

Cooperación en Sistemas Distribuidos de Robots Reactivos Minimizando la Cantidad de Información Comunicada

Fernando Fernández, Germán Gutiérrez y José Manuel Molina



Universidad Carlos III de Madrid.
Avda. de la Universidad, 30.28911 Leganés, Madrid, Spain,
ffernand@grial.uc3m.es, gutierr@inf.uc3m.es, molina@ia.uc3m.es

Resumen

La coordinación emergente pretende obtener comportamientos colaborativos entre diversos agentes sin que eso implique que cada individuo deba tener un conocimiento global del dominio, y sin que ese conocimiento deba estar centralizado. Al no requerir conocimiento global, se minimiza la comunicación entre los agentes de forma que cada uno de ellos puede comportarse de forma reactiva y totalmente autónoma. En este trabajo se presenta una primera aproximación a este modelo de coordinación aplicado al dominio de la *RoboCup*.

Palabras clave: Sistemas multiagente, comportamientos cooperativos, comportamientos reactivos, *RoboCup*.

1. Introducción

Un sistema compuesto por varios robots es por propia definición un sistema distribuido [Durfee *et al*, 1989] ya que existe una distribución: (a) espacial, ya que cada uno de los robots se encuentra en una situación física distinta, (b) de información, ya que la información es interna a cada robot, de manera que no toda la información es común a todos los robots y (c) de toma de decisiones, ya que la decisión última sobre la tarea a realizar es siempre interna al robot que en función de su situación interna ejecuta las tareas. Las decisiones que involucran a todo el sistema no se toman en un único punto sino que aparecen como agregación de todas las decisiones locales a los robots [Bond and Gasser]. El proceso de toma de decisiones en el sistema es tanto más distribuido cuanto mayor es la capacidad de decisión existente en cada robot del sistema, desde organizaciones centralizadas a descentralizadas.

Las arquitecturas de gestión que se pueden desarrollar sobre un conjunto de robots se diferencian básicamente en la capacidad de control interna de cada robot. No existe una diferenciación exacta entre arquitectura centralizada y descentralizada, ya que el paso de una a otra es gradual a medida que se dota al robot de mayor autonomía [Kamel and Ghenniwa]. En general, se pueden definir los sistemas centralizados como aquellos en los que las decisiones que afectan a todo el sistema se toman en un único nodo (robot coordinador) y los descentralizados como aquellos donde estas se toman entre todos los nodos (todos los robots son igual de importantes) sin que exista una supeditación de unos nodos a otros.

En los sistemas centralizados uno de los robots es el decisor de todo el grupo, las decisiones fluyen desde él al resto de robots y son aceptadas en el proceso de decisión de cada robot.

A medida que se permite a los robots una mayor capacidad de autonomía y por lo tanto se les dota de una mayor capacidad para la toma de decisiones, resulta menos necesario centralizar la toma de decisiones del sistema en un punto [Molina *et al*, 1996a]. Esta autonomía redundante en la arquitectura de gestión del conjunto, al permitir que las decisiones se tomen mediante la puesta en común de decisiones locales. Este es un problema asimilable a los problemas de sistemas multiagentes cooperantes [Wesson *et al*, 1988].

Para la concepción de esquemas de gestión de robots autónomos los sistemas multiagente (SM) resultan apropiados [Molina *et al.*, 1996a]. En general los SM tratan sobre la coordinación inteligente entre una colección de 'agentes' autónomos, cómo pueden coordinar sus conocimientos, metas, propiedades y planes para tomar una decisión o resolver un problema [Bond and Gasser, 1998].

El objetivo central de este trabajo es el desarrollo de un sistema de gestión que permita coordinar un conjunto de robots, donde cada robot presente un alto grado de autonomía, es decir, que la decisión última sobre las tareas a realizar sea tomada en su propio sistema de gestión. El proceso de coordinación debe organizar el flujo de informaciones de tal manera que la comunicación entre los distintos robots se traduzca en una optimización en el uso global de los recursos. La arquitectura que da soporte al proceso de coordinación hará uso de técnicas de Inteligencia Artificial Distribuida [Decker, 1987], en particular de la teoría de multiagentes [Luxa and Esteren, 1995]. El problema al que nos enfrentamos se ve agravado por el hecho de necesitar estructuras de control en los robots relativamente simples basadas en la información accesible por los sensores en ese instante de decisión. En resumen se va a presentar una estructura de gestión de robots basada en técnicas multiagente que permite coordinar un conjunto de robots controlados de forma reactiva de manera que el comportamiento global del sistema sea coherente sin necesitar de un decisor global.

En este trabajo se presenta una primera aproximación a estas ideas dentro del dominio de la *RoboCup*. La *Robot World Cup Initiative* [Gitano et al., 1995] es una iniciativa para desarrollar el estudio de la Inteligencia Artificial y los robots inteligentes proponiendo un problema estándar donde un amplio rango de tecnologías pueden ser integradas y desarrolladas.

En la siguiente sección se presenta en más profundidad el dominio de la *RoboCup* y el simulador *Soccer Server*. En la sección 3 se presenta la arquitectura y el comportamiento de los agentes de esta propuesta, así como el método utilizado para obtener comportamientos colaborativos. En la sección 4 se muestran los primeros resultados de su aplicación en el dominio de la *RoboCup*. Por último, en la sección 5 se describen las principales conclusiones y posibles trabajos futuros.

2. Entorno de Experimentación: La RoboCup

La *RoboCup* plantea un problema estándar consistente en dos equipos de robots que juegan al fútbol. Entre las tecnologías que pueden ser integradas en estos dos equipos están: el diseño de agentes autónomos, colaboración multiagente, adquisición de estrategias, razonamiento en tiempo real, etc.

A pesar de que el verdadero objetivo de *RoboCup* es realizar una copa mundial entre robots reales, ofrece un soporte software en el que se pueden estudiar aspectos relacionados con *RoboCup*: el simulador *Soccer Server*. *Soccer Server Simulator* [Noda 1995] es un sistema que permite que dos equipos implementados en diversos lenguajes de programación jueguen al fútbol entre ellos. Para ello, se utiliza una arquitectura cliente-servidor.

La información visual que los agentes reciben del entorno es variada. Pueden recibir información sobre la posición del balón, de los jugadores, y sobre marcas situadas por el campo que dan idea a los jugadores de dónde están situados los diversos elementos del entorno (como las porterías y el centro del campo). Toda esta información que reciben los agentes es subjetiva, es decir, relativa a su propia posición. Esto significa que un agente no recibe nunca un mensaje visual de que, por ejemplo, el balón está en la posición x e y del campo, sino que el balón está a una distancia d de él, y que le está "viendo" con un ángulo a . Si se añade a esto que la información que se recibe del simulador, se recibe con un ruido proporcional a la distancia de los objetos, se obtiene una de las principales dificultades del dominio consistente en obtener una representación completa de la realidad.

Además, las acciones que ejecutan los agentes no son deterministas, lo que quiere decir que un agente que se encuentra en un determinado estado en el juego, si ejecuta una acción aislada, puede llegar a estados distintos en diferentes ejecuciones de la acción.

Hasta ahora, los principales trabajos que se han desarrollado sobre la RoboCup, en lo que a la liga de simulación se refiere, han estado muy orientados al desarrollo de sistemas de control de agentes autónomos en sistemas multiagente, que además permita la colaboración entre ellos [Iso and Inazumi, 1997, Girault and Stinckwich, 1999]. Entre estos, cabe destacar el modelo ABC^2 [Matellán, 1998, Matellán *et al.*, 1998], que se rige por un sistema oportunista de elección de actos a realizar en cada momento. También son de gran

¹ <http://ci.etl.go.jp/~noda/soccer/server/index.html>

importancia las aportaciones sobre los sistemas de control en tiempo real, como el descrito en el modelo desarrollado por [Zhang and Mackworth, 1997], y los sistemas de aprendizaje de habilidades de alto y bajo nivel [Stone, 1999, Fernández and Borrajo 1999].

3. Cooperación Basada en Controladores Reactivos

La arquitectura de cada robot estará basada en comportamientos reactivos [Brooks, 1986, Brooks, 1991]. Entre los diferentes tipos de arquitecturas de control propuestas a lo largo de los últimos años existen dos perspectivas mayoritarias: aquellas que mantienen la posibilidad de alcanzar la consecución de objetivos complejos basándose únicamente en un conjunto de reacciones simples, denominense instintos, módulos, agentes, agencias, controladores o comportamientos, y aquellas que necesitan tener un modelo interno del mundo sobre el que razonar. La segunda de estas arquitecturas obliga a una representación fehaciente del mundo. Únicamente, a partir de resultados intermedios de razonamiento sobre dicha representación será posible la resolución de problemas complejos.

En ambos casos es necesario de tener un conjunto de acciones básicas que permitan al robot interactuar con el entorno en el que se encuentra inmerso. A la hora de integrar todas las acciones básicas para dar respuesta a problemas complejos el investigador debe decidir si en realidad deben o no existir niveles jerárquicos en la estructura de control [Matellán *et al*, 1995]. Para evaluar esta necesidad es necesario comprender las distintas clases de representación del mundo haciendo especial hincapié en la aplicación última del robot desarrollado. Cuando el entorno en que se desarrollan las pruebas está perfectamente definido (bien porque sea una simulación bien porque el entorno es constante y definido a priori) se puede obtener una alta precisión en la representación interna del mundo sobre la que trabajará el robot, pero si el entorno es real y no constante se debe proveer al robot de la capacidad de generar su propia representación a partir de los valores que obtiene de los sensores. Este problema no es fácilmente abordable debido a los problemas de ruido en la medida e indeterminación de las posiciones por las que va pasando el robot, en estos casos la representación del entorno es en muchos casos aproximada y el razonamiento sobre ella no produce los resultados que se obtienen en simulación.

Por todo esto las arquitecturas reactivas donde no existe una representación del mundo resultan más robustas. La robustez de estas propuestas presenta la limitación en la capacidad de razonamiento de las mismas. Para coordinar un conjunto de robots reactivos se necesita o bien una gestión centralizada que, en todo momento, sepa que debe hacer cada robot y que le envíe una tarea concreta, o bien que la cooperación aparezca de manera emergente.

Los comportamientos cooperativos emergentes permiten la resolución de problemas de forma robusta pero no permiten conocer el conjunto de decisiones concretas que han llevado a esa solución en particular. El grado de indefinición (estocástica) introducido por la emergencia del comportamiento cooperativo no resulta asumible en muchas aplicaciones, como por ejemplo en el caso del fútbol. En este caso se puede definir una estrategia de juego conjunto, es decir no es emergente (ni se desea que lo sea) basada en estrategias individuales, donde cada estrategia individual debe estar basada en lo que sucede en el entorno en ese momento (debe ser reactiva).

3.1 Comportamiento de los agentes

El comportamiento de los agentes es totalmente reactivo. Este comportamiento se resume en mantenerse colocado en una zona del campo relativamente amplia, tal y como se muestra en la figura 1.

Estas zonas pueden estar definidas a priori, o variar según distintas situaciones de juego. Para mantener la coherencia del equipo, sin mantener información global del resto de los agentes, se establece para cada agente un líder (*mi_lider*) al que debe seguir en todo momento a una distancia suficientemente grande como para mantener la formación de equipo, y suficientemente pequeña como para que esté siempre en su rango de visión. De esta forma, el comportamiento reactivo de cada agente que le sitúa en una posición adecuada dentro del campo viene definida por la regla dada en la tabla 1.



Figura 1. Posible definición de las zonas de los jugadores para una disposición 4-3-3.

En principio, este es el controlador mínimo necesario para mantener a todos los jugadores en formación dentro del campo, e incluso para que se muevan en función de los movimientos de otros jugadores. Como se puede observar, para que este controlador sea ejecutable, cada agente sólo necesita mantener información sobre sí mismo (dónde está situado en el campo) y dónde está su líder particular, al que siempre debería mantener bajo su rango de visión (habría que definir también controladores que permitan buscar al líder, en caso de que el agente se pierda).

Controlador: Mantener_Posición
SI (cerca(mi_lider)=CIERTO) ENTONCES IR(centro_zona) SINO IR(Posicion_lider)

Tabla 1. Controlador Mantener Posición. Mantiene al agente situado en su zona a una distancia "prudencial" de su líder local

3.2 Coordinación y comunicación

Para la coordinación de los agentes entra en juego el único conocimiento global que es transmitido por los agentes: el líder global. Se define el líder global del equipo como aquél jugador que posee el balón o que tiene más posibilidades de poseerlo. En cada momento sólo debe existir un líder global de todo el equipo, y la información sobre quién es tal líder debe ser recibida por todos los jugadores. Así, una vez que cada agente sabe quién es el líder global, puede calcular quién es su líder local. En la figura 1 se muestra un equipo en formación. El líder del equipo es el jugador número 11. La flecha que sale de cada jugador indica quién es su líder local.

El cálculo del líder local de cada jugador a partir del líder global del equipo puede ser calculado mediante heurísticas. Una heurística sencilla viene dada en la tabla 2.

Jugador/Líder_Global	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	1	2	3	3	3	6	6	6	6	6	6
3	1	2	3	4	4	7	7	7	7	7	7
4	1	3	3	5	5	7	7	7	7	7	7
5	1	4	4	5	5	8	8	8	8	8	8
6	2	2	2	2	2	6	7	7	9	9	9
7	3	3	3	3	3	6	7	8	10	10	10
8	5	5	5	5	5	7	7	8	11	11	11
9	6	6	6	6	6	6	6	6	9	10	10
10	7	7	7	7	7	7	7	7	9	10	11
11	8	8	8	8	8	8	8	8	10	10	11

Tabla 2. Heurística que define los líderes locales de cada jugador a partir del líder global.

En dicha tabla se muestra para cada jugador, quién es el líder local en función del líder global. Así, si suponemos que el líder global es el jugador número 11, de la última columna se obtiene que para los jugadores 8, 10 y 11, el líder local coincide con el líder global; para los jugadores 7 y 9, el líder local es el 10; para el 6, el 9; etc. Como se puede observar, la situación mostrada en la figura 1 es obtenida mediante esta heurística. Cabe destacar, que el portero (jugador número 1) tiene un comportamiento independiente.

El último punto que queda por definir es quién es el líder del equipo en cada momento.

Controlador: Quitar_Líder(num)
SI (muy_cerca(bola)=CIERTO) ENTONCES Cambiar_Líder(num)
<i>Controlador: Ceder_Líder(num)</i>
SI (he_pasado(num)=CIERTO) ENTONCES Cambiar_Líder(num)

Tabla 3. Controladores Quitar Líder y Ceder líder. Permiten cambiar el líder local del equipo.

El controlador Quitar_Líder(num) (ver tabla 3) permite al jugador *num* quitarle el liderazgo al líder que hay en ese momento. Por contra, el controlador Ceder_Líder(num) permite ceder el liderazgo a un jugador al que acabo de pasar el balón. En ambas situaciones, el cambio de líder debe incluir el único proceso de comunicación entre los agentes que hay prevista en esta arquitectura, que es el de transmisión de quién es el líder en cada momento. Para asegurar que todos los agentes conocen al líder global (coherencia del equipo) este conocimiento es transmitido de unos agentes a otros: es decir, cada vez que un agente advierte un cambio de líder, retransmite el mensaje de cambio de líder.

La figura 2 muestra una primera aproximación de cómo se unen todos los controladores reactivos para cada agente. Como se puede comprobar, hay un primer bloque de acciones que deben ejecutar todos los agentes: verificar quién es el líder global, para así obtener su líder local y su zona asociada, y verificar si el jugador está bloqueado. El segundo bloque discrimina el comportamiento del agente que hace de líder del equipo (tiene prioridad para actuar sobre el balón) y de aquellos que no son líder y tienen acciones dirigidas a mantener la coherencia y el orden dentro del equipo.

4. Resultados de Juego

En la figura 3 se muestran algunas instantáneas de diferentes situaciones de juego durante una simulación del equipo realizada sin contrarios. Todos los jugadores a excepción del portero son iguales, es decir, los mismos controladores (las mismas reglas) y las mismas habilidades para cada agente.

Como se puede ver en las instantáneas cada jugador tiene una zona del campo asignada, también se indica, con una flecha, quien es el líder particular ("mi_lider ") para cada uno de los jugadores. El líder del equipo en cada caso es aquel del que no sale ninguna flecha. En una primera aproximación a las zonas dinámicas, la zona del campo dentro de la cual el jugador trata de mantenerse depende de la línea a la que

pertenezca el líder del equipo: línea de defensas, de medios o delanteros. Por ejemplo si el líder del equipo es un delantero (jugadores 9, 10 y 11) las zonas asignadas abarcan principalmente el campo contrario como en la instantánea c) donde el líder es el jugador numero 9. En las instantáneas a) y b) los líderes del equipo son respectivamente el número 2 (defensa) y el numero 8 (medio). Si el líder es un defensa (jugadores 2, 3, 4 y 5) las zonas están distribuidas de modo que la formación cubre el campo propio, y en el caso de que el líder sea un medio (jugadores 6, 7,y 8) se cubre la parte central del terreno de juego.

Con las zonas dinámicas se pretende una formación más compacta, en la que el equipo no tiene por qué abarcar todo el campo sino sólo aquella parte que conviene en cada instante y según la estrategia de juego. Así los jugadores se pueden mantener más agrupados.

El líder del equipo buscará e irá hacia la pelota, o avanzará con ella hasta que salga de su zona asignada. En ese momento buscará algún compañero y si lo encuentra le pasara la pelota, convirtiéndose este último en el único líder del equipo. Si no, seguirá avanzando con ella hacia la portería, llegando a salir de su zona . El resto de los jugadores, si se encuentran muy cerca de la pelota intentarán obtener el liderato; si no, mantendrán su posición dentro de la formación siguiendo a su líder particular ("mi_lider") o situándose en el centro de la zona si es que no tiene información acerca de él. En el momento que un agente detecta un cambio de líder recalcula su zona asignada.

Como se puede ver en la primera instantánea el jugador numero 11 está fuera de su zona asignada. Esta situación es posible porque previamente fue líder y se esta situando correctamente. La disposición mostrada en la simulación es una "4-3-3", pero cambiar a cualquier otra estrategia, como "4-4-2" o "5-3-2" por ejemplo, es sumamente sencillo. Además modificando las zonas, el tamaño, la disposición, etc., se podría obtener distintos comportamientos como grupo, más conservadores, más ofensivos, etc.

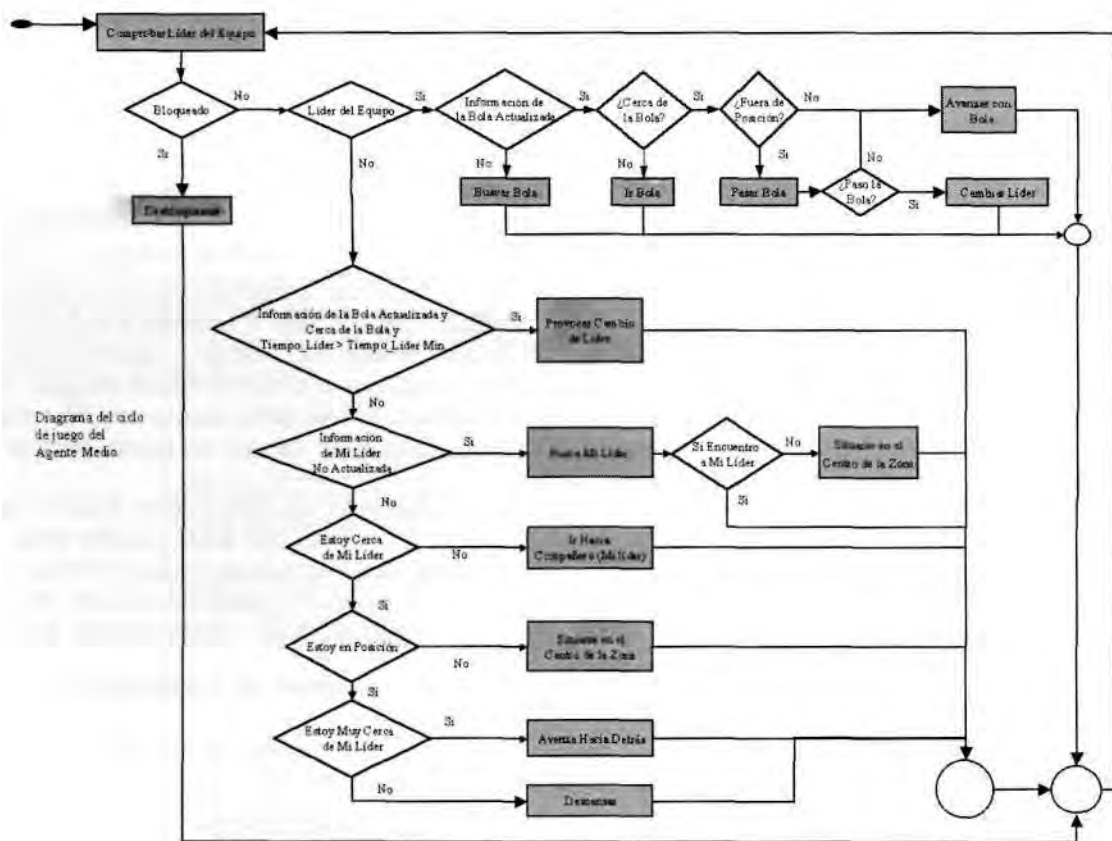


Figura 2. Ciclo de Ejecución de un Agente.

5. Conclusiones

Esta arquitectura está abierta a comportamientos y habilidades muy amplias dentro del dominio de la RoboCup, y que pueden ser fácilmente introducidas en esta arquitectura. A continuación se describen algunos de ellos.

Zonas dinámicas. Las zonas definidas para cada agente pueden ser dinámicas y dependientes de parámetros tales como quién es el líder, si se está atacando o defendiendo, de la propia estrategia de juego, etc.

Cambio de Estrategias de Juego. Dado que las zonas son dinámicas, se pueden establecer distintas estrategias (ofensiva, defensiva) y/o formaciones (5-3-2, 4-4-2, etc.) de juego, simplemente variando las zonas y la heurística de cálculo de líderes locales.

Cambio de Roles. El papel del agente en el campo puede ser variado si los números (o dorsales) de los agentes y su papel en el campo es una relación que varía en el tiempo y que debe ser informada al resto de agentes. Dado que la cantidad de información intercambiada entre los agentes para mantener el conocimiento del entorno es mínima, se puede explotar la capacidad de comunicación para estas habilidades.

Adaptación a las Situaciones de Juego. Las zonas de los agentes, la heurística de quién es el líder, etc. puede variar también en función de la situación de juego: si aun no ha empezado el partido, si hay un saque de falta, etc.

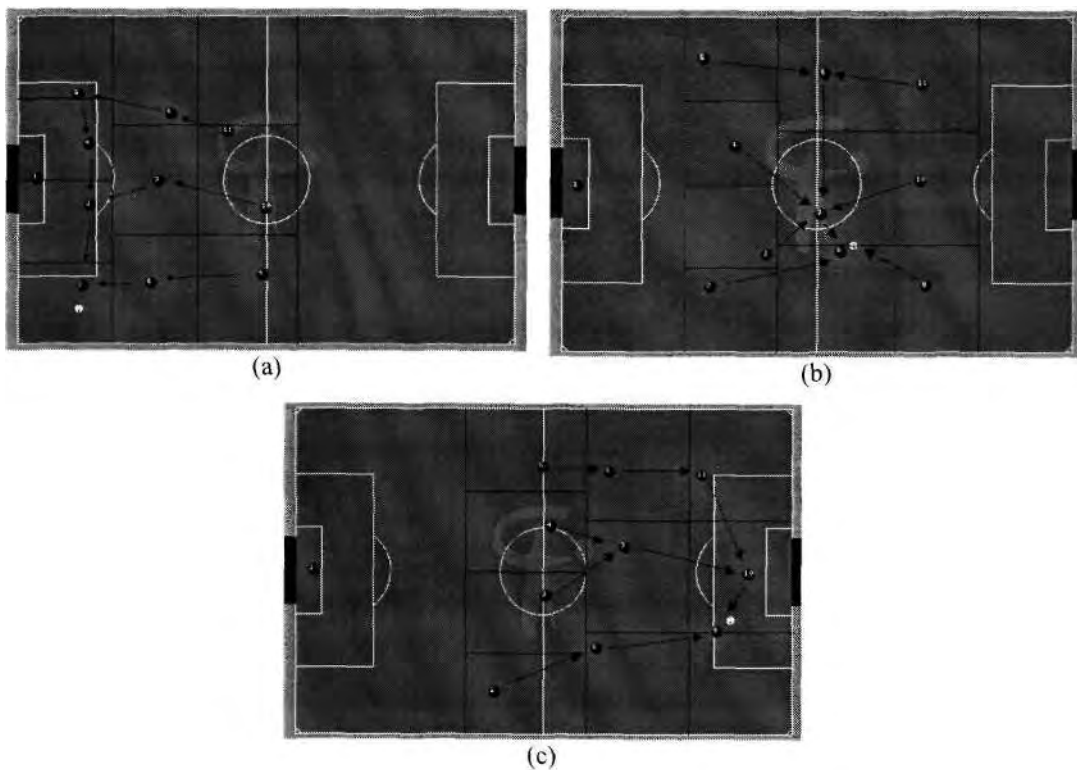


Figura 3. Distintas instantáneas de juego. (a) muestra una situación de juego defensiva. (b) muestra un control del medio campo. (c) muestra una situación de juego ofensiva.

Desarrollo de las Habilidades Individuales. Los controladores son totalmente independientes, por lo que pueden implementarse controladores individuales pre-programadas o aprendidas [Fernández and Borrajo, 1999].

Refinamiento de las Heurísticas de definición de líderes. Las heurísticas de definición de líderes locales a partir del líder global pueden ser ampliadas, mejoradas o aprendidas incluyendo otros parámetros, como la línea de juego del líder (si es delantero, centro, etc.), si el equipo está atacando o no, etc.

Otro punto importante es el refinamiento del comportamiento global del equipo a partir del refinamiento del comportamiento individual de cada agente, es decir, el controlador reactivo. Para ello, es imprescindible además la experimentación con otros equipos en un entorno competitivo.

Referencias

- [Bond and Gasser, 1988] Alan H. Bond and Les Gasser. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann, 1988.
- [Brooks, 1986] Rodney A. Brooks. A Roboust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1) : 14-23, March 1986.
- [Brooks, 1991] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47 :139-160, 1991.
- [Deker 1987] Keirh A. Brooks. Distributed problem-solving techniques: A survey. *IEEE Transactions on Systems, Man and Cybernetics*, Sept/Oct 1987.
- [Durfee *et al.*, 1989] E.H. Durfee, V. R. Lesser, and D. D. Corkill. *The Hand Book of AI*, chapter Cooperative Distributed Problem Soving. Addison-Wesley, 1989.
- [Fernández and Borrajo, 1999] Fernando Fernández and Daniel Borrajo. Vector quantization applied to reinforcement learning. In Manuela M. Veloso, editor, *Proceedings of the Third International Workshop on Robocup*, pages 97-102, Stockholm, Sweden, August 1999. IJCAI'99.
- [Girault and Stinckwich, 1999] François Girault and Serge Stinckwich. Footux team description: A hybrid recursive based agent architecture. In Manuela M. Veloso, editor, *Proceedings of the Third International Workshop on Robocup*, pages 103-108, Stockholm, Sweden, August 1999. IJCAI'99.
- [Iso and Inazumi, 1997] Ransui Iso and Hiroshige Inazuni. A multi-layered planning architecture for soccer agent. In Hirokai Kitano, editor, *Robocup-97: Robot Soccer World Cup I*, pages 513-518, 1997.
- [Kamel and Ghenniwa, 1994] M.Kamel and H. Ghenniwa. "Coordination of distributed intelligence systems". In Aminzadeh and Jamshidi, editors, *Soft Computing*. Prentice Hall, 1994.
- [Kitano *et al.*, 1995] Hirokai Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. Robocup: The robot world cup initiative. In *Proceedings of the IJCAI-95 Workshop on Learning Robots*, pages 19-24, December 1995.
- [Lux and Steiner, 1995] A.Lux and D.Steiner. Understanding cooperation: an agent's perspective. In *Proc. International Conference on Multiagents Systems, ICMAS '95*. AAAI Press., June 1995.
- [Matellán *et al.*, 1995] Vicente Matellán, Daniel Borrajo, and Camino Fernández. Fusion of fuzzy behaviors for autonomous robots. In *Proceedings of the Third International Symposium on Intelligent Robotic System*, Pisa, Italia, 1995.
- [Matellán, 1998] V. Matellán. ABC²: *Un Modelo para el Control de Robots Autónomos*. PhD thesis, Universidad Politécnica de Madrid, 1998.
- [Matellán *et al.*, 1998] Vicente Matellán, Daniel Borrajo, and Camino Fernández. Using ABC² in the robocup domain . In Hirokai [Citano, editor, *Robocup-97: Robot Soccer World Cup I*, Lecture Notes in Artificial Intelligence, pages 475-483. Springer Verlag, 1998.
- [Molina *et al.*, 1996a] J.M. Molina, V. Matellán, C. Fernández. Distributed fuzzy decisión making systems. In *Proc. International Conference on Intelligent Technologies in Human-Related Sciences*, volume II, pages 191-196, León, Spain, 1996.
- [Molina *et al.*, 1996b] J.M. Molina, V. Matellán, and L. Sommaruga. Fuzzy multi agent interaction. I *Proc. IEEE International Conference on Systems, Man and Cybernetics*, volume III, pages 1950-1955, Beijing, China, 1996.
- [Noda, 1995] Itsuki Noda. Soccer server: a simulator of robocup. In *Proceedings of the Fourth International Symposium'95*, December 1995.
- [Stone, 1999] Peter Stone. *Layered Learning in Multi-Agent Systems*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1999.
- [Wesson *et al.*, 1988] R. Wesson. *Readings in Distributed Artificial Intelligence*, chapter Network Structures for Distributed Situation Assessment.
- [Zhang and Mackworth, 1997] Yu Zhang and Alan K. Mackworth. Using reactive deliberation for real-time control of soccer-playing robots. In Hirokai Kitano, editor, *Robocup-97: Robot Soccer World Cup I*, pages 508-512, 1997.