

Fútbol de robots para la Liga Humanoide de RoboCup

Informe Final

23 de mayo de 2013

Marcelo Baliero - Gerardo Pias

<http://www.fing.edu.uy/~pgsalimoo/>

Tutores:
Gonzalo Tejera
Facundo Benavides
Serrana Casella

Instituto de Computación
Facultad de Ingeniería - Universidad de la República
Montevideo - Uruguay

Resumen

El presente documento se inscribe en el marco del proyecto de grado Salimoo que se propuso la construcción de un equipo de fútbol de robots para la Liga Humanoide KidSize de RoboCup. Los grandes desafíos del proyecto fueron desarrollar las habilidades individuales básicas de los jugadores, y dotarlos de comportamiento que les permita actuar como equipo. Como punto de partida se utilizaron los robots del kit robótico Bioloid Premium de Robotis junto con el sistema de visión HaViMo y el módulo de visión desarrollado en el proyecto VisRob II que permite el reconocimiento de los elementos del entorno. Asimismo sirvieron de referencia otros proyectos de la Facultad de Ingeniería de la Universidad de la República vinculados a la robótica humanoide y al fútbol de robots.

Para abordar el diseño de los movimientos básicos de los robots se utilizó una técnica que obtiene los movimientos a partir de transiciones entre distintas posturas, similar a la técnica de *stop motion*. La asignación de comportamiento a los robots fue realizada según una distribución de tareas basadas en roles. Es así que se construyeron estrategias acordes a la situación, distinguiendo a los jugadores según su rol. Se profundizó en algunos aspectos buscando mejorar la solución. En este sentido se trabajó en localización, a través de un modelo del entorno en el que se almacenan datos sobre los elementos fijos de la cancha. Además se incorporó la predicción de la trayectoria de la pelota, donde se procesan las tomas de la misma y se aproxima su movimiento mediante técnicas como mínimos cuadrados e interpolación lineal.

Durante el transcurso del proyecto se alcanzaron los objetivos planteados. Este trabajo aporta la constitución de los movimientos básicos: desplazarse, atajar y patear la pelota y levantarse del piso. Además se logró integrar con el módulo de visión generando operaciones de mayor nivel para la búsqueda y seguimiento de los elementos del entorno. Finalmente los comportamientos contruidos sobre esto abarcan la toma de posesión de la pelota y disparo al arco en el ataque, la interposición entre la pelota y el arco en la defensa y el seguimiento, identificación de la trayectoria y reacción del golero. En las pruebas se observa que los robots identifican estas situaciones, mayormente en el atacante y el golero.

Palabras Clave: Fútbol de Robots, RoboCup, Liga Humanoide, KidSize, Bioloid.

Agradecimientos

El equipo de Salimoo quiere agradecer a las personas que han acompañado el proyecto durante este difícil y enriquecedor proceso.

A los tutores Gonzalo Tejera, Facundo Benavides y Serrana Casella por su ayuda, dedicación y comprensión. A Gonzalo Gismero quien siempre estuvo dispuesto a evacuar nuestras dudas del uso del módulo de visión y a incorporar nuevas funcionalidades que pudieran colaborar con nuestra causa. También a Federico Andrade y Martin Llofriu por sus ideas, aportes y tiempo compartido en el laboratorio. A Juan “Paco” Saavedra por el aguante constante. Y a todo el equipo de robótica que siempre nos dio una mano cuando necesitábamos en esas largas horas en el laboratorio.

A lo largo de este camino ha sido fundamental el apoyo de nuestros afectos y seres queridos por lo que no podemos dejar de agradecerles también a ellos.

Gerardo quiere agradecer y dedicar este trabajo a sus padres y familia. A Lu por la compañía incondicional. A Leo y Lu por el apoyo. A mis amigos de siempre. Y a mis compañeros de trabajo que aguantaron las locuras y huidas.

Marcelo quiere agradecer a sus padres y a su hermana por su constante apoyo y por todos los sacrificios realizados. A Victoria por su apoyo y comprensión en todo momento. Por último, agradecer a todos aquellos amigos que siempre estuvieron presentes en todo este tiempo.

A todos ellos, ¡Muchas Gracias!

Índice general

1. Introducción	1
1.1. Objetivos del Proyecto	2
1.2. Organización del Documento	2
2. Marco de trabajo	5
2.1. RoboCup	5
2.2. Liga Humanoide - Categoría Kid Size	6
2.3. Recursos	9
2.3.1. Robots	9
2.3.2. Entorno de juego	12
2.4. Trabajos relacionados	13
2.4.1. Proyecto VisRob II	13
2.4.2. Proyecto de Construcción de Robots Bípedos	13
2.4.3. Proyecto Forrest	14
2.4.4. Tesis Fútbol de Robots UBASot	14
2.4.5. Proyecto Fútbol de Robots Uruguayo para Torneos.	14
2.4.6. Proyecto FIbRA.	15
3. Solución propuesta	17
3.1. Movimientos	17
3.1.1. Definición de movimientos	17
3.1.2. Realización de los movimientos	18
3.1.2.1. Caminar	21
3.1.2.2. Desplazamiento lateral	23
3.1.2.3. Rotación en el lugar	24
3.1.2.4. Levantarse del piso	25
3.1.2.5. Atajar el balón	28
3.1.2.6. Patear el balón	28
3.1.2.7. Movimientos de la cabeza	30
3.1.3. Obtención de poses	30
3.1.4. Transición entre poses	33

3.2.	Estrategia	35
3.2.1.	Ataque	35
3.2.1.1.	Acciones en ataque	36
3.2.2.	Defensa	36
3.2.2.1.	Acciones en defensa	37
3.2.3.	Neutral	37
3.2.3.1.	Acciones en situación neutral	38
3.3.	Estado del mundo	38
3.3.1.	Actualización de datos	40
3.3.1.1.	Rotaciones	40
3.3.1.2.	Caminata	40
3.3.1.3.	Pasos laterales	42
3.3.2.	Pérdida de precisión	45
4.	Aspectos de implementación	47
4.1.	Arquitectura	47
4.2.	Toma de decisiones y cálculos realizados	48
4.2.1.	Posesión de la pelota	48
4.2.2.	Interceptar ataque	50
4.3.	Caídas	52
4.3.1.	Detección	52
4.3.2.	Sentido	54
4.3.3.	Corrección	55
4.4.	Predicción de trayectoria de la pelota	56
5.	Pruebas y resultados	61
5.1.	Pruebas funcionales	61
5.1.1.	Indicadores	61
5.1.2.	Escenarios	63
5.1.2.1.	Escenario I - Ataque	63
5.1.2.2.	Escenario II - Defensa	65
5.1.2.3.	Escenario III - Golero atajando	66
5.1.2.4.	Escenario IV - Golero Perdido	68
5.2.	Pruebas no funcionales	68
5.3.	Resultados	69
5.3.1.	Resultados de las pruebas funcionales	69
5.3.1.1.	Escenario I - Ataque	69
5.3.1.2.	Escenario II - Defensa	72
5.3.1.3.	Escenario III - Golero Atajando	74
5.3.1.4.	Escenario IV - Golero perdido	76
5.3.2.	Resultados de las pruebas no funcionales	77

6. Conclusiones y trabajo futuro	83
6.1. Conclusiones	83
6.1.1. Habilidades básicas de los agentes.	84
6.1.2. Integración con el módulo de visión	85
6.1.3. Conformación del equipo	85
6.2. Trabajo futuro	86
6.2.1. Mejoras y profundización	86
6.2.2. Ideas para el futuro	87
Bibliografía	89
A. Glosario	93

Índice de figuras

2.1. Dimensiones del campo de juego Kid Size.	8
2.2. Visión del campo de juego a escala.	9
2.3. Tipos de robots humanoides Bioloid.	10
2.4. Cámara HaViMo.	10
2.5. Modificaciones realizadas a los robots BIOLOID.	11
2.6. Indumentaria de los robots.	12
2.7. Campo de juego.	12
3.1. Rango de ángulos válidos de los motores AX-12 [Rob06]	18
3.2. Postura corporal erguida del agente que se obtiene con la pose inicial (p_0)	19
3.3. Secuencia de imágenes fijas de <i>stop motion</i> [LOH]	20
3.4. Estados inicial y final para dar un paso con el pie izquierdo. .	21
3.5. Estados y transiciones de las plantas de los pies durante la caminata.	23
3.6. Estados y transiciones de las plantas de los pies durante el desplazamiento lateral derecho.	24
3.7. Estados y transiciones de las plantas de los pies durante la rotación derecha.	25
3.8. Secuencia de poses del robot al levantarse del piso de forma frontal.	26
3.9. Secuencia de poses del robot al levantarse del piso de espalda. .	27
3.10. Regiones donde la pelota puede estar posicionada con respecto al robot, y las acciones que se toman para ubicarse con el fin de patear.	29
3.11. Secuencia de poses del robot para patear la pelota con el pie derecho.	29
3.12. Captura de Roboplus Motion, donde se ven en las columnas derechas las posiciones de los 18 motores corporales.	31
3.13. Mapa de motores corporales del robot tipo A.	32
3.14. Actualización de los datos en el estado del mundo al realizar una caminata.	42

3.15. Obtención del ángulo para realizar la triangulación durante el desplazamiento lateral hacia la derecha.	43
3.16. Obtención del nuevo ángulo hacia el objeto, luego del desplazamiento lateral hacia la derecha.	45
4.1. Diagrama de los diferentes componentes de la arquitectura. . .	48
4.2. Cálculo de la distancia entre el agente B y la pelota.	49
4.3. Cálculos al momento de interceptar un ataque.	51
4.4. Sensor de giro	52
4.5. Esquema de referenciación de la pelota en la predicción de trayectoria.	57
5.1. Escenario I - Estado inicial I	64
5.2. Escenario I - Estado inicial II.	64
5.3. Escenario II - Estado inicial I.	65
5.4. Escenario II - Estado inicial II.	66
5.5. Escenario III.	67
5.6. Escenario IV.	68
5.7. Resultados generales obtenidos del escenario I.	70
5.8. Resultados generales obtenidos del estado inicial I.	70
5.9. Resultados generales obtenidos del estado inicial II.	72
5.10. Resultados generales obtenidos del escenario II.	72
5.11. Defensa detectada e interceptada durante las pruebas.	73
5.12. Resultados generales obtenidos del estado inicial I.	73
5.13. Resultados generales obtenidos del estado inicial II.	74
5.14. Comparación de modelos Doraemon - Golero.	76

Índice de cuadros

2.1. Dimensiones del campo de juego Kid Size (cm)	8
5.1. Detección del disparo	74
5.2. Resultados pruebas no funcionales, temperatura.	78
5.3. Estadísticas de las gráficas de temperatura.	78
5.4. Resultados pruebas no funcionales, voltaje.	80
5.5. Estadísticas de las gráficas de voltaje.	80
5.6. Resultados pruebas no funcionales, relación de la carga.	81
5.7. Estadísticas de las gráficas de relación de la carga.	81

Capítulo 1

Introducción

El estudio de robots humanoides es un área de investigación que concentra mucho interés principalmente por sus diversas aplicaciones en la vida cotidiana, como pueden ser en la medicina, en la industria, en las tareas del hogar, entre tantas otras. La robótica está integrándose cada vez más en la vida de las personas donde crecen las exigencias en la resolución de tareas cada vez más complejas y con la mayor autonomía posible.

RoboCup [Roba] es una de las organizaciones vinculadas a la robótica más importantes a nivel mundial. Entre sus fines, persigue la promoción del estudio e investigación de la inteligencia artificial y robótica, buscando soluciones a estos problemas a través del juego. En particular propone competencias de fútbol de robots como medio para motivar investigaciones y medir sus avances. El entorno y las características que plantea un partido de fútbol son el escenario ideal para resolver problemas complejos que atañen a la robótica de una forma lúdica y motivada también por la competencia.

El Instituto de Computación ha comenzado a trabajar en RoboCup desde el año 2005 y tiene interés en el desarrollo de sistemas robóticos que puedan competir en la Liga Humanoide acompañando los objetivos planteados por la misma lo que motiva este proyecto.

En el presente trabajo se describe la concepción de un equipo de fútbol de robots humanoides, cumpliendo con las reglas establecidas por RoboCup [Roba] para la liga de fútbol KidSize. Los robots usados son los del kit robótico Bio-loid Premium de Robotis [Robd], con modificaciones tanto para cumplir con el reglamento que impone la liga como para mejorarlos según las necesidades que surgieron en el transcurso del proyecto. Este trabajo es el fruto del proyecto de grado Salimoo de la Facultad de Ingeniería [Fac] de la Universidad de la República [Unib]. Este documento resume el trabajo realizado donde se destacan el análisis de los problemas a resolver, las soluciones propuestas y el sendero de trabajo que plantea este proyecto.

1.1. Objetivos del Proyecto

Con la motivación de poder lograr avances en el área de fútbol de robots humanoides autónomos, surge el proyecto Salimoo. Los objetivos planteados dentro del mismo son:

1. Relevamiento del estado del arte de los problemas a abordar al trabajar en la liga Humanoide de RoboCup.
2. Lograr definir e implementar las habilidades básicas necesarias para que los agentes puedan desempeñarse en este entorno.
3. Definir e implementar roles y establecer una estrategia de juego, logrando de esta forma, conformar un equipo de robots humanoides capaces de jugar al fútbol de forma autónoma.

1.2. Organización del Documento

El documento está organizado en diferentes capítulos, cuyo contenido se detalla a continuación.

- Capítulo 2: Se detalla el marco de trabajo en el cual se realizó el proyecto: las características de la competencia para la que se desarrolla el equipo, los recursos disponibles y algunos trabajos previos relacionados.
- Capítulo 3: Se enumeran los diferentes aspectos que componen la solución llevada a cabo. Se especifican los movimientos que son capaces de realizar los agentes y la técnica utilizada para implementarlos. Se detallan los roles y las acciones definidas, así como la representación del entorno que los robots realizan.
- Capítulo 4: Los aspectos de implementación son listados, tales como la arquitectura desarrollada, la toma de decisiones y los cálculos realizados por los agentes durante el juego. También se incluye en este capítulo, los métodos utilizados para la detección de las caídas y para la predicción de la trayectoria de la pelota.
- Capítulo 5: Se detalla el plan de pruebas que se realizó y los resultados que dichas pruebas arrojaron.
- Capítulo 6: Son expuestas las conclusiones del proyecto y el trabajo futuro a realizar en el marco del fútbol de robot humanoides.

- Glosario: Se definen los principales términos y conceptos utilizados a lo largo del documento.
- Bibliografía: Se presentan las referencias bibliográficas.

Capítulo 2

Marco de trabajo

En este capítulo se mencionan las condiciones que enmarcan el proyecto y las características de los diferentes elementos utilizados para la construcción del equipo de fútbol. Dentro de estos elementos se encuentran: los robots utilizados para ser los jugadores, la cancha de fútbol donde se realizaron las pruebas y el proyecto de visión con el módulo HaViMo[Mob07], cuyos resultados fueron utilizados.

2.1. RoboCup

RoboCup[Roba] es un proyecto internacional que promueve, a través del juego en formato de competencias, la investigación y educación sobre las diferentes áreas que componen a la robótica. Como cuenta su historia, la idea de que los robots jugaran al fútbol fue mencionada por primera vez por el profesor Alan Mackworth (Universidad de British Columbia, Canadá) en el artículo titulado «On Seeing Robots» de 1992 [Ala92], y más tarde publicado en el libro «Computer Vision: System, Theory, and Applications» de 1993. Es en ese año que surge la iniciativa para realizar una competencia de carácter internacional donde se utilizara el fútbol para la promoción de la ciencia y la tecnología. El proyecto resultante fue nombrado como Robot World Cup (abreviado RoboCup).

Las competencias comenzaron oficialmente en el año 1997, contemporáneamente a grandes sucesos de la historia de la robótica y la inteligencia artificial. Ese año IBM Deep Blue[IBM] derrotó al campeón mundial de ajedrez Gary Kasparov, y la NASA a través de su misión Pathfinder[NAS] colocó en la superficie de Marte al primer sistema autónomo robótico conocido como Sojourner. Continuando con el desafío que se ha planteado la comunidad robótica de enfrentar en competencias a los seres humanos con los robots,

RoboCup tiene como objetivo final desarrollar para el año 2050 un equipo de jugadores de fútbol robótico capaz de vencer al campeón de la Copa del Mundo humano.

El foco principal de las competencias ha sido desde el comienzo el fútbol teniendo como meta principal la investigación de sistemas cooperativos multi-robot y multi-agente en entornos dinámicos adversos. Sin embargo se han ido ampliando las líneas de investigación como ser en la ayuda del hogar o en la asistencia frente a accidentes o catástrofes. También cuenta con una categoría para jóvenes motivándolos a aprender sobre ciencia, tecnología, ingeniería y matemática, así como fomentar sus habilidades mediante la participación en el proceso creativo de la construcción y programación de robots autónomos.

Dentro de estas temáticas se definen diferentes ligas en las que se desarrollan las distintas competencias que plantea RoboCup. Es de interés en este trabajo focalizarse en en la Liga Humanoide dentro de la categoría de fútbol que se describe a continuación.

2.2. Liga Humanoide - Categoría Kid Size

Los robots que participan en las competencias de la Liga Humanoide[Rob11] deben actuar de forma autónoma mientras el partido se está llevando a cabo. No se permite ninguna fuente de alimentación externa, al igual que no se permite la teleoperación o control remoto de los agentes. Sí está permitido que los robots de un equipo se comuniquen entre sí en cualquier momento durante un juego.

En la Liga Humanoide los robots deben tener un aspecto similar al cuerpo humano, es por este motivo que los mismos deben estar compuestos por dos piernas, dos brazos y una cabeza que están unidos a un tronco. Los robots deben ser capaces de mantenerse en pie y caminar. Los únicos modos de locomoción que se permiten son caminar y/o correr. En lo referente a la altura de los mismos, para la categoría Kid Size, la medida mientras el robot se encuentra de pie, debe ser entre 30 cm y 60 cm.

Los equipos participantes en las competencias de la Liga Humanoide deben dotar a sus robots con sensores que tengan un equivalente en los sentidos humanos. Estos sensores se deben colocar en una posición equivalente a la ubicación de los sensores biológicos del humano. Por tales motivos se establecen las siguientes restricciones sobre los sensores a usar y su ubicación en el robot.

1. No se admiten dispositivos capaces de emitir luz, sonido u ondas electromagnéticas que interfieran en el entorno. Sólo se permite el uso de un altavoz que puede ser colocado en la cabeza, el cuello o el tronco

del robot, emitiendo sonidos similares a los de un humano en volumen y frecuencia .

2. Sensores externos, como cámaras y micrófonos (hasta dos permitidos), no se pueden colocar en las piernas, los brazos o el torso de los robots. Deben ser colocados en la cabeza del robot y por encima de cualquier articulación del cuello.
3. El campo de visión de los robots está limitado en todo momento a 180 grados. Esto significa que el ángulo máximo entre dos puntos cualesquiera de la superposición del campo de visión de todas las cámaras montadas en el robot debe ser inferior a 180 grados. También el movimiento de giro e inclinación de la cabeza y las cámaras montadas en el robot se limita a fin de mantener un modelo similar al humano. Por lo tanto, el mecanismo para girar la cámara se limita a 270 grados lo que significa ± 135 grados de la posición mirando hacia delante. El mecanismo para inclinar la cámara se limita a ± 90 grados (medidos respecto al plano horizontal).
4. El número de cámaras se limita a una configuración de visión estéreo (es decir, máximo 2 cámaras) solamente. Visión monocular también está permitido.
5. Sensores de contacto, de fuerza, y de temperatura pueden ser colocados en cualquier parte del robot.
6. Sensores dentro del robot pueden medir todas las magnitudes de interés, incluyendo (pero no limitado) voltajes, corrientes, fuerzas, movimientos, aceleraciones y velocidades de rotación. Los mismos pueden estar en cualquier parte dentro del robot.

El campo de juego consiste de una alfombra verde sobre un suelo plano y uniforme. Las especificaciones de las dimensiones se pueden ver en la imagen 2.1 y sus valores en la tabla 2.1.

Las líneas del campo son de color blanco de un ancho de 5 cm. Los arcos deben estar centrados respecto a la línea del ancho del campo, siendo uno de color amarillo y el otro de color azul, con una altura de 80 cm. Los postes laterales se colocan enfrentados en la mitad de la cancha y cada uno tiene tres franjas de color de 15 cm de alto. Las franjas superior e inferior son del mismo color del arco que se encuentra a la izquierda del poste, siendo la restante franja del color del otro arco como se puede observar en la figura 2.2.

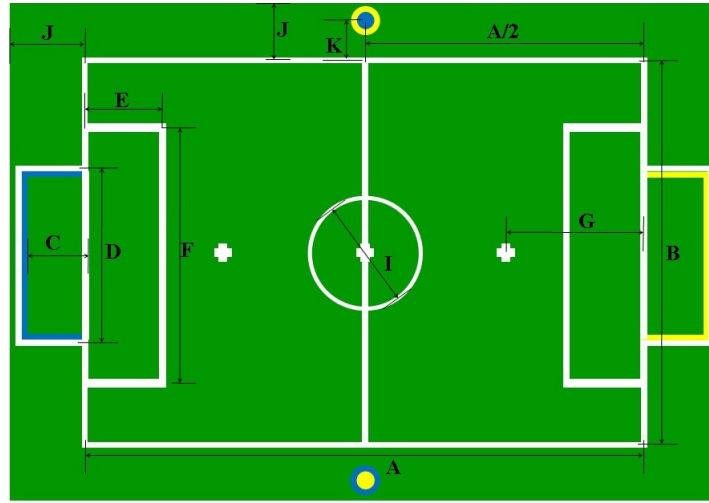


Figura 2.1: Dimensiones del campo de juego Kid Size.

A	largo del campo	600
B	ancho del campo	400
C	profundidad del arco	50
D	ancho del arco	150
E	largo del área	60
F	ancho del área	300
G	distancia punto penal	180
I	diámetro círculo central	120
J	ancho externo del campo (min)	70
K	distancia del poste al campo	40

Cuadro 2.1: Dimensiones del campo de juego Kid Size (cm)

La pelota utilizada en los partidos de la categoría Kid Size es una pelota de tenis estándar de color anaranjado. En los partidos de la categoría participan dos equipos conformados por tres integrantes, donde uno de ellos se distingue como golero. Las reglas generales del fútbol de robots están orientadas a ser las mismas reglas establecidas por la FIFA para el fútbol, con determinadas salvedades. Cada equipo debe identificarse con un color: uno con cían y el otro con magenta. De no existir un acuerdo entre los equipos se sortea el color utilizando una moneda. Estos colores deben estar presentes al menos en los brazos y piernas de los jugadores, teniendo en cuenta que el resto del cuerpo debe presentar en su mayoría color gris o negro y que lógicamente no se aceptan los colores destinados a los restantes elementos del campo. La duración del partido es de dos tiempos de 10 minutos con un entretiempo

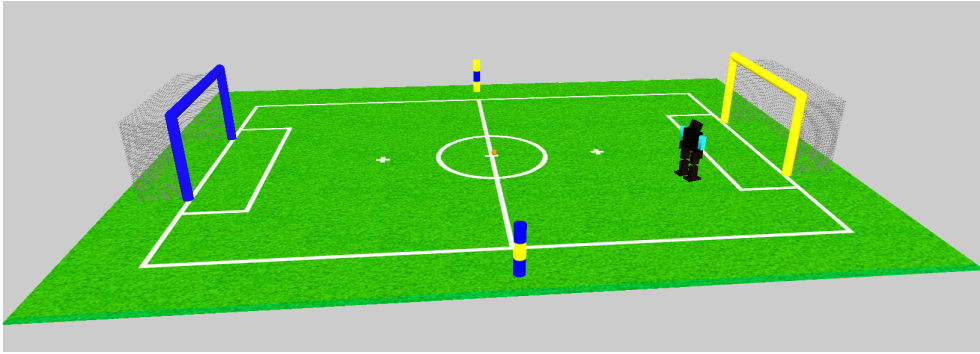


Figura 2.2: Visión del campo de juego a escala.

de 5 minutos. Si el partido termina empatado y es necesario determinar un ganador, se agregan dos periodos extras de 5 minutos de duración. En caso de que el empate persista en el período extra el partido se define por tanda de penales. Sin embargo si durante el tiempo reglamentario ninguno de los equipos logró patear al arco contrario, el tiempo extra es saltado y se pasa directamente a los penales. Cada equipo puede solicitar lo que se denomina tiempo muerto (máximo uno por período) donde se puede brindar servicio a los robots. Cada tiempo muerto dura a lo sumo 2 minutos.

2.3. Recursos

En esta sección se describen los recursos disponibles para llevar a cabo el proyecto.

2.3.1. Robots

Para la realización del proyecto se utilizaron los robots humanoides Bioloid del kit Premium de Robotis[Robd] y el módulo de visión HaViMo[Mob07]. El kit cuenta con un manual[Robb], en el cual se detalla paso a paso el armado de tres tipos diferentes de humanoides, los cuales se pueden apreciar en la figura 2.3. La principal diferencia entre las tres opciones de robots es la disposición de los motores en la cadera, lo cual impacta directamente sobre los movimientos realizables con las piernas.

Se puede ver en la figura 2.4 la cámara que incluye el módulo de visión HaViMo[Mob07] utilizada para la visión de los robots.

El robot que se decidió construir fue el robot tipo A, por ser el que cuenta con 18 grados de libertad, a diferencia de los otros dos modelos que poseen sólo 16 grados de libertad. Por contar con dos grados de libertad



Figura 2.3: Tipos de robots humanoides Bioloid (dimensiones: 24 cm de ancho y 38 cm de alto).

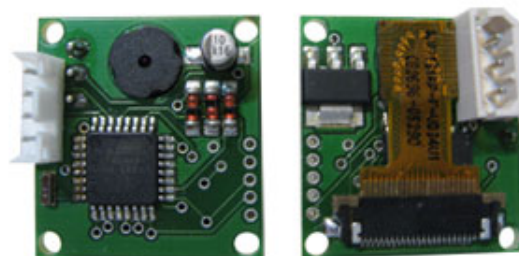


Figura 2.4: Cámara que incluye el módulo de visión HaViMo (dimensiones: 2,2 cm de ancho y 2,2 cm de alto).

adicionales en la cadera, posibilita así una mayor cantidad de movimientos con las piernas.

Se debieron realizar diferentes modificaciones a los robots para ajustarlos a las necesidades requeridas y principalmente a las reglas establecidas por RoboCup [Roba, Rob11]. Se detalla a continuación la lista de modificaciones realizadas.

- Se quitó el sensor de distancia que tenía ubicado en el pecho, dado que el mismo no se correspondía con ninguno de los seis sentidos que disponen los humanos. Esta es una de las restricciones sobre los sensores fijada por RoboCup [Roba, Rob11].
- Se quitó el receptor de las señales del joystick ubicado en la cabeza, garantizando así la autonomía del robot.
- Para poder integrar las cámaras que incluye el módulo de visión HaViMo [Mob07] (figura 2.4) a los robots, se mandó construir, en acrílico, una carcasa que permitiera la sujeción de las cámaras a las piezas del kit de forma segura (al ser un elemento externo al kit, el encastre no

era directo) y al mismo tiempo que pudiera protegerlas de eventuales golpes.

- Se removió el casco de plástico que tenía el robot por cabeza y se incorporaron dos motores en su lugar para permitir movilidad horizontal y vertical a la cámara. Sin la movilidad vertical, la cámara no podría capturar la pelota en el piso si se encontrase a menos de un metro del robot. Con esta incorporación de motores se alcanzó un total de 20 grados de libertad para cada uno de los robots y una altura final de 39 cm.

Los cambios realizados se pueden apreciar en la figura 2.5.

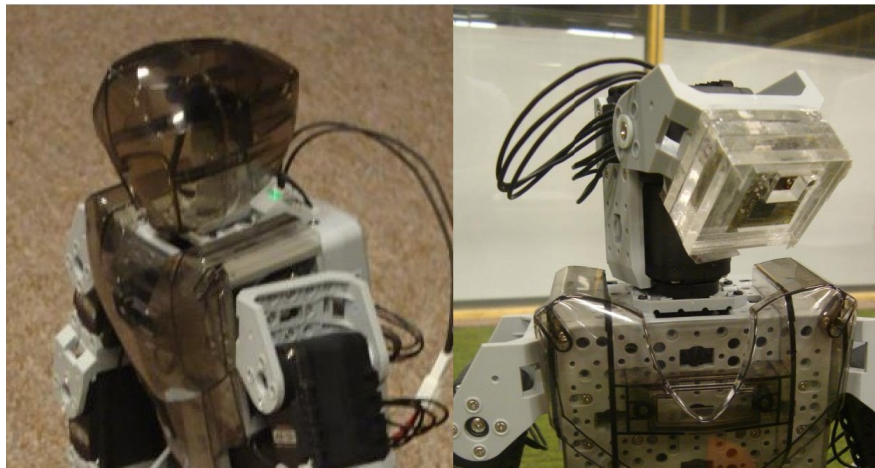


Figura 2.5: Modificaciones realizadas a los robots BIOLOID (izq: cabeza según viene en el kit - der: cabeza modificada para el proyecto).

Por otra parte, el laboratorio de robótica facilitó la indumentaria de los robots (ver figura 2.6), que permiten la distinción de los dos equipos en el juego. Dicha indumentaria consiste en un pantalón y una remera para el robot. Los colores que se disponían eran magenta y negro para un equipo y cían y negro para el otro.



Figura 2.6: Indumentaria de los robots.

2.3.2. Entorno de juego

Para la realización de pruebas, se dispuso de una cancha de fútbol de 3 metros de largo por 4 metros de ancho. Este tamaño es equivalente a medio campo de juego de una cancha oficial, según las dimensiones establecidas por RoboCup[Rob11]. Además se cuenta con dos arcos, uno amarillo y otro azul, cuyas dimensiones son de 150 cm de ancho y 80 cm de alto, dos postes laterales de 45 cm de alto y 10 cm de diámetro y una pelota anaranjada. Todos los colores de los elementos son los colores indicados en las reglas de la liga.

En la figura 2.7 se puede apreciar el campo de juego.

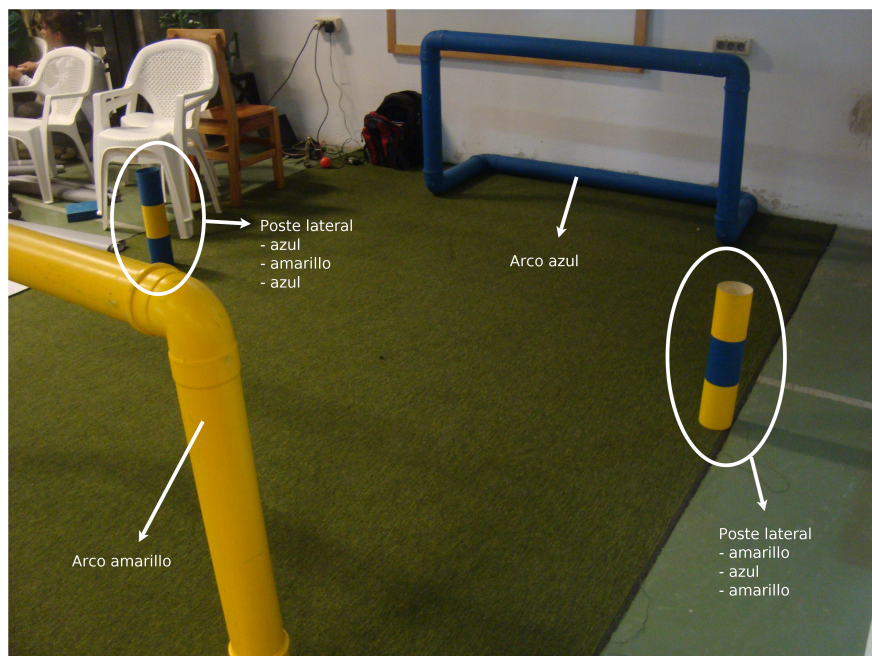


Figura 2.7: Campo de juego.

2.4. Trabajos relacionados

2.4.1. Proyecto VisRob II

VisRob II[Gon] es un proyecto de grado de la Facultad de Ingeniería[Fac] de la Universidad de la República[Unib] finalizado en el año 2012, sobre visión de robots utilizando el módulo de visión HaViMo[Mob07]. El resultado logrado en dicho proyecto, le permite al agente identificar y recabar información de los diferentes objetos que se encuentran en el entorno y que son captados por la cámara. El ángulo en grados y la distancia en centímetros, conforman la información que se obtiene de cada objeto reconocido.

Los objetos identificados por la aplicación son los siguientes:

- Pelota de fútbol.
- Postes laterales de la cancha, diferenciando el poste lateral derecho del poste lateral izquierdo.
- Compañeros de equipo.
- Agentes del equipo contrario.
- Arcos, diferenciando el arco del equipo contrario del arco propio.
- Postes laterales de los arcos.

La identificación de los objetos se realiza por medio de los colores que presentan los mismos, en el caso de los jugadores, la distinción es en base al color de la indumentaria de los equipos.

Un aspecto importante en la identificación de objetos, es la discriminación entre el objeto arco y el objeto poste de arco. Esta importancia radica en que para distinguir el arco, se requiere ver al menos un palo y el travesaño o los dos palos en una misma captura, con lo que se requiere de cierta distancia entre la cámara y el arco para que este pueda ser identificado. El hecho de identificar el poste lateral del arco como objeto en sí, permite tener una noción del arco cuando la distancia al mismo no es suficiente para que sea identificado.

2.4.2. Proyecto de Construcción de Robots Bípedos

Construcción de Robots Bípedos[Dam05] es un proyecto de grado de 2004 de la Facultad de Ingeniería[Fac] de la Universidad de la República[Unib], que abarcó el estudio de robótica bípeda, logrando la construcción de un prototipo de robot bípedo capaz de caminar cuasiestáticamente. Entre los

aportes originales del proyecto se encuentra la estrategia para caminar del prototipo construido, que utiliza un enfoque análogo a la técnica de *stop motion* utilizada en animación. Dicha técnica de caminata implementada, fue la utilizada para la realización de los diferentes movimientos de los agentes en el presente proyecto.

2.4.3. Proyecto Forrest

El proyecto de grado Forrest[Ser06b] de la Facultad de Ingeniería[Fac] de la Universidad de la República[Unib], logró desarrollar en el 2006 un equipo de fútbol de la liga de simulación 2D de RoboCup, formado por once agentes de software autónomos que operan en un simulador de fútbol. Entre las principales contribuciones del proyecto se incluye el estudio e implementación de las técnicas grafos de coordinación y planificación en línea de dos niveles, así como su aplicación combinada en la estrategia de Forrest (nombre dado al equipo de fútbol desarrollado en el proyecto). Si bien estas técnicas no fueron utilizadas, Forrest fue usado como referencia a la hora de realizar el modelado de la realidad[Ser06a].

2.4.4. Tesis Fútbol de Robots UBASot

UBASot[Cla02] es un equipo de fútbol simulado realizado en el marco de una tesis de licenciatura de la Universidad de Buenos Aires[Unia] del año 2002. En dicho trabajo se realizó un estudio del problema desde el análisis de cada uno de los aspectos que entran en juego en el desarrollo de un equipo de fútbol de robots, así como también de las distintas técnicas utilizadas en referencia a sistemas multi-agentes y al desarrollo de habilidades especiales en robots móviles. Esta tesis fue usada como referencia en este trabajo, al momento de definir las acciones y los roles de los diferentes agentes. También sirvió de referencia al igual que el proyecto Forrest, para realizar el modelado de la realidad.

2.4.5. Proyecto Fútbol de Robots Uruguayo para Torneos.

Este proyecto de grado realizado en la Facultad de Ingeniería[Fac] de la Universidad de la República[Unib] finalizado en el año 2004, tuvo como resultado la creación del equipo F.R.U.T.O.[Alv04], desarrollado para competir en la categoría Middle League SimuroSot de FIRA [FIR]. Este trabajo recorre los pasos de la construcción de un equipo de fútbol robótico de manera

integral. Desde el control de movimiento pasando por las acciones y la estrategia. Resulta una interesante fuente de información para la construcción de un equipo de fútbol robótico. Entre otros aportes, el proyecto brinda herramientas auxiliares útiles para el desarrollo de equipos simulados: un simulador, un debugger y proxys que permiten la interacción con el simulador oficial de la categoría antes mencionada. Si bien este trabajo no pertenece al área de simulación, la construcción de F.R.U.T.O. formó parte de las referencias utilizadas para armar el equipo. Además, como se menciona más adelante en la sección 4.4, la implementación de Traking de pelota fue utilizada para realizar la predicción de la trayectoria de la pelota.

2.4.6. Proyecto FIbRA.

Este proyecto de grado realizado en la Facultad de Ingeniería[Fac] de la Universidad de la República[Unib] finalizado en el año 2006, logró la construcción de un equipo de fútbol de robots para la categoría Middle League SimuroSot de FIRA [FIR]. El equipo FIbRA [Gus06] surge del trabajo concentrado sobre tres líneas: reconocimiento y predicción del juego oponente, planificación de movimientos y toma de decisiones. En el trabajo se utilizan algunas técnicas interesantes como el uso de lógica difusa para la toma de decisiones e ideas basadas en el comportamiento de colonias de hormigas para la resolución del problema de reconocimiento de la formación del equipo oponente. Este trabajo no fue utilizado concretamente ya que las técnicas expuestas presentan un nivel de complejidad que supera el alcance propuesto para este proyecto. Sin embargo sirvió de guía para la definición de roles y acciones para los agentes y estuvo presente durante la construcción del equipo.

Capítulo 3

Solución propuesta

En el presente capítulo se detalla toda la construcción del equipo de fútbol de robots humanoides. En primera instancia se explica la obtención de la movilidad de los robots, incluyendo la definición de los movimientos que son capaces de realizar y la técnica para lograrlos. La segunda parte del capítulo detalla la estrategia que sigue el equipo al momento de estar jugando un partido de fútbol.

3.1. Movimientos

3.1.1. Definición de movimientos

Para poder definir el conjunto de movimientos necesarios que deben poder ser realizados por los agentes, se debe tener en cuenta qué roles cumplen cada uno de ellos dentro del equipo. Por regla de la liga, uno de ellos indefectiblemente debe desempeñarse como el golero del equipo y los otros dos agentes como jugadores de campo, que en principio deberían poder realizar los mismos movimientos, dado que la distinción de sus roles se da por diferencias en su comportamiento y no por diferencias en sus movimientos.

Dada la amplia variedad de movimientos que pueden ser realizados por los robots y dada la gran complejidad que implica implementarlos, se decidió descartar todos aquellos que aportaran al desempeño del jugador pero que no fueran indispensables para la tarea de jugar un partido de fútbol, como es el caso, por ejemplo, de caminar hacia atrás. Se definió en una primera instancia el conjunto de movimientos mínimos necesarios para que los robots se pudieran desempeñar como jugadores de fútbol. La lista de movimientos resultantes fue la siguiente:

- Caminar hacia adelante.

- Desplazarse de forma lateral (a la izquierda y a la derecha).
- Rotar en el lugar (a la izquierda y a la derecha).
- Levantarse del piso (de forma frontal y de espaldas).
- Tirarse para atajar el balón (a la izquierda y a la derecha).
- Patear el balón.
- Movimientos de la cabeza (hacia ambos lados y hacia arriba y abajo).

3.1.2. Realización de los movimientos

Los diferentes movimientos de los robots fueron realizados mediante secuencias de diferentes poses o posturas. Las mismas no son más que un conjunto de pares (*motor, posición*) que relacionan a cada motor con una ubicación determinada.

Al momento de indicar una posición a los motores, se debe considerar que los motores AX-12 [Rob06] tienen un rango de 300° alcanzables, que se encuentran discretizados en 1024 posiciones. El rango de posiciones alcanzables se puede visualizar en la figura 3.1.

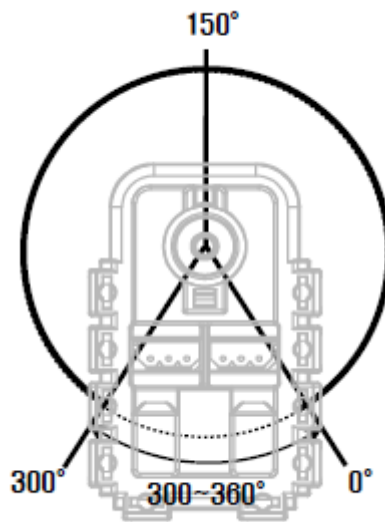


Figura 3.1: Rango de ángulos válidos de los motores AX-12 [Rob06]

De esta forma quedan definidas las siguientes posiciones:

- La posición 511 ubica al motor en la posición central con respecto a su rango de movilidad, alcanzando así los 150° .
- La posición 0 ubica al motor en el ángulo de 0° .
- La posición 1023 ubica al motor en el ángulo de los 300° .

Considerando esta discretización en el movimiento de los motores, se puede establecer la siguiente pose:

$$p_0 = \{(1, 235), (2, 788), (3, 279), (4, 744), (5, 462), (6, 561), (7, 358), (8, 666), (9, 507), \\ (10, 516), (11, 341), (12, 682), (13, 240), (14, 783), (15, 647), (16, 376), (17, 507), (18, 516)\}$$

Ésta ubica a cada motor en la posición indicada para lograr la postura erguida del robot y en equilibrio (ver figura 3.2), donde los pies se encuentran alineados, las piernas juntas y los brazos junto al cuerpo. Dicha pose se toma como pose inicial y es precondition para realizar cualquier movimiento. Es por este motivo que todos los movimientos deben finalizar en p_0 , de forma de garantizar que el próximo movimiento que se realice comience desde la misma.



Figura 3.2: Postura corporal erguida del agente que se obtiene con la pose inicial (p_0)

De ahora en adelante y por motivos de simplicidad, las poses son representadas como un vector de posiciones, donde el lugar dentro del vector indica

a qué motor corresponde la posición. Considerando esto, la pose inicial se representa como el vector:

$$p_0 = \{235, 788, 279, 744, 462, 561, 358, 666, 507, 516, 341, 682, 240, 783, 647, 376, 507, 516\}$$

Notar que las posturas sólo asocian una ubicación a 18 de los 20 motores que contienen los robot, esto debido a que quedan excluidos los dos motores que permiten el movimiento de la cámara, ya que se independizó el movimiento de la cabeza del movimiento corporal. Esta independencia permite realizar el «barrido» del entorno sin establecer una postura corporal puntual.

Una vez determinado el conjunto de poses que tienen que ser alcanzadas por el robot para lograr cierto movimiento, se realizan las transiciones entre las mismas en el orden correspondiente. Los cambios de ubicación de los motores entre los valores establecidos por las diferentes posturas, posibilitan la realización del movimiento por parte del agente. La cantidad de poses que integran la secuencia varía según la complejidad del movimiento que se pretenda realizar.

Esta técnica de lograr un movimiento a partir de la transición entre diferentes posturas es similar a la técnica de animación *stop motion*, donde se logra aparentar el movimiento de objetos estáticos mediante la transición de imágenes fijas. Se muestra un ejemplo en la figura 3.3 de una serie de imágenes utilizadas para *stop motion*.



Figura 3.3: Secuencia de imágenes fijas de *stop motion*[LOH]

Si los movimientos implican un gran número de poses para lograrlos o si son repetitivos (como caminar) se descomponen en submovimientos más

simples, como por ejemplo *Caminar*, que puede ser descompuesto en una primera instancia en los submovimientos *dar un paso con el pie derecho* y *dar un paso con el pie izquierdo*. Los beneficios que brinda realizar la descomposición de movimientos en submovimientos más simples son esencialmente, simplificar la implementación y facilitar las pruebas de los mismos.

A continuación se detallan cada uno de los movimientos implementados y las diferentes descomposiciones en submovimientos realizadas.

3.1.2.1. Caminar

Caminar implica que los robots realicen reiteradamente pasos hacia adelante, con el fin de poder desplazarse una distancia deseada. Es por este motivo que el caminar es indispensable para los robots dentro del campo de juego durante un partido de fútbol.

Debido a que la cantidad de pasos que se realizan durante una caminata es arbitraria y depende directamente de la distancia a ser recorrida, se debe contar con un submovimiento que permitan realizar a los robots un solo paso y cuya repetición permita realizar la caminata y alcanzar las distancias requeridas. En una primera instancia, el movimiento caminar puede verse como la sucesión de pasos izquierdos y derechos, dado que con esos submovimientos alcanza para lograr el desplazamiento de los robots. Para dar un paso con el pie izquierdo el robot se debe encontrar en una pose donde tenga el pie izquierdo por detrás de la línea del pie derecho para finalizar con el pie izquierdo por delante del pie derecho y así lograr el paso como se muestra en la figura 3.4. De forma análoga, para poder dar un paso con el pie derecho debe comenzar en una pose con el pie derecho por detrás del izquierdo y finalizar en una pose con el pie derecho por delante del izquierdo.

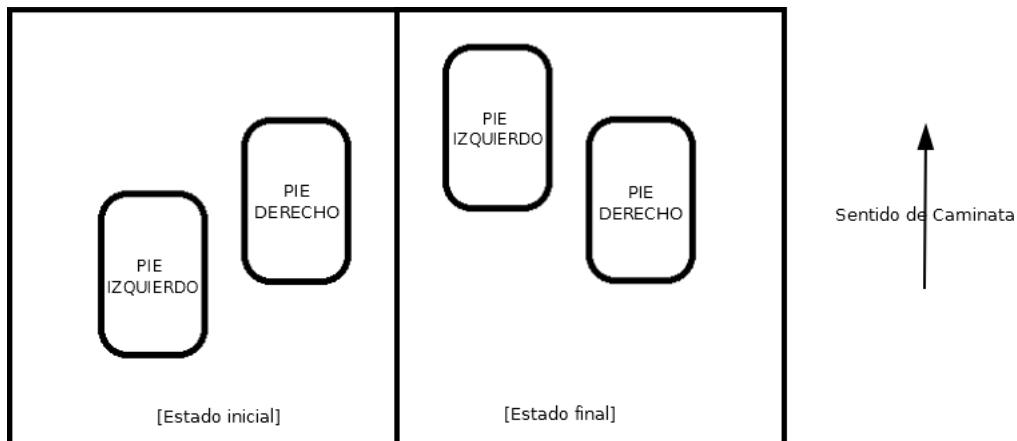


Figura 3.4: Estados inicial y final para dar un paso con el pie izquierdo.

Considerando el estado de los pies en el estado inicial y final al dar un paso, sin importar cual es el pie con el que se realiza el paso, en ninguno de los dos estados se tiene a los pies alineados. Esto implican que si la caminata se descompone solamente en los submovimientos *dar un paso con el pie izquierdo* y *dar un paso con el pie derecho*, el robot comenzaría y finalizaría el movimiento *caminar* con los pies desalineados. Esto no es conveniente considerando que el robot debe realizar otros movimientos que no sean caminar. Por el contrario, es conveniente que el robot comience los diferentes movimiento desde una posición conocida, en la que permanezca en equilibrio y con los pies alineados, de tal forma que le facilite la realización de los diferentes movimientos. Con lo cual, como se mencionó en la sección 3.1.2, se considera que para realizar cualquier movimiento el robot debe encontrarse en la pose inicial p_0 , en la cual se mantiene erguido y con los pies alineados. Debido a que el robot debe iniciar la caminata partiendo con los pies alineados y finalizar la misma de igual forma, el movimiento caminar debe contener dos submovimientos iniciales (uno por cada pie) que permitan desalinear los pies a partir de la pose inicial y dos submovimientos finales (uno por cada pie) que permitan alinear los pies para retomar la pose inicial y garantizar que cualquiera sea el movimiento a ejecutarse posterior a la caminata, comience desde p_0 .

En resumen, el movimiento caminar se subdivide en seis submovimientos más simples: dos submovimientos iniciales que consisten en el adelantamiento de un pie a partir de la postura inicial para comenzar la caminata, dos submovimientos intermedios que consisten en la realización de un paso y cuya repetición determina la distancia recorrida en la caminata y dos submovimientos finales que permiten alinear los pies del robot y volver a la postura inicial. La cantidad par de cada uno de los submovimientos se debe a que cada uno se realiza tanto con el pie izquierdo como con el derecho.

Se puede ver que cualquier caminata que se realiza iniciando con el pie izquierdo, puede ser realizada iniciando con el pie derecho, simplemente se debe invertir de pie todos los pasos con que se realiza la caminata. De esta forma, se puede concluir que iniciar la caminata con el pie derecho o con el izquierdo es indiferente y no es necesario que los robots sean capaces de realizar ambos submovimientos iniciales. Se decidió arbitrariamente que los robots inicien las caminatas siempre con el pie derecho y por tal motivo, el submovimiento inicial con el pie izquierdo no fue implementado, reduciendo así a cinco los submovimientos que componen la caminata.

La simplificación que se realizó en el comienzo de la caminata no es aplicable al final de la misma, debido a que se necesita tanto finalizar con el pie derecho como con el pie izquierdo, por no saber apriori cuantos pasos son necesarios para cubrir la distancia a recorrer, por lo que no se puede deter-

minar cual de los dos pies va a quedar por delante del otro luego de realizar el último paso.

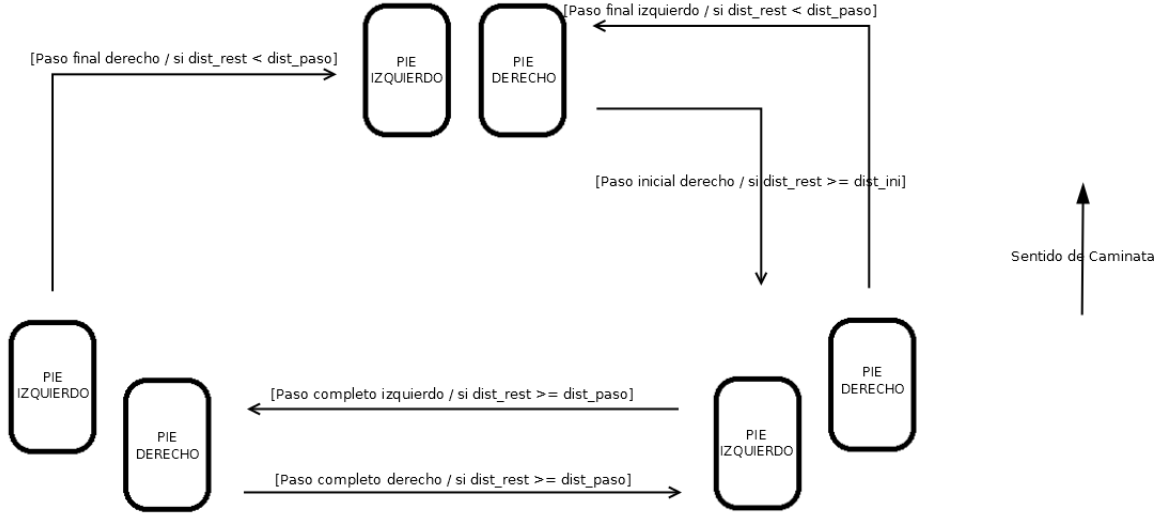


Figura 3.5: Estados y transiciones de las plantas de los pies durante la caminata.

Se puede ver en la figura 3.5, los diferentes estados que van adoptando los pies del robot al caminar y las transiciones que fueron implementadas. Las diferentes transiciones están determinadas dependiendo de la distancia que resta por recorrer durante la caminata ($dist_rest$), por la distancia que logra desplazarse el robot al momento de dar un paso ($dist_paso$, equivalente a 7 cm) y por la distancia que recorre al adelantar un pie sobre el otro ($dist_ini$, equivalente a 5 cm). El sentido de la caminata en la figura está indicado por la flecha al costado derecho de la imagen.

3.1.2.2. Desplazamiento lateral

Al igual que la caminata, el desplazamiento lateral es un movimiento de carácter repetitivo, cuya duración depende de la distancia a ser recorrida. Por este motivo también se subdividió en submovimientos más simples que se repiten hasta alcanzar la distancia requerida. En este caso se descompuso al movimiento solamente en dos submovimientos, uno que inicia el paso lateral, en el cual se abre la pierna correspondiente a la dirección del desplazamiento, y otro en el que se finaliza el paso lateral, moviendo la segunda pierna y

recobrando la posición inicial. Notar que a diferencia de la caminata, no hay submovimientos de inicio y finalización del desplazamiento, debido a que el paso lateral, comienza y termina con el robot en posición erguida (p_0), posibilitando así que el desplazamiento lateral pueda ser implementado como una sucesión de estos dos submovimientos definidos.

Si bien se puede implementar el paso lateral completo como un solo submovimiento, e implementar al desplazamiento lateral como una repetición de este, la subdivisión en dos submovimientos facilita la evaluación de la realización del movimiento, dado que las pruebas se pueden dividir por cada uno de los submovimientos implementados.

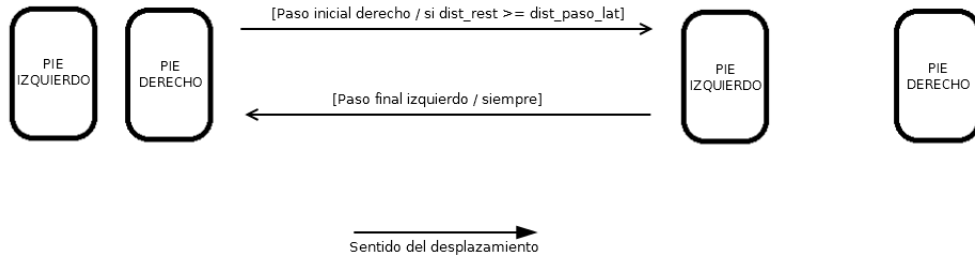


Figura 3.6: Estados y transiciones de las plantas de los pies durante el desplazamiento lateral derecho.

En la figura 3.6 se pueden ver las dos posiciones que adoptan los pies al momento de desplazarse lateralmente hacia la derecha y de forma análoga a la caminata. Las transiciones dependen de la distancia que resta por recorrer ($dist_rest$) y de la distancia que se desplaza el agente al dar un paso ($dist_paso_lat$), equivalente a 3 cm cuando el paso es a la izquierda y a 1 cm cuando es a la derecha. Si bien los movimientos de paso lateral izquierdo y paso lateral derecho son simétricos, diferencias estructurales provocan que el desplazamiento no sea el mismo en ambas direcciones. El submovimiento que finaliza el paso lateral siempre se ejecuta de forma de terminar el desplazamiento lateral en p_0 . Durante el desplazamiento lateral izquierdo las plantas de los pies adquieren una posición análoga al desplazamiento lateral derecho.

3.1.2.3. Rotación en el lugar

La rotación en el lugar, es el tercer movimiento de carácter repetitivo que realiza el agente, se aclara que la rotación implementada es realizada con el robot mirando hacia «afuera» del giro. Al igual que el desplazamiento lateral,

la rotación se subdivide solamente en dos submovimientos más simples, uno que inicia el paso de la rotación, moviendo el pie del lado correspondiente al sentido del giro y otro submovimiento que lo finaliza moviendo el pie opuesto y retornando a la pose inicial p_0 . La sucesiva repetición de estos pasos posibilita la rotación en el lugar de los robots.

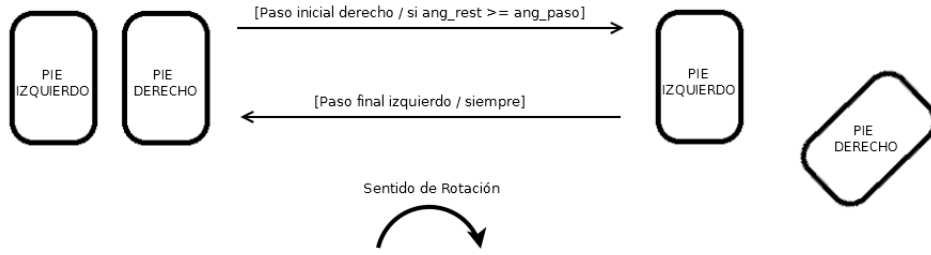


Figura 3.7: Estados y transiciones de las plantas de los pies durante la rotación derecha.

En la figura 3.7 se pueden ver las dos posiciones que adoptan los pies al momento de rotar hacia la derecha, siendo análoga la rotación hacia la izquierda. En este caso, la realización de las transiciones depende de la distancia angular que resta por alcanzar (ang_rest) y del ángulo que rota el robot al dar un paso (ang_paso), equivalente a 11° cuando es a la izquierda y a 15° cuando es a la derecha. Como ocurre con el desplazamiento lateral, los pasos de la rotación presentan diferentes desplazamientos según si es a la izquierda o a la derecha debido a diferencias estructurales. Al igual que en el desplazamiento lateral, siempre se ejecuta el submovimiento que finaliza el paso de la rotación para que el robot termine el movimiento en p_0 .

3.1.2.4. Levantarse del piso

Una vez que se produce una caída por parte del robot, el mismo puede finalizar en el piso boca abajo o boca arriba, por lo que se tienen dos movimientos para realizar la incorporación del agente: uno que levanta al robot de forma frontal, y otro de espalda. Ambos logran volver a posicionar al robot de pie en la pose erguida p_0 . Dichos movimientos no fueron subdivididos en submovimientos debido a que las subdivisiones posibles no generaban submovimientos reutilizables.

A continuación se puede apreciar en las figuras 3.8 y 3.9 las secuencias de poses que ejecuta el robot para poder levantarse del piso tras sufrir caídas hacia adelante y hacia atrás.

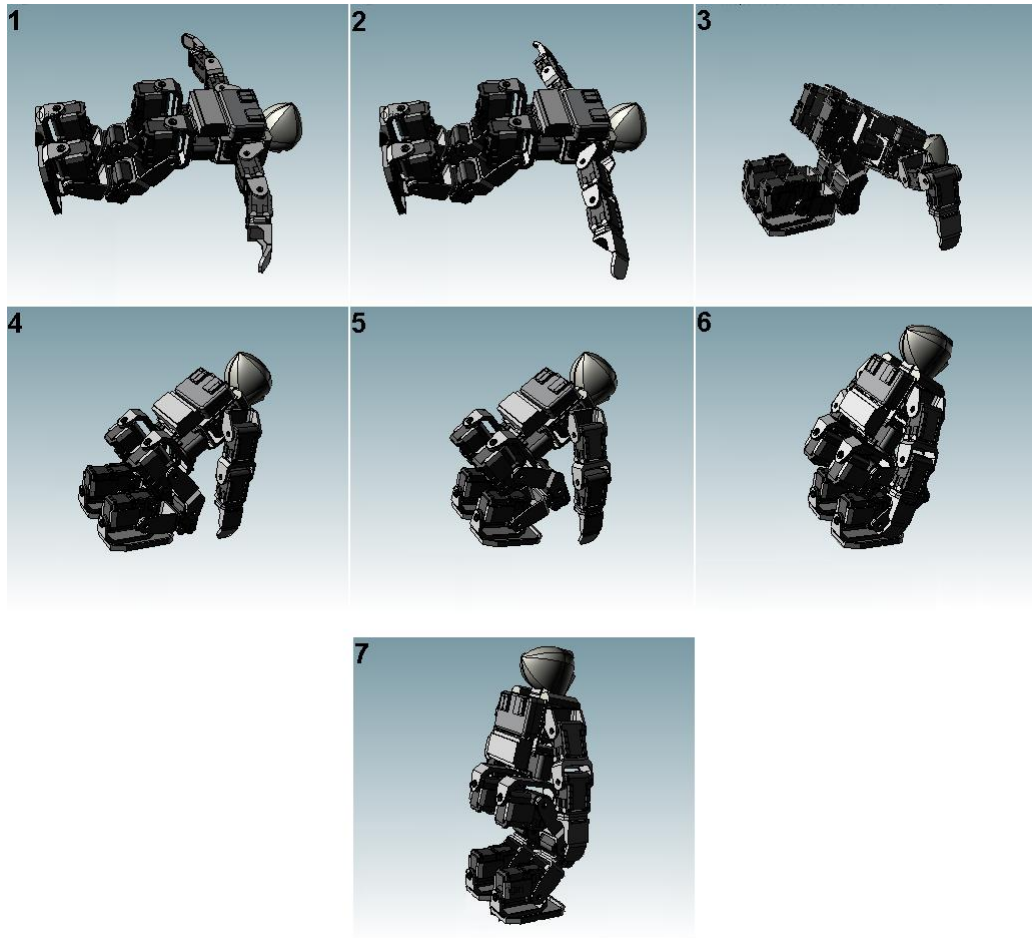


Figura 3.8: Secuencia de poses del robot al levantarse del piso de forma frontal.



Figura 3.9: Secuencia de poses del robot al levantarse del piso de espalda.

3.1.2.5. Atajar el balón

Los dos movimientos implementados para atajar el balón por parte del golero, tanto hacia la izquierda como a la derecha, corresponden a movimientos en sentido lateral, simétricos entre ellos, donde el robot estira uno de sus brazos hacia el piso tratando de cubrir la mayor distancia horizontal del lado donde se aproxima la pelota, mientras que simultáneamente estira el pie opuesto en dirección contraria. Se buscó que el robot no se arroje al piso al momento de atajar sino que, manteniendo su posición erguida, intente cubrir la mayor parte del arco que le sea posible. Esta decisión se tomó tratando de minimizar las veces que el robot debe incorporarse del piso, debido al tiempo que le insume levantarse y reubicarse con respecto al arco. En contrapartida, el hecho de no tirarse reduce las posibilidades de atajar el balón, debido a que el espacio del arco que el golero obstruye cuando se arroja al piso, es mayor al espacio del arco que cubre con la posición erguida.

3.1.2.6. Patear el balón

Al momento de implementar el movimiento para patear el balón, no se realizó ninguna descomposición en submovimientos, debido a que las subdivisiones posibles no generaban submovimientos reutilizables. El agente es capaz de patear el balón tanto con el pie izquierdo como con el derecho. Si bien implementando una sola de las patadas es capaz de desempeñarse como jugador, el hecho de disponer de los dos movimientos ahorra tiempo al momento de acomodarse frente al balón, dado que si solamente se contara con una de las piernas para patear, por ejemplo la izquierda, el agente debería indefectiblemente acomodarse para esa pierna, incluso en ocasiones en que estuviera en perfecta ubicación para patear con la derecha.

En la figura 3.10 se puede apreciar la forma con la cual el robot determina con qué pie debe patear, en base a la posición que visualiza la pelota en relación con sus pies. Cabe destacar que al momento de determinar con que pie patear el robot ya está posicionado correctamente para efectuar el remate, es decir que tiene la pelota a su alcance y ya se encuentra enfrenteado al lugar donde desea patear.

En la figura 3.11 se muestra la secuencia de poses que el agente ejecuta para lograr patear la pelota con el pie derecho.

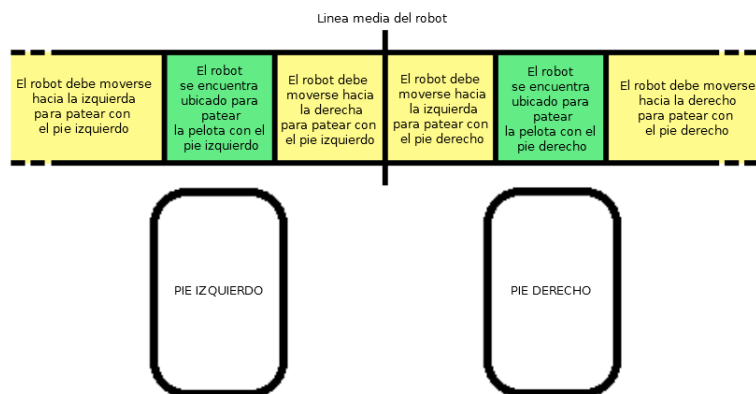


Figura 3.10: Acciones al patear según posición de la pelota.

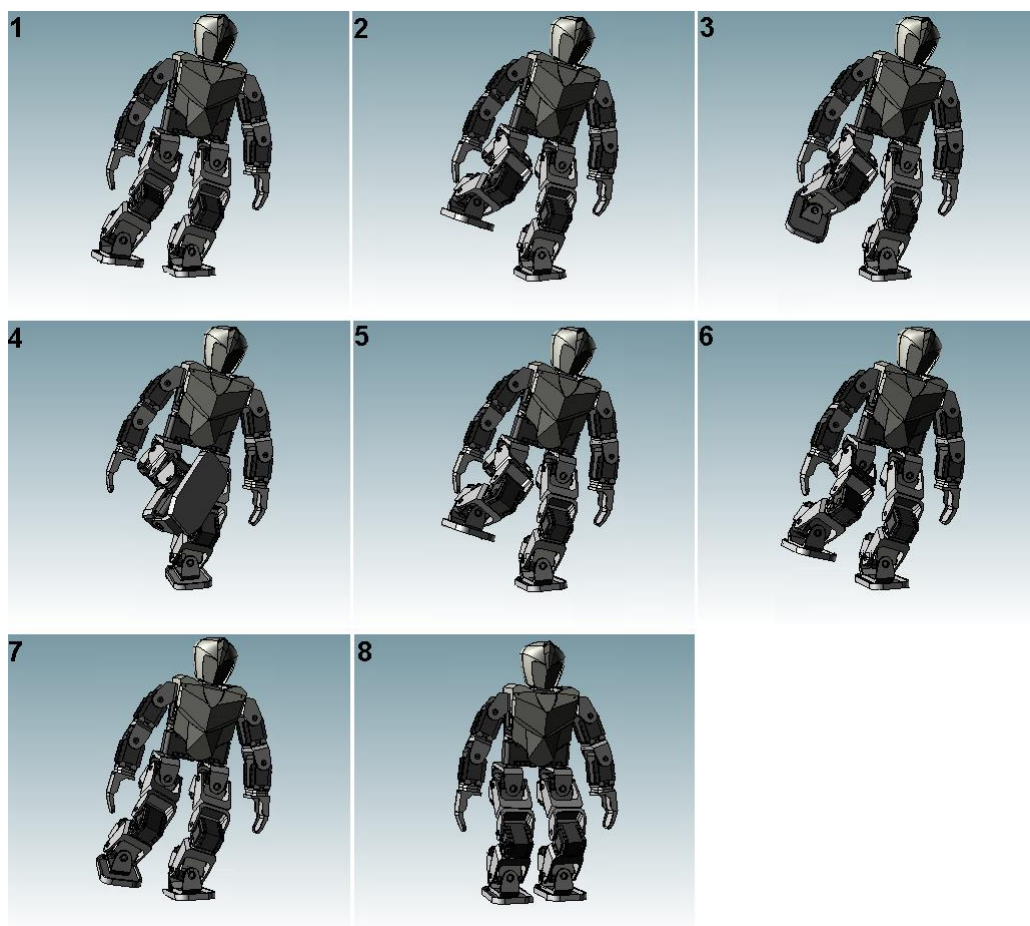


Figura 3.11: Secuencia de poses del robot para patear la pelota con el pie derecho.

3.1.2.7. Movimientos de la cabeza

Se implementaron movimientos de la cabeza que permiten que esta se mueva hacia ambos lados del robot y hacia arriba y abajo, logrando de esta forma desplazar la cámara que se incorporó en la cabeza, permitiendo «barrer» el campo de juego en busca de determinados objetos, tales como la pelota, sin necesidad de mover el cuerpo del robot. Estos movimientos son realizados por los dos motores que se incorporaron a los robots y que sustituyeron al casco de plástico que tenían por cabeza.

3.1.3. Obtención de poses

Un aspecto fundamental a la hora de lograr la realización de movimientos mediante secuenciamiento de poses, es la obtención de los diferentes valores de las ubicaciones que debe alcanzar cada uno de los motores, determinando así el conjunto de pares que componen a cada una de las poses necesarias. La obtención de dichos valores no es trivial, las técnicas para su abordaje fueron detalladas en el documento de estado del arte[Mar12]. Debido a que este paso si bien es claramente necesario para la confección de un equipo de fútbol, no era el objetivo primario de este proyecto, sino un paso más para el cumplimiento de los objetivos, se decidió no partir de cero en la obtención de las poses. Por este motivo se tomó como punto de partida el software RoboPlus[Robc] que trae el kit Bioloid de fábrica, con una gran variedad de movimientos pregrabados, dentro de los cuales se encuentran movimientos de fútbol. Si bien el código que implementa dichos movimientos no está a disposición, sí se encuentran disponibles en la aplicación RoboPlus Motion[Robc] las secuencias de poses utilizadas para realizarlo.

Se indican a estas posturas como punto de partida, debido a que los robots utilizados en el proyecto fueron modificados y no coinciden con los robots cuya construcción se sugiere en los manuales del kit Bioloid[Robb], y para los cuales fue desarrollado el software de fábrica. La principal diferencia entre robots es la incorporación de los dos motores ubicados en la cabeza, lo cual afecta directamente el peso del agente y la distribución del mismo, imposibilitando el uso directo de las poses.

Parte de la modificación de las diferentes poses se realizó por medio de la aplicación RoboPlus Motion que permite ir «moldeando» al robot a una determinada posición y ver al mismo tiempo la posición que adquiere cada uno de los motores. De esta forma se podía visualizar el recorrido que debería hacer cada uno de los motores para lograr un movimiento.

Otro aspecto importante que debía ser considerado a la hora de la modificación de las posturas, era el uso de la información aportada por los

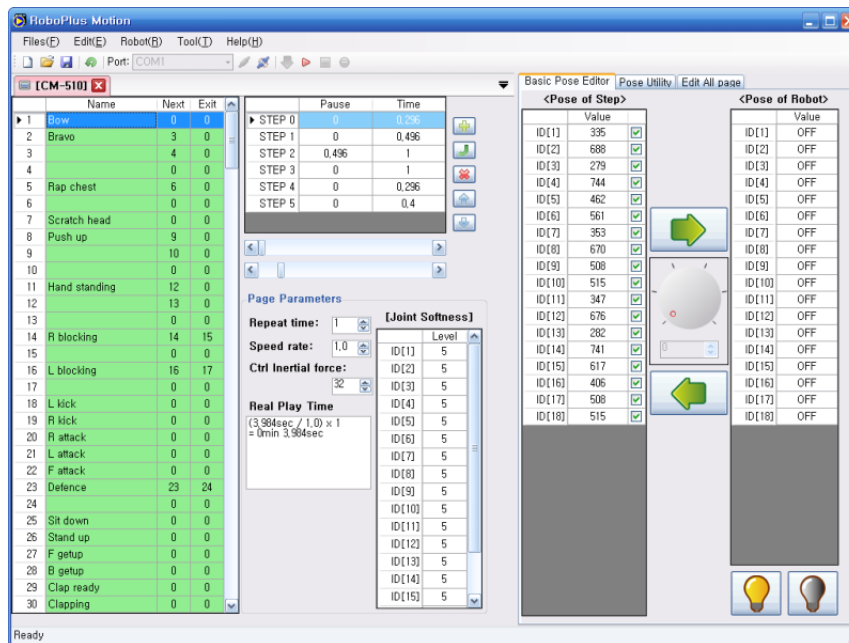


Figura 3.12: Captura de Roboplus Motion, donde se ven en las columnas derechas las posiciones de los 18 motores corporales.

movimientos que se pretendía lograr, como lo es el hecho de la simetría en determinados movimientos.

Teniendo como referencia la figura 3.13, los motores para los cuales se cumple la simetría son aquellos que están en la misma ubicación pero en lados opuestos del robot, por lo que los 18 motores ubicados en el cuerpo del robot son simétricos con algún otro. Siguiendo la numeración de motores de la figura, la simetría se produce entre los motores (1, 2), (3, 4), (5, 6), (7, 8), (9, 10), (11, 12), (13, 14), (15, 16) y (17, 18). Esta simetría se puede aprovechar en movimientos de esa naturaleza, como lo son dar un paso con el pie derecho y dar un paso con el pie izquierdo, atajar el balón hacia la derecha y atajar el balón hacia la izquierda, etc. La simetría implica que el desplazamiento de un motor entre dos poses dentro de la secuencia de un determinado movimiento, tiene que ser de igual magnitud, pero de sentido contrario, al desplazamiento del motor simétrico, entre las poses ubicadas en la misma posición de la secuencia del movimiento simétrico.

Es decir, dada las posturas 1 y 2 que conforman la secuencia de dar un paso con el pie derecho:

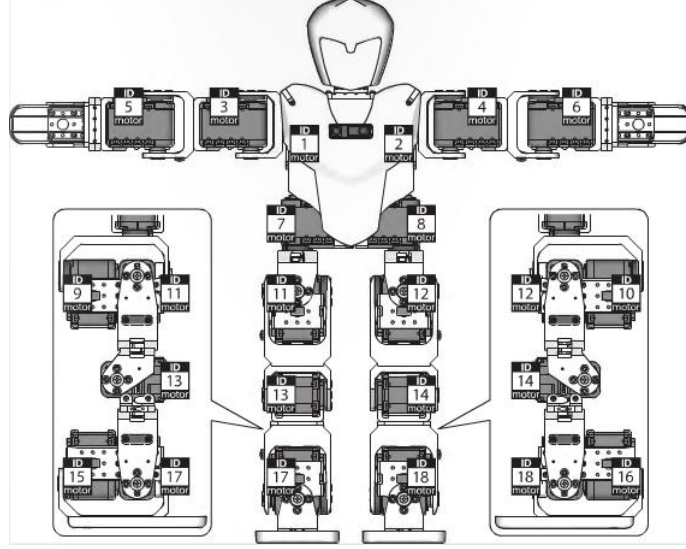


Figura 3.13: Mapa de motores corporales del robot tipo A.

$$p_{1R} = (300, 853, 279, 744, 462, 561, 358, 666, 501, 510, 363, 721, 254, 777, 675, 402, 501, 510)$$

$$p_{2R} = (303, 856, 279, 744, 462, 561, 358, 666, 496, 505, 372, 715, 262, 775, 675, 398, 496, 505)$$

los desplazamientos de cada uno de los motores están dados por el siguiente vector (donde el signo indica el sentido)

$$d_R = (3, 3, 0, 0, 0, 0, 0, 0, -5, -5, 9, -6, 8, -2, 0, -4, -5, -5)$$

Por lo que dichos desplazamientos tienen que mantenerse para los motores simétricos, entre las posturas 1 y 2 de la secuencia de dar un paso con el pie izquierdo. Intercambiando los desplazamientos según los motores simétricos y modificando los sentidos, el vector de desplazamientos para el paso izquierdo resulta en

$$d_L = (-3, -3, 0, 0, 0, 0, 0, 0, 5, 5, 6, -9, 2, -8, 4, 0, 5, 5)$$

Conociendo dicho vector d_L con los desplazamientos de los motores y una de las dos primeras posturas del paso con el pie izquierdo, queda determinada en su totalidad la postura restante. Se pueden ver a continuación las dos primeras posturas de la secuencia del movimiento del paso con el pie izquierdo, y ver que la distancia entre las mismas respeta la simetría del movimiento.

$$p_{1L} = (169, 722, 279, 744, 462, 561, 358, 666, 513, 522, 302, 660, 246, 769, 621, 348, 513, 532)$$

$$p_{2L} = (166, 719, 279, 744, 462, 561, 358, 666, 518, 527, 308, 651, 248, 761, 625, 348, 518, 537)$$

3.1.4. Transición entre poses

Si bien un movimiento queda establecido por la secuencia de poses que lo integran, la ejecución del mismo depende de como se realice la transición o barrido entre las diferentes poses que lo componen, dado que las posturas indican en cada momento hasta qué posición debe ir cada uno de los motores, pero no indican cómo deben ir. Es en este punto que entra en juego la velocidad con la que cada motor del robot va a recorrer las distancias establecidas.

La secuencia de poses define etapas ordenadas que el agente debe cumplir en su movimiento, es por este motivo que es necesario que el robot complete una postura antes de iniciar la siguiente, por lo cual cada uno de los motores deben alcanzar de forma sincronizada la ubicación indicada en cada pose antes de comenzar a moverse a la siguiente ubicación en la secuencia. Los desplazamientos de cada motor generalmente son de diferentes magnitud, por lo que las velocidades de los mismos no pueden ser uniformes; si lo fueran, los motores con menores distancias a recorrer alcanzarían el objetivo antes que los motores con mayores distancias.

La sincronización de los mismos se realiza calculando para cada motor una velocidad proporcional a la distancia que tenga que recorrer. Para ello se asocia a cada pose una velocidad, la cual es fijada al motor que tenga la mayor distancia por recorrer, y a partir de la relación distancia - velocidad que este defina, se realiza el cálculo proporcional para los restantes motores. Para aquellos motores cuya distancia a recorrer sea muy reducida, y por consecuencia el cálculo de velocidad retorne una velocidad despreciable, se les fija una mínima establecida *MIN_VEL*.

El siguiente es el algoritmo utilizado para el cálculo y seteo de las velocidades, así como el seteo de las posiciones a cada uno de los motores. Toma como entrada el vector *POSE*, de tamaño *TAM_VECTORES*, que define a la pose a alcanzar, y la velocidad asociada a dicha pose, denominada *MAX_VEL*.

Otro aspecto importante en la ejecución de poses, es el establecer cuándo una pose es alcanzada y debe continuar con la siguiente en la secuencia. Si bien una postura es claramente alcanzada cuando todos los motores alcancen la posición que la pose establece, no es viable esperar a que todos los motores lleguen exactamente a la posición fijada. Esto es debido a que los motores no están libres, sino que soportan parte del peso del robot, en especial los motores ubicados en la piernas del agente. Para resolver esta situación se tomaron dos medidas: en primer lugar, se estableció un margen de error *MARGEN*, con lo cual si un motor está a una distancia menor del margen fijado, se da por alcanzado la ubicación destino, y en segundo lugar se establece para

Algoritmo 3.1 Algoritmo de cálculo y seteo de velocidades y posiciones de los motores.

```

Maxima_Distancia = 0
// OBTENCIÓN DE LA MAYOR DISTANCIA A RECORRER POR UN MOTOR
Para cada uno de los motores {
    Posicion_Actual = Obtengo_Posicion_Motor(id_Motor)
    Distancia = | Posicion_a_Alcanzar[id_Motor] - Posicion_Actual |
    Si (Distancia > Mayor_Distancia){
        Mayor_Distancia = Distancia
    }
}
// CÁLCULO Y SETEO DE VELOCIDADES A CADA MOTOR
Para cada uno de los motores {
    Posicion_Actual = Obtengo_Posicion_Motor(id_Motor)
    Distancia = | Posicion_a_Alcanzar[id_Motor] - Posicion_Actual |
    Velocidad = ((Distancia * MAX_VEL) / Mayor_Distancia)
    Si (Velocidad < MIN_VEL){
        Velocidad = MIN_VEL
    }
    Fijo_Velocidad_al_Motor(Velocidad)
}
// SETEO POSICIONES
Para cada uno de los motores{
    Fijo_Posicion_al_Motor(Posicion_a_Alcanzar[id_Motor])
}

```

las diferentes poses un tiempo de *timeout*, para el cual la pose se da por alcanzada aunque exista algún motor que no se haya reportado en la posición objetivo.

Algoritmo 3.2 Algoritmo de espera de obtención de pose.

```

Fin = false
Mientras (NOT Fin){
    Para cada uno de los motores {
        Vector_Posicion_Actual[id_Motor] = Obtengo_Posicion_Motor(id_Motor)
    }
    Fin = true
    Para cada uno de los motores {
        Si (NOT (Posicion_a_Alcanzar[id_Motor] - MARGEN <= Vector_Posicion_Actual[id_Motor]) OR
            NOT (Vector_Posicion_Actual[id_Motor] <= Posicion_a_Alcanzar[id_Motor] + MARGEN)){
            Fin = false
        }
    }
    Si (TIMEOUT){
        Fin = true
    }
}

```

Si bien se realizó la espera para que las posiciones indicadas por una pose sean alcanzadas por todos los motores, en determinadas transiciones que implicaban pequeños recorridos por parte de todos los motores, dicha espera provocaba que el robot se moviera de forma intermitente, perdiendo continuidad. Es por este motivo que en esas transiciones de pequeños recorridos, no se aguarda hasta que los motores alcancen el objetivo, sino que

una vez indicadas todas las posiciones, se continúa con la siguiente pose en la secuencia.

3.2. Estrategia

Durante la ejecución de un partido de fútbol, se definió que la estrategia a realizar por los diferentes jugadores dependerá si el equipo se encuentra atacando o si está defendiendo. Los jugadores determinan en que situación se encuentra el equipo en base a la identificación de diferentes condiciones que caracterizan ambos escenarios de juego.

Las acciones a ejecutar por cada agente, no dependerán solamente de qué escenario de juego se está distinguiendo, sino que además dependerán del rol que cumplan los agentes dentro del equipo. Los jugadores de cancha son aquellos robots que son capaces de tomar la posesión de la pelota para poner al equipo en situación de ataque y son los encargados de interceptar el ataque contrario al momento de estar defendiendo. Dentro de los dos jugadores que cumplen el rol de jugador de cancha, se encuentra el delantero y el defensa. El delantero es aquel que se encuentra ubicado en la mitad de la cancha correspondiente al equipo contrario y es el encargado de marcar los goles del equipo, mientras que el defensa es aquel que se encuentra ubicado en la mitad de la cancha correspondiente al equipo propio. Si bien el nombre de los roles hace referencia a la posición en la cancha donde se ubican, el defensa no sólo se limita a defender sino que realiza acciones en ataque, y de igual modo el delantero no sólo se limita a atacar sino que además realiza acciones defensivas.

El golero es aquel robot que se encarga de evitar las anotaciones del equipo contrario y es el único agente del equipo que es capaz de tocar la pelota con las manos. El golero se desplaza solamente en las inmediaciones del arco que defiende.

3.2.1. Ataque

Los jugadores identifican que están atacando si se producen algunas de las siguientes situaciones.

- El jugador tiene la pelota en su posesión, es decir que la distancia a la misma es menor que una constante fijada.
- El jugador no tiene la pelota, pero determina que es el jugador más próximo a la misma.

3.2.1.1. Acciones en ataque

Una vez que el agente identifica que el equipo se encuentra en ataque, determina qué acción ejecutar con el fin de que el ataque finalice con la anotación de un gol en el arco del equipo contrario. Se describe a continuación el conjunto de acciones que los agentes tienen disponibles a realizar en ataque, discriminadas por el rol que los agentes cumplan en el equipo. Se aclara que las acciones que se encuentran listadas como acciones de los jugadores de cancha, son aquellas que son realizables tanto por el delantero como por el defensa.

Jugador de cancha:

- Avance en cancha: con el balón localizado en cancha, y habiendo determinado que es el jugador más próximo, *camina* hacia la pelota para lograr la posesión de la misma.
- Adelantamiento del balón: con el balón en posesión, el jugador *patea la pelota* para adelantarla en la cancha, manteniendo de referencia el arco contrario para evitar que la pelota salga del campo de juego.
- Espera por la pelota: cuando el equipo está atacando y el robot determina que es un compañero de equipo el agente que se encuentra más cerca de la pelota o que está en posesión de la misma, el robot aguarda a que pueda tomar posesión de la pelota luego de la acción de su compañero.

Delantero:

- Tiro al arco: Con el balón en posesión, el arco contrario ubicado y a una distancia próxima, el jugador *patea la pelota* hacia el arco.

Se aclara que el golero mantiene una actitud pasiva mientras se esté desarrollando el ataque del equipo y no participa del mismo.

3.2.2. Defensa

Los jugadores identifican que están defendiendo de forma análoga a como identifican que están en ataque. Las situaciones que se consideran son las siguientes:

- Un jugador del equipo contrario tiene la pelota en su posesión.

- Un jugador del equipo contrario no tiene la pelota, pero es el jugador más próximo a la misma.
- Un jugador del equipo contrario efectúa un tiro al arco.

3.2.2.1. Acciones en defensa

Las acciones que el agente debe tener disponible al momento de estar defendiendo, deben permitirle evitar la anotación de un gol por parte del equipo contrario y poder quitarle el balón a un jugador oponente cuando tenga la posesión de la pelota. A continuación se listan las acciones disponibles en defensa.

Jugador de cancha

- Interceptar: el jugador identifica que un jugador oponente tiene la pelota o es el jugador más próximo al balón. Frente a esta situación, el agente se posiciona en el camino del contrario hacia el arco para evitar un posible tiro o para interceptar el posible avance del contrario.
- Avance para producir quite: Luego de interceptar a un contrario, el agente avanza hacia el mismo, con el fin de provocar la pérdida del balón y poder revertir la situación de defensa para estar en ataque.

Golero

- Posicionamiento del golero: con la pelota identificada en las proximidades del arco, el golero se posiciona en la ubicación sobre la línea del arco frente a la pelota para aumentar las posibilidades de atajar el balón en caso de producirse un tiro al arco por parte del contrario.
- Atajar: Una vez que el golero identifica que se ha producido un tiro al arco por parte de un jugador rival, el mismo intenta *atajar* el disparo, prediciendo la trayectoria del balón. Esta acción es la que identifica al golero del equipo.

3.2.3. Neutral

La única condición para poder determinar en que situación está el juego, es decir, para determinar si el equipo se encuentra atacando o defendiendo, es la identificación de la ubicación de la pelota dentro de la cancha. Es por este motivo que mientras los jugadores no localicen la pelota, estarán en una situación neutral. Cada vez que el robot detecte que sufrió una caída va a

pasar a estar en la situación neutral, debiendo ubicar nuevamente la pelota luego de levantarse.

3.2.3.1. Acciones en situación neutral

Se listan a continuación las acciones realizadas por los agentes mientras no se haya determinado la ubicación de la pelota.

Jugador de cancha

- Posicionamiento en cancha: Los jugadores se ubican en el centro del área donde desempeñan su actividad. El delantero se ubica en la media cancha donde se encuentra el arco rival y el defensa en la media cancha donde se encuentra el arco propio.
- Búsqueda del balón: Los jugadores luego de ubicarse en su posición dentro de la cancha, barren visualmente el campo de juego con la finalidad de encontrar la pelota.

Golero

- Posicionamiento del golero: El golero se posiciona en el centro del arco paralelo a la línea de fondo a una distancia que le permita ante una caída continuar dentro del campo. Se mantiene buscando la pelota, barriendo el campo con la cámara hasta el momento que divise al balón.

3.3. Estado del mundo

Uno aspecto importante al momento de actuar en un determinado entorno, es mantener información que lo describa. En el caso del fútbol de robots esta información se hace vital al momento de ubicarse dentro del campo de juego. Si bien los robots disponen de una cámara que les permite obtener información, no siempre se dan las condiciones para que por medio de esta puedan obtener toda la información que necesitan de forma rápida, dado que los objetos pueden estar por fuera de su campo visual. Esta limitación se ve reflejada claramente en dos situaciones que componen la estrategia de juego:

1. Al momento de realizar un tiro al arco, si el robot se encuentra a una distancia muy corta del arco contrario, la cámara no es capaz de identificar el objeto arco, por lo que no se tiene una noción clara de hacia donde patear.

2. Al momento de interceptar un ataque del equipo contrario, el agente se debe ubicar entre el arco propio y el jugador con la posesión de la pelota. En esta situación es bastante probable que el defensa se encuentre de espaldas al arco propio y el mismo no se encuentre en su campo visual, por lo que no tiene la información necesaria para poder ubicarse en la posición requerida.

Estas dos situaciones del juego fueron la principal motivación para que el agente mantenga y actualice información del estado del mundo.

La información que el agente mantiene en este estado del mundo, es información de los objetos estáticos que conforman el campo de juego y que son visibles por los robots. Estos son:

- El arco contrario.
- El arco propio.
- Los dos postes laterales que se encuentran por fuera de la cancha.

La información que se almacena de cada objeto, es la distancia del robot hacia el objeto y el ángulo desde el frente del robot hacia el centro del objeto, siendo el ángulo 0 cuando el objeto se encuentra enfrente, negativo cuando está hacia la derecha y positivo cuando está hacia la izquierda. Por este motivo el ángulo de los objetos siempre va a estar en el rango de $[180, -180)$.

Si bien cabía la posibilidad de mantener en el estado del mundo solamente las coordenadas del agente dentro de la cancha y el ángulo que indicara hacia que lado se encontraba el frente del robot, dado que con estos datos se podría calcular la distancia hacia los diferentes objetos fijos en la cancha, se optó por almacenar la distancia y el ángulo de cada uno de los objetos, dado que era directo almacenarlos a partir de los datos retornados por el módulo de visión, y de la misma forma, era directo utilizarlos, pues el agente sabe cuanto tiene que girar o cuanto avanzar para dirigirse hacia uno de los mismos. La desventaja que tiene almacenar los datos de cada uno de los objetos es la mayor cantidad de cálculos para actualizar el estado del mundo que se deben realizar cuando no se tiene información del módulo de visión, dado que se debe calcular los nuevos datos para cada uno de los objetos, mientras que si se guardara la información propia de la localización del agente, solamente ésta debería ser actualizada. Esta mayor cantidad de cálculos a realizar trae aparejado un mayor tiempo de reacción por parte del robot al momento de actualizar el estado del mundo, debido al mayor tiempo que le lleva dicha actualización.

3.3.1. Actualización de datos

Los datos del estado del mundo son actualizados en dos momentos:

- Cuando se dispone de nueva información proveniente de una captura de la cámara, la misma es almacenada en el estado del mundo de forma directa, debido a que la visión permite obtener el ángulo y la distancia de los objetos que se identifican en la captura.
- Cuando se realiza un movimiento, toda la información del estado del mundo es actualizada en base al movimiento que se esté realizando. Las nuevas distancias y ángulos son calculadas en base a la información previa que se encontraba en el estado del mundo y la información del movimiento realizado.

Dependiendo del movimiento que se realice es el método de actualización de datos realizado. Existen tres tipos de actualizaciones diferentes: la primera es llevada a cabo cuando el robot realiza el movimiento de rotación donde solamente se actualizan los ángulos de los objetos, la segunda está vinculada al agente caminando, debiendo actualizar las distancias y ángulos sabiendo que ningún objeto va a cambiar de lado en que se encuentra en referencia al robot y por último está la actualización que se realiza cuando el robot se mueve lateralmente. A continuación se detallan los tres métodos de actualización de datos implementados.

3.3.1.1. Rotaciones

Cuando se realiza una rotación, ya sea hacia la derecha o hacia la izquierda, se asume que la distancia hacia los objetos que conforman el estado del mundo no va a variar, lo que si va a cambiar es el ángulo a los mismos. Se definieron dos constantes *ANG_ROT_DER* y *ANG_ROT_IZQ* que son sumadas a los ángulos de los objetos dependiendo el sentido de la rotación y que corresponden al ángulo que el robot logra desplazar al momento de rotar. Se definieron dos constantes en vez de una debido a que el ángulo rotado no es igual en ambos movimientos.

La constante *ANG_ROT_IZQ* correspondiente a la rotación izquierda, tiene un valor negativo, de forma que al ser sumada a los ángulos de los objetos del estado del mundo los mismos disminuyan, indicando así que los objetos van a tender a quedar hacia la derecha del robot.

3.3.1.2. Caminata

Al momento de realizar la caminata, los datos del estado del mundo no son actualizados hasta finalizar, evitando de esta forma actualizar los datos

en cada paso que el robot realiza. Esta única actualización evita que se estén realizando cálculos con pequeñas distancias recorridas equivalentes a cada paso, disminuyendo de esa forma el error en los cálculos.

Se definieron para este propósito las constantes *DISTANCIA_PASO_INI* y *DISTANCIA_PASO* equivalentes a las distancias logradas al dar el paso inicial y al dar un paso completo, respectivamente. Mientras se realiza la caminata se va contabilizando la cantidad de pasos dados y al finalizar se calcula la distancia total recorrida mediante la siguiente ecuación:

$$dist_{recorrida} = DISTANCIA_PASO_INI + (Cant_Pasos * DISTANCIA_PASO)$$

Una vez que se cuenta con la distancia total recorrida, se procede a actualizar para cada objeto en el estado del mundo, su nueva distancia y su nuevo ángulo. Los nuevos valores son calculados mediante trigonometría. Para ello se considera el triángulo formado por la posición del robot anterior al movimiento, la posición del robot posterior al movimiento y el objeto para el cual se actualizan los valores de distancia y ángulo (ver figura 3.14). Sean $dist_{ant}$ y ang_{ant} la distancia y el ángulo anteriores del objeto que figuran en el estado del mundo. La nueva distancia y el nuevo ángulo se calculan realizando los siguientes cálculos aplicando el teorema del coseno.

$$dist_{nueva} = \sqrt{dist_{ant}^2 + dist_{recorrida}^2 - 2 * dist_{ant} * dist_{recorrida} * \cos(ang_{ant})}$$

$$ang_{nuevo} = 180 - \alpha$$

Siendo α (alpha) el ángulo interno del triángulo considerado en el vértice de la posición del robot luego del movimiento. El valor de dicho ángulo interno es

$$\alpha = \arccos\left(\frac{dist_{ant}^2 + dist_{recorrida}^2 - dist_{nueva}^2}{2 * dist_{nueva} * dist_{recorrida}}\right)$$

El ángulo ang_{nuevo} corresponde al ángulo suplementario del ángulo α . El sentido del nuevo ángulo del objeto actualizado siempre va a ser igual al sentido del ángulo al que se encontraba el objeto inicialmente, esto debido a que ningún objeto del estado del mundo va a cambiar de lado mientras el robot esté caminando. Se puede apreciar en la imagen 3.14 los diferentes ángulos y distancias utilizadas.

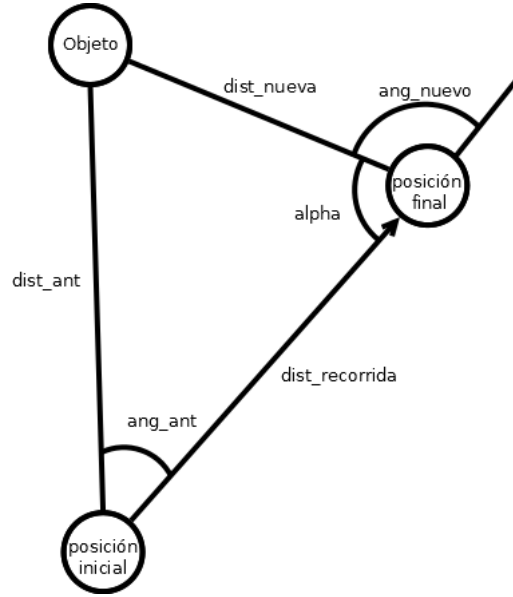


Figura 3.14: Actualización de los datos en el estado del mundo al realizar una caminata.

3.3.1.3. Pasos laterales

Al momento de dar un paso lateral, los nuevos ángulos y distancias de los objetos en el estado del mundo, son actualizadas en base a los ángulos y distancias existentes en él previo a la actualización y la distancia recorrida en el paso lateral. Para este fin, se definieron las constantes `DIST_LAT_R` y `DIST_LAT_L` que indican la distancia recorrida en el paso lateral derecho e izquierdo respectivamente. Al igual que la actualización realizada al realizar la caminata, los nuevos datos son obtenidos en base a la triangulación entre la posición anterior al movimiento del robot, la nueva posición luego del movimiento y el objeto del estado del mundo. De esta forma, mediante la aplicación del teorema del coseno se pueden calcular los nuevos valores. La principal diferencia con el cálculo que se realiza en la actualización del estado del mundo al caminar, es que cuando se camina la dirección del movimiento coincide con el lugar desde donde están medidos los ángulos (el frente del robot), permitiendo así usar los ángulos del estado del mundo de forma directa para realizar la triangulación. En el movimiento lateral, los ángulos que existen en el estado del mundo no pueden ser usados directamente para triangular debido a que el movimiento no es hacia el frente del robot, sino que justamente es lateral.

El ángulo α utilizado para realizar la triangulación depende del rango en que se encuentre el ángulo β , siendo este, el ángulo en que se encuentra el

objeto previo al movimiento. Si el desplazamiento lateral es hacia la derecha del robot (ver figura 3.15) entonces el ángulo para triangular se calcula de la siguiente forma :

- Si $\beta \in (90, 180]$ entonces $\alpha = 180 - (\beta - 90)$
- Si $\beta \in (0, 90]$ entonces $\alpha = \beta + 90$
- Si $\beta \in (-90, 0]$ entonces $\alpha = 90 - |\beta|$
- Si $\beta \in (-180, -90]$ entonces $\alpha = |\beta| - 90$

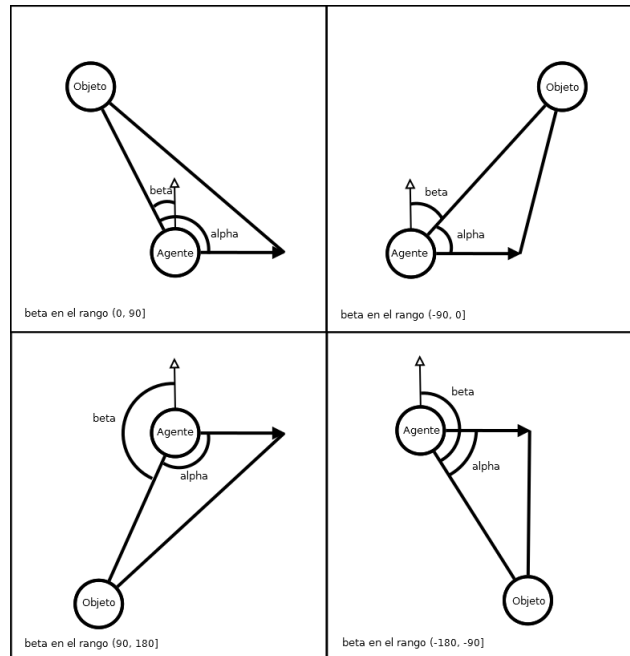


Figura 3.15: Obtención del ángulo alpha para realizar la triangulación durante el desplazamiento lateral hacia la derecha (arriba izquierda: $\beta \in (0, 90]$ - arriba derecha: $\beta \in (-90, 0]$ - abajo izquierda: $\beta \in (90, 180]$ - abajo derecha: $\beta \in (-180, -90]$).

Si el desplazamiento es a la izquierda, el razonamiento es el mismo pero de forma simétrica, determinando que el ángulo para triangular sea calculado de la siguiente forma:

- Si $\beta \in (90, 180]$ entonces $\alpha = \beta - 90$
- Si $\beta \in (0, 90]$ entonces $\alpha = 90 - \beta$

- Si $\beta \in (-90, 0]$ entonces $\alpha = |\beta| + 90$
- Si $\beta \in (-180, -90]$ entonces $\alpha = 180 - (|\beta| - 90)$

Una vez calculado α , con la distancia al objeto existente en el estado del mundo, y con la distancia desplazada por el agente, se puede proceder a calcular el ángulo γ (gama: ángulo interno del triángulo en la posición del agente luego del movimiento) y la nueva distancia hacia el objeto en base al teorema del coseno.

$$dist_{nueva} = \sqrt{dist_{recorrida}^2 + dist_{ant}^2 - 2 * dist_{recorrida} * dist_{ant} * \cos(\alpha)}$$

$$\gamma = \arccos\left(\frac{dist_{recorrida}^2 + dist_{nueva}^2 - dist_{ant}^2}{2 * dist_{recorrida} * dist_{nueva}}\right)$$

Una vez calculado γ se calcula el nuevo ángulo δ (delta) hacia el objeto, teniendo en cuenta nuevamente el rango en el cual se encuentra β .

En el caso del desplazamiento hacia la derecha (ver figura 3.16) δ es calculado de la siguiente forma:

- si $\beta \in (90, 180]$ entonces $\delta = \gamma + 90$
- si $\beta \in (0, 90]$ entonces $\delta = 90 - \gamma$
- si $\beta \in (-90, 0]$ entonces $\delta = \gamma - 90$
- si $\beta \in (-180, -90]$ entonces $\delta = (180 - \gamma) + 90$

De forma análoga, para el caso del desplazamiento hacia la izquierda, δ se obtiene de la siguiente manera.

- si $\beta \in (90, 180]$ entonces $\delta = (180 - \gamma) + 90$
- si $\beta \in (0, 90]$ entonces $\delta = \gamma - 90$
- si $\beta \in (-90, 0]$ entonces $\delta = 90 - \gamma$
- si $\beta \in (-180, -90]$ entonces $\delta = \gamma + 90$

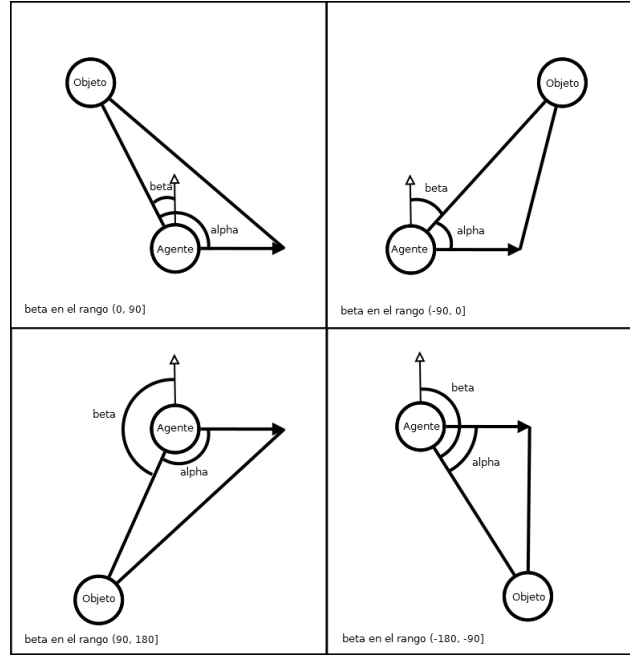


Figura 3.16: Obtención del nuevo ángulo (medido desde el frente del robot) hacia el objeto, luego del desplazamiento lateral hacia la derecha (arriba izquierda: $\beta \in (0, 90]$ - arriba derecha: $\beta \in (-90, 0]$ - abajo izquierda: $\beta \in (90, 180]$ - abajo derecha: $\beta \in (-180, -90]$).

3.3.2. Pérdida de precisión

Si bien el estado del mundo es actualizado cuando no se tiene información proveniente del módulo de visión, cuando más tiempo pase sin datos provenientes de dicho módulo y los datos del estado del mundo se sigan actualizando en base a su estado previo y a los movimientos que el agente vaya realizando, la información almacenada con cada actualización va perdiendo precisión, generando la degradación de los datos y que la información que dispone el robot del estado del mundo no corresponda con la realidad en la que se encuentra. Para representar como los datos de cada objeto van «envejeciendo», se les asigna un contador que indica cuantas actualizaciones se llevan efectuadas, sin información desde el módulo de visión. Una vez que la cantidad de actualizaciones supere una cantidad determinada, la información se considera envejecida y se resetean los datos asociados al objeto considerándolos desconocidos. La cantidad máxima de actualizaciones permitidas, esta indicada por la constante `CANT_MAX_ACTUALIZA`.

La segunda situación donde los datos asociados al estado del mundo deben de ser descartados, es cuando ocurre una caída por parte del agente. La

diferencia en esta situación radica en que todos los datos almacenados en el estado del mundo son descartados y no solamente los de algunos objetos en particular. Este descarte de datos del estado del mundo se realiza debido a que se desconoce la posición y la dirección en que se va encontrar el robot dentro del campo de juego luego de sufrir una caída, por lo que no se puede realizar actualización alguna sobre los datos del estado del mundo ni asumir que los mismos van a permanecer incambiados luego de la caída.

Capítulo 4

Aspectos de implementación

4.1. Arquitectura

La estructura principal de la arquitectura del sistema fue definida en base a diferentes niveles de abstracción, resultando finalmente en una arquitectura de tres niveles que se detallan a continuación (ver figura 4.1).

- Nivel I - Interfaz de Motores: El nivel inferior es el que implementa la interfaz con los motores y permite realizar la comunicación con los mismos. Se encarga de establecer en cada una de las poses, la velocidad y posición objetivo de cada uno de los motores, así como de controlar que cada uno alcance la posición objetivo que se le indicó, determinando de esta forma la finalización de la pose que le fue indicada por el nivel II.
- Nivel II - Movimientos: El segundo nivel es el que implementa los movimientos básicos del agente, los cuales fueron presentados en la sección 3.1.2.
- Nivel III - Comportamiento: El nivel superior es el que implementa el comportamiento del robot y realiza la toma de decisiones en base a la información que dispone. Este nivel utiliza los submovimientos del nivel dos para lograr el movimiento del agente.

Además de estos tres niveles, existen dos módulos auxiliares implementados, que son los encargados de brindar la información para la toma de decisiones realizada por el nivel III de comportamientos. Estos son:

- Interfaz de Visión: Este módulo permite la comunicación con el módulo externo de visión realizado por el proyecto de grado VisRob II[Gon].

Esta interfaz permite acceder a la información que el robot percibe mediante la cámara y que es procesada por el módulo externo.

- Estado del Mundo: Este módulo es el encargado de mantener un modelo del mundo, brindando métodos para su actualización como para su consulta. Utiliza las primitivas que brinda la interfaz de visión al momento de actualizar el entorno con nueva información visual.

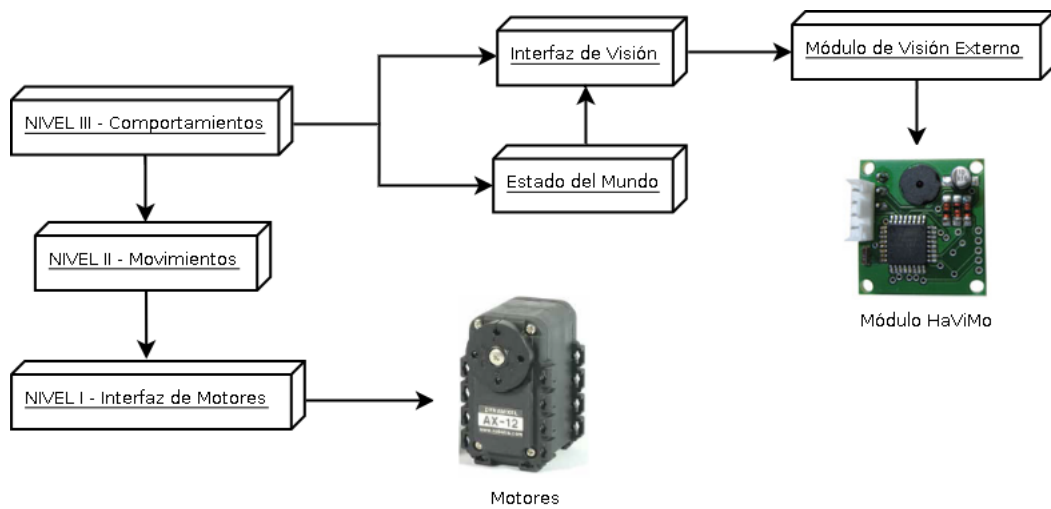


Figura 4.1: Diagrama de los diferentes componentes de la arquitectura.

4.2. Toma de decisiones y cálculos realizados

Los diferentes agentes que conforman el equipo deben identificar cuál es la situación del juego para poder decidir la acción a realizar. Se detalla a continuación los métodos utilizados para que los robots puedan determinar la decisión más acorde a la situación que se están enfrentando y los cálculos que realizan durante el desarrollo del juego.

4.2.1. Posesión de la pelota

Una vez que el jugador de cancha identifica la pelota, debe determinar en primera instancia si alguien tiene la posesión de la misma, y en caso de que así sea, determinar quien la tiene bajo su control. Debido a que la posesión de la pelota se determina en base a la distancia entre el agente y la misma,

el robot debe disponer de la distancia a la cual se encuentra la pelota y de la distancia entre otro agente y la pelota.

La distancia a la cual se encuentra la pelota del agente es provista por el módulo de visión, sin embargo, la aparición de un jugador que pueda disputar la posesión de la pelota implica que el agente deba calcular la distancia de ese jugador hacia la pelota, para así poder determinar si alguno de los dos tiene la pelota bajo control o en caso de que ninguno tenga su posesión, poder determinar cuál es el más próximo al balón.

Para calcular la distancia entre otro agente y la pelota (ver figura 4.2), el robot debe aprovechar los datos que conoce, como son el ángulo y la distancia a la que se encuentran. La idea utilizada es triangular y calcular la distancia entre ellos. Debido a que el ángulo retornado por el módulo de visión siempre parte desde el medio de la cámara, siendo positivos a la izquierda y negativos a la derecha, el robot debe calcular el ángulo absoluto entre el robot detectado y la pelota. Esto se hace restando los ángulos dependiendo del sentido de los mismos. Si α (alpha) es el ángulo a la pelota y β (beta) el ángulo al restante agente involucrado, el ángulo absoluto entre ellos γ (gama) se calcula de la siguiente manera

$$\gamma = \alpha - \beta \text{ si } \alpha \geq \beta$$

$$\gamma = \beta - \alpha \text{ si } \beta \geq \alpha$$

Una vez que el robot dispone del ángulo entre la pelota y el otro jugador, y conociendo la distancia hacia la pelota y hacia el jugador mediante el módulo de visión, el robot dispone de todos los elementos para realizar la triangulación.

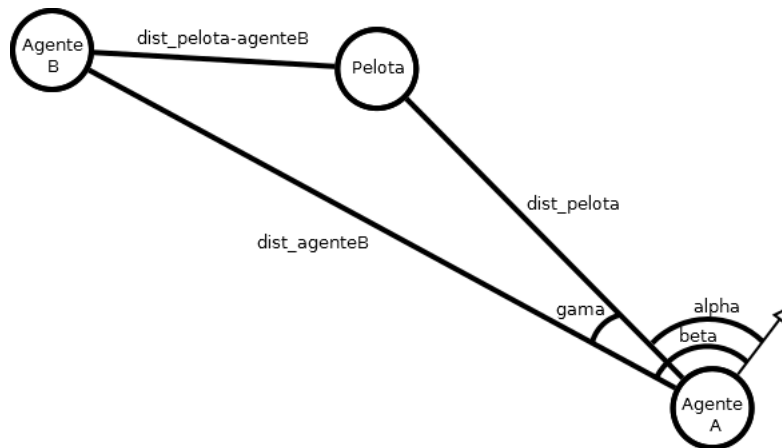


Figura 4.2: Cálculo de la distancia entre el agente B y la pelota.

Mediante el teorema del coseno, la distancia entre el agente B y la pelota se calcula

$$dist_{pelota-agenteB} = \sqrt{dist_{pelota}^2 + dist_{agenteB}^2 - 2 * dist_{pelota} * dist_{agenteB} * \cos(\gamma)}$$

Una vez que se obtiene la distancia entre la pelota y el otro robot que aparece en la captura, y disponiendo de la distancia a la que se encuentra la pelota, el robot es capaz de determinar quien se encuentra más cerca de la pelota comparando ambas distancias. Al mismo tiempo, se dispone de la constante *DISTANCIA_POSESION* con la cual se determina la posesión de la pelota. Si la distancia del robot que se encuentra más cerca a la pelota es menor a *DISTANCIA_POSESION*, se determina que dicho robot tiene posesión del balón.

4.2.2. Interceptar ataque

Una vez que el robot determina que un robot del equipo contrario tiene la posesión del balón o se encuentra a una menor distancia de la pelota, el robot determina que tiene que interceptar el ataque, y con tal fin, debe interponerse en el camino del jugador contrario hacia el arco propio para evitar su posible avance. El problema que se detalla en esta sección, es el cálculo de la posición que el robot debe alcanzar para lograr interponerse en el camino que se estima realizará el adversario.

Bajo un paradigma deliberativo, una vez que el agente determina que debe interceptar el ataque del equipo contrario, se desplaza por la mediana del triángulo formado entre la pelota, el agente y su arco propio (ver figura 4.3) para ubicarse entre la pelota y el arco propio. La distancia que va a recorrer el robot hacia el punto de intercepción es la longitud de dicha mediana *M*.

Los datos con los que el robot cuenta al momento de realizar los cálculos, son el ángulo y la distancia hacia la pelota (mediante la información que le provee el módulo de visión), y el ángulo y la distancia hacia el arco propio, ambos provenientes del entorno que el robot mantiene. Antes de poder calcular la longitud de la mediana, el robot precisa calcular la distancia de la pelota hacia el arco propio, y para ello se aplica el teorema del coseno.

$$dist_{arco-pelota} = \sqrt{dist_{pelota}^2 + dist_{arco}^2 - 2 * dist_{pelota} * dist_{arco} * \cos(\alpha)}$$

Siendo α (alpha) el ángulo entre la pelota y el arco, calculado con el ángulo del agente a la pelota y el ángulo del agente al arco, siguiendo la misma lógica que en 4.2.1.

Una vez que se dispone de la distancia entre el arco y la pelota, se puede calcular el valor de la mediana mediante la siguiente ecuación:

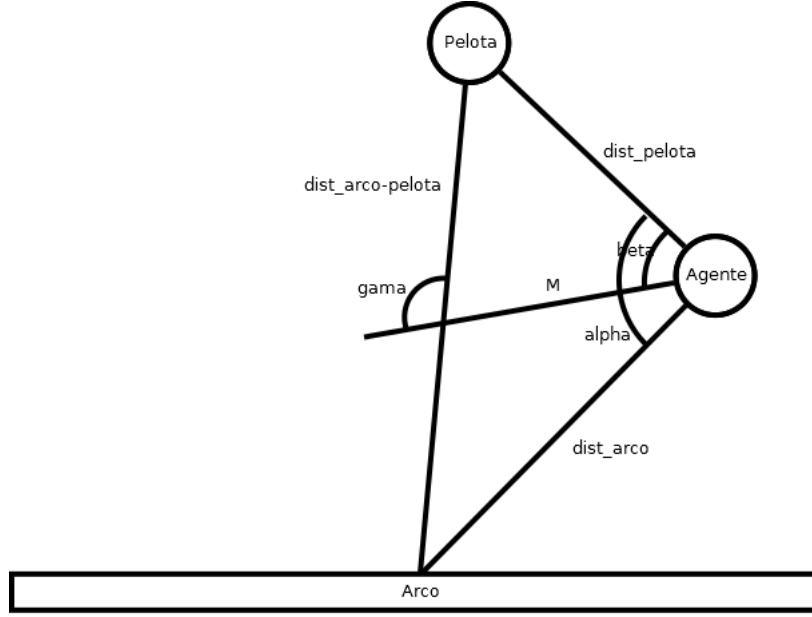


Figura 4.3: Cálculos al momento de interceptar un ataque.

$$M = (1/2) * \sqrt{2(dist_{arco}^2 + dist_{pelota}^2) - dist_{arco-pelota}^2}$$

Una vez calculada la distancia a recorrer, se debe determinar en que sentido recorrer dicha distancia. Sea β (beta) el ángulo a girar antes de empezar a caminar. El mismo se obtiene aplicando nuevamente el teorema del coseno de la siguiente forma.

$$\beta = \arccoseno\left(\frac{M^2 + dist_{pelota}^2 - (dist_{arco-pelota}/2)^2}{2 * M * dist_{pelota}}\right)$$

Si bien la función de arcocoseno no es biyectiva, si lo es si se restringe el dominio de la función coseno a $[0, \pi]$, dicha restricción se cumple en la función arcocoseno utilizada en la programación. El sentido del giro va a depender del sentido al cual esté ubicado el arco propio.

En este punto el robot es capaz de dirigirse al punto de intercepción girando β grados y avanzando M centímetros.

Una vez alcanzado el punto de intercepción, lo que resta por realizar al agente, es enfrentar a la pelota y al jugador contrario girando un ángulo γ (gama). Dicho giro va a ser en sentido opuesto al que se realizó al momento de recorrer la mediana. Aplicando nuevamente el teorema del coseno se puede calcular la magnitud de dicho ángulo.

$$\gamma = 180 - \arccoseno\left(\frac{(dist_{arco-pelota}/2)^2 + M^2 - dist_{pelota}^2}{2 * (dist_{arco-pelota}/2) * M}\right)$$

4.3. Caídas

El entorno donde deben actuar los agentes y la dinámica que presenta un partido de fútbol robótico son terreno fértil para las caídas. Las deformaciones del piso o los choques con los contrarios son ambas situaciones frecuentes que generan caídas a los robots. Abordar el problema de las caídas y obtener estrategias de recuperación son esenciales para asegurar la continuidad de los jugadores en la cancha. En esta sección se presenta el resultado del análisis y la estrategia con la que se identifica una caída, así como también las herramientas utilizadas. Aunque inicialmente se consideró un algoritmo más sencillo, la progresión de las pruebas hizo que finalmente el algoritmo para analizar las caídas tuviera tres aspectos fundamentales: la detección de la caída, la predicción del sentido en que se produce y finalmente la confirmación o corrección. Los dos primeros se determinan durante la caída y la última se determina una vez que el robot está en el suelo.

4.3.1. Detección

Primero que nada es necesario contar con una herramienta que permita detectar las caídas. Para ello se contó con un sensor de giro (Figura 4.4) incluido en el kit Premium de Bioloid[Robb] que permite tomar valores de la velocidad angular en los ejes x e y . Este sensor se encuentra ubicado en la base del pecho del robot lo que permite analizar el aumento o disminución de la velocidad sobre los ejes lo que podría determinar si se está produciendo una caída. Para este caso el eje x representa la dirección de una caída frontal o de espaldas mientras que el eje y representa la dirección lateral izquierda-derecha.

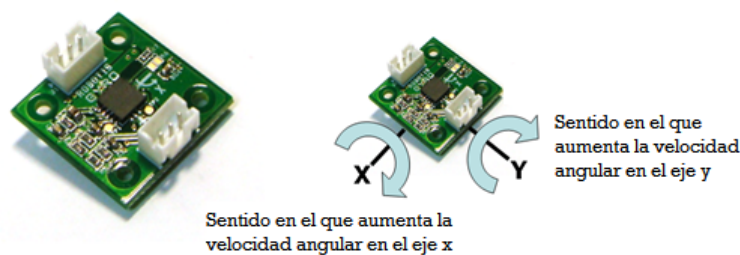


Figura 4.4: Sensor de giro

El sensor proporciona información sobre la velocidad angular en los ejes, pero no cualquier variación de las mismas representa una caída. Existen variaciones en la velocidad sensada inherentes a los movimientos del robot,

dados por la velocidad y la posición de los motores. Éstas deben ser tenidas en cuenta para no confundirlas con una caída. En vista de esto fue elaborado un criterio para distinguir según el movimiento que se está ejecutando, cuales valores son normales y cuales indican una caída.

Para ello se realizó un análisis de las mediciones del sensor, en el que fue definido como punto de partida el valor de equilibrio que se corresponde con la lectura del sensor cuando el robot se encuentra erguido. Este valor es tomado al inicio de cada ejecución ya que puede presentar variaciones en diferentes ejecuciones y pueden existir diferencias entre un robot y otro (diferentes sensores). Además se definen márgenes de tolerancia que incluyen los valores que son considerados normales, es decir que son ocasionados por los movimientos del robot y que no representan una potencial caída. Se considera caída cuando una cierta cantidad de valores consecutivos se encuentran por fuera de los márgenes de tolerancia establecidos. El hecho de que exista un valor fuera de los márgenes puede deberse a una mala medición del sensor o a un tropezón del jugador por esto en la práctica se exigen tres valores fuera de los márgenes para confirmar la tendencia de los mismos y reconocer la caída.

A su vez se deben tener en cuenta algunos movimientos más bruscos que pueden exceder los márgenes de tolerancia por lo que es necesario manejar diferentes niveles de sensibilidad que reajustarán los márgenes. Para comprender con mayor claridad este análisis se puede observar el algoritmo 4.1 de detección de caídas. Para facilitar su lectura resumimos las nociones que acabamos de introducir:

- EQUILIBRIO: como ya se mencionó, es el valor del sensor cuando el robot está en la pose erguida.
- MARGEN: se utiliza para determinar el rango de los valores del sensor que no se consideran de caída. MARGEN representa la diferencia aceptada respecto al equilibrio de los posibles valores devueltos. Es decir, si la diferencia entre un valor devuelto por el sensor y EQUILIBRIO es menor a MARGEN, no se considera caída. El MARGEN se define para cada dirección y sentido(no tiene por qué ser simétrico).
- CANT_VALORES: es la cantidad de veces que deben medirse valores fuera del rango para considerarse caída.
- SENSIBILIDAD: se definieron diferentes niveles de sensibilidad para evaluar las caídas. En algunos movimientos se debe tener menor sensibilidad ya que los movimientos son más bruscos y provocan variaciones mayores en los valores. Sin embargo los valores que se alcanzan en esos

movimientos en otros indicarían una caída, por esto es necesario realizar esta distinción.

- TOLERANCIA: la tolerancia amplía el MARGEN para los casos en que la sensibilidad debe ser menor. Se define para cada dirección y sentido.

Algoritmo 4.1 Algoritmo de detección de caídas.

```
int lectura = leoValorGyro(); // lectura del valor del giroscopio
int caidaFrontal;
int caidaAtras;
// ESTABLEZCO MARGENES SEGÚN SENSIBILIDAD
switch (sensibilidad){
case NORMAL:
    caidaFrontal = EQUILIBRIO - MARGEN_INFERIOR;
    caidaAtras = EQUILIBRIO + MARGEN_SUPERIOR;
    break;
case ESPECIAL:
    caidaFrontal = EQUILIBRIO - MARGEN_INFERIOR - TOLERANCIA_INFERIOR;
    caidaAtras = EQUILIBRIO + MARGEN_SUPERIOR + TOLERANCIA_SUPERIOR;
    break;
}
// VERIFICO LA LECTURA PARA DETECTAR UNA POSIBLE CAIDA
if (lectura < caidaFrontal){ // si el valor es menor al margen puedo estar cayendo hacia adelante
    cantValoresAdelante++;
    cantValoresAtras = 0;
    if(cantValoresAdelante > CANT_VALORES) // si el valor de caida se repite confirmo la caida
        return CAI_ADELANTE;
}else{
    if(lectura > caidaAtras){ // si el valor es mayor al margen puedo estar cayendo hacia atras
        cantValoresAtras++;
        cantValoresAdelante = 0;
        if(cantValoresAtras > CANT_VALORES) // si el valor de caida se repite confirmo la caida
            return CAI_ATRAS;
    }else{
        cantValoresAdelante = 0;
        cantValoresAtras = 0;
    }
}
```

4.3.2. Sentido

Si bien en el algoritmo 4.1 se puede observar como se determina el sentido de la caída es necesario aclarar algunos puntos más sobre este tema.

Debido a como se encuentra ubicado el sensor de giro, valores más bajos que el valor de equilibrio indican que el cuerpo del robot se está inclinando hacia adelante, mientras que si los valores sobrepasan el equilibrio la inclinación se da en el sentido contrario. Además de estos valores, se cuenta también con los valores sobre el eje y , donde valores más bajos al equilibrio indican

una inclinación del cuerpo hacia la izquierda, y lo opuesto para valores que lo sobrepasan.

Toda esta información es útil para hacer un análisis de la caída y poder determinar cómo se encuentra el robot una vez finalizada la misma. Poder realizar este análisis es de gran importancia ya que esto determina el movimiento siguiente a ejecutar, que a su vez define si el jugador retoma el juego, levantándose del piso, o debe ser retirado del campo. Sin embargo, algo tan sencillo como una caída puede resultar bastante complejo. Desde el inicio de la caída pueden suceder diferentes acontecimientos que determinan si el robot cae de frente o de espaldas. Por ejemplo, puede suceder que el robot gire durante la caída, lo que cambia drásticamente el resultado final. Por situaciones como estas es que no es suficiente confiar en los valores de un eje para determinar el final de la caída. Complementando en este caso con los valores de lectura sobre el eje y se podría rectificar si se produjo un giro.

Si bien esto representa una mejor aproximación, aún se están dejando fuera otros aspectos del entorno que influyen también en la caída. ¿Qué pasa si en la caída se produce un choque con otro robot? ¿Y si un brazo trava contra el piso e invierte el giro? Estos acontecimientos pueden suceder y aún utilizando el eje y no podríamos asegurar el resultado final. Reformulando, incorporar la lectura del eje y agrega mucha complejidad al algoritmo y aún con su aporte quedan escenarios sin ser desambiguados.

Esto condujo a simplificar el uso de los valores del sensor utilizando sólo los valores sobre el eje x para determinar si la caída es frontal o de espaldas, implementando adicionalmente un mecanismo que permita corroborar la predicción realizada y que permita lidiar con el problema de los giros en la caída y lograr una predicción correcta.

4.3.3. Corrección

Una vez finalizada la caída se debe ejecutar la maniobra que ponga al robot nuevamente de pie para continuar con el juego. El análisis de la caída da un indicio de la posición en la que se encuentra el robot en el piso pero esto puede no ser correcto. Para determinar si el análisis falló se utilizan como herramientas los propios movimientos para levantarse. En las primeras poses de estos movimientos los motores «realizan fuerza» para levantar el cuerpo del piso lo que provoca una carga importante en los motores por un tiempo. Si este mismo esfuerzo se realiza contra el aire, en caso de estar realizando la maniobra equivocada, la carga de los motores y el tiempo que requiere alcanzar la pose final serán menores a los anteriores. Estas diferencias permiten saber con relativa facilidad cuando el análisis es equivocado y corregir la decisión a tiempo. Con este mecanismo se podría prescindir de la detección

del sentido: sencillamente se detectaría la caída y se ejecutaría una maniobra de reincorporación que tendría su corrección en caso de corresponder. Esta solución sin embargo tendría un mayor costo en tiempo lo que se evita analizando el sentido de la caída mientras se produce.

4.4. Predicción de trayectoria de la pelota

Conocer que sucederá en un futuro próximo resulta de gran ayuda para tomar decisiones a corto plazo. El golero elige sus acciones en función de las observaciones que hace de la pelota y el entorno. Cuanto mejor sea la predicción de la trayectoria de la pelota, más acertadas serán las decisiones que toma. En el escenario de un disparo al arco, existen diferentes aproximaciones posibles. La más básica puede ser elegir una acción basándose únicamente en la última posición observada. Aunque puede ser válida, esta aproximación deja fuera información que puede complementar y mejorar la toma de decisiones. Incorporar observaciones anteriores de la pelota para describir la trayectoria que realiza permite tener una noción de cómo evolucionará su posición en el futuro cercano. Solamente teniendo en cuenta la posición anterior de la pelota, ya es posible aproximar su movimiento por una recta y estimar la velocidad con que se mueve (obviamente teniendo en cuenta las marcas de tiempo en que se obtuvieron ambas posiciones).

Este razonamiento puede continuar incorporando más puntos y así obtener mayor precisión en la aproximación de la trayectoria. Si bien esto agrega complejidad es un paso importante en la búsqueda de mejores resultados. Pero antes de explicar como se ha realizado esta mejora, es necesario aclarar lo siguiente. En este trabajo se asume, como es razonable hacer, que una vez que la pelota abandona el pie del pateador describe una trayectoria en línea recta y se mueve con aceleración constante. Es decir, el modelo considerado se ajusta al de un movimiento rectilíneo uniformemente acelerado.

En los párrafos que siguen se describe el análisis que deriva en la estimación de la trayectoria de la pelota, poniendo el foco en el golero, donde es deseable frente a un ataque rival predecir la trayectoria que tendrá el balón para así poder proyectar la jugada y mantener el arco a salvo. Cabe aclarar que aunque se ha utilizado solamente para el golero, el análisis podría ser aplicado a otros escenarios donde la pelota se mueve libremente. Para la predicción de la trayectoria se utiliza como base la idea general planteada en [Alv04] ajustada a la realidad de este proyecto. Como se observa en la figura 4.5, el estado de la pelota es modelado con la terna (t, x, y) que representa la posición de la pelota en el campo en el momento t , siendo x la coordenada en el eje paralelo a la línea del arco e y la coordenada en el eje paralelo a la línea

lateral de la cancha. Los ejes se cruzan en la posición del golero, el origen del sistema de coordenadas, los valores de x aumentan hacia la izquierda del golero y los de y hacia el arco contrario.

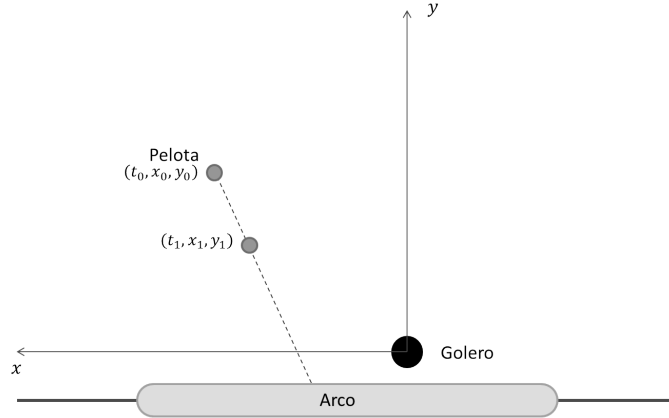


Figura 4.5: Esquema de referenciación de la pelota en la predicción de trayectoria.

Como se mencionó anteriormente, el modelo se ajusta al de un movimiento rectilíneo uniformemente acelerado, entonces las coordenadas de la posición de la pelota en un tiempo t dado responden a la ecuación:

$$x(t) = x_0 + v_0 t + \frac{a}{2} t^2$$

Para realizar la predicción se utiliza, como se sugiere en [Alv04], el método de Mínimos Cuadrados. Esto permite, en base a las observaciones realizadas por el golero al seguir la pelota, obtener el polinomio de grado 2 que mejor aproxima la trayectoria, minimizando el error cuadrático.

Previo a los cálculos se debe tener en cuenta las limitantes del sistema para incorporar estas restricciones y ajustar el modelo a la realidad del problema.

- Capacidad de cálculo y espacio de almacenamiento. Es necesario esforzarse en simplificar las cuentas a realizar y acotar la información almacenada ya que el espacio de almacenamiento disponible es restringido.
- Limitaciones de la visión. La relación entre velocidad de procesamiento de frames y alcance del campo visual, hace que estén disponibles pocas tomas en el lapso de un disparo al arco.

Dadas estas limitaciones es necesario simplificar los cálculos y obtener resultados del método con la menor cantidad de puntos posibles sin perder confiabilidad. Para simplificar las cuentas en vez de determinar todos los coeficientes del método en el momento de la ejecución como se hace en [Alv04], se toma la aceleración como dato. Es posible asumir que la aceleración sólo depende del coeficiente de rozamiento entre la cancha y la pelota, y por lo tanto no varía entre distintos disparos al arco por lo que puede ser obtenida previo a la ejecución del algoritmo. La aceleración (su módulo) se transforma en un parámetro más del sistema y ahorra cálculos y tiempo, sin renunciar a aproximar la posición por un polinomio de segundo grado. Para determinar la aceleración registramos los valores de observación del golero en reiterados tiros al arco. Con estos valores podemos utilizar el método de Mínimos Cuadrados para obtener la aceleración. Para cada tiro podemos formar una tabla con los valores del tiempo y la posición de la pelota en ese instante (t_i, x_i, y_i) y podemos plantear el sistema:

$$x(t_i) = x_i = x_0 + v_{0x}t_i + \frac{a_x}{2}t_i^2$$

$$y(t_i) = y_i = y_0 + v_{0y}t_i + \frac{a_y}{2}t_i^2$$

Resolviendo éste utilizando Mínimos Cuadrados¹ es posible hallar las componentes de la aceleración sobre los ejes y por lo tanto determinar el módulo de la misma. Con estos valores obtenidos de las distintas repeticiones se toma el promedio como el valor de la aceleración del sistema. Lo anteriormente descrito constituye un método para determinar la aceleración, dada una pelota y una cancha y por lo tanto permite determinar el parámetro a utilizar, siendo tolerantes a un cambio de pelota o de cancha.

Al quitar la aceleración como incógnita, se cambia el balance entre cantidad de puntos y confiabilidad, lo que implica una estimación de la trayectoria en un tiempo menor. Esto ahorra tiempo crítico cuando el golero debe tomar la decisión de tirarse y la pelota se dirige al arco. Además, esto permite simplificar los cálculos obteniendo las ecuaciones:

$$x_i - \frac{a_x}{2}t_i^2 = x_0 + v_{0x}t_i$$

$$y_i - \frac{a_y}{2}t_i^2 = y_0 + v_{0y}t_i$$

¹Dado un conjunto de puntos $\{(t_i, x_i)\}$ se aproxima la trayectoria tal que $x_i \approx \sum c_i t_i^n$ determinando los coeficientes c_i que minimicen $\|Ac - b\|$ donde A es la matriz formada por los vectores fila $(1 \ t_i \ t_i^n)$, b el vector formado por las observaciones x_i y c el vector de los coeficientes c_i .

donde a_x y a_y son conocidos.

Con los resultados obtenidos se predice la posición del balón en un tiempo determinado, así como el tiempo que le llevará en cruzar la línea del arco.

Esto le permite al golero tener una noción de cuándo y dónde la pelota cruzará la línea del arco dándole la posibilidad de tomar acciones para evitarlo.

Capítulo 5

Pruebas y resultados

Se detalla en este capítulo el plan de pruebas seguido para la evaluación del equipo de fútbol desarrollado y los resultados obtenidos en el mismo.

Las pruebas realizadas se dividieron en dos tipos: pruebas funcionales, que tienen por finalidad probar la capacidad de los robots para enfrentar las diferentes situaciones que se le presenten durante un partido de fútbol, y pruebas no funcionales, cuya finalidad es la de evaluar el desempeño de determinados elementos que componen el equipo, en particular se evaluó el desempeño de los motores Ax-12.

5.1. Pruebas funcionales

Las pruebas funcionales realizadas fueron en base a situaciones. En ellas se evaluó la capacidad del robot para identificar las diferentes situaciones y su reacción frente a las mismas. Cada prueba consistió en la repetición de una situación reiteradas veces, acompañada de un análisis posterior de indicadores que permitieron evaluar el accionar del robot. La evaluación se realizó en base a las estadísticas obtenidas del total de las ejecuciones realizadas.

Para todas las pruebas realizadas, se utilizó el software de visión global Doraemon[Joh02], con el fin de tener un marco de referencia al analizar las pruebas y poder comparar los resultados obtenidos.

No se realizaron pruebas de integración debido a que no se contaba el hardware necesario para realizar las mismas.

5.1.1. Indicadores

Con el fin de analizar los resultados obtenidos en las pruebas funcionales, se definió un conjunto de indicadores considerados significativos para tales

finés teniendo en cuenta las diferentes pruebas realizadas. A continuación se presentan cada uno de estos indicadores:

- **Tiros al arco y tiros por fuera del arco:** Estos dos indicadores contabilizan la cantidad de tiros realizados que se consideran al arco y por fuera del arco respectivamente. Se considera que un tiro es al arco, cuando la trayectoria de la pelota luego de ser pateada la dirige al interior del arco contando inclusive aquellos casos en que la pelota no llega al interior por falta de velocidad. Estos indicadores permiten evaluar la capacidad del robot atacante para realizar tiros al arco y para ubicar el arco contrario al momento de realizar el tiro.
- **Posesiones de la pelota logradas:** Este indicador contabiliza la cantidad de veces que se logra la posesión de la pelota. Este indicador permite evaluar la capacidad del robot de ubicar, identificar y desplazarse hacia la pelota hasta lograr la posesión de la misma.
- **Tiempo entre localización de la pelota y posesión de la misma:** Se registra el tiempo que pasa desde que el robot comienza a ubicar la pelota, hasta que logra hacerse con la misma. De esta forma se evalúa la capacidad de localización de la pelota por parte del agente y la movilidad del mismo en la cancha.
- **Ataques detectados e interceptados:** Este indicador contabiliza la cantidad de ataques que fueron exitosamente detectados e interceptados. Se evalúa con el mismo la capacidad del defensa de detectar cuando se produce un ataque del equipo contrario y su reacción frente a dicha situación. Un ataque se considera detectado cuando el defensa identifica que un jugador del equipo contrario dispone del balón y un ataque se considera interceptado cuando el defensa se ubica entre el contrincante y el arco propio.
- **Tiempo en interceptar ataques:** Se registra el tiempo que demora el agente en lograr interceptar un ataque del equipo contrario. Evaluando la velocidad de respuesta del agente frente a ataques del equipo contrario.
- **Tiros detectados por el golero:** Se evalúa con este indicador, la capacidad del golero de identificar cuando se produce un tiro al arco. Se considera que el golero identifica el tiro, cuando reacciona para atajarlo (lanzándose hacia el lado correcto), sin considerar si logra su cometido de atajar el tiro o si el gol es convertido, ya que no se está evaluando su efectividad al momento de atajar.

- **Seguimientos efectuados por el golero:** Con este indicador se verifica la capacidad del golero para acompañar la pelota cuando su trayectoria no implica un tiro al arco. La efectividad del seguimiento queda determinada por el acompañamiento con movimientos laterales siguiendo el sentido de la trayectoria del balón.
- **Tiempo y capacidad de reubicación del golero:** En caso de una pérdida de referencia se evalúa la velocidad y efectividad del mecanismo de reubicación. Se mide el tiempo que requiere el mecanismo y en cuantas oportunidades se recupera el centro del arco manteniendo la posición correcta (mirando de frente hacia la cancha).
- **Caídas sufridas y caídas detectadas:** Dos aspectos importantes en lo que refiere a movilidad de robots humanoides es su capacidad para mantener el equilibrio y su capacidad de detectar posibles caídas. Estos dos aspectos son evaluados en base a la cantidad de caídas sufridas y la cantidad de caídas detectadas.

5.1.2. Escenarios

En las ejecuciones fueron considerados cuatro escenarios de juego independientes. Dichas ejecuciones fueron realizadas 40 veces para cada uno de los escenarios con el fin de obtener valores para cada uno de los indicadores con los cuales se pudiera realizar un análisis del desempeño de los robots.

Se aclara que el indicador de caídas sufridas y caídas detectadas se obtiene de todas las ejecuciones realizadas sin importar a cual de los cuatro escenarios probados corresponda.

A continuación se detalla cada uno de los escenarios que fueron probados y los diferentes estados iniciales de cada uno.

5.1.2.1. Escenario I - Ataque

El atacante, sin sufrir la defensa de otro agente, arranca de una posición en el medio campo contrario. La pelota estará ubicada a una distancia fija del arco contrario, suficientemente cerca, para que el robot, una vez que se haga de la posesión de la pelota, realice un tiro al arco. En esta ejecución, se obtienen los indicadores de tiempo entre localización de la pelota y la posesión de la misma y tiros al arco y tiros por fuera del arco.



Figura 5.1: Escenario I - Estado inicial I

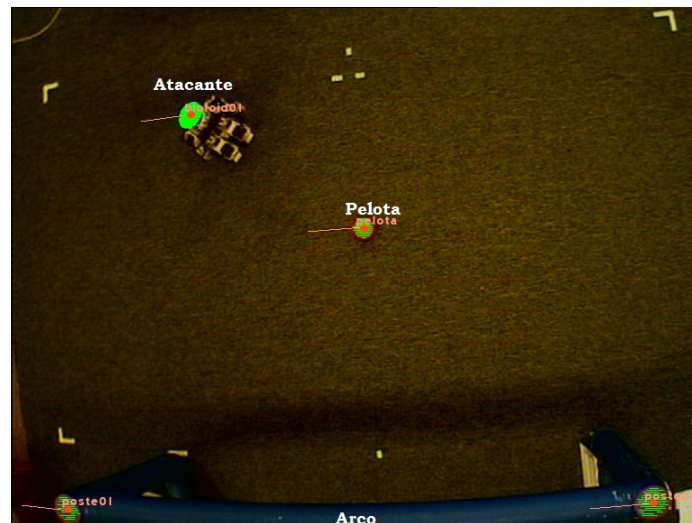


Figura 5.2: Escenario I - Estado inicial II.

En primera instancia se buscó replicar la ejecución de penal que se estipula en las reglas de la liga Kid Size (ver figura 5.1), se ubicó al robot frontal al arco de forma que el agente, la pelota y el punto medio del arco contrario se encontraran alineados. Con respecto a las distancias, se fijaron de tal forma que el escenario fuera abarcado por el sistema de visión Doraemon. El agente se ubicó a una distancia de 137 cm del arco, quedando en el borde del campo registrado por el sistema de visión, y la pelota a 54 cm del agente. Si bien la distancia entre la pelota y el agente es equivalente a 1.38 veces la altura

del robot erguido, y no respeta la distancia establecida por robocup para la ejecución de un penal, la cual es equivalente a 1.5 la altura del robot erguido, la posición de la pelota se eligió de forma que el robot tuviera que recorrer una distancia considerable para hacerse con la posesión de la misma pero que al mismo tiempo la pelota no estuviera a una corta distancia del arco de forma de no facilitar el remate del agente. En segunda instancia se fijó el mismo escenario pero posicionando al agente formando un ángulo de 45° con el arco (ver figura 5.2), con esto se buscó aumentar la dificultad del escenario al momento del remate.

Se realizaron 20 ejecuciones para cada uno de los estados iniciales definidos.

5.1.2.2. Escenario II - Defensa

El defensa se ubica en posiciones entre el arco propio y el jugador del equipo contrario que se encuentre más cerca de la pelota, de forma que no se encuentren alineados los dos agentes y el arco. Se obtiene de esta forma el indicador de ataques detectados e interceptados y el tiempo en interceptar ataques. Se aclara que en este escenario el agente atacante se ubicó estático en el campo de juego, por lo que no realiza ningún avance en el campo ni efectúa remates al arco.



Figura 5.3: Escenario II - Estado inicial I.

Para este escenario se ubicó al jugador contrario en el borde del campo que abarcaba el sistema de visión Doraemon, a 137 cm del arco y el agente defensa a un metro de distancia del centro del arco, formando un ángulo de

45°. La posición del arco que defiende fue precargada en el entorno del agente defensor para que este tenga la referencia del mismo dado lo aislada que es la situación que se está probando con respecto al desarrollo normal de un partido de fútbol. En una primera instancia se ubicó a la pelota a 10 cm del robot atacante (ver figura 5.3) de tal forma que tenga la posesión de la misma y en segunda instancia se ubicó a 30 cm del robot atacante (ver figura 5.4) de tal forma que no tenga la posesión de la pelota pero sí que sea el atacante el más cercano de los dos agentes. Se probaron estos dos estados iniciales cuya diferencia era la posesión de la pelota, realizando 20 ejecuciones para cada una de ellas.



Figura 5.4: Escenario II - Estado inicial II.

5.1.2.3. Escenario III - Golero atajando

El golero ubicado en el centro del arco recibe tiros al arco desde distintos puntos y percibe desplazamientos de la pelota. Este escenario permite medir el indicador de tiros detectados por el golero y el de seguimiento. Paralelamente, se contabilizan los casos en que pierde la referencia y debe relocalizarse.

El área de pruebas fue dividida en tres bandas, como puede apreciarse en la figura 5.5, que aparecen delimitadas por líneas verdes. Dentro de éstas se identificaron regiones donde la pelota no es visible por el golero desde la posición inicial (ver línea gris en la figura 5.5). Así, se consideraron 4 zonas para el punto de partida del tiro al arco. Una de ellas es la zona donde el arquero no ve la pelota, y las tres restantes corresponden a las formadas por

la intersección de cada una de las bandas con la zona donde el golero sí la visualiza.

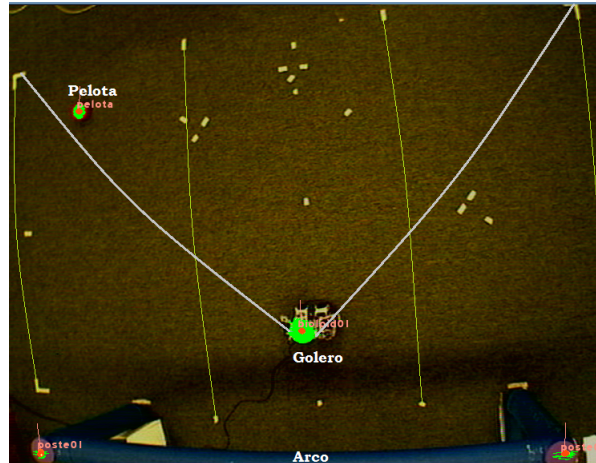


Figura 5.5: Escenario III.

Desde cada zona se realizaron 10 disparos al arco, con el fin de probar las distintas características del comportamiento del golero. Los disparos fueron variados para probar distintas situaciones. Direccionados hacia ambos lados del golero, los disparos se clasifican en cuatro situaciones posibles:

- Enfrentado: donde el golero está quieto observando la pelota y en la mejor posición para atajar ante un eventual disparo.
- No enfrentado: donde la pelota está dentro de su campo visual y el golero está quieto pero no en la mejor posición para atajar. Esta situación se da cuando el golero ha detectado que puede mejorar su posición («achicar el arco»).
- En movimiento: donde el golero se encuentra desplazándose. Este caso se da cuando el golero busca «achicar el arco», es decir se desplaza al lugar donde tiene mejores posibilidades para interponerse entre la pelota y el arco.
- En búsqueda: Cuando el golero no visualiza la pelota, la busca barriendo todas las zonas que están a su alcance con el objetivo de visualizar la pelota. Esta situación se da en las pruebas para la zona de no visibilidad.

5.1.2.4. Escenario IV - Golero Perdido

El golero cae y pierde su referencia debiendo reubicarse. Se obtiene el tiempo en recuperar el centro del arco y la efectividad de la maniobra.

Si bien durante las ejecuciones para el Escenario III fue posible probar esta situación, se realizaron 10 ejecuciones más para evaluar de mejor forma los indicadores. El golero se ubica en algún punto sobre su línea de acción (línea verde figura 5.6) y es empujado probocándole una caída, lo que dispara el algoritmo de reubicación.



Figura 5.6: Escenario IV.

5.2. Pruebas no funcionales

Las pruebas no funcionales se centraron en la evaluación del desempeño de los motores Ax-12. Esto se debe a que uno de los problemas que surgieron en el transcurso del proyecto, fue la rotura de algunos motores que se quemaron en diferentes momentos de las ejecuciones de los agentes. Las pruebas, además de permitir evaluar su desempeño, fueron realizadas para determinar medidas paliativas frente al quemado de motores. Con este fin, fueron analizados tres indicadores para cada uno de los motores durante diferentes ejecuciones, los mismos se presentan a continuación:

Indicadores:

- La temperatura alcanzada, medida en grados celsius.
- La relación de carga con la máxima soportada.

- El voltaje recibido, medido en voltios por diez¹.

Los valores que devuelve el motor con respecto a la carga soportada, no son una medida de la misma, sino que indica la relación que existe entre la carga soportada (o peso soportado) y la carga máxima tolerada. El rango del valor devuelto varía entre 0 y 1023, con lo cual si el valor del indicador es 512, implica que la carga soportada por el motor es el 50 % de la carga máxima tolerada.

Las ejecuciones realizadas consistían en que el robot atacante localizara la pelota, avanzara hasta tener su posesión sin defensa de otro agente y realizara un tiro al arco, para luego continuar con el ataque. Se realizaron tres corridas de aproximadamente siete minutos cada una, registrando el valor de los tres indicadores mencionados para cada uno de los motores, y el tiempo transcurrido desde el comienzo de la ejecución.

5.3. Resultados

A continuación se detallan los resultados obtenidos de las diferentes pruebas realizadas. Si bien los resultados obtenidos cumplen con lo esperado, resulta interesante compararlos con los de otros equipos para obtener una noción de competitividad. Lamentablemente no se logró conseguir resultados que sean comparables: los documentos técnicos de los equipos analizados son más generales y no presentan resultados de su desempeño mientras que los trabajos de investigación que desarrollan son más específicos como algoritmos de caminatas, localización, etc. que tampoco presentan resultados útiles para la comparación.

5.3.1. Resultados de las pruebas funcionales

5.3.1.1. Escenario I - Ataque

De las 40 ejecuciones realizadas para este escenario, en 29 oportunidades el agente fue capaz de lograr la posesión de la pelota, representando el 72,5 % de las ejecuciones, en un tiempo promedio de 1:04 minutos. En las 11 ejecuciones restantes, el agente no fue capaz de lograr la posesión debido a caídas no detectadas o debido a la no identificación de la pelota en el campo de juego.

Con respecto a los remates, el robot logró realizar 16 remates del balón, siendo el 40 % de las ejecuciones totales y el 55,2 % si restringimos las ejecuciones a aquellas que logró la posesión de la pelota. De los 16 remates, 11

¹Los motores Ax-12 manejan el voltaje multiplicándolo por 10.

fueron tiros al arco y 5 tiros fueron afuera del arco, por lo que el 27,5 % de las ejecuciones culminó con el agente realizando un tiro al arco y el 12,5 % de las ejecuciones terminó con un remate por fuera del arco.

En lo que refiere a las caídas, se registraron 15 caídas en total a lo largo de las 40 ejecuciones, detectando 11 de las mismas.

En la figura 5.7 se pueden ver representados gráficamente los resultados obtenidos de las pruebas realizadas para este escenario.

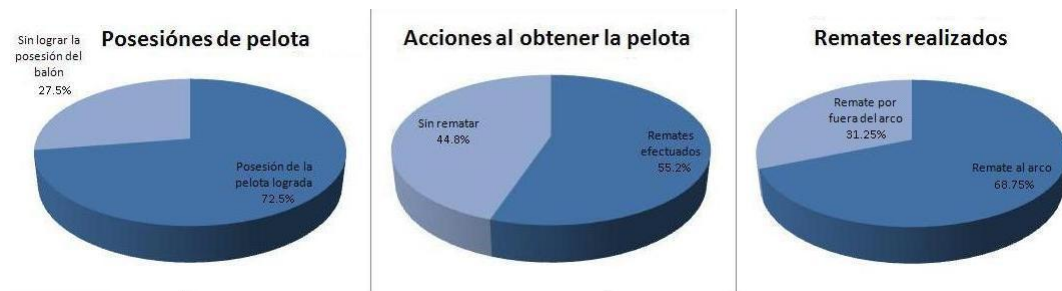


Figura 5.7: Resultados generales obtenidos del escenario I.

Se detallan a continuación los resultados en cada estado inicial ejecutado.

Estado Inicial I De las 20 ejecuciones realizadas con el atacante frontal al arco, en 14 oportunidades, es decir el 70 %, el agente fue capaz de ubicar la pelota y hacerse con la posesión de la misma. El tiempo promedio que transcurrió en este caso desde el inicio de la ejecución hasta la posesión de la pelota, fue de 1:05 minutos con una desviación estándar de 23,2 segundos, donde el motivo en las diferencias de tiempos se daba en la perdida de la pelota al dirigirse hacia la misma.

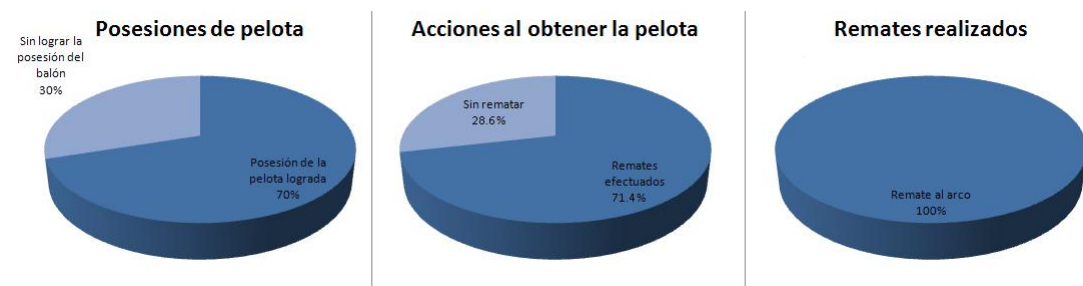


Figura 5.8: Resultados generales obtenidos del estado inicial I.

En 10 oportunidades (el 50 % de las ejecuciones) se culminó con el agente rematando el balón, de los cuales, todos fueron tiros al arco, esto facilitado

por ser un remate frontal. Si contamos solamente aquellas ejecuciones que el robot logró hacerse con la posesión de la pelota, el porcentaje de ejecuciones que finalizaron con el remate del balón asciende al 71,4 %.

Con respecto a caídas, se registraron 8 en total a lo largo de las 20 ejecuciones, siendo capaz de detectar 6 de las mismas. El 25 % de las caídas ocurrió mientras el agente realizaba el tiro y luego de haber impactado a la pelota.

En la figura 5.8 se pueden ver representados gráficamente los resultados obtenidos de las pruebas realizadas para este estado inicial.

Estado Inicial II De las ejecuciones realizadas con el atacante en ángulo al arco, el 75 % de las mismas, es decir en 15 oportunidades, el robot fue capaz de hacerse con la posesión de la pelota, con un tiempo promedio de 1:04 minutos con una desviación estándar de 17,7 segundos.

El agente logró realizar 6 remates del balón, de los cuales 5 fueron tiros fuera del arco y 1 fue un tiro al arco, con lo que en el 30 % de las ejecuciones el agente fue capaz de realizar un disparo, el 40 % de las ejecuciones que consiguió la posesión de la pelota. Los bajos números de remates luego de lograr la posesión de la pelota se deben a caídas sufridas al momento de realizar el remate, sumado a las pérdidas del balón al buscar el arco y a las pérdidas del balón al posicionarse para patear. Estas pérdidas se producían debido a que el robot rozaba con sus pies la pelota al moverse provocando que la misma se alejara. Por otra parte, el bajo número de remates al arco se debe a que la posición que tiene el agente con respecto al arco al momento de rematar, no le permite identificar el segundo palo del arco por más que el agente desplace la cámara hacia ambos lados para ubicarlo, con lo cual, solamente tiene un palo de referencia para dirigir el remate. A esto se suma que dado que la situación comienza con el agente muy cerca del arco, por lo que el entorno no aporta mayor información para detectarlo que la sensada al momento de tomar la posesión de la pelota. Si se consideran todas las ejecuciones, sin contar si el robot logró rematar o no, solamente el 5 % de las mismas culminaron con un tiro al arco y el 25 % culminó con un tiro fuera del arco.

En lo referente al equilibrio, el agente sufrió caídas en 7 oportunidades durante las 20 ejecuciones, siendo el 14,3 % de las mismas mientras remataba y luego de impactar la pelota. Cinco fue el número de caídas que el robot logró identificar.

En la figura 5.9 se pueden ver representados gráficamente los resultados obtenidos de las pruebas realizadas para este estado inicial.

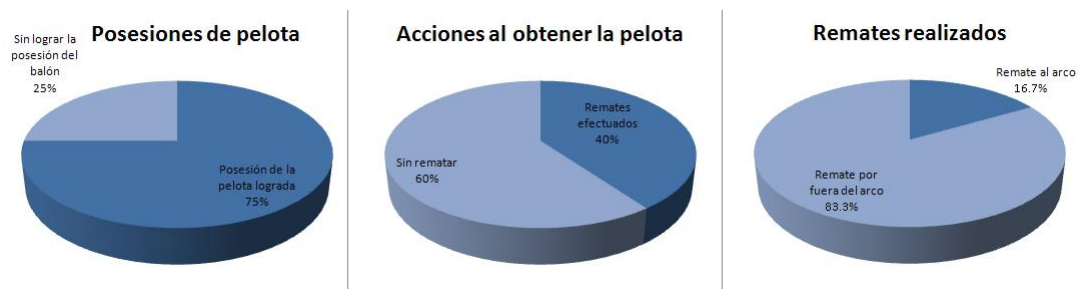


Figura 5.9: Resultados generales obtenidos del estado inicial II.

5.3.1.2. Escenario II - Defensa

De las 40 ejecuciones realizadas dentro de este escenario, el agente fue capaz de detectar 15 situaciones correctamente, es decir que el 37,5 % de las ejecuciones procedía a interceptar el ataque contrario, logrando interceptar 14 de las 15 detectadas (93.3 % de éxito luego de identificada la situación). Por lo que el 35 % de las ejecuciones culminaron con la detección e intercepción del ataque del equipo contrario de forma exitosa, al ubicarse de forma correcta entre la pelota y el centro del arco propio.

Se registraron 9 caídas en total a lo largo de todas las ejecuciones y 6 de las mismas fueron detectadas.

En la figura 5.10 se pueden ver representados gráficamente los resultados obtenidos de las pruebas realizadas para este escenario.

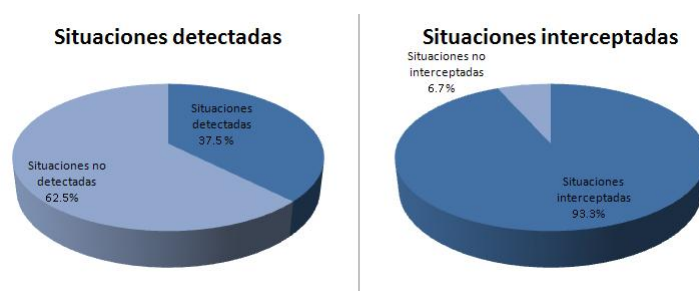


Figura 5.10: Resultados generales obtenidos del escenario II.

A continuación se detallan los resultados de acuerdo a los dos estados iniciales que se probaron.

Estado Inicial I De las ejecuciones que tienen al agente atacante con la posesión de la pelota, el defensa identificó en 10 oportunidades el ataque del equipo contrario correctamente, es decir que el 50 % de las ejecuciones

logró identificar la situación. De las identificadas, interceptó exitosamente 9 de las 10, logrando así cumplir su tarea en el 45 % del total de ejecuciones. El tiempo promedio que registró el agente para interceptar el ataque fue de 0:38 minutos con una desviación estándar de 1,7 segundos. La baja tasa de identificación de los ataques se debe a la dificultad del robot de identificar al agente del equipo contrario. Se puede ver en la figura 5.11 la posición que adopta el defensa luego de detectar e interceptar un ataque de forma exitosa.

Se registraron un total de 4 caídas durante las 20 ejecuciones, llegando a identificar 2 de las mismas.



Figura 5.11: Defensa detectada e interceptada durante las pruebas.

En la figura 5.12 se pueden ver representados gráficamente los resultados obtenidos de las pruebas realizadas para este estado inicial.



Figura 5.12: Resultados generales obtenidos del estado inicial I.

Estado Inicial II Con el agente del equipo contrario ubicado más próximo a la pelota, el porcentaje de ejecuciones que el defensa logró detectar e

interceptar los ataques fue de un 25 %, dado que en 5 oportunidades, detectó correctamente la situación a la que se enfrentaba, logrando interceptar las 5 en un tiempo promedio de 0:37 minutos con una desviación estándar de 8,4 segundos. Al igual que en las ejecuciones del estado inicial I, la baja tasa de identificación de los ataques se debe a la dificultad por parte del robot de identificar al agente del equipo contrario.

Durante estas pruebas el agente sufrió un total de 5 caídas y logró detectar 4 de ellas.

En la figura 5.13 se pueden ver representados gráficamente los resultados obtenidos de las pruebas realizadas para este estado inicial.

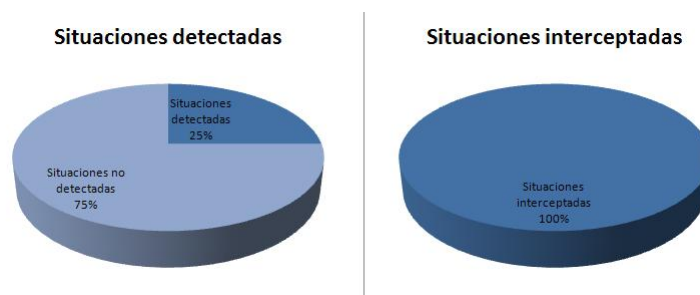


Figura 5.13: Resultados generales obtenidos del estado inicial II.

5.3.1.3. Escenario III - Golero Atajando

Como este escenario se constituye de situaciones diversas, incluimos un cuadro comparativo a fin de facilitar la interpretación de los resultados.

		E		NE		EM		EB		Total
		I	D	I	D	I	D	I	D	
Detectados	Zona Izquierda	2	2	1	2				2	27
	Zona Centro	3	2	2	2	1				
	Zona Derecha	2	2	1	1	1		1		
No Detectados	Zona Izquierda			1		1	1	2	1	13
	Zona Centro									
	Zona Derecha			1	1	1		1	3	
Error en detección	Zona Izquierda								1	3
	Zona Centro	1								
	Zona Derecha			1						

E: Enfrentado - NE: No Enfrentado - EM: EnMovimiento - EB: En Búsqueda

Cuadro 5.1: Detección del disparo

Del total de disparos al arco realizados para este escenario, el 67,5 % fueron detectados por el golero donde en el 89 % de los casos el golero acertó la dirección del disparo. Pero este resultado no es homogéneo, ya que si el disparo parte de alguna de las zonas donde tiene visibilidad, el índice de detección es del 80 %, entre los cuales acertó la orientación en el 85 % de los casos. En contraposición, las tasas de acierto bajan drásticamente si el disparo parte desde un lugar donde no puede ser visto (30 % y 66 % respectivamente).

A su vez, se desprende de los resultados que el golero predice mejor cuando el tiro es frontal, y que esto tiene mayor incidencia que si está o no enfrenteado a la pelota, aunque la primera de estas situaciones es la que más lo favorece, como es de esperar. Que el golero tenga tan baja efectividad cuando el tiro parte mientras está en movimiento (2 aciertos en 5) se explica porque durante el desplazamiento no obtiene tomas por tener un error demasiado grande, y cuando vuelve a realizar tomas puede haber perdido la pelota de vista.

Paralelamente, durante estas pruebas, se estudió el seguimiento de la pelota, en el que hubo acierto en todos los casos. Finalmente, respecto de las caídas, en las pruebas se produjeron 4, de las cuales el golero detectó 3. Cabe aclarar que el golero tiene movimientos más restringidos que un jugador de cancha, y por lo tanto menos posibilidades de caerse.

Además de los resultados obtenidos de los indicadores, gracias a la utilización del módulo Doraemon podemos evaluar el modelo interno que genera el golero y las predicciones que realiza. El gráfico 5.14 muestra por un lado como Doraemon capta la trayectoria de la pelota y por otro como el golero modela internamente esta trayectoria con los puntos obtenidos en sucesivas capturas (utilizando el módulo de visión VisRobII). Si bien las trayectorias difieren en sus coordenadas, las curvas que describen son similares. De las pruebas se obtiene que la distancia entre las mismas ronda los 15 cm. Esta distancia es considerable pero debemos tener en cuenta que existen diferencias en la base de las pruebas que aportan a este error. Por ejemplo, existe una diferencia entre los ejes de los dos modelos comparados. Además los puntos captados por el robot presentan un error mayor que los captados por Doraemon. Si a esto agregamos que cuando el robot realiza un movimiento se deben recalcular los valores de los puntos, estimando la distancia de desplazamiento en el movimiento, el error aumenta. A pesar de estas diferencias los resultados resultan satisfactorios.

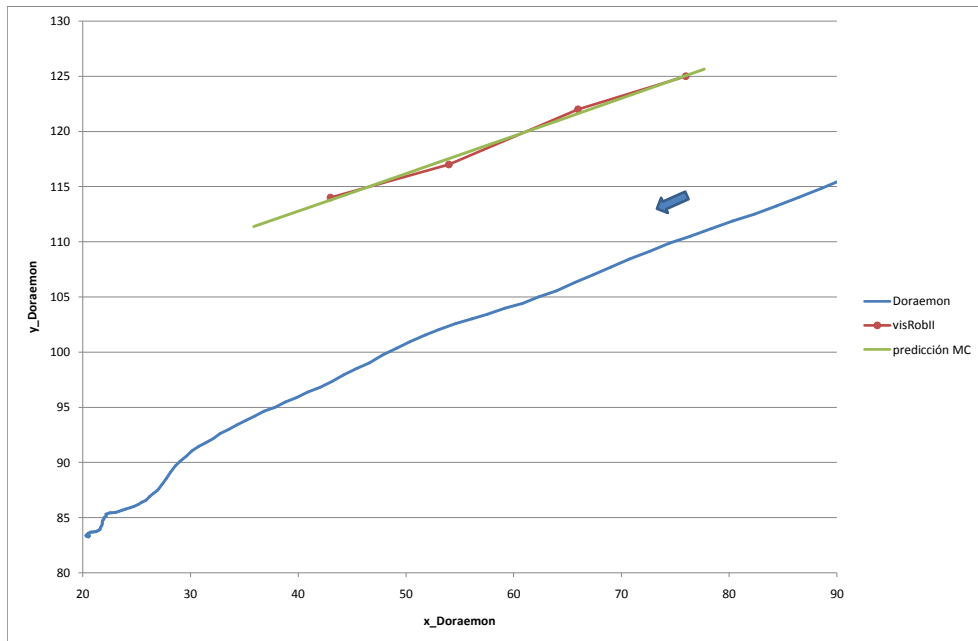


Figura 5.14: Comparación de modelos Doraemon - Golero.

A su vez podemos analizar las predicciones que realiza el golero sobre la trayectoria de la pelota. Con la predicción, el golero estima en qué punto cortará la pelota a la recta imaginaria sobre la que él se desplaza (paralela a la línea del arco). En el gráfico 5.14 se puede observar la aproximación de la trayectoria utilizando la predicción por mínimos cuadrados. La cantidad de puntos empleada en el cálculo juega un papel importante. El resultado, en algunos casos, mejora la predicción lineal pero existen casos que al calcular la trayectoria con 3 puntos la curva estimada se aleja de la realidad por lo que este resultado no puede ser usado para tomar decisiones. Las limitaciones de visión para seguir la pelota a una distancia grande sumadas a la dificultad para determinar si la pelota está en movimiento (el error en la posición de la pelota marca un umbral alto de distancia entre sus posiciones para considerarla en movimiento) generan que el seguimiento y modelado captado por el golero sea de pocos puntos (3 o 4 para un disparo) lo que restringe la capacidad de mejora de la predicción.

5.3.1.4. Escenario IV - Golero perdido

Para las 3 caídas detectadas en el escenario anterior, sólo en un caso se logró ejecutar el algoritmo de relocalización ya que en las otras 2 oportunidades la caída se produjo hacia adentro del arco, lo que no cumple con las precondiciones del algoritmo de recuperación del golero. Fueron realizadas

entonces 10 ejecuciones adicionales para evaluar mejor el algoritmo. En 7 de las 10 ejecuciones el golero logró ubicar el arco para recuperar su posición inicial pero sólo en 4 oportunidades alcanzó esa ubicación ya que los errores de cálculo de distancia a los postes sumado a los errores de los movimientos no permitieron que se alcanzaran en los otros casos. En promedio la maniobra consume 1:24 minutos ($\sigma = 14,1$ segundos) incluyendo la ubicación del arco y el movimiento de relocalización. La diferencia entre las medidas de tiempo se debe principalmente al tiempo que se emplea en identificar el primer poste del arco que depende del lugar donde haya caído el golero, el resto de la maniobra no presenta grandes variaciones en diferentes ejecuciones.

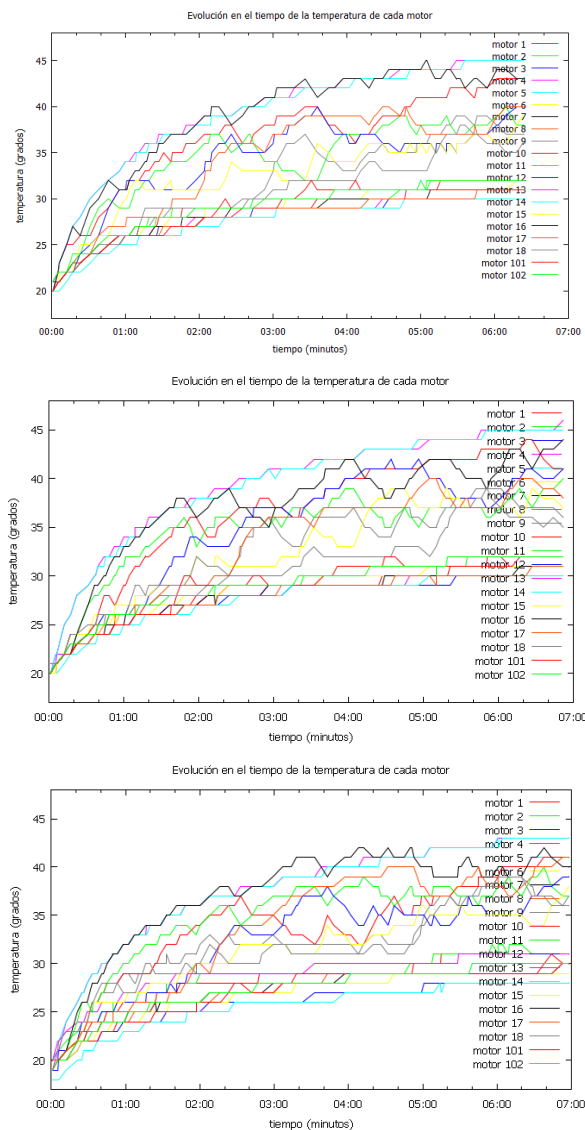
La no detección de los postes del arco explica los 3 casos en que el golero no pudo ubicar la posición deseada, ya que, al ser la única referencia en el entorno que puede ser captada por la visión, su detección es crítica para el éxito del objetivo en cuestión.

5.3.2. Resultados de las pruebas no funcionales

Se presentan en los cuadros 5.2, 5.4 y 5.6 las gráficas realizadas a partir de las mediciones obtenidas en las pruebas no funcionales acompañadas de los cuadros donde se calculan para cada, el valor mínimo, el valor máximo, el promedio de los valores y su desviación estándar. La numeración de los motores corresponde a la que se indicó en la figura 3.13 en la sección 3.1.3.

La primera conclusión que se puede obtener al analizar las gráficas, es el gran incremento de la temperatura que sufren los motores, donde algunos alcanzan en dos minutos prácticamente el doble de su temperatura inicial, pasando de una temperatura de 20 grados a una temperatura entre 37 y 39 grados. Este incremento si bien continúa, tiende a disminuir con el correr de los minutos, alcanzando así temperaturas finales que van desde los 30 hasta los 46 grados, siendo los motores correspondientes a las rodillas del robot (motores 13 y 14) los que alcanzan las temperaturas más altas.

Dado este aumento de temperatura que sufren los motores, se decidió limitar la temperatura máxima a 50 grados para tener un margen con respecto a la temperatura máxima soportada de 70 grados. Dicho limite puede ser seteado en la memoria EEPROM de los motores [Rob06], con lo cual, si un motor alcanza una temperatura por arriba de ese valor, se producirá un error sobre temperatura (Over Heating Error) y quedará sin torque, evitando así que siga funcionando frente a una situación que puede provocar que se queme. De esta forma se logró un mínimo control frente a situaciones de riesgo.



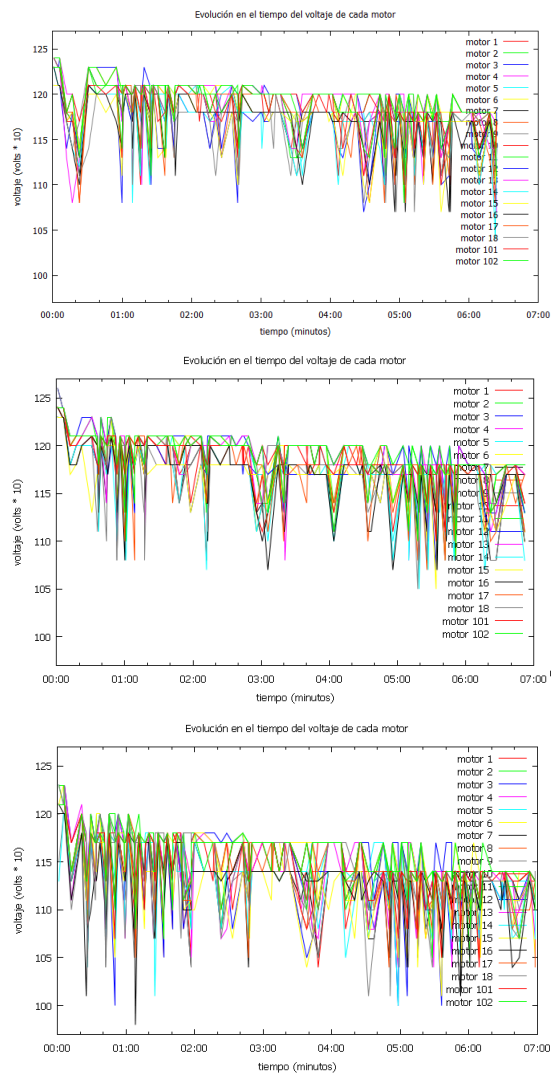
Cuadro 5.2: Gráficas de temperatura (arriba: *Ejecución I*, medio: *Ejecución II*, abajo: *Ejecución III*).

	Mínimo	Máximo	Valor promedio	Desviación estándar
Ejecución I	20	45	31.956	5.617 (17.58 %)
Ejecución II	20	47	31.917	5.874 (18.40 %)
Ejecución III	18	44	30.659	5.581 (18.20 %)

Cuadro 5.3: Estadísticas de las gráficas de temperatura.

Otro aspecto importante que se puede apreciar en las gráficas, en particular en las gráficas correspondientes a los voltajes, es que en ningún momento hay picos de alto voltaje que puedan provocar que el motor se queme. Esto se debe a que la fuente regula el voltaje evitando que existan picos de alto voltaje. Además se ve reflejado gráficamente como el voltaje va disminuyendo a medida que pasa el tiempo, producto de que la batería se va descargando. El rango de voltaje que manejan los motores por defecto es de 60 a 140 ($6V \sim 14V$), y el voltaje en las ejecuciones se encuentra siempre en el rango de los 98 y 126 ($9.8V \sim 12.6V$), partiendo siempre con baterías completamente cargadas.

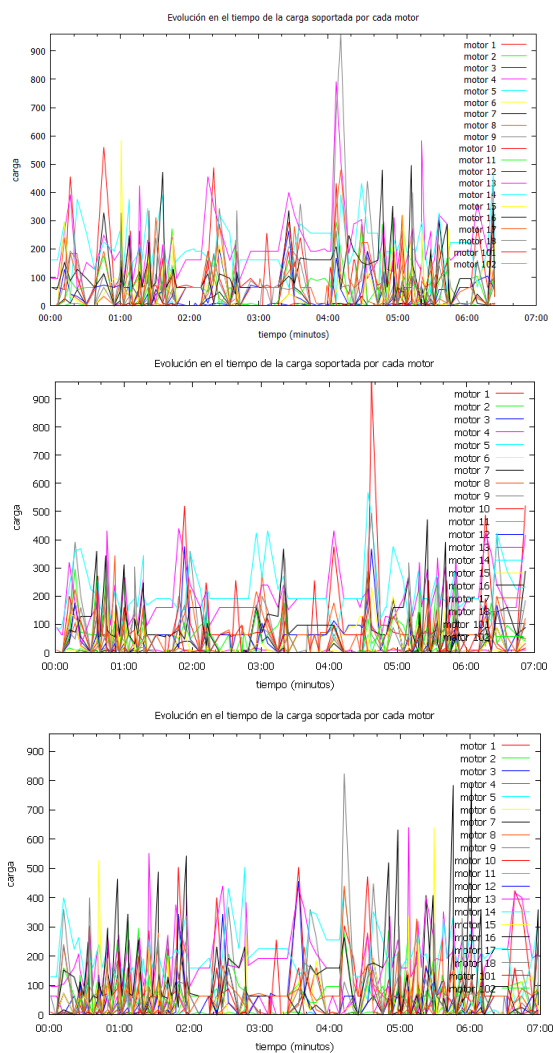
Con respecto a las gráficas de la relación de la carga, se ve la oscilación de los valores que alcanza en cada motor, dado que la carga soportada depende del movimiento real que se esté realizando en cada momento. Un punto importante que se puede apreciar en las dos primeras gráficas, es el pico que se produce pasado los cuatro minutos. En la ejecución I se produce en los motores 9 y 13, correspondientes a la pierna derecha (cadera y rodilla), debido a que en dicho momento el robot pateó la pelota con el pie izquierdo y el cuerpo era sostenido solamente por la pierna derecha. En la ejecución II, el pico se produce en los motores 10 y 14, correspondientes a la pierna izquierda, debido a que en la misma el robot pateó con el pie derecho, produciendo el mismo pico de carga pero en los motores simétricos.



Cuadro 5.4: Gráficas de voltaje (arriba: *Ejecución I*, medio: *Ejecución II*, abajo: *Ejecución III*).

	Mínimo	Máximo	Valor promedio	Desviación estándar
Ejecución I	104	124	117.898	3.142 (2.66 %)
Ejecución II	105	126	117.795	3.065 (2.60 %)
Ejecución III	98	123	113.803	3.656 (3.21 %)

Cuadro 5.5: Estadísticas de las gráficas de voltaje.



Cuadro 5.6: Gráficas de relación de la carga (arriba: *Ejecución I*, medio: *Ejecución II*, abajo: *Ejecución III*).

	Mínimo	Máximo	Valor promedio	Desviación estándar
Ejecución I	0	959	57.662	95.078 (164.89 %)
Ejecución II	0	967	51.560	85.349 (165.53 %)
Ejecución III	0	824	58.257	94.968 (163.01 %)

Cuadro 5.7: Estadísticas de las gráficas de relación de la carga.

Capítulo 6

Conclusiones y trabajo futuro

A lo largo del documento, partiendo de la idea de conformar un equipo de fútbol de robots humanoides se han recorrido los distintos aspectos de este proyecto que abarcan distintas áreas de la robótica y la ingeniería más en general. Dada su complejidad, no todos estos aspectos fueron abordados con la misma profundidad. Se hizo especial énfasis en aquellos que resultaban más interesantes o fundamentales dado el alcance del proyecto. La diversidad de temáticas que toca habla de la magnitud de un proyecto con este objetivo y como el abordaje de cada temática puede constituir un proyecto en sí mismo. Por esto el presente trabajo fundó sus bases en proyectos anteriores relacionados y pretende abrir camino hacia la construcción de un equipo competitivo.

En cuanto a los objetivos centrales del proyecto se logró cumplirlos y a su vez los avances obtenidos abren un abanico de posibles trabajos futuros a considerar. En las siguientes subsecciones se exponen las conclusiones resultantes del trabajo realizado y propuestas de trabajo futuro.

6.1. Conclusiones

Conformar un equipo de fútbol humanoide implica atacar un amplio espectro de problemas incluyendo la construcción de los robots, la correcta selección del hardware en función de las cualidades del equipo; el diseño, la implementación y la optimización de los movimientos; establecer las estrategias que gobiernan las decisiones de los robots y darle una coherencia a todos estos para lograr que se potencien en la solución final. Todas estas áreas son objeto de investigación en la comunidad científica de hoy y lograr formar un equipo a nivel competitivo es una tarea que usualmente involucra a grupos enteros de investigadores.

Cada uno de los aspectos que mencionados influyen directamente en el desempeño general y la manera como se balancean puede dar lugar a equipos completamente distintos. Este abanico de opciones y oportunidades de profundización hace difícil determinar el alcance del proyecto para darle un punto final. A su vez, más allá del conocimiento que se pueda alcanzar sobre cada uno de los aspectos, el paso de la teoría a la práctica en el marco de agentes robóticos constituye otro desafío en sí mismo pudiendo introducir variables que no habían sido tenidas en cuenta o que en un principio no representaban un problema.

Algunas de las problemáticas fueron atacadas y resueltas en proyectos anteriores, lo que permitió que el proyecto fuera abarcable. Por otro lado hubo que atacar toda otra serie de desafíos, siendo algunos de ellos imprescindibles para la construcción del equipo. En este proceso se pueden identificar tres etapas. La primera de estas comprendió desarrollar las habilidades básicas de los agentes. La segunda fue la de integración del módulo de visión que constituye un elemento crítico para la toma de decisiones. Y finalmente se atacó la conformación del equipo, definiendo reglas para la interacción, roles y una coherencia a las acciones de los agentes. Para cada una de estas etapas se presentan a continuación las principales conclusiones obtenidas.

6.1.1. Habilidades básicas de los agentes.

Se pudo cumplir con el primer objetivo central del proyecto: definir e implementar los diferentes movimientos básicos de los robots humanoides. Esto constituyó el cimiento sobre el cual se desarrollaron las acciones de los agentes para lograr el objetivo de construir un equipo de fútbol robótico. Además de permitir el desarrollo del proyecto pueden ser útiles para resolver otros tipos de actividades por fuera del fútbol de robots.

Durante la construcción y el trabajo sobre estos movimientos se debió profundizar en el conocimiento de la plataforma Bioloid Premium y por lo tanto en la mayoría de sus componentes. El aprendizaje en el manejo y los problemas enfrentados y resueltos en el uso de motores AX-12, giroscopio y el propio bloque CM-510 genera también una fuente de conocimiento para otros proyectos.

Lograr este objetivo constituyó el primer escalón fundamental en el desarrollo del proyecto donde se pudo evaluar físicamente la complejidad de la tarea y lo que ésta influye en el resto del desarrollo. Profundizar en las mejoras de estas habilidades y perfeccionarlas pueden influir positivamente en el desempeño que tendrán los jugadores en el campo y es necesario encontrar un equilibrio entre la exactitud de los movimientos y la efectividad de los mismos. Por ejemplo una caminata que busca la perfección en sus pasos

puede ser muy vistosa pero no cumplir con el objetivo de alcanzar el balón antes que el contrario. Para el caso concreto del proyecto los movimientos de rotación se simplificaron al máximo siendo menos «prolijos» pero más veloces y por lo tanto más efectivos a la hora de ubicar al jugador frente a la pelota. Otro punto importante es lograr una adaptabilidad de los movimientos a pequeños cambios en la configuración. Un problema que se debió enfrentar fue que pequeños cambios en la postura del robot debidos sencillamente al uso del mismo o a cambios de algún motor quemado que presentaba diferencia mínima de fabricación con el sustituido, provocaba variaciones en los movimientos afectando el buen desempeño y provocando en algunos casos caídas. Por lo que es muy importante tener en cuenta esto al momento de trabajar en la construcción de movimientos.

6.1.2. Integración con el módulo de visión

Si bien el módulo de visión era un insumo más para el proyecto, la integración del mismo significó un gran desafío. Fue necesario adaptar los robots para lograr un buen uso del sistema de visión, lo que implicó grandes cambios en los movimientos desarrollados. Como aprendizaje se vio que la integración puede conllevar un trabajo importante más que simplemente aprender a utilizar el sistema integrado y debe ser tenida en cuenta ya sea para pautar reglas con el sistema integrado o para mitigar el impacto que pueda provocar.

Este paso constituye el primer uso práctico del módulo de visión, sobre el cual además se construyó una capa de operaciones de más alto nivel reutilizables para un futuro. Se desarrollaron algoritmos básicos de búsqueda de objetos y seguimiento de la pelota dándole uso al sistema de visión y probando su desempeño.

Este módulo tiene gran influencia sobre el desempeño del agente ya que es el punto principal de ingreso de información y sobre esa información se toman decisiones. Claramente, si la información capturada difiere de la realidad esto se verá reflejado en las acciones realizadas. Trabajar en lograr mejores resultados en este módulo sin duda aportará para lograr mejores resultados en el juego.

6.1.3. Conformación del equipo

Cada uno de los puntos discutidos anteriormente tiene influencia en el buen desempeño de los jugadores y por lo tanto del equipo. Sin embargo es importante integrarlos y utilizarlos sacando su mayor provecho. Se logró el objetivo final de conformar un equipo de fútbol robótico. Si bien es necesario mejorar el accionar conjunto del equipo y profundizar mejoras de algunos

movimientos y técnicas de juego para hacer de este un equipo competitivo se constituyó un equipo base sobre el cual construir en el futuro.

En lo que implica la implementación del equipo obtenido, se elaboró una estrategia de equipo definiendo roles particulares para cada agente y las acciones a realizar en diferentes momentos del desarrollo de un partido. Se profundizó en un algoritmo de seguimiento de la trayectoria de la pelota por parte del golero, el cual permite obtener un valor agregado en el desempeño del mismo durante un partido de fútbol. Se trabajó también en una representación básica del entorno por parte de los agentes, el cual permite agregar información del ambiente al momento de tomar una decisión.

Más allá que el proyecto se desarrolló para una plataforma particular y cumpliendo las reglas de RoboCup es destacable la utilidad que puede tener como referencia para trabajos futuros tanto para el área de robots humanoides como en el área de fútbol de robots.

6.2. Trabajo futuro

Dada la amplia cantidad de aspectos que involucra la elaboración de un equipo de fútbol de robots humanoides, es posible llevar a cabo una amplia variedad de trabajos en el futuro. A lo largo de este documento se expresan varios indicios de mejoras y trabajos a realizar que quedaron fuera del alcance de este proyecto. A continuación se presentan propuestas de mejoras y profundización de temas ya desarrollados así como ideas que parece interesante abordar en el transcurso de otros proyectos o trabajos relacionados.

6.2.1. Mejoras y profundización

Una posible mejora a tener en cuenta dando continuidad al proyecto es trabajar sobre los movimientos y habilidades básicas de los agentes. Un ejemplo destacable de esto podría ser aplicar un algoritmo evolutivo ¹ para la obtención de los movimientos, mejorando su desempeño y posibilitando una mejor adaptación de los mismos frente a cambios del hardware o cambios del entorno (como puede ser el cambio de la superficie de juego). También podría modificarse el movimiento atajar intentando cubrir mejor el arco o tirándose, siempre teniendo en cuenta que si el movimiento implica una caída no se podrá contar con el golero mientras este se recupera, dejando desprotegido el arco. A su vez como ejemplo de nuevo movimiento podría agregarse el de pase diferenciándolo del disparo dándole mayor precisión, mejorando el

¹Los algoritmos evolutivos son métodos de optimización y búsqueda de soluciones basados en los postulados de la evolución biológica.

direccionamiento y la potencia de tal forma que el balón llegue a los pies (o lo suficientemente cerca) del compañero receptor del pase.

Otro aspecto importante a considerar podría ser mejorar el hardware, esto sobre todo a nivel de cámaras y motores, puntos vitales para el correcto accionar de los agentes a la hora de desempeñarse de forma autónoma. Se podría evaluar la utilización de kits más completos y robustos donde el módulo de visión se encuentra integrado. E incluso considerar construir los robots como lo hacen algunos equipos. Sin duda mejorar la capacidad del módulo de visión permitiendo un mayor procesamiento de *frames* por unidad de tiempo y con un alcance mayor logrando reconocer objetos a distancias mayores.

Un área importante para profundizar y desarrollar en el equipo es la localización, contar con un sistema de localización sería un gran aporte para el buen rendimiento del equipo. Se podría disponer de un algoritmo de localización basado por ejemplo en filtro de partículas ², lo cual evitaría que los agentes almacenen información en su representación del mundo sobre los objetos fijos, ya sean arcos o postes laterales. En este punto, se cuenta con el proyecto de grado de la Universidad de la República SLAM: Construcción de mapas y localización simultánea[Fed] en el cual se implementó un algoritmo que resuelve la localización de un agente en un ambiente desconocido. Referente a este punto, la identificación de las líneas del campo de juego por parte del módulo de visión sería un punto que aportaría mayor información del entorno a estos algoritmos.

Finalmente, en lo que respecta al comportamiento de los robots, el mismo puede ser extendido permitiendo por ejemplo realizar pases entre agentes, o incorporar el despeje del balón por parte del golero, entre otros aspectos. Dicho comportamiento también puede ser mejorado al profundizar en el seguimiento de adversarios y compañeros de equipo, de manera que permitan tener una mejor noción del desarrollo del juego, aportando a la toma de decisiones del agente.

6.2.2. Ideas para el futuro

Dentro de los puntos que no fueron manejados en el proyecto debido al alcance del mismo, se encuentra la comunicación explícita de los robots mediante el estándar de comunicación inalámbrica ZigBee[Zig], para el cual ya se dispone de un módulo implementado dentro del proyecto de grado VisRobII[Gon]. De esta forma, los robots serían capaces de comunicarse la información sensada de su entorno, permitiendo así, lograr una mejor repre-

²El filtro de partículas es un método de Montecarlo empleado para estimar el estado de un sistema que cambia a lo largo del tiempo.

sentación del mundo e identificar de forma más fácil y rápida el estado en que se encuentra el juego.

A su vez sería interesante trabajar sobre algunas líneas de cooperación entre los agentes. Por ejemplo tener en cuenta a la hora de llevar a acabo una acción quién es el agente más apto, por ejemplo qué jugador se encuentra más cerca de la pelota para ir a buscarla o para cerrar el ataque contrario. Contar con la capacidad de cubrir el puesto de un compañero si no está disponible por alguna razón, abriendo el paso también al intercambio de roles y dándole valor a un equipo donde los agentes son capaces de cambiar de rol si el equipo así lo necesita. También se podría intentar aplicar algunas de las soluciones analizadas en el documento de estado del arte[Mar12] y/o explorar otras técnicas donde se logre que los agentes tomen sus decisiones como parte de un equipo y no independientemente.

Lógicamente sería muy interesante realizar pruebas del funcionamiento del equipo, con los agentes interactuando y compitiendo con otros equipos, y cumpliendo el objetivo deseado de participar en la RoboCup.

Bibliografía

- [Ala92] Alan K. Mackworth. On Seeing Robots. *University of British Columbia*, 1992.
- [Alv04] Alvaro Castromán y Ernesto Copello. *Fútbol de Robots Uruguayo para Torneos: el Equipo*. Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay, Mayo 2004.
- [Cla02] Claudia C. Castelo, Héctor R. Fassi y Flavio E. Scarpettini. *Tesis de Licenciatura - Fútbol de Robot: Revisión del Estado del Arte y Desarrollo del Equipo UBASot de Simulación*. Universidad de Buenos Aires, 2002.
- [Dam05] Damián Lezama y Alexander Sklar. *Informe Final - Proyecto Construcción de Robots Bipedos*. Universidad de la República Oriental del Uruguay, 2005.
- [Fac] Facultad de Ingeniería. Portal de la FIng. <http://www.fing.edu.uy/>, accedida Febrero 2013.
- [Fed] Federico Andrade y Martin Llofriu. Sitio web Proyecto de Grado SLAM. http://www.fing.edu.uy/~pgslam/wiki/index.php/P%C3%A1gina_principal, accedida Febrero 2013.
- [FIR] FIRA. Sitio web FIRA - Liga Simurosot. <http://www.fira.net/?mid=simurosot>, accedida Febrero 2013.
- [Gon] Gonzalo Gismero. Sitio web Proyecto de grado VisRobII. <http://www.fing.edu.uy/~pgvisrob/>, accedida Julio 2012.
- [Gus06] Gustavo Armagno, Facundo Benavides y Claudia Rostagnol. *Informe Final - Proyecto FIbRA*. Universidad de la República Oriental del Uruguay, 2006.

- [IBM] IBM. Sitio web Proyecto Deep Blue. <http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/deepblue/>, accedida Febrero 2013.
- [Joh02] John Anderson y Jacky Baltes. *Doraemon User's Guide V.2.0*. University of Manitoba, 2002.
- [LOH] LOHI Digital Arts. Imagen publicada. <http://lohidigitalarts.snappages.com>, accedida Noviembre 2012.
- [Mar12] Marcelo Baliero y Gerardo Pias. *Estado del Arte - Proyecto Fútbol de robots para la Liga Humanoide de RoboCup*. Universidad de la República Oriental del Uruguay, 2012.
- [Mob07] Hamid Mobalegh. *Embedded Vision module for Bioloid Quick Start*. Freie Universität, Berlin, 2007.
- [NAS] NASA. Sitio web Misión Mars Pathfinder. <http://mars.jpl.nasa.gov/MPF/index0.html>, accedida Febrero 2013.
- [Roba] RoboCup. Sitio web de RoboCup. <http://www.robocup.org/>, accedida Julio 2012.
- [Robb] Robotis. *Manual Bioloid Premium Kit*.
- [Robc] Robotis. RoboPlus e-Manual. http://support.robotis.com/en/software/roboplus_main.htm, accedida Julio 2012.
- [Robd] Robotis. Sitio web de Robotis. <http://www.robotis.com>, accedida Julio 2012.
- [Rob06] Robotis. *User's Manual Dynamixel AX-12*, 2006.
- [Rob11] RoboCup. *RoboCup Soccer Humanoid League Rules and Setup*, 2011.
- [Ser06a] Serrana Casella, Pablo Rodríguez y Raúl Canales. *Estado del Arte - Proyecto Forrest*. Universidad de la República Oriental del Uruguay, 2006.
- [Ser06b] Serrana Casella, Pablo Rodríguez y Raúl Canales. *Informe Final - Proyecto Forrest*. Universidad de la República Oriental del Uruguay, 2006.

- [Stu04] Stuart J. Russell, Peter Norvig. *Inteligencia Artificial: Un enfoque moderno*. Number ISBN: 84-205-4003-X. Segunda edición. edition, 2004.
- [Unia] Universidad de Buenos Aires. Sitio web de la UBA Universidad de Buenos Aires. <http://www.uba.ar/>, accedida Febrero 2013.
- [Unib] Universidad de la República Oriental del Uruguay. Portal de la Ude-laR. <http://www.universidad.edu.uy/>, accedida Febrero 2013.
- [Zig] ZigBee Alliance. Sitio web ZigBee Alliance. <http://www.zigbee.org/Home.aspx>, accedida Febrero 2013.

Apéndice A

Glosario

- **Agente.** Cualquier cosa capaz de percibir su medioambiente con la ayuda de sensores y actuar en este medio utilizando actuadores [Stu04].
- **BIOLOID.** Kit robótico [Robb].
- **Entorno.** Contexto en el cual el robot desarrolla su actividad.
- **Estrategia.** Conjunto de acciones planificadas por el equipo para lograr una determinada finalidad durante el transcurso del partido.
- **HaViMo.** Módulo de visión diseñado para la plataforma Bioloid [Mob07].
- **KidSize.** Categoría de fútbol de robots cuya altura varía entre 30 cm y 60 cm de alto, de la Liga Humanoide de RoboCup [Rob11].
- **Movimiento.** Acción básica que culmina con el agente en la postura corporal erguida.
- **Pose.** Postura que toma el robot al ubicar cada uno de sus motores en una posición determinada.
- **RoboCup.** Conjunto de competencias internacionales integradas por robots autónomos [Roba].
- **Rol.** Función que cumple un robot en el marco de la actividad en la cual se desempeña.
- **Sistema Multi-agente.** Sistema compuesto por múltiples agentes racionales que interactúan entre ellos.
- **Submovimiento.** Acción que define una etapa particular de un determinado movimiento.