

# Coordinación global basada en controladores locales reactivos en la *RoboCup*



Fernando Fernández, Germán Gutiérrez y Jose Manuel Molina  
Universidad Carlos III de Madrid. Avda. de la Universidad, 30 • 28911 Leganés, Madrid  
Correo-e: [ffernand@grial.uc3m.es](mailto:ffernand@grial.uc3m.es), [g Gutierr@inf.uc3m.es](mailto:g Gutierr@inf.uc3m.es), [molina@ia.uc3m.es](mailto:molina@ia.uc3m.es)

## Resumen

El principal problema a que nos enfrentamos al diseñar sistemas multi-agente es cómo coordinar los agentes que pertenecen a este sistema para obtener un comportamiento global eficiente. En algunos trabajos, el comportamiento coordinado de los agentes se obtiene gracias a un conocimiento total del dominio en el que se pueden aplicar planificadores tradicionales. Sin embargo, en muchos dominios esta aproximación no es posible por tratarse de dominios demasiado grandes y la comunicación y capacidad de memoria de los agentes muy limitada. En este trabajo se presenta un diseño multi-agente que permite obtener un comportamiento global coordinado basado en los comportamientos reactivos de los agentes que son controlados o dirigidos por la información local que poseen. En el diseño se ha minimizado la comunicación entre los agentes, ya que el proceso de coordinación se basa en reglas locales reactivas. Esta propuesta tiene una primera aplicación en el dominio de la RoboCup (*Robot World Cup Initiative*), proyecto ampliamente utilizado por la comunidad investigadora.

Palabras Clave: Inteligencia artificial distribuida, sistemas multi-agente, comportamientos cooperativos, comportamientos reactivos, RoboCup.

## 1. Introducción

La mayoría de sistemas inteligentes distribuidos [3] [1] puede clasificarse principalmente en dos ramas. En la primera se incluyen aquellos que basan su comportamiento en un conocimiento total del dominio. Esta aproximación tiene como principal inconveniente que la obtención de dicho conocimiento no es posible en todos los entornos. En la mayoría de entornos reales, el conocimiento del dominio suele estar limitado por la percepción del mundo real que reciben los agentes, ya que ésta es inexacta e incompleta. Por otro lado, en un sistema multiagente, la cantidad de información necesaria para representar todo el dominio (la unión del conocimiento de

todos los agentes) puede ser demasiado grande como para que sea comunicada y contenida en un agente central.

La segunda rama incluye los sistemas que trabajan con un conocimiento local del dominio [2]. Es decir, limitan su conocimiento a la información captada por los sensores. Ésta suele ser la situación *a priori* de cualquier agente en cualquier sistema en el que los agentes obtienen información local del entorno mediante sensores. El paso de un conocimiento local a un conocimiento global del dominio por parte de los agentes suele resolverse mediante la comunicación entre dichos agentes, es decir, mediante un intercambio de información que a veces implica un coste demasiado alto, ya sea en tiempo o en recursos del sistema.

El proceso de toma de decisiones en un sistema es tanto más distribuido cuanto mayor es la capacidad de decisión de cada agente del sistema, desde organizaciones centralizadas a descentralizadas. No existe una diferenciación exacta entre ambas arquitecturas, ya que el paso de una a otra es gradual a medida que se dota al agente de mayor autonomía. 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 ser requerido un 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 [7].

La *Robot World Cup Initiative* (RoboCup) [8] es una iniciativa para desarrollar el estudio de la Inteligencia Artificial y los robots inteligentes que plantea un problema estándar donde un amplio rango de tecnologías pueden ser integradas y desarrolladas. Este problema estándar consiste 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, la colaboración multi-agente, la adquisición de estrategias, el razonamiento en tiempo real, etc.

Hasta ahora, los principales trabajos que se han desarrollado sobre la RoboCup, por lo que respecta a la liga de simulación, han estado muy orientados al desarrollo de sistemas de control de agentes autónomos en sistemas multi-agente, que además permitan la colaboración entre

ellos [6] [5]. Entre éstos cabe destacar el modelo ABC<sup>2</sup> [9] [5], que se rige por un sistema oportunista de elección de los actos que deben ser realizados en cada momento.

Sin embargo, es obvio que el aprendizaje de habilidades de bajo o alto nivel es un elemento imprescindible para el buen funcionamiento de sistemas de comportamiento como los nombrados anteriormente. En este sentido, hay también una gran cantidad de trabajos, como el desarrollado por Stone [12], que introduce el aprendizaje por niveles, es decir, el aprendizaje en las distintas capas del plan expuesto en [4], el modelo VQQL, que permite el aprendizaje de habilidades en dominios continuos e indeterministas.

También son de gran importancia las aportaciones que sobre los sistemas de control en tiempo real han llevado a cabo Zhang y Mackworth [13]. En el modelo desarrollado en [13] se integran acciones reactivas al control en tiempo real.

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. La RoboCup**

La primera RoboCup se celebró en el año 1997 en Nagoya (Japón) y contó con la participación de más de 40 equipos. La segunda, Robocup-98, se celebró en París en julio de 1998 y participaron en ella más de 60 equipos. La tercera, Robocup-99, tuvo lugar en Estocolmo en julio-agosto de 1999.

A pesar de que el verdadero objetivo de RoboCup es disputar una copa mundial de fútbol entre robots reales, el proyecto ofrece un soporte software en el que se pueden estudiar aspectos relacionados con RoboCup: el simulador *Soccer Server*<sup>1</sup>. *Soccer Server Simulator* [11] 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.

El servidor proporciona un campo virtual y simula los movimientos de los jugadores y el balón. La comunicación entre el servidor y los clientes se realiza mediante *sockets* UDP/IP. Así, los clientes pueden ser implementados en cualquier arquitectura que permita comunicaciones de este tipo. Este servidor está compuesto por dos módulos principales: el servidor propiamente dicho, que establece las comunicaciones con los clientes y controla el juego, y un monitor que se encarga de generar la representación gráfica de lo que ocurre en el servidor y que, por tanto, muestra un campo de fútbol con los agentes disputando un partido.

Los clientes son los "cerebros" de los jugadores, controlan sus movimientos de forma que cada uno de ellos dirigen a un jugador. Mediante *sockets* se recibe información sensorial (visual, auditiva y de estado) desde el servidor, y también a través de *sockets* se envían las órdenes, o lo que es lo mismo, las acciones que se quieren ejecutar sobre el entorno, incluida la comunicación con el resto de los agentes.

La información visual que los agentes reciben del entorno es variada. Pueden recibir información sobre la posición del balón, sobre la posición de los jugadores, y sobre marcas situadas en el campo que informan a los jugadores de donde 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 que le informe, por ejemplo, de que el balón está en la posición  $x$  e  $y$  del campo, sino que el balón está a una distancia  $d$  de él, y que lo está "*viendo*" con un ángulo  $a$ . Si se añade a esto que la información que llega del simulador, se recibe con un ruido proporcional a la distancia de los objetos, nos hallamos ante 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. Este indeterminismo se debe a que sobre las acciones de los jugadores también se ejerce un ruido que produce aleatoriedad en las acciones.

## 2. Coordinación global de agentes basada en información local en la RoboCup

Tenemos un sistema multi-agente autónomo en el que hay un grupo de agentes que colaboran para conseguir un objetivo común, y otro grupo de agentes con un objetivo opuesto al del primer grupo. La pertenencia a un grupo u otro (es decir, qué agentes colaboran y cuáles se enfrentan) es conocida *a priori*.

### 2.1 Arquitectura

Los módulos esenciales de los que consta la arquitectura de un agente que permita una coordinación global con otros agentes utilizando únicamente controladores locales son:

- Comunicación. Módulo que permite la comunicación del agente con el exterior. Así, un agente recibe información sensorial del entorno y realiza acciones dentro de él. La comunicación con otros agentes queda limitada a acciones que se pueden realizar sobre el entorno, ya que se supone que se utiliza ese entorno como medio de transmisión.
- Controlador reactivo. Este controlador sustituye al planificador existente en otras arquitecturas, y es el encargado de lanzar las distintas habilidades o acciones según un número muy limitado de situaciones posibles. Esta limitación es la que permite un control reactivo sencillo.
- Habilidades o acciones. Conjunto de posibles acciones que puede ejecutar un agente. Con el fin de que el controlador reactivo sea sencillo, el número de habilidades también debe ser pequeño. Para compensar este pequeño número, las habilidades pueden ser tan complejas como se desee sin que deban limitarse a pequeñas acciones de bajo nivel.

### 2.2 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 conocer información global del resto de los agentes, se

establece para cada agente un líder (mi\_líder) 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 que viene definida por la regla dada en la Tabla 1. La idea del controlador es que si el agente está lejos de su líder se dirija hacia él, y si no, se dirija hacia el centro de su zona.

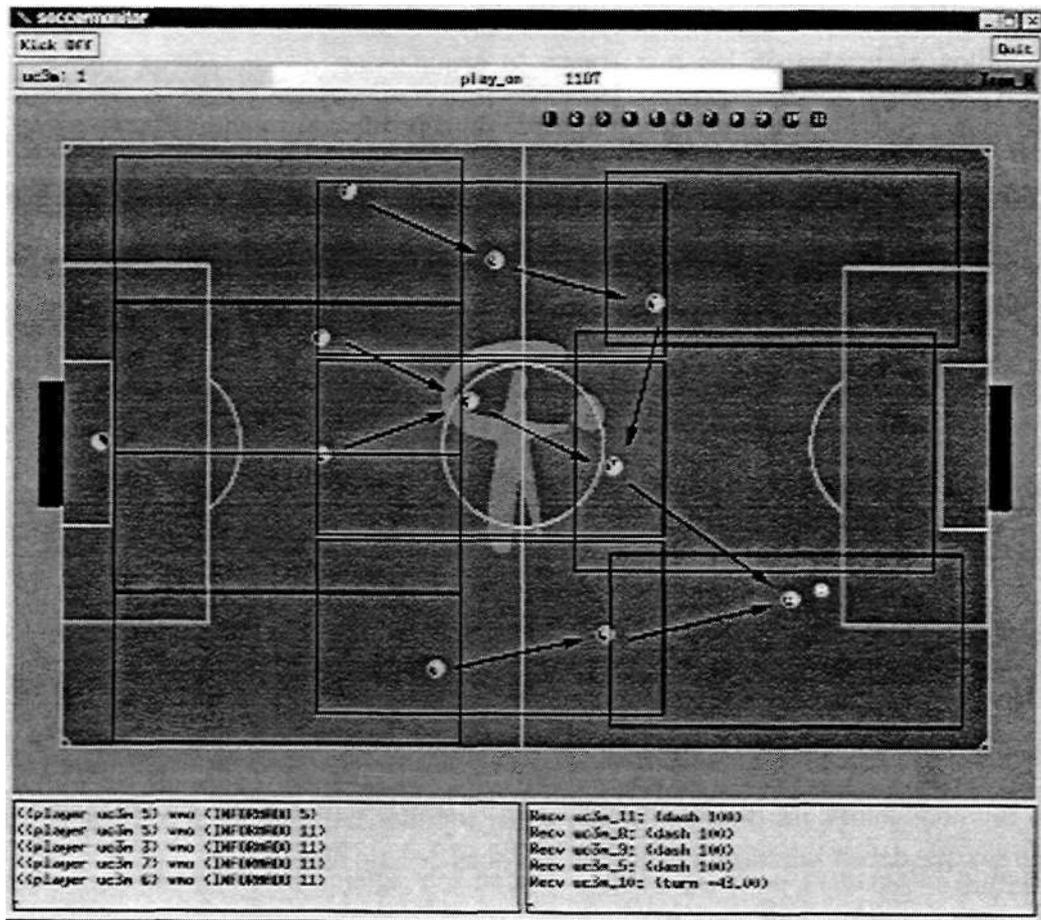


Figura 1. Posible definición de las zonas de los jugadores para una disposición 4-3-3.

En principio, éste 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 sobre dónde está su líder particular, al que siempre deberá mantener bajo su rango de visión; habrá

que definir también controladores que permitan buscar al líder en caso de que el agente se pierda.

Controlador: Mantener\_Posición

SI (cerca (mi\_líder)=CIERTO) ENTONCES IR(centro\_zona)

SINO IR (Posición\_líder)

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

### 2.3 Coordinación y comunicación

En este punto falta definir cómo se establece el líder de cada agente. Aquí entra en juego el único conocimiento global que es transmitido por los agentes: el líder global. Se define el líder global del equipo como aquel jugador que posee el balón o que tiene más posibilidades de poseerlo. En cada momento sólo debe existir un líder global en cada equipo, y la información sobre quién es tal líder en cada 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 líder local de cada jugador puede ser calculado mediante heurísticas, a partir del líder global del equipo. 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 líderes locales a partir del líder global.*

En la Tabla 2 se muestra quién es el líder local para cada jugador en función del líder global. Así, si suponemos que el líder global es el jugador número 11, de los datos 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 se obtiene mediante esta heurística. Cabe destacar que el portero (jugador número 1) presenta un comportamiento independiente.

Esta heurística puede ser ampliada, mejorada o aprendida incluyendo otros parámetros, como la línea de juego del líder (si es delantero, centro, etc.), si el equipo está atacando o no, etc.

El último punto que queda por definir es quién es el líder del equipo en cada momento. Esta decisión viene dada también por heurísticas o controladores sencillos como los que se muestran en la Tabla 3:

Controlador: Quitar\_Líder (num)

SI & (muy\_cerca(bola)=CIERTO) ENTONCES Cambiar\_lider (num)

Controlador: Ceder\_Líder (num)

SI & (he\_pasado(num)=CIERTO) ENTONCES Cambiar\_lider (num)

Tabla 3. *Controladores Quitar\_Líder y Ceder\_líder. Permiten cambiar el líder local del equipo.*

El controlador Quitar\_Líder (num) 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 se acaba de pasar el balón. En ambas situaciones, el cambio de líder debe incluir el único proceso de comunicación entre los agentes que hay previsto 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.

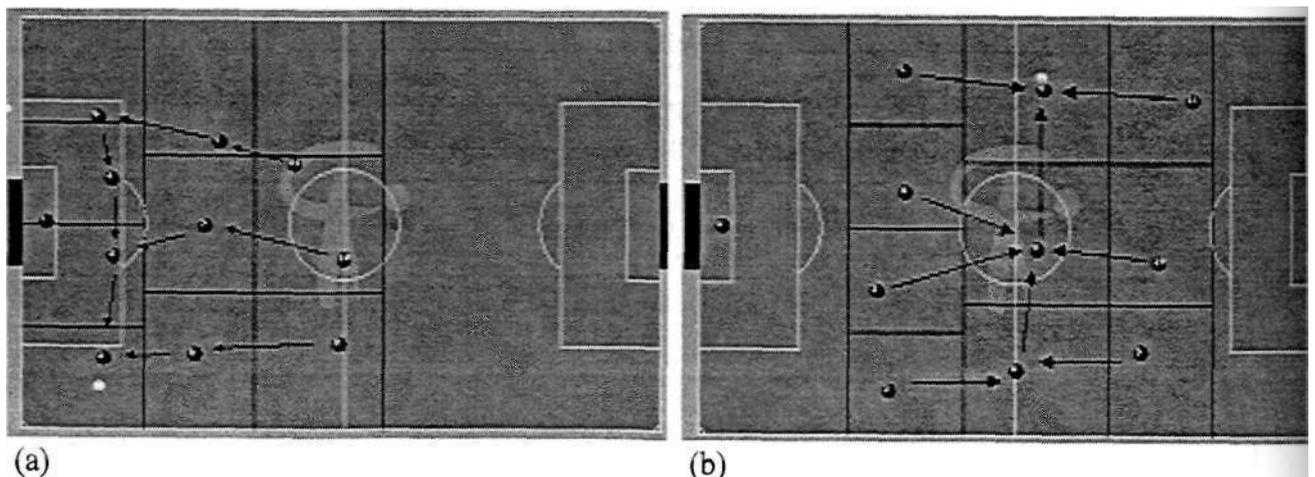
La Figura 2 muestra una primera aproximación de cómo se unen todos los controladores reactivos para cada agente.

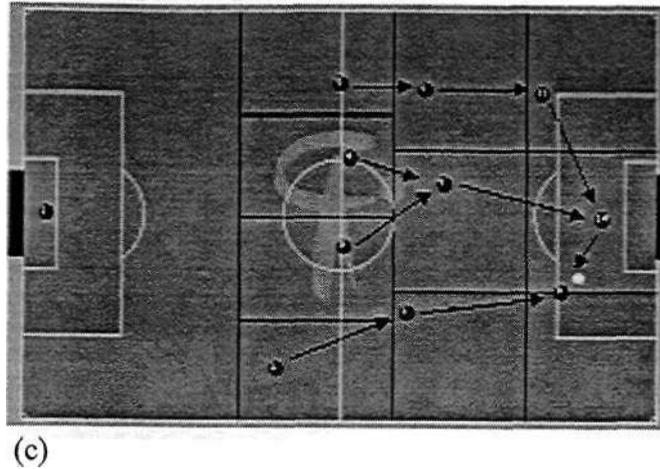
Como se puede comprobar, hay un primer bloque de acciones que deben ejecutar todos los agentes: verificar quién es el líder global, para así obtener su líder local y su zona asociada, y verificar si el jugador está bloqueado. El segundo bloque discrimina el comportamiento del agente que hace de líder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



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 jugador y, si lo encuentra, le pasará la pelota, pasando éste a ser el líder del equipo. Si no pasa la pelota a otro jugador, seguirá avanzando con ella hacia la portería, y llegará a salir de su zona. El resto de los jugadores, si se encuentran cerca de la pelota, intentarán obtener el liderazgo; si no, mantendrán su posición dentro de la formación siguiendo a su líder particular ("mi\_líder") o situándose en el centro de su zona si no tienen información acerca de él.

La disposición aquí mostrada es un 4-3-3, pero cambiar de estrategia es sumamente sencillo. En la Figura 4 podemos ver la disposición de los jugadores en el caso de que el líder sea el número 2, un defensa. En este caso las zonas de los jugadores están distribuidas abarcando el campo propio. Si el líder del equipo es un jugador de la línea de medios, las zonas asignadas ocupan la parte central del terreno de juego. Y por último, si el líder es un delantero, las zonas cubren principalmente el campo contrario. Como se puede ver en la primera instantánea, el jugador número 11 está fuera de su zona asignada. Esta situación es posible porque previamente fue líder y se está situando correctamente. Se permite que un jugador siguiendo a su líder particular salga de su zona asignada, regresando a ésta si se diera la circunstancia de que perdiera el contacto con aquél. Modificando estas zonas, el tamaño, la disposición, etc., se podrían obtener distintos comportamientos como grupo, más conservadores, más ofensivos, etc.





(c)  
 Figura 4. 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.

#### 4. Conclusiones

El sistema actúa en un entorno dinámico cuyo estado cambia en tiempo real, el control es autónomo y el conocimiento del dominio es incompleto (cada agente sólo obtiene información sensorial local) e inexacto (la información recibida tiene ruido).

El objetivo del sistema es minimizar la comunicación entre los agentes sin perder el comportamiento coherente tanto del individuo como del grupo, es decir, minimizar la información transmitida por los agentes al resto del grupo, aunque manteniendo una información global mínima que permita el correcto desarrollo de la estrategia de juego.

La información global junto con la local son los datos de entrada de un controlador reactivo que permite que cada agente sea autónomo y no dependiente de otros jugadores. La información global permite que las estrategias de cada robot se conjunten en una estrategia global, es decir, en un comportamiento conjunto coherente. En este sistema se ha minimizado al máximo esa información global.

Esta arquitectura permite una coordinación global basada en información local y 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. pueden variar también en función de la situación de juego: si aún no ha empezado el partido, si hay un saque de falta, etc.

Desarrollo de las habilidades individuales. Los controladores son totalmente independientes, por lo que pueden implementarse controladores individuales pre-programados o aprendidos [4].

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

- [1] Bond, A. H., y Gasser L., *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, 1988.
- [2] Brooks, R. A., "Intelligence without representation", *Artificial Intelligence*, 47, 1991, pp. 139-160.
- [3] Brooks, K. A., "Distributed problem-solving techniques: A survey", *IEEE Transactions on Systems, Man and Cybernetics*, Sept/Oct 1987.

- [4] Fernández, F., y Borrajo, D. "Vector quantization applied to reinforcement learning", en M. M. Veloso, (ed.), *Proceedings of the Third International Workshop on Robocup*, Estocolmo, (Suecia), agosto 1999, pp. 97-102.
- [5] Girault, F., y Stinckwich, S., "Footux team description: A hybrid recursive based agent architecture", en M. M. Veloso, (ed.), *Proceedings of the Third International Workshop on Robocup*, Estocolmo, (Suecia), agosto 1999, pp. 103-108.
- [6] Iso, R., y Inazuni, H., "A multi-layered planning architecture for soccer agent", en H. Kitano, (ed.), *Robocup-97: Robot Soccer World Cup I*, 1997, pp. 513-518.
- [7] Kamel, M., y Ghenniwa, H., "Coordination of distributed intelligence systems", en Aminzadeh y Jamshidi (eds.), *Soft Computing*, Prentice Hall, 1994.
- [8] Kitano, H.; Asada, M; Kuniyoshi, Y.; Noda, I., y Osawa, E., "Robocup: The robot world cup initiative", en *Proceedings of the IJCAI-95 Workshop on Learning Robots*, diciembre 1995, pp. 19-24.
- [9] Matellán, V., "ABC<sup>2</sup>: Un Modelo para el Control de Robots Autónomos", PhD thesis, Universidad Politécnica de Madrid, 1998.
- [10] Matellán, V. *et alii*, "Using ABC<sup>2</sup> in the robocup domain", en H. Kitano, (ed.), *Robocup-97: Robot Soccer World Cup I, Lecture Notes in Artificial Intelligence*, Springer Verlag, 1998, pp. 475-483.
- [11] Noda, I., "Soccer server a simulator of robocup", en *Proceedings of the Fourth International Symposium'95*, diciembre 1995.
- [12] Stone, P., "Layered Learning in Multi-Agent Systems", Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1999.
- [13] Zhang, Y., y Mackworth, A. K. "Using reactive deliberation for real-time control of soccer-playing robots", en H. Kitano, (ed.), *Robocup-97: Robot Soccer World Cup I*, 1997, pp. 508-512.

## Notas

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