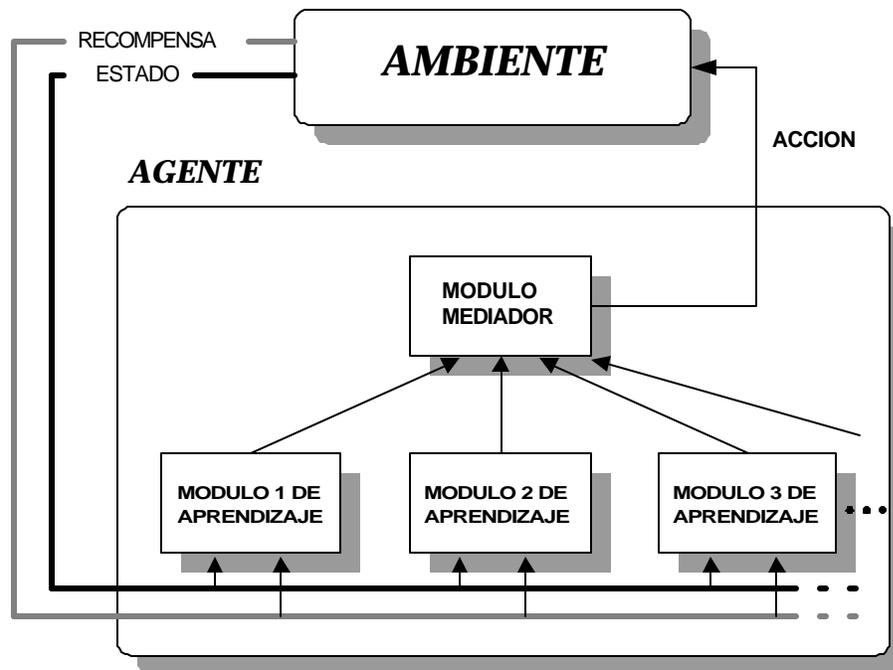


figura 4.2.- Arquitectura de Aprendizaje – Q modular [17].

En el aprendizaje Q modular cada agente incluye un módulo de aprendizaje Q , donde se concentra el aprendizaje de un solo agente y no el de otros agentes. Para completar la meta global de aprendizaje multiagente, un *módulo mediador*, realiza la decisión final seleccionando la acción más conveniente, basándose en los valores Q recibidos desde cada uno de los módulos de aprendizaje. En [17], cuando el módulo mediador hace esta elección considerando el más alto valor Q recibido desde cada uno de los módulos de aprendizaje, es llamada *la estrategia de la cantidad más grande*. Sin embargo, en experimentos en un ambiente real, converger al valor Q más óptimo durante una iteración finita no es usualmente posible. Así que, es deseable

seleccionar la mejor acción considerando una apropiada función la cual es calculada usando el valor Q y la información de los estados.

La Ec. 4.7 presenta la función usada para hacer la decisión final en el módulo mediador:

$$\arg \max_a f(Q_i(s_i, a), \mathbf{q}_i, d_i) \quad (4.7)$$

donde a y s son la acción y el estado del agente respectivamente, mientras que i es el número de módulos de aprendizaje. El valor Q_i es obtenido desde el módulo de aprendizaje y \hat{e}_i es calculado como $(90^\circ - |\hat{e}_i|)$, \hat{e} es el ángulo de error entre la pelota y el robot. Considerando d_1 como la distancia entre robot_1 y el balón, y d_2 como la distancia entre robot_2 y el balón, d_i para el robot_1 es calculado como $d_i = d_2 - d_1$. Puede notarse que \hat{e}_i y d_i son considerados en el módulo mediador para seleccionar la acción final.

4.3.2 Descripción del algoritmo implementado.

El aprendizaje Q modular es empleado en los sistemas de fútbol robótico para proveer cooperación entre los robots de un mismo equipo, así como para llevar a cabo una estrategia de juego en el equipo.

En el fútbol robótico, los robots cumplen roles de atacantes, defensas y arquero. Los atacantes y defensas tienen solo dos acciones, patear la pelota o seguir el campo potencial univectorial descrito en el capítulo 3. Mientras que, el arquero tiene sus propias acciones dentro de su área de gol, tal como se menciona en los capítulos 3 y 5.

Cuando los robots atacante y defensa siguen el campo potencial univectorial, el punto objetivo hacia donde se dirige el robot es la posición del balón. Si el robot se encuentra en una buena posición para patear directamente la pelota, el *módulo mediador*, actúa como un coordinador, seleccionando la acción de patear la pelota (por ejemplo). En otras circunstancias, bajo condiciones normales, el robot sigue el campo univectorial [17].

En la acción de patear la pelota, el objetivo hacia donde el robot quiere enviar el balón normalmente es hacia la portería del equipo oponente. En este caso, la velocidad del robot es más rápida que la velocidad cuando el robot sigue el campo potencial univectorial.

Bajo el esquema de un atacante y un defensa, se presenta un conflicto entre los robots de un mismo equipo, este conflicto ocurre cuando los robots atacante y defensa se dirigen hacia un objetivo común, la pelota, situada dentro de la Región 1, tal como se muestra en la figura 4.3.

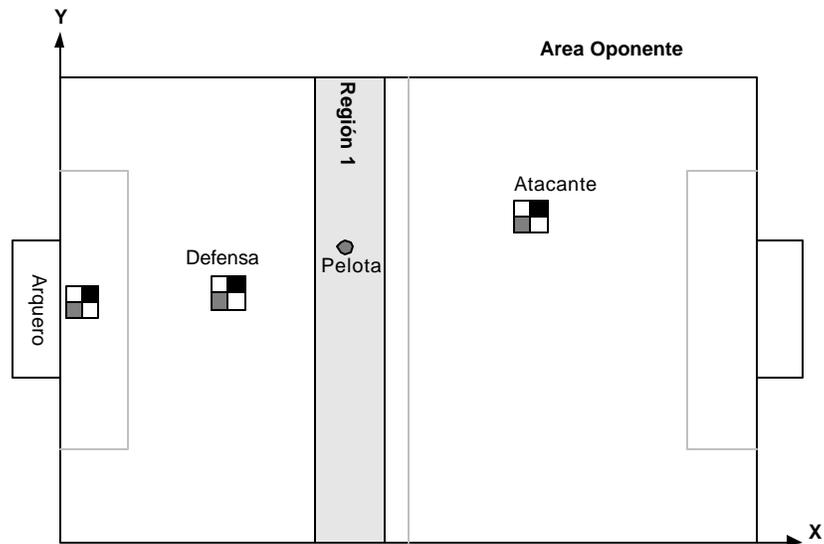


figura 4.3.- La Región 1 muestra una zona de conflicto para los robots atacante y defensa de un mismo equipo.

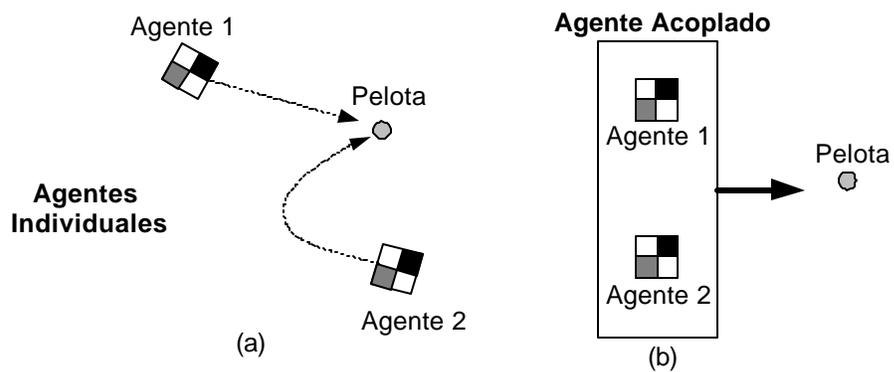


figura 4.4.- (a) Agentes individuales y (b) Agentes acoplados.

Para aplicar el aprendizaje Q modular sobre un sistema de fútbol robótico es necesario introducir dos conceptos importantes: *agentes individuales* y *agentes acoplados*. Los agentes individuales se caracterizan por que cada agente ejecuta su acción de forma independiente del resto de agentes, figura.

4.4 (a). Mientras que, en los agentes acoplados dos agentes son considerados como un solo agente para ejecutar una acción, figura. 4.4 (b).

En la figura 4.3, cuando el balón está dentro de la Región 1, la posibilidad de colisión entre los robots atacante y defensa aumenta, puesto que ambos robots pueden adoptar la posición adecuada para patear el balón. Para resolver este problema, el concepto de agentes acoplados es aplicado. Por ejemplo, si el balón está localizado en la Región 1, los robots atacante y defensa son considerados como un agente acoplado, y por tanto se asumen los dos comportamientos de estos agentes como una sola acción. De esta forma, el módulo mediador asignará una acción a cada agente acoplado basándose en la ecuación (4.7).

Para la implementación del algoritmo de aprendizaje Q modular dos fases son requeridas: *fase de aprendizaje individual* y la *fase de aprendizaje modular*.

En la fase de aprendizaje individual cada agente explora el ambiente de trabajo para obtener información de los estados, el jugador cambia su acción de forma aleatoria cada vez que se encuentra con la pelota, para actualizar de esta manera la tabla Q con valores reales. Estos valores son usados por el módulo mediador en la fase de aprendizaje modular, para seleccionar la acción que considere más adecuada utilizando la (4.7).

Para la fase de aprendizaje individual de cada agente, el valor Q inicial de la función definida en la figura 4.1 es aleatorio, en el rango $[0, 0.02]$. La tasa de aprendizaje α es inversamente proporcional a la frecuencia de pares estado-acción visitados, para proveer la convergencia a los valores Q deseados. El factor de descuento $\tilde{\alpha}$ es seteado a 0.3, un valor relativamente bajo para reducir el posible efecto del ruido (valores aberrantes en la adquisición de información del sistema de visión). El efecto del ruido surge porque $\tilde{\alpha}$ es multiplicado por el máximo valor Q del próximo estado. La razón para elegir un valor $\tilde{\alpha}$ bajo es porque en experimentos reales es posible que el robot pateo inesperadamente la pelota. En tal caso, el valor Q es actualizado con una recompensa, lo cual no es deseado cuando se requiere un aprendizaje preciso en el cual se requiere un número grande de pruebas [11] [17] [27].

Una vez culminado un número variable de iteraciones en la fase de aprendizaje individual, se procede a activar el módulo mediador, para que seleccione la acción más adecuada del agente acoplado.

Los estados y acciones de la fase de aprendizaje individual y de la fase de aprendizaje modular son definidos de la siguiente forma:

Asumiendo el ejemplo de una zona de conflicto definida por la Región 1, donde un estado en la función de aprendizaje Q de un agente individual consiste de cinco componentes que son: localización del robot con 2 niveles

(Area 1 o Area 2 ocupadas por el robot, figura 5(a)), diferencia de distancia d con 4 niveles (figura 5(b)), ángulo de error entre la dirección del robot y el balón con 3 niveles (figura 4.5(c)), bandera binaria (figura 4.5(d)) definida como:

$$\begin{cases} 1 & \text{Si } R_x < B_x, \text{ y} \\ & (B_y - \frac{3}{4} B_w) < R_y < (B_y + \frac{3}{4} B_w), \\ 0 & \text{otros casos} \end{cases}$$

y bandera binaria (figura 4.5(e)) definida como:

$$\begin{cases} 1 & \text{Si el otro robot en el agente acoplado está para patear el balón,} \\ 0 & \text{otros casos} \end{cases}$$

Considerando d_1 , la distancia entre el robot_1 y el balón, y d_2 , la distancia entre robot_2 y el balón, d es calculado como la diferencia entre d_1 y d_2 , tal que, $d = d_1 - d_2$. Así, d en efecto es el negativo de d_i en la ecuación (4.7). Por otro lado, R_x (R_y) es la coordenada X (Y) del robot, B_x (B_y) es la coordenada X (Y) del balón, y B_w es el ancho del balón. Desde el módulo de aprendizaje del robot_2, $d = d_2 - d_1$.

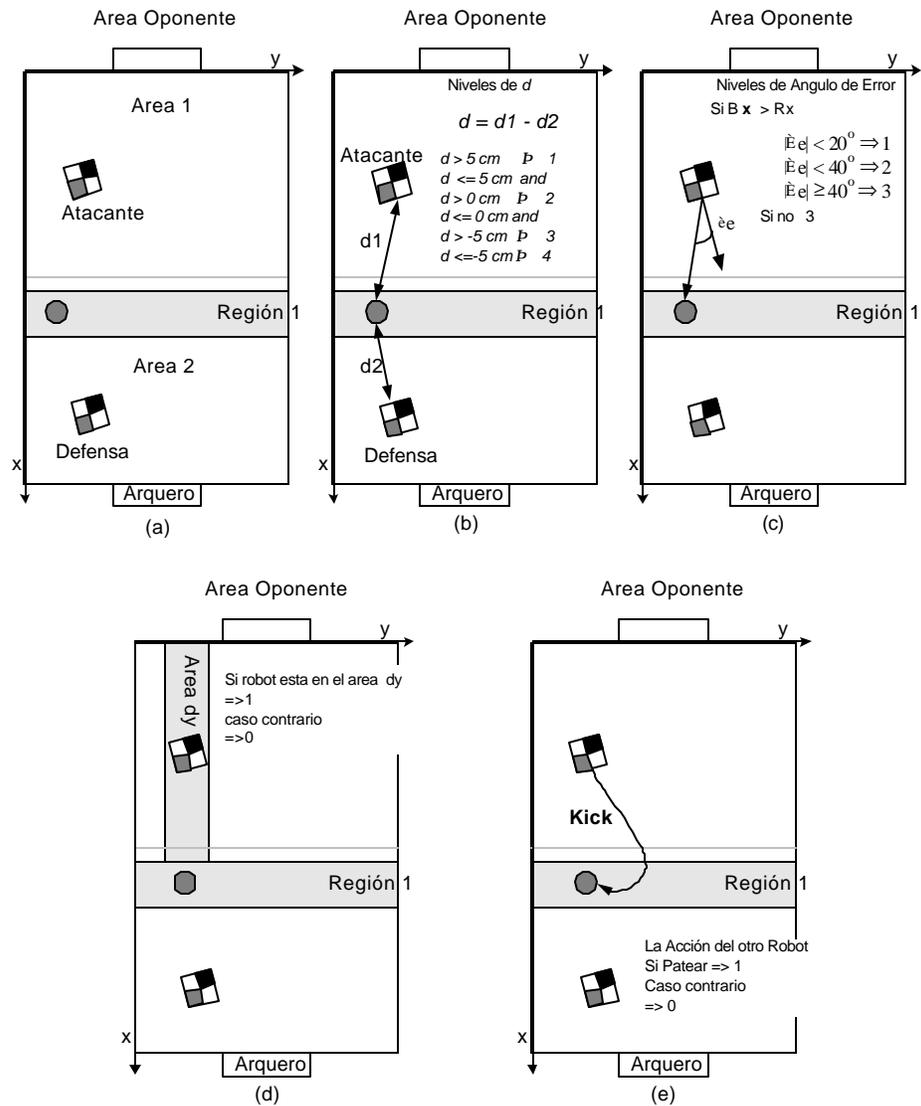


figura 4.5.- Componentes de estados en la Región 1

La figura 4.5 muestra cada componente de estado, para calcular el número de estados posibles se realiza la multiplicación del número de posibles valores de cada componente así de esta forma: $2 \times 4 \times 3 \times 2 \times 2 = 96$ estados en total. Por ejemplo, en la Región 1, el robot_1 y el robot_2 forman el agente

acoplado y cada uno de los robots tiene un módulo de aprendizaje el cual tiene 96 estados.

La tabla 4.1 muestra la lista de acciones que el agente acoplado puede seleccionar en la Región 1. Por ejemplo, si la Acción 0 es seleccionada, el robot_1 será atacante y el robot_2 será defensa.

Región 1		
Acciones	robot_1	robot_2
0	Atacar	Defender
1	Patear	Atacar
2	Ir a Posición Ataque	Defender

tabla 4.1.- Acciones del agente acoplado.

La función recompensa r que es utilizada para la actualización de los valores Q es definido como sigue:

$$r = \left(\frac{a}{t_1 + t_{\text{const1}}} \right) m \quad (4.8)$$

Donde t_1 es el tiempo que toma el robot para patear el balón, a , t_{const1} son constantes, m toma el valor de 1 si es que el movimiento del agente acoplado conllevó a una anotación, o -1 si provocó un autogol. De la ecuación (4.8) se puede ver que la constante t_{const1} es usada para prohibir que el valor de recompensa se incremente a un valor infinito.

Considerando el número de estados igual a 96 y el número de acciones posibles igual a 3, el tamaño de la tabla de valores Q para cada agente será de tamaño $96 \times 3 = 288$ valores.

4.4 ANÁLISIS DE RESULTADOS.

Para la implementación y evaluación del algoritmo de aprendizaje Q modular en la fase de entrenamiento el número de pruebas fue de 300, durante los cuales los valores Q eran guardados en un archivo de texto. Estos valores eran leídos una vez que el número de iteraciones de prueba culminaba, de esta forma comprobábamos la eficiencia del algoritmo en línea usando el algoritmo Q modular para definir la estrategia del equipo.

Los valores de las constantes para el cálculo de la recompensa fueron los siguientes: $a = 12000$, $t_{\text{const } 1} = 18$. Considerando que para propósitos de implementación, cada toma de la tarjeta de digitalización de video es una iteración y ocurren aproximadamente 33 iteraciones cada segundo, una iteración, por tanto, se presentará cada $1/33$ segundo, con lo que podemos calcular el tiempo que aproximadamente le toma a cada robot llegar a la pelota.

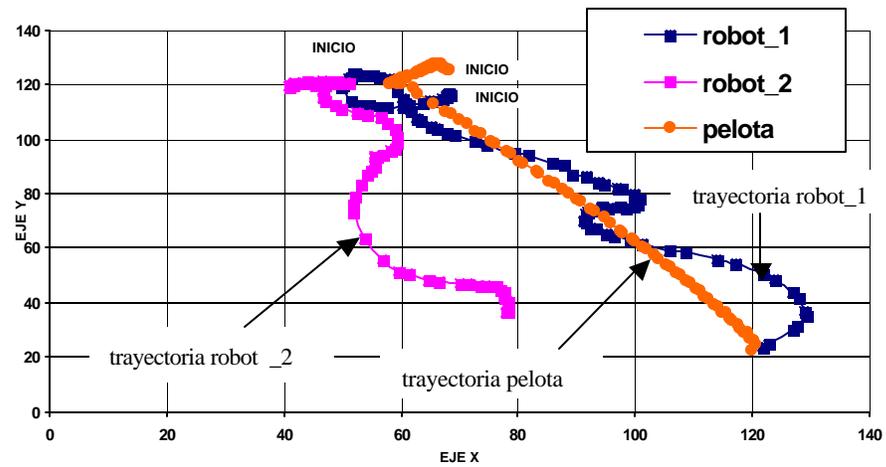


figura 4.6.- Trayectorias realizadas por los robots atacante y defensa para alcanzar a la pelota, después de ejecutar el algoritmo de aprendizaje Q modular implementado

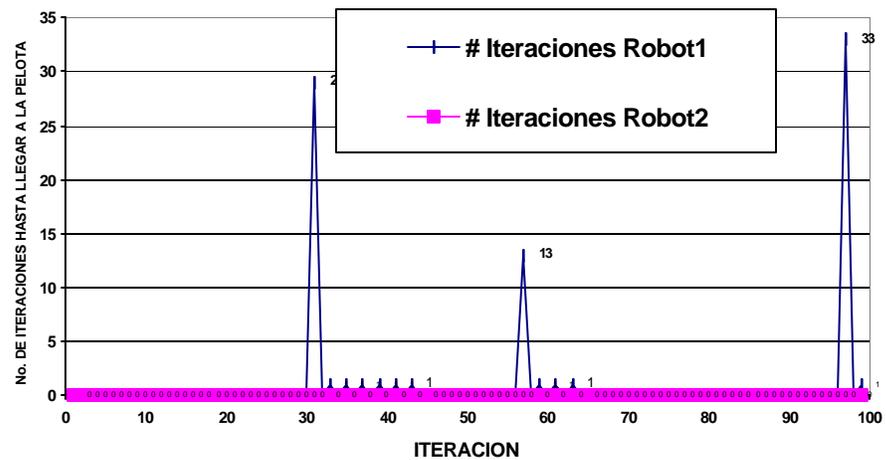


figura 4.7.- Número de iteraciones tomadas para que cada robot alcance la pelota.

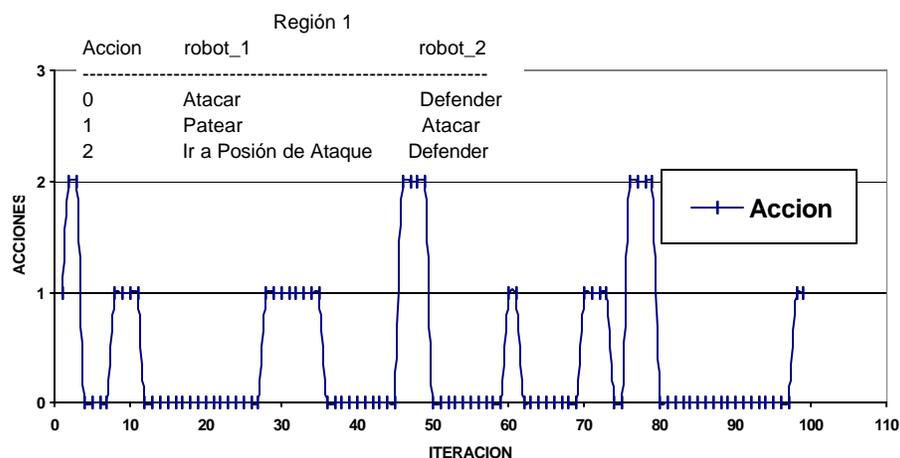


figura 4.8.- Acciones ejecutadas por el agente acoplado durante las trayectorias de la figura 4.6

La figura 4.6 muestra las trayectorias realizadas por dos robots durante la ejecución del algoritmo implementado, mientras que en la figura 4.7 se puede observar el tiempo que tomó a cada robot llegar a la pelota, por ejemplo, para el caso del robot_1, 29 iteraciones (878 mseg.) fueron necesarias para que el robot toque la pelota por primera vez, mientras que el robot_2 no toca la pelota en ningún momento.

La recompensa está basada en los valores de los tiempos para ir a la pelota, el agente acoplado calcula la recompensa cuando uno de los dos robots llega a la pelota, el robot que llega primero recibe el mayor valor de recompensa según la ecuación (4.8), y por consecuencia tendrá el mayor valor Q entre los dos, finalmente, el modulo mediador escogerá la acción del agente que cumpla con la ecuación (4.7). La figura 4.8 muestra las acciones ejecutadas por el agente

acoplado del ejemplo de la figura 4.6. En esta figura, en la iteración 29 el robot_1 asumirá la acción de patear la pelota.

Se puede apreciar que la última acción que el robot_1 va a realizar es patear la pelota, mientras que el robot_2 cambia su rol de defensa a atacante. Los repentinos cambios de rol en el agente acoplado, provocan que los robots modifiquen su trayectoria inesperadamente, esta actuación del robot disminuye conforme se ejecutan un número mayor de fases de entrenamiento individual.

4.5 CONCLUSIONES.

Para que los sistemas multiagente realicen tareas complejas con la finalidad de cumplir un objetivo común se necesita que exista coordinación y cooperación entre cada uno de los agentes que lo conforman. En un ambiente de trabajo tan complejo y dinámico se deben añadir características al sistema multiagente que les permita aprender interactuando con su entorno. En este capítulo se implementó una técnica de aprendizaje por refuerzo, de tal forma de que el agente aprenda a partir de aciertos y errores, recompensas y castigos. Concretamente se usó el concepto de aprendizaje Q modular para seleccionar de entre varios agentes la mejor acción a ejecutar en cualquier estado del juego.

Mediante la aplicación del aprendizaje Q modular, se consiguió mejorar la falta de coordinación entre dos robots, puesto que permite el intercambio de roles entre un robot defensa y un robot atacante. Con esto se reduce la posibilidad de bloqueo mutuo entre los dos robots, y aumenta la posibilidad de que el ataque sea ejecutado por el robot que se encuentra en la posición más idónea para realizar esta función.

Futuras implementaciones considerarán la posibilidad de seleccionar la acción que el agente acoplado deba realizar, tomando en cuenta las posiciones del oponente y de la pelota dentro de la Región 1, con la finalidad de resolver el problema de que el jugador que no se encuentre bloqueado por el oponente asuma la función de ataque.

CAPITULO V

Técnicas de movimiento utilizadas por el equipo FUROEC en las competiciones de la FIRA 2002 World Cup

Este trabajo describe las técnicas de movimiento usadas por el equipo de robots que representó al Ecuador en las competiciones mundiales 2002 de fútbol robótico realizado en Corea y bajo la organización de la Federación Internacional de Fútbol Robótico Asociado (FIRA). Las técnicas implementadas permiten a los robots: atacantes, defensas y arquero, poseer habilidades para patear, salir de los bordes de la cancha y rechazar la pelota. Estas técnicas se basan en algoritmos que actúan dependiendo de la posición de la pelota dentro del campo de juego. Todas las técnicas implementadas fueron aplicadas tanto en las competiciones de fútbol robótico simulado (SIMUROSOT) como en las competiciones de fútbol robótico con robots reales (MIROSOT), ambas en la categoría de 5 vs. 5 robots.

5.1. INTRODUCCIÓN.

Dentro de los campos de investigación de la Inteligencia y Visión Artificial y Robótica, nuestro grupo escogió trabajar en Fútbol Robótico, puesto que es

una interesante aplicación en tiempo real para sistemas multiagente cooperativos.

Debido a la naturaleza del Fútbol Robótico, se necesita crear un conjunto de comportamientos y procesos de tomas de decisión en tiempo real rápidamente. Pero, como los comportamientos y procesos de tomas de decisión pueden variar desde los más simples comportamientos reactivos, como ir detrás de la pelota; a razonamientos complejos, comportamientos deliberativos, que toman en cuenta las acciones y estrategias de los compañeros y los oponentes, así como predecir la trayectoria de la pelota; nuestra investigación se ha enfocado en construir técnicas básicas de movimientos sobre las cuales se diseñarán otras técnicas más complejas.

En [22] Peter Stone emplea técnicas de aprendizaje de máquina para que los robots aprendan comportamientos y desarrollen estrategias en línea para su equipo CMUnited (Carnegie Mellon University), mientras Jong Hwan Kim en [14] se enfoca más en esquemas de control de movimientos y diseño de supervisores para seleccionar cada comportamiento [15]. Nuestra implementación incluye algunos criterios basados principalmente en las ideas presentadas por estos dos investigadores, y además aprovecha algunas técnicas de movimiento implementadas para las competiciones organizadas por la FIRA.

En este sentido, el primer paso consistió en analizar e investigar la cinemática de los robots usados en estas competiciones, luego se estudió el funcionamiento general del sistema (visión artificial, transmisión de señales de control y sistemas multiagente), así como también se analizó las reglas de juego planteadas por la FIRA para la categoría MIROSOT y SIMUROSOT de 5 vs. 5 robots. La duración de este trabajo fue de aproximadamente 5 meses.

5.2. DESCRIPCIÓN DE LOS ALGORITMOS UTILIZADOS EN LAS COMPETENCIAS.

Para el desarrollo de las habilidades de cada jugador se utilizaron diagramas similares a los diagramas de estado o a los diagramas Petri-Net mostrados en [12]. Así mismo, se definieron zonas de cobertura, posiciones pasivas (en donde el jugador simplemente espera para realizar su acción), y roles para cada jugador, que en conjunto forman la estrategia general del equipo.

Un breve resumen de este capítulo se detalla a continuación: la sección 5.2.1 presenta el primer algoritmo y sus diagramas de estado para patear la pelota. La sección 5.2.2 describe un algoritmo para evitar obstáculos móviles, definiendo puntos intermedios al objetivo para evitar las colisiones. La sección 5.2.3 muestra un algoritmo sencillo para salir de los bordes del campo de juego. En la sección 5.2.4 se presenta un algoritmo para definir los movimientos de un arquero dependiendo de la posición de la pelota. En la sección 5.2.5 se implementa un algoritmo que utilizan los defensas para

rechazar la pelota cuando esta se encuentra cerca del jugador. En la sección 5.2.6 se utilizan las habilidades desarrolladas en las secciones anteriores para construir la estrategia del atacante y defensa. Finalmente, en la sección 5.2.7 se desarrolla la estrategia general del equipo, definiendo zonas de juego, posiciones pasivas y roles para cada jugador.

5.2.1 Algoritmo básico para patear la pelota (pelota quieta).

Para controlar el movimiento de cada robot primeramente tenemos que tomar en cuenta como se calculan las velocidades de sus motores, analizando para esto la cinemática del robot. Al respecto en el capítulo III se presentó un análisis detallado sobre la cinemática del robot. A continuación la figura 5.1. presenta una breve descripción de esta teoría.

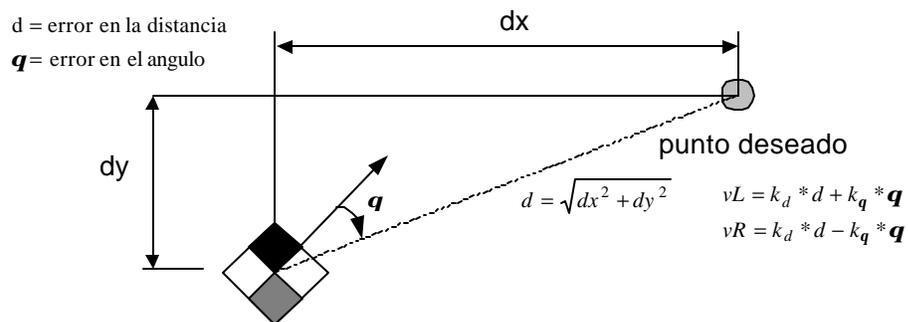


Figura 5.1.- Análisis vectorial del movimiento de un robot.

Donde vL es la velocidad lineal de la rueda izquierda y vR es la velocidad lineal de la rueda derecha del robot, d es la distancia hacia el punto destino. Como se puede apreciar en las ecuaciones de la figura 5.1 los movimientos del robot dependerán de la distancia y la orientación del robot hacia el objetivo, por tanto la idea es calcular adecuadamente los valores de las constantes K_d , y K_θ para que su orientación sea más precisa hacia el punto deseado.

La figura 5.2 muestra el algoritmo implementado para que un robot patee la pelota hacia el arco (un algoritmo mejorado de esta implementación consistió en el uso de campos potenciales, el cual es presentado en el capítulo III):

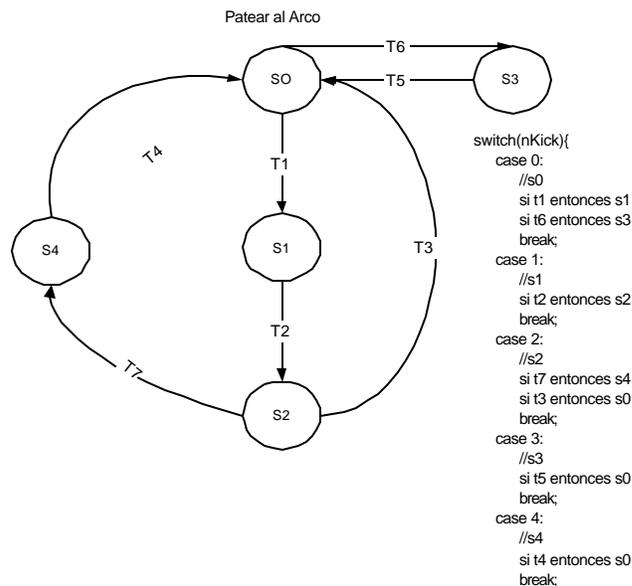


Figura 5.2.- Algoritmo para que un robot patee la pelota hacia el arco[33].

De la gráfica anterior, el *conjunto de estados internos* de cada robot se define como $S=\{S0,S1,S2,S3,S4\}$, y las *transiciones entre estados*, que hacen referencia a las condiciones que permiten ir de un estado a otro, como $T=\{T1,T2,T3,T4,T5,T6\}$; el significado de cada estado se describe a continuación:

S0: En este estado el robot calculará su posición óptima para patear la pelota.

S1: Rotar al robot para orientarlo hacia la pelota.

S2: Ir con velocidad máxima hacia la pelota, para este caso se define vL y vR con un valor de 127.

S3: Ir hacia la pelota cuando está cerca de la pared, para este caso la posición deseada es la posición de la pelota.

S4: Cuando el robot pierde la pelota, se ajusta el ángulo del robot q_c perpendicularmente a la pelota para volver al estado S0.

Mientras que el significado de las transiciones es:

T1: El robot está en la posición adecuada para patear la pelota.

T2 : El ángulo entre el robot y la pelota es casi cero.

T3 : El robot sobrepasó el límite del arco, o el ángulo entre el robot y la pelota sobrepasó un valor de 60° .

T4 : El ángulo entre el robot y la pelota está entre 60° y 120° .

T5: La pelota está lejos del borde del campo de juego.

T6: La pelota está cerca del borde del campo de juego.

5.2.2 Algoritmo para evitar obstáculos móviles mediante la generación de objetivos intermedios.

Un camino debe ser planificado de forma que los robots puedan evitar obstáculos. Definir toda la trayectoria desde el origen hasta el punto destino puede conllevar a un excesivo costo computacional y además esta solución no podría ser muy eficiente, tomando en cuenta que el ambiente de trabajo donde el robot se mueve cambia constantemente. Esto último es debido a que los obstáculos están definidos por los robots compañeros y robots oponentes. Es preferible entonces contar con un algoritmo de planificación en línea.

En esta sección se describe el algoritmo para evitar obstáculos que se usó en las competiciones de la FIRA categoría SIMUROSOT, el cual trabaja calculando objetivos intermedios cuando existen obstáculos frente al robot, estos objetivos intermedios representan puntos que están ubicados a una distancia predefinida del robot obstáculo. Una implementación más eficiente de este algoritmo es presentada en la sección 3.5.

Para el caso en que el obstáculo es fijo, tal como por ejemplo el borde del campo de juego, un algoritmo sencillo ha sido implementado en la sección 5.2.3.

En base a la figura 5.3, que muestra el movimiento de un robot hacia un objetivo, el conjunto de estados se describe como sigue:

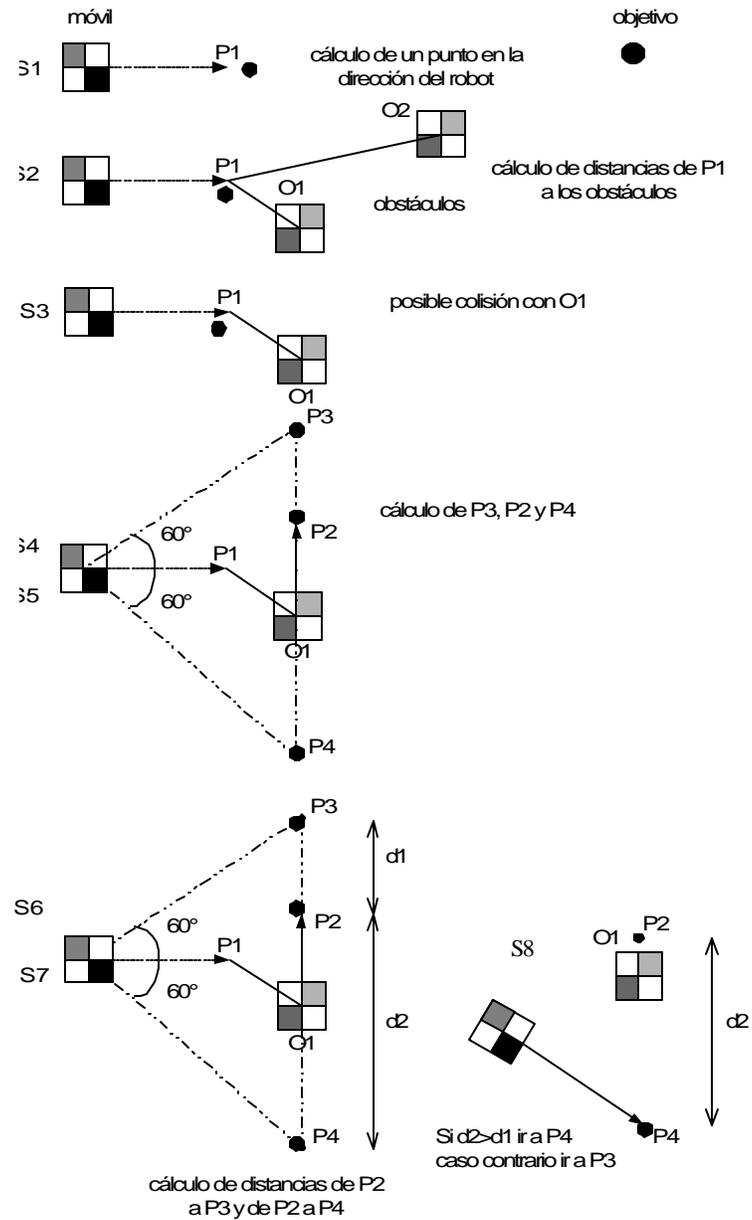


Figura 5.3.- Algoritmo implementado por FUROEC para evitar obstáculos

S0: Se predice un punto P1 a una distancia definida en la dirección del robot (móvil).

S1: Se calcula la distancia de cada obstáculo (otros robots) a este punto.

S2: Si alguna de estas distancias es menor que un umbral, entonces habrá colisión.

S3: Se identifica la posición y orientación del obstáculo.

S4: Se calculan dos puntos, el primero P3 sumando 60° al ángulo del robot y el segundo P4 restando 60° al ángulo del robot a una distancia de 10 unidades.

S5: Se predice un punto P2 en la dirección del obstáculo.

S 6: Se obtienen las distancias d_1 y d_2 ; entre P2 y P3 y entre P2 y P4 respectivamente.

S7: La distancia más grande, d_1 ó d_2 , determinará cual punto, P3 ó P4, será la posición hacia la que debe ir el robot para evitar la colisión.

S8: Continuar al estado S0 y así sucesivamente hasta que el robot llegue al objetivo final.

Cuando existen varios obstáculos en movimiento, puede ocurrir que perdemos nuestra prioridad de ir hacia el objetivo, con lo cual en estos casos nuestro algoritmo presenta deficiencias, es por ello que se debe diseñar mejor el algoritmo donde la trayectoria del robot se modifique constantemente en presencia de obstáculos, una vez libre de esos obstáculos la trayectoria vuelve a corregirse para que el robot pueda ir hacia el objetivo con la orientación y la velocidad adecuada. Para solucionar estos casos la teoría de campos potenciales [14] puede ser aplicada, la sección 3.3 hace uso de esta técnica y mejora el algoritmo descrito arriba.

5.2.3 Algoritmo para salir de los bordes del campo de juego.

Esta sección proponemos un algoritmo básico para evitar que el robot choque o quede atrapado contra los bordes del campo de juego. En este caso, los bordes son considerados obstáculos fijos que deben ser evitados. Existen dos casos en los cuales el robot puede quedar atrapado o chocar contra los bordes. El primer caso, es cuando el robot está frente a un *borde cancha* y el otro caso es cuando el robot está frente a un *borde arco*. Ambos casos son detallados en la figura 5.4

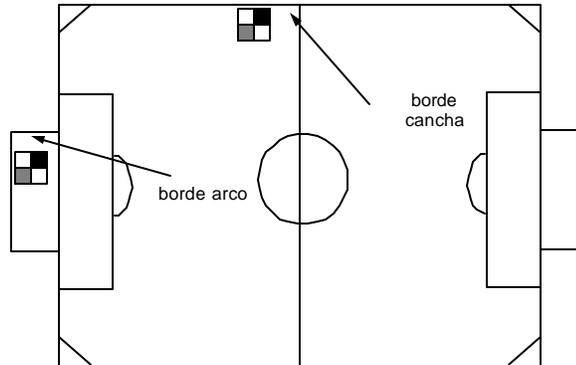


Figura 5.4.- Tipos de bordes

Cuando el primer caso ocurre, *evitar un borde cancha*, el objetivo consiste en hacer girar al robot un valor de ángulo predefinido por el usuario y de esta forma evitar que el robot choque contra el borde. Este giro debe ser realizado a una distancia de $3/2 L$ entre el robot y el borde, donde L es la longitud del robot igual a 7.5 cm. De esta forma, conseguimos que el robot choque contra el borde obstáculo.

Para conseguir que el robot gire, se mantiene uno de sus dos motores con velocidad cero y el otro diferente de cero. La selección del motor que se mantiene con velocidad cero dependerá de la ubicación del punto destino hacia donde se quiere que el robot vaya. Después de una serie de pruebas se llegó a la conclusión que el valor del ángulo que debe girar el robot es de 60° .

La función *Velocity* es la encargada de hacer girar al robot, esta función asigna velocidades a cada uno de los motores que controlan las ruedas del robot, por

ejemplo: Velocity(0,10) significa velocidad 0 a la rueda izquierda y velocidad 10 a la de la rueda derecha, lo que hace que el robot empiece a girar, si las velocidades de los robots tienen el mismo valor, ejemplo Velocity(10,10), el robot se mueve en línea recta.

Para este mismo caso, cuando ocurra que el robot ya ha chocado contra el borde de la cancha se aplica sólo la función Velocity.

Por otro lado, cuando el segundo caso ocurre, *evitar borde arco*, sólo se considera que el robot ya ha chocado contra el borde de una portería y entonces únicamente se aplica la función Velocity. Se debe destacar que la opción de hacer girar al robot a una distancia de $3/2 L$ para este segundo caso no ha sido considerada debido a que la profundidad del arco es aproximadamente igual a L . La figura 5.5 muestra gráficamente los algoritmos para salir de los bordes del campo de juego:

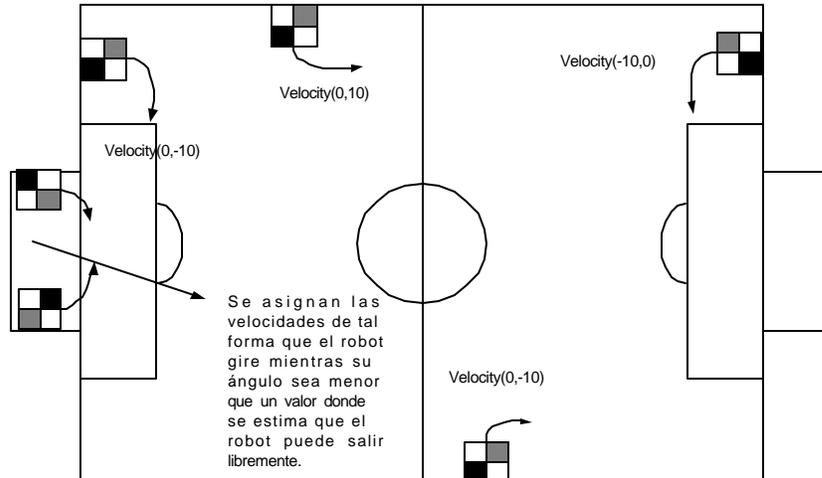


Figura 5.5.- Algoritmo para evitar un borde cancha y borde arco.

5.2.4 Algoritmo para mover el arquero de acuerdo a la posición de la pelota.

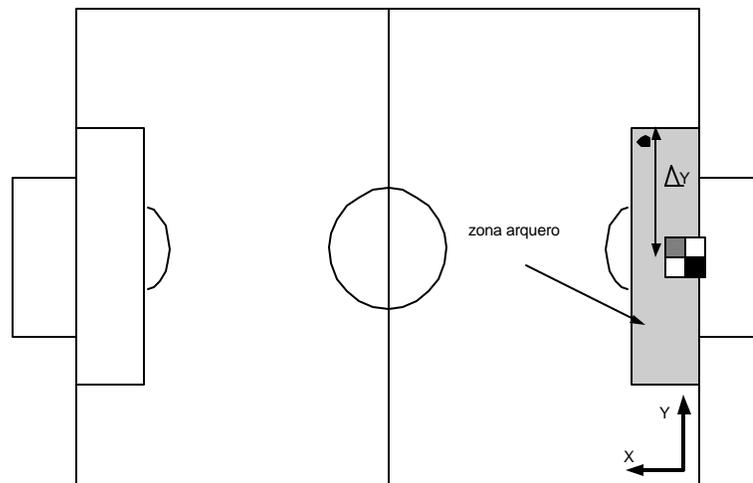


Figura 5.6.- Algoritmo del arquero de acuerdo a la posición de la pelota

Este algoritmo tiene como objetivo principal permitir al arquero bloquear el paso de la pelota a su arco. El algoritmo implementado usa algunos de los conceptos definidos en la figura 5.6.

Para impedir que la pelota ingrese al arco, el robot arquero deberá realizar movimientos constantes en la dirección del eje y . Para esto la orientación de este robot siempre será de $90^\circ \pm 1^\circ$, donde el frente del robot está hacia el *borde superior*. De esta forma, la velocidad lineal del robot arquero con la que realiza sus movimientos varía en función de la distancia entre el robot y la pelota, en dirección del eje y , Dy en la figura 5.6 Esta velocidad es definida positiva si la pelota está por encima del robot y negativa cuando está por debajo de este. Actualmente, los valores de las velocidades tanto positivas como negativas han sido fijadas a vL y vR con un valor de 50. Cuando $\Delta y=0$ el valor de las velocidades es fijada a 0.

Se debe mencionar que el proceso anterior es ejecutado siempre y cuando la pelota esté dentro de la *zona de arquero*. De otro modo, el robot siempre tiende a posicionarse en el centro del arco.

Al algoritmo anterior se le ha añadido también la capacidad para que el robot arquero pueda rechazar la pelota. Este proceso se ejecuta cuando el robot está cerca de la pelota. En este caso, el robot gira rápidamente para rechazar la pelota.

Debido a que este algoritmo no toma en cuenta las posiciones estimadas de la pelota no se puede controlar eficientemente la posición del robot en la zona del arquero. Una modificación al respecto fue crear posiciones apropiadas para tres ubicaciones de la pelota tal como se muestra en la figura 5.7.

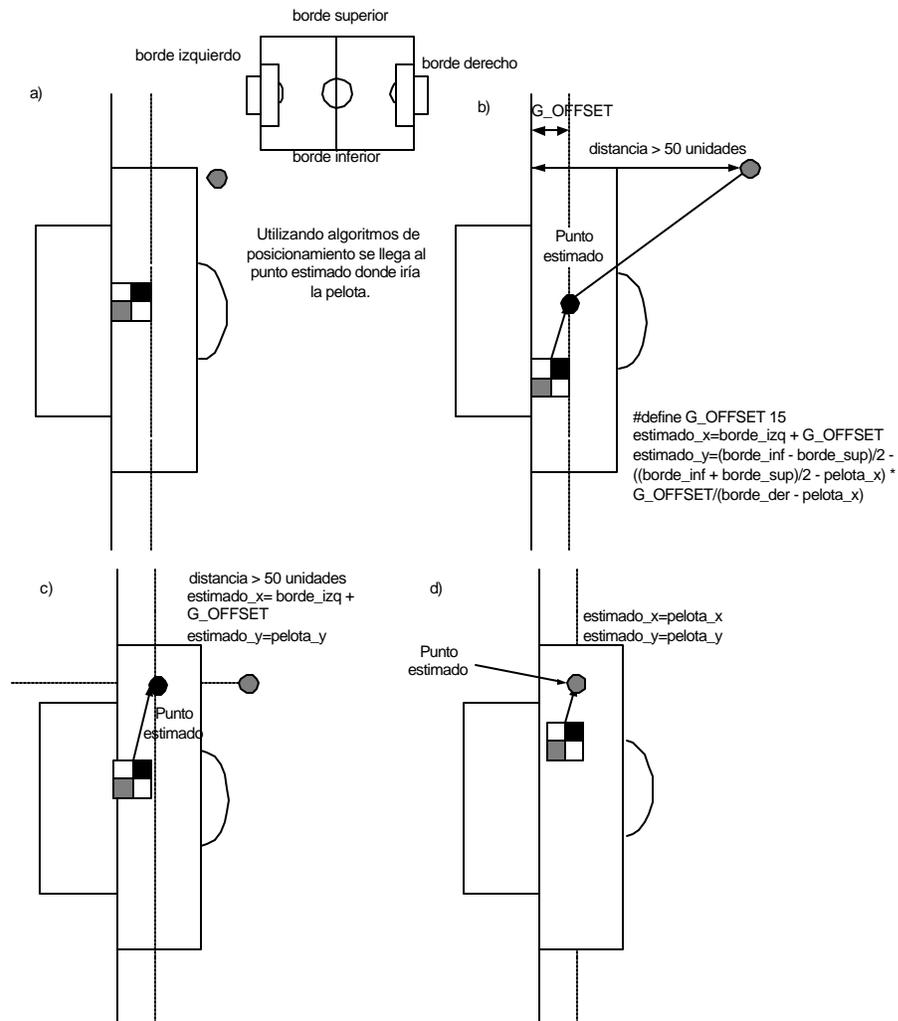


Figura 5.7.- Algoritmo para bloquear la pelota

Estimado_x : posición estimada de la pelota en el eje x.

Estimado_y : posición estimada de la pelota en el eje y.

G_OFFSET: es la distancia entre el borde del arco y la posición del robot arquero.

Borde_inf: borde inferior de la cancha.

Borde_sup: borde superior de la cancha.

Borde_inf: borde izquierdo de la cancha.

Borde_der: borde derecho de la cancha.

En la figura 5.7.a muestra la ubicación del robot arquero en la cancha. Cuando la pelota se encuentra a una distancia mayor a 50 píxeles del borde izquierdo, el punto estimado de la ubicación deseada del robot arquero, se calcula mediante la fórmula de la figura 5.7.b. La figura 5.7.c muestra el punto estimado cuando la pelota esta a una distancia menor a 50 unidades, y cuando la pelota esta dentro de la zona del arquero el punto se calcula como muestra la figura 5.7.d [33].

Para que el robot pueda aprovechar los dos frentes, se implementó una mejora a este algoritmo, un cambio automático en la dirección del robot de los valores del ángulo entre el robot y la pelota, si la pelota se encuentra más cerca de la parte posterior del robot, el frente del robot es su parte trasera, caso contrario queda inalterable, tal como se muestra en la figura 5.8.

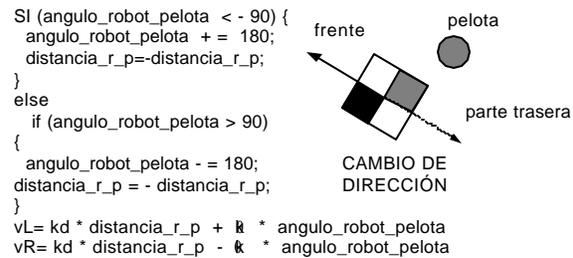


Figura 5.8.- Cambio automático en la dirección del robot

Angulo_robot_pelota : ángulo entre el robot y la pelota.

K_q = constante de rotación

K_d = constante de ganancia

Distancia_r_p: distancia entre el robot y la pelota.

5.2.5 Descripción del movimiento de giro y rechazo cuando la pelota se encuentra bien cerca del jugador.

Tanto la posición del arquero y defensas dependen del estado del juego, se sitúan de forma perpendicular a la trayectoria de la pelota con el objetivo de bloquear su paso hacia la portería, es recomendable en esos casos enviar la pelota hacia los bordes. Esta habilidad se implementa girando al robot para golpear la pelota sin enviarla hacia su propio lado, y es utilizada cuando el jugador esta situado cerca de los bordes y de la pelota.

El algoritmo se describe gráficamente a continuación en la figura 5.9.

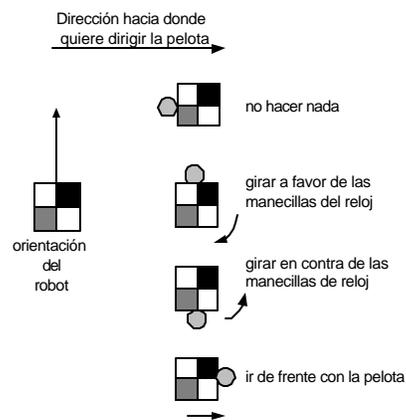


Figura 5.9.- Descripción del movimiento de rechazo

5.2.6 Implementación del comportamiento para el atacante.

Nuestro primer algoritmo del jugador atacante se basa en el movimiento PATEAR_AL_ARCO, y el movimiento de rechazo anteriormente mencionado, consiste en generar objetivos de forma dinámica dependiendo de la posición de la pelota en la cancha con respecto al jugador.

La figura 5.10 muestra el punto TARGET_DINAMICO donde el robot se ubicará para girar y empujar la pelota hacia la portería contraria, este punto varía en el entorno de trabajo, para que el robot alcance la pelota con mayor o menor fuerza, para lo cual se ha definido el radio de acción del robot igual a 25.

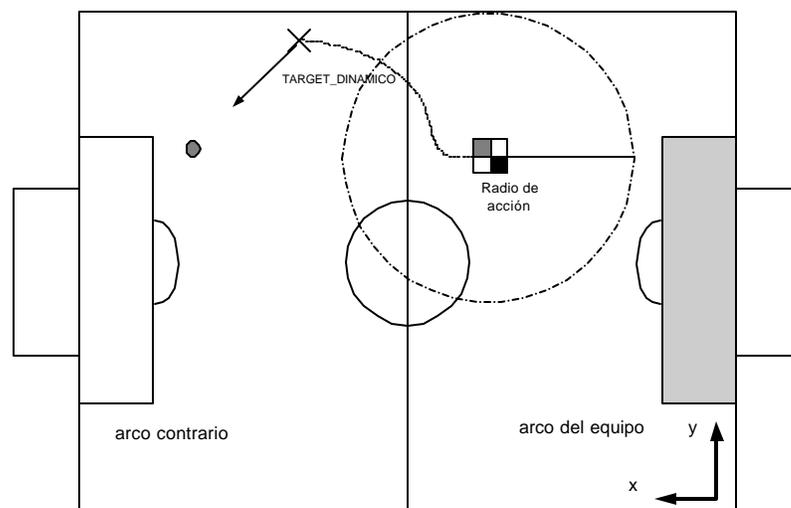


Figura 5.10.- Definición del TARGET_DINAMICO

A continuación se describe el algoritmo para el atacante con sus respectivos estados:

S 0: Calcula una posición futura del robot (calcular un punto definido por la función TARGET_DINAMICO).

Si la distancia entre el robot y la pelota es menor al radio de acción

→ ir al estado S 1

caso contrario → ir al estado S 4.

S 1: Ir al punto TARGET_DINAMICO.

Si robot alcanzó el TARGET_DINAMICO y el ángulo entre el robot y la pelota es menor a 3° (según pruebas experimentales en el simulador).

→ ir al estado S 3

si la Posición-X-Robot es mayor a Posición-X-Pelota y $\text{AnguloRobotPelota} < 30^\circ$

→ ir al punto definido en el estado S 0

si la distancia entre el robot y pelota es casi cero

→ rechazar → ir al estado S 0.

Si la distancia entre el robot y el TARGET_DINAMICO es casi cero

→ ir al estado S 2.

S 2: Reducir el anguloRobotPelota a un valor pequeño (aprox. 3°).

Si el anguloRobotPelota $\leq 3^\circ$ → ir al estado S 3.

S 3: Ir a la pelota

si la posXRobot es mayor a posXPelota y AnguloRobotPelota $< 30^\circ$ y la distancia entre la pelota y el robot es menor a un umbral igual a tamaño del robot/2+diámetro de la pelota+1

→ rechazar → ir al estado S0

caso contrario

si la $posXRobot$ es menor a $posXPelota$ y $AnguloRobotPelota < 30^\circ \rightarrow$ ir hacia la pelota con la máxima velocidad

Si el $AnguloRobotPelota > 40^\circ \rightarrow$ retroceder ($vL=-70$, $vR=-70$) \rightarrow ir al estado S 2

Si la distancia entre el robot y la pelota es mayor a su radio de acción \rightarrow ir al estado S 4

S 4: Reducir el ángulo entre el robot y la pelota.

Si la distancia entre robot y pelota es menor al radio de acción \rightarrow ir al estado S 0

El movimiento descrito lo hemos denominado Atacante B, siendo probado en el simulador de 5 vs. 5 jugadores.

5.2.7 Implementación de la estrategia general del equipo para un equipo de cinco robots usando estructuras de decisión.

El diseño de la estrategia general del equipo se basa en la ubicación de la pelota dentro de la cancha. Para esto, se definen modos de juego (defensa, normal y ataque), zonas de juego (arquero, defensa 1, defensa 2, atacante 3 y atacante 4), posiciones pasivas y roles de cada jugador, toda esta información se encuentra guardada en la memoria.

El módulo de selección de los roles (comportamientos) para cada jugador se implementó mediante algoritmos jerárquicos IF-THEN-ELSE similares a un árbol de decisión, que obtiene información de una lista enlazada de dos nodos que guarda el estado actual y anterior de cada robot. Los últimos nodos del árbol representan comportamientos básicos que fueron descritos anteriormente en este capítulo y son guardados en la lista.

Cuando un robot ejecuta un movimiento para ir desde una posición inicial a una posición final este toma en cuenta los obstáculos presentes en el trayecto.

Un planificador de movimiento en línea es usado para ayudar a definir las trayectorias que debe seguir el robot. De esta forma, dependiendo del tipo movimiento que tenga que realizar el robot se define un valor de velocidad para cada motor (v_L = motor izquierdo, v_R = motor derecho). Se debe mencionar que las trayectorias que sigue el robot son definidas en todo momento y dependen de la situación actual del partido.

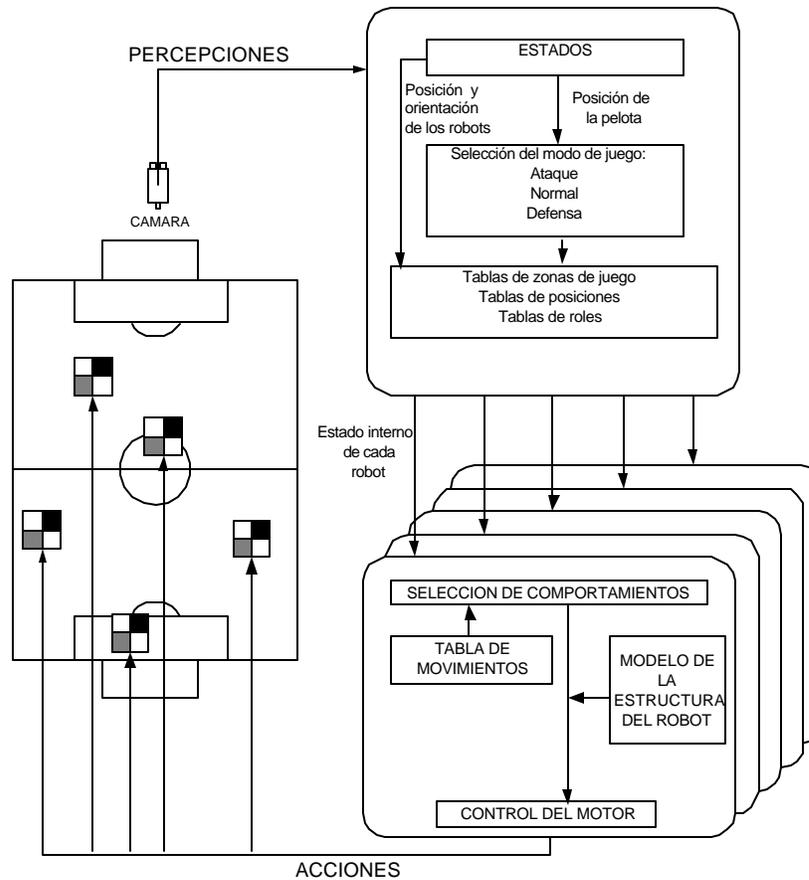


Figura 5.11.- Diagrama de bloques implementado por FUROEC para el componente inteligente.

En la figura 5.11 el diagrama de bloques describe en resumen de la implementación de nuestro sistema de selección de estrategias. Donde los *estados*, representan las diferentes situaciones del partido (posiciones, orientaciones y las dimensiones de la cancha), las cuales son captadas por la cámara. Dependiendo de la ubicación de la pelota en la cancha se selecciona el modo de juego.

El módulo de *selección de comportamientos* selecciona el movimiento adecuado para cada robot, y tomando en cuenta la cinemática del robot se calculan los valores de velocidad de cada motor transmitidos por radiofrecuencia.

Cada robot cumple un rol fijo dentro de su equipo, en el cual se definen cuales son sus movimientos básicos. La Tabla 5.1 muestra el contenido de la tabla de roles:

	HOME 0	HOME 1	HOME 2	HOME 3	HOME 4
ATAQUE	Arquero	Defensa	Defensa	AtacanteB	AtacanteA
NORMAL	Arquero	Defensa	Defensa	AtacanteA	AtacanteA
DEFENSA	Arquero	Defensa	Defensa	AtacanteB	AtacanteB

Tabla 5.1.- Estrategias para cada modo de juego

La figura 5.12 muestra los límites que definen las zonas para cada modo de juego:

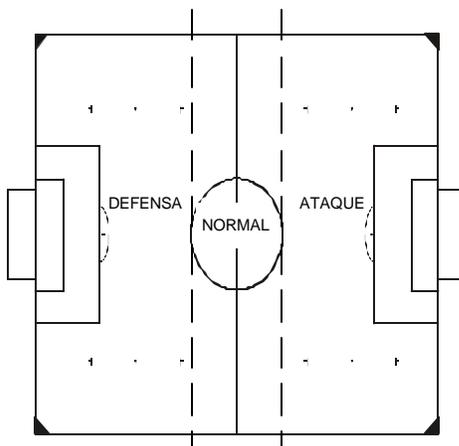


Figura 5.12.- Límites de las zonas para cada modo de juego

Un ejemplo de las posiciones de los robots (un arquero, dos defensas, y dos atacantes) en el campo de juego es mostrado en la figura 5.13. La figura 5.14 muestra las zonas de juego para cada robot.

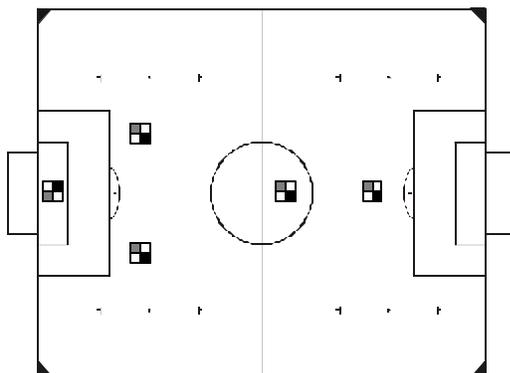


Figura 5.13.- Posiciones pasivas

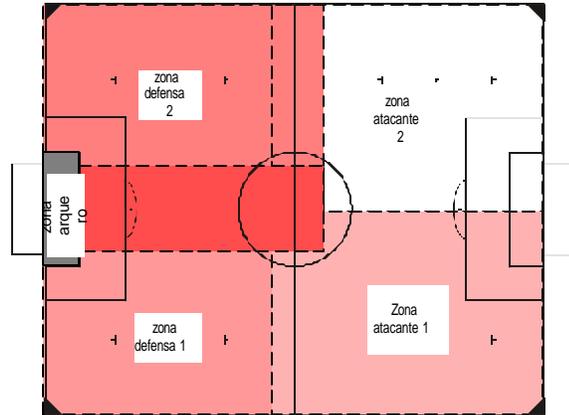


Figura 5.14.- Zonas de juego de cada robot.

Una vez definidos estos parámetros dentro de la memoria de trabajo se procede a definir la estrategia general del equipo. Para la implementación del módulo de selección de comportamientos de cada rol se utilizó una estructura de decisión, algoritmo jerárquico IF-THEN-ELSE. Para el caso del robot *arquero* la estructura de decisión se muestra en la figura 5.15.

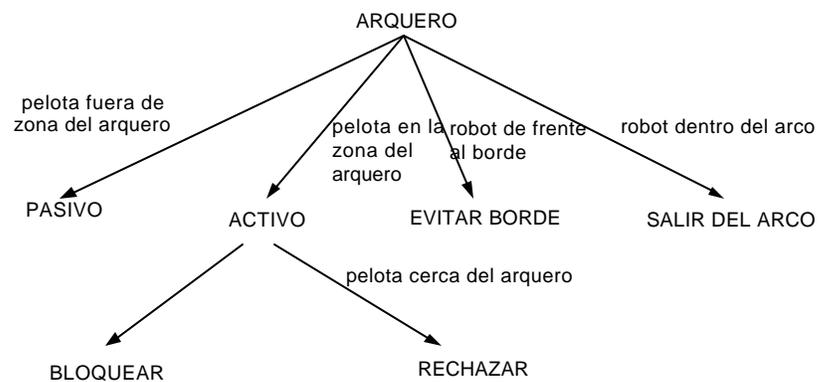


Figura 5.15.- Árbol de decisión para el arquero

Donde los comportamientos básicos se encuentran en los últimos nodos del árbol el nodo principal representa el rol que cumple este robot en el equipo.

Para los *defensas* nuestra estructura de decisión IF-THEN-ELSE se muestra en la figura 5.16.

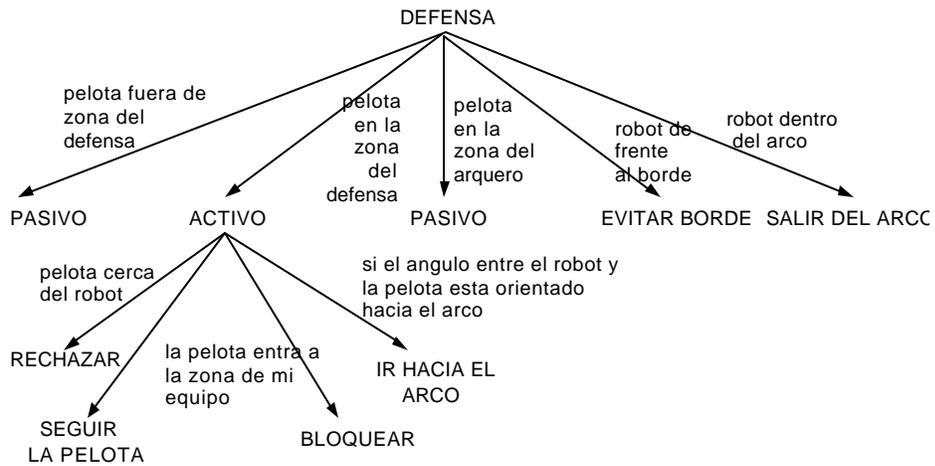


Figura 5.16.- Árbol de decisión para los defensas

Para la construcción del rol del *atacante* se utilizó la siguiente estructura de decisión, figura 5.17.

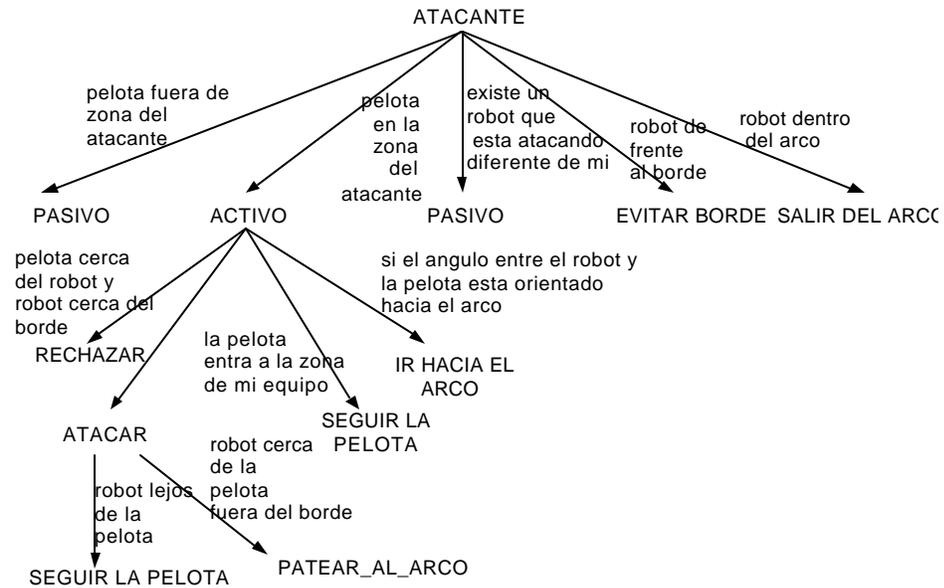


Figura 5.17.- Árbol de decisión para los atacantes

5.3. ANÁLISIS DE RESULTADOS EN LAS COMPETICIONES.

Antes de realizar un detalle de los resultados obtenidos en las competiciones de la FIRA 2002 (*Federación Internacional de Fútbol Robótico Asociado*) World Cup en la categoría de MIROSOT se debe destacar el principal inconveniente que sufrió el equipo de FUROEC en el momento de realizar sus partidos. Según el reglamento de la FIRA se establece que las condiciones de iluminación a la cual los sistemas de fútbol robótico deben funcionar adecuadamente son de aproximadamente 1000 lux. Sin embargo, en el primer partido que realizó FUROEC las condiciones de iluminación estuvieron a 2.5 veces por encima de lo normal. A pesar de que para los otros partidos las

condiciones de iluminación fueron reducidas, la intensidad lumínica fue de 1.7 veces por encima de lo reglamentario.

Durante la realización de los partidos se pudo apreciar que el control de movimiento y planificación de trayectoria en los robots juega un importante rol dentro del funcionamiento del sistema, puesto que una mejor implementación en estos campos mejorará la eficiencia global del equipo.

Nuestro sistema de control no incluía algoritmos para evitar obstáculos de manera eficaz, ni tampoco se aplicaron métodos de control tal como los de campos potenciales, descritos en el capítulo 3.

También se observó según los marcadores de la Tabla 5.2 que se necesitaba mejorar el control de movimiento de los robots para conseguir más precisión, así como también mejorar la estrategia general del equipo.

La mayoría de los equipos participantes trabajaron sobre asignación dinámica de las posiciones de los jugadores, contaban con un algoritmo de control de movimiento que les permitía trabajar a mayor velocidad, estrategias predefinidas de acuerdo a la situación del juego (similar a los de *plays off* como los mencionados por el equipo de CMUnited en [22], o por el FCPortugal [18] respecto a su *sistema de posicionamiento estratégico* basado en la situación).

Al momento de realizar las competiciones pudimos observar que nuestro equipo tuvo un buen desempeño a pesar de los resultados obtenidos y la falta de preparación en partidos reales con un equipo contrario, debido a la ausencia de otro equipo de transmisión para la construcción de un equipo oponente.

	UBASOT (Argentina)	Las Vegas REBEL (EEUU)	ITESM (México)
FUROEC (SIMUROSOT)	5-2	----	2-1
FUROEC (MIROSOT)	19-2	5-3	----

Tabla 5.2.- Marcadores de los partidos jugados en la FIRA 2002 World Cup

5.4. CONCLUSIONES

Diseñar estrategias de juego para un equipo de robots puede ser muy complicado, por ello es recomendable comenzar con algoritmos sencillos y probarlos en ambientes simulados, luego una vez que estos hayan sido suficientemente probados implementarlos en ambientes reales. Una importante plataforma de prueba para esto podría ser utilizar el simulador de fútbol robótico desarrollado por la FIRA.

Para mejorar el comportamiento de los robots es necesario aplicar técnicas de aprendizaje de máquina que permitan a los jugadores tener autonomía y realizar la acción más adecuada para alcanzar la meta.

Los algoritmos jerárquicos de decisión tales como árboles de decisión no fueron apropiados para definir los comportamientos de los robots, los cuales dependían de la posición de la pelota y de sus robots compañeros. Además, tomando en cuenta que los sistemas de fútbol robótico son ambientes dinámicos e impredecibles, resulta difícil construir una función que considere todas las posibilidades de juego, se requiere por tanto, aplicar otras técnicas como aprendizaje de máquina [32], como las implementadas en el capítulo IV.

A pesar de que en las competencias de la FIRA 2002 World Cup se trabajó con valores de intensidad lumínica diferentes a las impuestas por el reglamento, se pudo notar que nuestro sistema de visión depende en alto grado de las condiciones de iluminación en el campo de juego. Esto afectó al sistema de control el cual no recibía correctamente la información acerca de las posiciones de los objetos.

Finalmente, se debe destacar que estos sistemas son excelentes bancos de pruebas para desarrollar teorías relacionadas con inteligencia artificial distribuida, control de movimientos, planificación de trayectorias, visión artificial, mecatrónica, telecomunicaciones, y otras áreas afines.

CAPITULO VI

Sumario, Contribuciones y Futuros Trabajos

En este capítulo se presenta un resumen de las técnicas y algoritmos implementados en el desarrollo de este proyecto enfocándonos en mayor parte a resolver los problemas de control de robots móviles y al aprendizaje en sistemas multiagente, muestra también las aportaciones y mejoras que se hicieron después de la participación de nuestro equipo FUROEC en el campeonato mundial de fútbol robótico organizado por la FIRA y desarrollado en Corea en mayo del 2002. Futuras líneas de trabajo que pueden mejorar o complementar las técnicas aquí desarrolladas son propuestas al final del capítulo.

6.1 INTRODUCCIÓN

En el fútbol robótico cada equipo debe tener una estrategia de juego que le permita obtener el mayor número de goles sobre su equipo oponente. Esta estrategia de juego incluye darle a los robots un posicionamiento, una formación, un rol específico, etc. Cuanto mejor se aproveche la estrategia, el rendimiento del equipo se perfeccionará.

Al inicio de nuestro proyecto se desarrollaron un conjunto de nuevas técnicas básicas de movimiento para por ejemplo patear la pelota, evitar obstáculos, las cuales fueron mejoradas al complementarse con técnicas de control y aprendizaje de máquina que contribuyeron a mejorar la estrategia de juego del equipo. Una descripción detalla de todos los algoritmos implementados fue presentado en los capítulos anteriores.

En el transcurso de este capítulo abordaremos de forma resumida los nuevos algoritmos implementados y las contribuciones obtenidas de estos, también se describen los futuros trabajos que se pueden realizar en áreas relacionadas con el fútbol robótico.

6.2 SUMARIO.

6.2.1 Implementación del algoritmo para patear la pelota

Uno de los primeros algoritmos implementados por nuestro equipo permitía al robot patear la pelota, para esto se requirió de un estudio previo de la cinemática de los robots de forma que se pueda controlar adecuadamente sus movimientos dentro del campo de juego.

El objetivo al patear, es dirigir el robot hacia la posición de la pelota, para lo cual se requiere calcular las velocidades para cada rueda, y el ángulo de

dirección del robot para orientarlo hacia la pelota. Nuestro primer algoritmo define un punto hacia donde el robot debe ir, para luego orientarlo en dirección al arco, y enviar al robot hacia la pelota con la velocidad máxima.

6.2.2 Algoritmo para evitar obstáculos mediante la generación de objetivos intermedios.

Los robots tratan de evitar colisiones planificando un camino entre los obstáculos. Este algoritmo trabaja calculando objetivos intermedios cuando existen obstáculos enfrente del jugador, los cuales pueden ser compañeros o adversarios. Básicamente el robot se dirige hacia una posición objetivo y al encontrar un obstáculo en su camino calcula un objetivo intermedio, se dirige hacia él, para luego continuar su camino hacia el objetivo original.

La importancia de este algoritmo radica en que los robots pueden realizar sus funciones con mayor eficiencia en el momento de efectuar una jugada en conjunto y además evitar averías en los robots, debido a los continuos choques entre ellos.

Un segundo algoritmo para evitar obstáculos mediante la aplicación del método de campos potenciales fue también desarrollado en este proyecto. El sumario de este algoritmo es descrito en la sección 6.2.7.

6.2.3 Algoritmo para salir de los bordes del campo de juego.

El objetivo de este algoritmo es evitar que el robot quede atrapado en los bordes de la cancha o dentro de los arcos. Este algoritmo funciona haciendo girar al robot manteniendo uno de los motores con velocidad cero, y el otro diferente de cero, hasta sobrepasar el ángulo de orientación con el cual el robot puede quedar atascado en el borde del tablero. Cabe señalar que este algoritmo está diseñado para ser ejecutado en el menor tiempo posible, con la finalidad de que ninguno de los robots oponentes tomen ventaja de esta situación.

6.2.4 Algoritmo para mover el arquero de acuerdo a la posición de la pelota.

Este algoritmo tiene como objetivo principal permitir al arquero bloquear el paso de la pelota a su arco. Para impedir que la pelota ingrese al arco, el robot arquero deberá realizar movimientos constantes en la dirección del eje y (Figura 5.6 y Figura 5.7). Para esto, la orientación de este robot siempre será de $90^\circ \pm 1^\circ$, donde el frente del robot está hacia el borde superior.

Cabe mencionar que el proceso anterior es ejecutado siempre y cuando la pelota esté dentro de la zona de arquero. De otro modo, el robot siempre tiende a posicionarse en el centro del arco.

6.2.5 Implementación de la estrategia de juego para un equipo de cinco robots usando estructuras de decisión.

Para la implementación de este algoritmo, considerando un equipo de 5 robots (competición *Middle League MIROSOT* de la *FIRA*), se definieron: modos de juego (defensa, normal y ataque), zonas de juego (arquero, defensa 1, defensa 2, atacante 3 y atacante 4), posiciones pasivas y roles de cada jugador.

Inicialmente la estrategia del equipo se desarrolló mediante algoritmos jerárquicos IF-THEN-ELSE similares a un árbol de decisión, que obtiene información de una lista enlazada de dos nodos que guarda el estado actual y anterior de cada robot. Los últimos nodos del árbol representan comportamientos básicos (seguir la pelota, por ejemplo) que son guardados en la lista. Se debe mencionar que las trayectorias que sigue el robot son definidas en todo momento y dependen de la situación actual del partido.

Una parte importante en la estrategia de juego es definir las zonas para cada jugador, esto es la zona de arquero, la zona para los defensas y la zona para los atacantes. La decisión del comportamiento que tomará un jugador en un momento del partido la realiza un módulo de selección, basándose en la posición de cada robot en la cancha y el rol (arquero, atacante o defensa) que tiene asignado cada uno de estos.

Para el desarrollo de una estrategia para un equipo de tres robots se introdujo una técnica de aprendizaje por refuerzo llamada aprendizaje Q modular, la cual se explica en la siguiente sección 6.2.6.

6.2.6 Implementación de la estrategia de juego para un equipo de tres robots usando aprendizaje Q modular.

Es relativamente sencillo construir un rol para cada robot (atacante, defensa, arquero) y que cada uno lo lleve a cabo individualmente durante un partido. El problema radica en hacer que los robots coordinen sus movimientos y cooperen todos para cumplir con el objetivo del juego, que es ganar el partido. Nosotros hemos afrontado este problema usando una técnica de aprendizaje por refuerzo llamada *aprendizaje Q*, debido a que es un método muy utilizado en este tipo de aplicaciones.

Para la construcción de una estrategia para un equipo de tres robots (competición *Small League MIROSOT* de la *FIRA*), visto como un sistema multiagente, se propuso aplicar “aprendizaje Q modular”, aquí cada agente incluye un módulo de aprendizaje cuyas acciones a realizar son decididas por un módulo que hace las veces de mediador. El módulo mediador elige el rol que tendrá cada robot y las acciones que ejecutarán en un determinado momento del juego, esto es retribuido con una recompensa, cuyo valor se incrementa conforme se ejecuta cada acción.

Para la construcción de este algoritmo, de acuerdo a las situaciones de juego (estados) se consideró a 2 agentes (robots) como uno solo, llamado *agente acoplado*, con el objetivo de resolver en gran parte el problema de gasto en espacio de memoria, reduciendo el número de estados y acciones que un agente individual necesitaría para aprender a realizar las mejores acciones en un partido de fútbol.

Debido a que no es conveniente describir todas las situaciones de un juego de fútbol robótico solamente con algunos enunciados de condición, debido a su naturaleza dinámica y compleja, resulta factible emplear el aprendizaje por refuerzo, ya que nos permite enseñar al agente una conducta a través de iteraciones de pruebas y errores.

Otras de las ventajas que presenta el aprendizaje Q es que al aplicarlo no se necesita de un modelo o patrón de juego que deba seguir el agente, las decisiones a tomar pueden aprenderse directamente a partir de la retroalimentación de recompensas obtenidas a lo largo del juego.

6.2.7 Campos potenciales univectoriales usados para evitar obstáculos y para ir hacia la pelota.

Los algoritmos basados en funciones potenciales permiten controlar a los robots en tiempo real, haciendo que el robot sea atraído al objetivo y repelido por los obstáculos, sean estos adversarios o compañeros de equipo.

Los campos potenciales es un método en el cual se generan diferentes tipos de vectores de tal forma que el robot es orientado en la dirección de esos vectores. Estos vectores son generados continuamente permitiendo controlar a los robots en tiempo real.

En este trabajo se propone la aplicación de campos potenciales univectoriales para patear el balón y evitar obstáculos. Una de las ventajas que le encontramos al algoritmo basado en campos potenciales consistió en mejorar la eficiencia del robot atacante para patear la pelota, ya que con este método el robot no requiere utilizar muchos pasos para situarse frente a la pelota y orientarse hacia el arco contrario con el objetivo de patear el balón, si no que al robot le bastará con seguir la dirección del campo potencial univectorial que lo llevará hacia la pelota con la orientación deseada.

Los campos potenciales se pueden utilizar también para evitar colisiones con otros robots del mismo equipo o contrarios, la idea aquí es de que nuestro

robot desvíe su trayectoria al estar en posibilidad de chocar con otro robot. Para lograrlo se hace variar el ángulo de dirección del robot de tal forma que éste rodee al obstáculo para luego continuar con su trayectoria hacia su objetivo original. En la práctica se observó que se debe reducir la velocidad del robot cuando se presente la posibilidad de colisión, de esta forma el robot ejecuta mejor el algoritmo y tiene mayor probabilidad de evitar el choque.

6.3 CONTRIBUCIONES

Las principales contribuciones obtenidas con el desarrollo de este proyecto se enfocan básicamente en mejorar los algoritmos implementados para controlar el movimiento de los microrobots móviles con ruedas, así como los algoritmos de aprendizaje aplicados a sistemas de fútbol robótico. Se destacan además, otros logros adicionales obtenidos como resultado de estos trabajos.

En forma detallada las principales contribuciones se señalan a continuación:

6.3.1 Movimiento rápido del robot para patear la pelota

Un punto clave en el desarrollo de la habilidad para patear la pelota, fue aumentar significativamente la velocidad del robot al máximo, en el momento que el robot se encuentre bastante cerca de la pelota y orientado en dirección

del arco contrario. Esto último permitió dar mayor eficiencia en el ataque ya que la pelota adquiere mayor velocidad al dirigirse al arco contrario.

Debido a que la pelota por lo general se encuentra en movimiento, el algoritmo original para patear el balón fue mejorado calculando varios puntos a diferentes distancias de la pelota, que dependen de la posición de ésta en el tablero de juego. Estos puntos son usados para indicar al robot la posición donde debe ubicarse para luego orientarse en dirección del arco adversario y empujar el balón hacia la portería.

6.3.2 Habilidad mejorada para salir de los bordes de la cancha

Para implementar esta habilidad en los robots, se determinó que el valor absoluto del ángulo entre el robot y los bordes del tablero de juego, a partir del cual un robot debe girar para salir será mayor de 60° .

Una modificación al algoritmo implementado tanto para las competiciones de tres y cinco robots, consistió en definir el ángulo de giro del robot para salir del borde de la cancha, y por consiguiente el cálculo de las velocidades de los motores para ejecutar el giro, dependerá de la posición del objetivo hacia donde el robot quiere ir. Esto fue necesario específicamente cuando el robot y el objetivo se encontraban cerca del borde de la cancha, ante cual situación el robot debía girar en un sentido ya preestablecido para alcanzar su meta final.

6.3.3 Algoritmo optimizado para definir el comportamiento del arquero usando estimación de trayectorias.

Para definir la habilidad de bloquear la pelota cuando ésta se dirige hacia la portería, se incluyó una función de predicción tanto para el arquero como para los defensas con el fin de estimar la trayectoria del balón. De este modo, la posición estimada de la pelota era usada en forma óptima para ubicar al robot arquero en diferentes posiciones, para esto se asumen tres situaciones distintas, descritas en detalle en la sección 5.2.4.

6.3.4 Especificación de las condiciones para aplicar campos potenciales univectoriales modificados.

El campo potencial univectorial se aplicó de forma exitosa en los robots que cumplen el rol de atacante, mejorando su habilidad para patear la pelota hacia la portería contraria.

Se implementó además, un algoritmo para evitar colisiones usando campos potenciales con el propósito de minimizar la posibilidad de que el robot choque tanto con los compañeros como con los adversarios cuando se dirige a un punto objetivo. Este algoritmo calcula el ángulo del campo univectorial definiendo una línea tangente al obstáculo, el cual se asume de forma circular para propósitos de diseño del algoritmo.

Una modificación implementada a este algoritmo permite especificar las condiciones para evitar obstáculos, utilizando como parámetros la distancia del obstáculo a la recta formada entre el objetivo y el robot, el radio del círculo formado por el obstáculo, las distancias al obstáculo y al objetivo, así como también los valores absolutos de los ángulos entre el objetivo-robot, obstáculo-robot, ángulo-robot. Para mantener un mejor control del robot móvil durante la ejecución del algoritmo, la velocidad lineal del robot se reduce en presencia del obstáculo.

Este método de planificación en línea para agentes reactivos se puede aplicar en ambientes donde se obtenga información de las posiciones absolutas y el tamaño relativo de cada obstáculo, para que el robot pueda llegar a su objetivo final usando varios campos univectoriales que lo lleven a su destino con la orientación y posición adecuados.

6.3.5 Aplicación del algoritmo de aprendizaje Q modular.

El algoritmo de aprendizaje Q modular fue aprovechado para permitir el intercambio de roles entre un robot defensa y un robot atacante, esto disminuye la posibilidad de que exista bloqueo mutuo entre ambos robots, y aumenta la posibilidad de que el ataque o la defensa sean ejecutados por el robot que se encuentra en la posición más idónea para realizar una u otra función. Este algoritmo genera una estrategia de juego diferente a la descrita

en la sección 5, el cual se basa en estructuras de decisión, debido a que se definió una nueva lista de acciones para el caso del agente acoplado.

En conclusión, las técnicas de aprendizaje de máquinas tales como el aprendizaje Q modular permitirán que los agentes de un sistema multiagente puedan desenvolverse de forma autónoma en ambientes adversos, con el fin de resolver problemas de alta complejidad. Estas técnicas de aprendizaje pueden ser implementadas tanto en sistemas con robots reales como en sistemas con agentes de software.

6.3.6 Logros adicionales obtenidos con la ejecución de este proyecto.

Otros resultados obtenidos con la ejecución de este proyecto son:

- Implementar un equipo de fútbol robótico que puede jugar en el sistema de 5 contra 5, e implementar dos equipos de fútbol robótico que pueden competir en el sistema de 3 contra 3.
- Incentivar la elaboración y ejecución de proyectos de investigación en el campo de Visión por Computador, Inteligencia Artificial, Robótica, Procesamiento de Imágenes entre otras áreas. Se ha comenzado a trabajar con otro tipo de robots para iniciar nuevas líneas de investigación, y desarrollar un curso introductorio de robótica en la

ESPOL. Los resultados obtenidos en este proyecto semilla, motivaron a que la FIEC participe con una propuesta para el Programa VLIR-ESPOL en el que se incluye un componente de Visión por Computador y Robótica.

- Formar un grupo de investigadores jóvenes en áreas afines a la robótica. A más de ocho estudiantes que iniciaron el proyecto como tesis, actualmente se han vinculado al mismo dos estudiantes más y un profesor de la ESPOL.
- Publicar los resultados del proyecto en diferentes congresos y eventos, internacionales y nacionales:
 - *Survey on Learning in Multi-Agent Systems: Robot Soccer*, presentado en el 2002 FIRA Robot World Congress organizado por la FIRA en Seúl – Corea. Este documento va a ser publicado también en el libro *Intelligent Robots: Vision, Learning and Interaction* editado por KAIST PRESS en la sección *Learning for Navigation and Control*.
 - *Técnicas de Movimiento utilizadas por el equipo FUROEC en las competiciones de la FIRA 2002 World Cup*, presentado en

la categoría Póster de ESPOLCIENCIAS 2002 - Guayaquil-Ecuador.

- *Hacia la Construcción de un MicroRobot*, versión final, presentado en las IV Jornadas Iberoamericanas de Robótica. Ciudad de Panamá – Panamá, 2003. Esta publicación en una versión resumida fue presentada también en ESPOLCIENCIAS 2002 - Guayaquil – Ecuador.
- *Sistemas Multi-Agentes Aplicados al Fútbol Robótico: Problemática Existente*, presentado en las IV Jornadas Iberoamericanas de Robótica. Ciudad de Panamá – Panamá, 2003.
- Sentar las bases para la creación de un Laboratorio de Robótica y Visión por Computador en la ESPOL. La implementación del sistema permitió a la FIEC a crear un laboratorio de Robótica y Visión por Computador, en el que actualmente existen a más de los equipos del sistema de fútbol robótico, otros equipos de un proyecto de investigación de reconstrucción de imágenes financiado por el FUNDACYT.

- Pertener al organismo FIRA (*Federación Internacional de Fútbol – Robot Asociado*) para participar en los campeonatos y eventos organizados por el mismo. Como se mencionó anteriormente, la ESPOL participó en su primer mundial de fútbol robótico y actualmente es miembro activo de la FIRA. Además, el Dr. Boris Vintimilla fue nominado como miembro del Comité Ejecutivo de la FIRA en representación del Ecuador.
- Desarrollar una página web sobre el proyecto:
<http://www.fiec.espol.edu.ec/futbol/>
- Planificar y organizar el primer Campeonato Politécnico de Fútbol Robótico Simulado, el mismo que se desarrolló el 6 de Febrero de 2003 con la participación de 6 equipos.
- Obtener la donación de equipos para el sistema de fútbol robótico por parte de la FIRA por un valor superior a los 5.000 dólares norteamericanos. Auspicios de la empresa privada por un valor superior a los 2.000 dólares norteamericanos, y ayudas de la ESPOL adicionales por un valor de 1.500 dólares norteamericanos.

6.4 FUTUROS TRABAJOS

Los resultados de este proyecto de graduación, abren futuras líneas de trabajo que pueden ser asumidas por investigadores y estudiantes de la ESPOL o de otros centros de estudio superior, algunos de los cuales a nuestro criterio podrían ser:

- Aplicar el método de campos potenciales aquí implementado sobre otro tipo de robots que usen sensores incorporados, donde las fuerzas virtuales de repulsión ejercidas por los obstáculos son generadas a partir de la información recibida a través de los sensores.
- Futuras implementaciones y experimentos en simulación y en partidos reales mejorarán los algoritmos de control con campos potenciales complementándolas con otras técnicas tales como lógica difusa, programación evolutiva o redes neuronales, que mejoren gradualmente el control sobre estos robots de competición y que consideren también todas las restricciones cinemáticas.
- Modificar la técnica realizada de aprendizaje Q modular para ser aplicada sobre robots móviles de servicio, tales como: robots caseros encargados del aseo, seguridad, mensajería en oficinas. Se debe destacar que este tipo de robots tienen un rango de metas diferentes, que en paralelo pueden

entrar en conflicto entre sí, para lo cual es necesario seleccionar la mejor acción en un determinado estado.

- Aplicar un método de aprendizaje de máquinas que permita seleccionar el camino más óptimo usando la técnica de campos potenciales univectoriales.
- Usar otras técnicas de aprendizaje por refuerzo para resolver problemas de colaboración, coordinación, y construcción de estrategias de juego en línea para un equipo de robots autónomos. Y comparar estos resultados con los obtenidos en los algoritmos aquí implementados.
- Diseñar estrategias que definan posiciones, zonas de juego, roles de forma automática utilizando aprendizaje del adversario.
- Construir otro tipo de robots móviles con la capacidad de comunicarse entre sí y con sensores incorporados, que sirvan como banco de pruebas para las técnicas implementadas en este proyecto.
- Realizar trabajos en otros entornos de fútbol robótico simulado, tales como el software del *Soccer Server de la RoboCup* o el *Java Soccer Server*. Estos simuladores poseen otras opciones adicionales a las del simulador *Robot Soccer* de la FIRA.

- Aprovechar los robots de la MIROSOT y las técnicas desarrolladas para realizar otro tipo de aplicaciones.
- Diseñar y construir algoritmos de planificación de trayectorias para otros tipos de robots móviles,
- Aprovechar el fútbol robótico para incentivar a los estudiantes al estudio de disciplinas tales como la inteligencia artificial, visión por computadora, control automático, entre otras.
- Continuar participando en las futuras competencias y congresos relacionados con los sistemas multiagente a nivel mundial.

6.5 CONCLUSIONES

El presente trabajo sirve como base para desarrollar futuros proyectos e investigaciones sobre temas relacionados con el fútbol robótico y se constituye en uno de los primeros logros en esta área dentro del país.

Como resultado de este trabajo pudimos observar que existe una mejora considerable en el comportamiento general del equipo de robots en cada partido . Por tal motivo se debe continuar con el estudio de planificación de trayectorias e inteligencia artificial para mejorar mucho más el sistema del componente inteligente.

Varios de los algoritmos de control de movimientos desarrollados al inicio del proyecto aún son utilizados y algunos fueron mejorados.

A través de este proyecto la Escuela Superior Politécnica del Litoral (ESPOL) consiguió reconocimiento internacional en nuevas áreas de estudio al pertenecer al organismo FIRA (*Federación Internacional de Fútbol – Robot Asociado*) y participar en el campeonato mundial de fútbol robótico desarrollado en mayo del 2002, además de presentar artículos acerca de nuestras investigaciones.

BIBLIOGRAFIA

- [1] Alvarez Juan Carlos, "Planificación del Movimiento de Vehículos Autónomos basada en Sensores", (Tesis Doctoral Universidad de Oviedo 1998)

- [2] Arkin Ronald C., "Integrating Behavioral, perceptual and World Knowledge in Reactive Navigation", *Robotics and Autonomous Systems*, vol 6, pp. 105-122, 1990.

- [3] Barbuceanu Mihai, Fox Mark S., "COOL: A language for describing coordination in multi agent systems", In *Proceedings of the First International Conference on Multi-Agent Systems*, pages 17–24, Menlo Park, California, June 1995. AAAI Press.

- [4] Borenstein J., Y. Koren. "Real-time obstacle avoidance for fast mobile robots", *IEEE, Trans. Systems Man and Cybernetics*, pp- 1179-1187, 1989.

- [5] Brooks Rodney A., "A Robust Layered Control System for a Mobile Robot", *IEEE Journal on Robotics and Automation*, vol, RA-2, no 1, March 1986.

- [6] Federation of International Robot-soccer Association (FIRA). Homepage: <http://www.fira.net>.

- [7] Finin Tim, McKay Don, Fritzson Rich, McEntire Robin. "KQML: An information and knowledge exchange protocol". Kazuhiro Fuchi and Toshio Yokoi, editors, Knowledge Building and Knowledge Sharing. Ohmsha and IOS Press, 1994.
- [8] Gasser L., "Social conceptions of knowledge and action", Artificial Intelligence, Vol. 47, No. 1, (1991), pp 107-138.
- [9] Genesereth M. R., Fikes R. E.. "Knowledge interchange format version 3.0 reference manual". Technical Report Logic-92-1, Computer Science Department, Stanford University, 1992.
- [10] Itsuki Noda, Hitoshi Matsubara and Kazuo Hiraki, " Learning Cooperative Behavior in Multi-agent Environment", Electrotechnical Laboratory, Tsukuba, Ibaraki 305, Japan, August, 1996.
- [11] Kaelbling L.P., M. Littman L., Moore A. W., "Reinforcement Learning: A survey", Journal of Artificial Intelligence Research, vol 4, 1996.
- [12] Kim DH, Kim K-C, Kim Y-J, Kim Jong Hwan and Vadakkepat P., "Vector Field Based Path Planning and Petri-Net Based Role Selection Mechanism with Q-Learning for the Soccer Robot System", Intelligent Automation and Soft Computing, Vol 6, No. 1, P. 75-88, 2000.

- [13] Kim Jong Hwan, Kim Yong-Jae. "Univector Field Navigation Method for Fast Mobile Robots in Dynamic Environment", 2002 FIRA Robot World Congress Proc., pp. 165-170. Seoul, 2002.
- [14] Kim Jong Hwan, Kim Kwan Choon, Vadakkepat Prahlad , "Real-Time Control Schemes for Mirosof", Departament of Electrical Engineering, KAIST, 2000.
- [15] Kim Jong-Hwan, Kim Kwang-Choon, Kim Dong-Han, Kim Yong-Kae, Vadakkepat Prahlad, "Path Planning and Role Selection Mechanism for Soccer Robots". Proceedings of the 1998 IEEE, International Conference on Robotics & Automation, Leuven, Belgium – May 1998.
- [16] Kim, J.-H., Shim, H.-S., Kim, H.-S., Jung, M.-J., Choi, I.-H., and Kim, J.-O. "Cooperative multi-agent system and its real time application to robot soccer." Proc. IEEE Int. Conf. on Robotics and Automation, Albuquerque, New Mexico, Vol. 1, (April 1997), pp 638-643.
- [17] Kui Hong Park , Jong Kwan Kim, "Modular Q learning based Action Selection for Narosof", Proc. 2002 FIRA Robot World Congress, Seoul, Korea, May 2002, pp. 222-226.

[18] Lau Nuno, Reis Luis Paulo, "FC Portugal Team Description: RoboCup 2000 Simulation League Champion", University of Aveiro, Portugal, University of Porto, Portugal 2000.

[19] Nagendra Prasad M.V., "Learning Situation Specific Control in Multi-Agent Systems.", Dept. of Computer Science, (Tesis Doctoral de la Universidad de Massachusetts Amherst Mayo 1997) , Dept. of Computer Science, Uni. of Massachusetts Amherst, May 1997.

[20] Robot World Cup Initiative (RoboCup). Homepage: <http://www.robocup.org>.

[21] Russell S.J., Norvig P., "Artificial Intelligence: A modern Approach". Prentice Hall, Englewood Cliffs, 1995.

[22] Stone P., "Layered Learning in Multiagent Systems: Robot Soccer", School of Computer Science Carnegie Mellon University, Pittsburgh 1998.

[23] Stone P., Veloso M., "Beating a Defender in Robotic Soccer: Memory-Based Learning of a Continuous Function", Neural Information Processing Systems 1995, <http://www.cs.cmu.edu/afs/cs/usr/pstone/mosaic/pstone-papers.html>.

[24] Stone P., Veloso M., "Towards Collaborative and Adversarial Learning: A Case Study in Robotic Soccer", International Journal of Human - Computer Studies (IJHCS),

Volume 48, Number 1, 1998. <http://www.cs.cmu.edu/afs/cs/usr/pstone/mosaic/pstone-papers.html>.

[25] Stone P., Veloso M., “Using Testing to Iteratively Improve Training”, 1995-AAAI Fall Symposium on Active Learning, <http://www.cs.cmu.edu/afs/cs/usr/pstone/mosaic/pstone-papers.html>.

[26] Stone P., Veloso M.. "Multiagent systems: A survey from a machine learning perspective", *Autonomous Robotics*, volumen 8, número 3, July, 2000.

[27] Sutton R. S., Barto A. G., "Reinforcement Learning: An Introduction", Bradford Brooks/MIT Press, 1998.

[28] Uchibe E., Asada M., Hosoda K., “Strategy Classification in Multi-agent Environment – Applying Reinforcement Learning to Soccer Agents”, 1996, <http://www-robotics.ccm.eng.osaka-u.ac.jp/user/uchibe>.

[29] Uchibe E., Nakamura M., Asada M., “Co-evolution for Cooperative Behavior Acquisition in a Multiple Mobile Robot Environment”, 1998., <http://www-robotics.ccm.eng.osaka-u.ac.jp/user/uchibe>.

[30] Villarroel C., Carrillo R., Calderón C., "Técnicas de movimiento utilizadas por el equipo FUROEC en las competiciones de la FIRA 2002 World Cup" , EspolCiencia 2002, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador, Octubre 2002.

[31] Vintimilla B., Monsalve A. C., Villarroel S. C., Calderón G. C., Carrillo C. R., Salamea V., Paillacho C. D., Chicala J., Atiencia L., Jacho R., "Sistemas multiagentes aplicados al futbol robotico: problematica existente", IV Jornadas Iberoamericanas de Robotica, Seminario 3, Panama, 27-31 Enero, 2003

[32] Vintimilla B.X., Villarroel C., Calderón C., Carrillo R., "Survey on Learning in Multi-Agent Systems", 2002 FIRA Robot World Congress, Seoul, Korea, May 2002, pg. 305-308. ISBN: 89-86522-47-0-93560 * Este artículo fue seleccionado para ser publicado en el Libro de *Intelligent Robots: Vision, Learning and Interaction*, Sección: *Learning for Navigation and Control*, Editorial: KAIST Press, año 2003.

[33] School of Electrical and Computer Engineering-Oklahoma State University (ECEN). Homepage: <http://elec-engr.okstate.edu>.

[34] Arkin R.C. and Balch T.R., "AuRA: Principles and Practice in Review", Journal of Experimental and Theoretical Artificial Intelligence, in press, 1997. Available on the web at: <http://www.cs.cmu.edu/~trb/papers/>.

[35] Vadakkepat Prahlad, "Evolucionary Multi-Agent Robotic Systems (E-MARS): From a Robot Soccer Perspective", Dept. of Electrical Engineering, KAIST, 1998.

[36] Robot Soccer News. Homepage: http://www.robot-soccer.co.kr/grand/news_view.asp?num=198&psize=10

[37] Pagina de K-Team. Homepage: <http://www.k-team.com>.

[38] Página del robot SONY AIBO. Homepage: <http://www.aibo.com>.

[39] Página de IEEE Entity Web Hosting http://www.ewh.ieee.org/reg/10/Training2001/Robot_soccer-Vadakkepat.pdf

[40] Vintimilla B.X., Villarroel C., Calderón C., Carrillo R., "Univector Fields for Motion Control in Mirosot Robots ", submitido en el 2003 FIRA Robot World Congress, Austria.