

# Arquitecturas de control sobre robots Nao en la SPL de Robocup

F. Blanes, M. Muñoz, J. Simó, H. Martínez, J. José Alcaraz  
Instituto de Automática e Informática Industrial Universitat Politècnica de Valencia  
Dep. Ingeniería de la Información y las Comunicaciones Universidad de Murcia

**Resumen**— La SPL (Standard Platform League) es la liga de la organización Robocup ([www.robocup.org](http://www.robocup.org)) en la que todos los equipos compiten usando el mismo hardware (desde el año 2008 el robot Nao de Aldebaran Robotics) y centran sus esfuerzos en el desarrollo de los sistemas software relacionados con la sensorización y el control del mismo. Si bien podríamos pensar que esta limitación en el tipo de hardware a utilizar puede producir un conjunto de sistemas con un nivel similar de juego, la realidad es que existen grandes diferencias entre los equipos ya que los aspectos de control y soporte de ejecución son decisivos a la hora del desarrollo. En el presente trabajo se realizará una revisión de la arquitectura de los sistemas con mejores éxitos en la competición y sus principales aportaciones a la robótica humanoide. Al mismo tiempo se presentará la arquitectura utilizada por el equipo "Los Hidalgos", sus puntos fuertes, limitaciones así como las líneas de progreso para próximos desarrollos.

## I. INTRODUCCIÓN

LA SPL (Standard Platform League) es una de las ligas con más equipos participantes dentro de la soccer league de humanoides (28 en 2011). Su atractivo radica principalmente en el uso de una plataforma robótica común a todos los equipos lo cual permite centrar los desarrollos en aspectos de sistemas software como sensorización, fusión de datos, visión artificial, control de procesos, comunicaciones, coordinación etc. Bajo esta perspectiva las competiciones son un buen referente sobre investigación en sistemas empotrados de control en tiempo real, ya que a la dificultad propia de los temas enumerados se añade el hecho de que la autonomía del sistema se sostiene sobre un computador con recursos bastante limitados.

Todos estos aspectos hacen que el diseño de un arquitectura con un buen equilibrio entre flexibilidad, capacidad de escalado de la misma (para adaptarse fácilmente a cambios en la normativa de competición o actualizaciones del hardware) y uso eficiente de recursos (para aprovecharlos en la medida de lo posible al control del sistema y no a la gestión del mismo) sea crucial en este caso de aplicación.

Si bien el fabricante del robot proporciona un framework

de desarrollo con un conjunto de librerías amplio (librerías Naoqi con interfaz C++ y Phytion) que abarcan muchos de los aspectos que se van a abordar en esta revisión, su ineficiencia fuente de la generalidad del diseño y en ocasiones probada inestabilidad, hacen recomendable limitar su uso a lo imprescindible siendo esta una cuestión que puede variar entre grupos en función de los recursos disponibles para el desarrollo.

Podemos aglutinar las tareas a realizar por el robot en varios grupos los cuales en función de la arquitectura diseñada se organizaran de diversas formas en módulos software. Estos serían:

- Sensorización, percepción del entorno y fusión de datos: obtención de datos de los sensores internos del robot, inferencia del estado del robot (postura, localización en el entorno), adquisición de imágenes, interpretación de la escena, localización de objetos de interés en el entorno, fusión temporal.
- Locomoción: uso del sistema de locomoción del fabricante o uno propio (gaits básicos), generación de movimientos específicos (kicks, levantarse y otros).
- Comportamientos y estrategias: bajo esta denominación se engloban aquellos bucles de control los cuales enlazan o fusionan movimientos básicos en función de la información sensorial con diverso grado de abstracción. Puede ser desde un comportamiento de localizar la pelota hasta un movimiento de pase con un robot del mismo equipo.
- Comunicaciones: esta funcionalidad abarca las comunicaciones con los sistemas de arbitraje de los partidos, las comunicaciones entre robots y con las aplicaciones externas de monitorización y calibrado, si bien estas últimas solo se usan fuera de los partidos.

## II. MODELOS DE ARQUITECTURAS

Intentar clasificar los modelos de arquitectura utilizados por los equipos es una labor sujeta a una gran complejidad desde el principio ya muchos de ellos no hacen referencia en sus documentos descriptivos al concepto de arquitectura ([1] , [2] , [3] , [4] , [5] , [6] , [7] , [8] , [9]) teniendo esta como módulos desarrollados, relaciones entre ellos, modelo de

implementación y soporte de ejecución.

Suele ser habitual que la organización en módulos siga el modelo Percibir-Planificar-Actuar siendo la principal diferencia entre ellos la forma de implementarlo. Muchos optan por el uso de un paradigma de agentes ([10],[11],[12],[13]) para implementar los bucles de control

Son minoría los grupos ([13],[14]) que se esfuerzan en el uso de modelos formales o el análisis del comportamiento de los sistemas desde el punto de vista de integridad, tolerancia a fallos y comportamiento temporal, aspecto este que suele vincularse al uso de modelos concurrentes de ejecución ([10],[14],[15]). Este último aspecto aporta un nivel de complejidad en el desarrollo (objetos compartidos, mecanismos de sincronización, etc.) y posterior verificación que hace que en muchos casos se adopte un modelo mono proceso del bucle principal ([11],[16],[17],[18])

### III. SISTEMAS DE PERCEPCIÓN

El sistema de percepción del robot nao está formado por tres conjuntos de sensores: los sensores de presión de planta, el sensor inercial y el sensor de visión.

Los sensores de presión junto con la unidad inercial son utilizados para cerrar los bucles de control del sistema de locomoción.

El sistema de visión aporta la información relativa al entorno del robot. El bípedo nao dispone de dos cámaras: una de ellas situada en la frente del humanoide orientada horizontalmente; mientras que la segunda situada bajo la “barbilla” está orientada a 45° bajo la horizontal. La utilización de las cámaras no puede ser simultánea, y la conmutación entre ellas conlleva un tiempo no despreciable, por lo que la mayoría de los equipos descarta la utilización de ambas y utiliza únicamente la inferior en sus desarrollos. No obstante algunos desarrollos ([15][9]) demuestran que con la utilización de ambas cámaras se aumenta el campo visual minimizando los movimientos de cuello del robot y mejorando las prestaciones.

El entorno en el que se desenvuelven los robots durante la competición es muy estructurado (porterías azul y amarilla, pelota naranja etc.) para facilitar la detección de objetos, siendo el objetivo general de la competición hacerlo cada vez más semejante al usado por los humanos (porterías iguales y pelota oficial FIFA). Para el desarrollo del partido una aproximación generalizada al problema de la identificación de elementos en la escena se basa distinguirlos por su color, es por ello que en la mayor parte de los desarrollos se utiliza clasificación mediante color ([15][1][2][3][16][4][10][12][7][13][9][18][20]).

Todos estos equipos utilizan LUT (Tablas de consulta del inglés Look Up Tables) para caracterizar el conjunto de tonos/intensidades que definen uno de los colores de interés. Estas tablas pueden ser obtenidas de forma semiautomática ([10][16][3]) donde el usuario a través de una aplicación indica que píxeles de la imagen forman parte

de un color. Otros utilizan métodos de autocalibración para obtener estas tablas de consulta de forma automatizada [15][12][7][9][18] y sin intervención del programador. Los cambios de iluminación son el principal inconveniente que presenta la utilización exclusiva del color para determinar los objetos, por ello se han desarrollado diferentes alternativas con el fin de evitar este problema. La primera de ellas es la utilización métodos de corrección de la iluminación ([10][8][9][19]) de modo que tras un cambio en la iluminación adapta la LUT a fin de poder seguir realizando la clasificación de forma adecuada. El segundo de los métodos utiliza la forma de los objetos para realizar la detección ([14][6][13][8][18]) y en algunos casos, adicionalmente se utiliza el color a posteriori a modo de validación [4]. Estos algoritmos deberán de ser introducidos por el resto de equipos puesto que se tiene a la eliminación de colores distintivos como ya sucede en otras ligas.

Actualmente uno de los retos se encuentra en la reducción del tiempo de computo de los algoritmos encargados de la detección de objetos, y la extracción de características de los mismos. Con este fin se han implementado diferentes soluciones en las que se pretende reducir la cantidad de información total a analizar. Para ello se analiza solo parte de la imagen seleccionando únicamente algunos de los píxeles de la imágenes bien repartidos de forma homogénea (líneas de escaneo [1][10][8], o realizando la llamada atención visual [19][9] que consiste en seleccionar una pequeña zona donde se encuentra el objeto de interés y el análisis es realizado únicamente sobre esta.

Debido a la gran especialización que requiere la competición y la necesidad de adaptarse a las condiciones de iluminación, la mayoría de equipos dispone de herramientas software propias que ayudan a realizar tareas de calibración, testeo, validación, teleoperación, etc. Aun así algunos equipos utilizan librerías generales de visión por computador (OpenCV en [2][3][19][12]) o aplicaciones de propósito general (Matlab en [18]) para este cometido.

### IV. SISTEMAS DE LOCOMOCIÓN

La locomoción de los robots humanoides es uno de los temas que más esfuerzo e investigación requieren por parte de los grupos y prueba de ello es la gran cantidad de artículos que cada año se dedican a este aspecto en el workshop de Robocup. Por otro lado la SPL ofrece un marco perfecto para la investigación en el desarrollo de sistemas de locomoción ya que el uso de la misma plataforma comercial permite por parte de todos los equipos un desarrollo más rápido de los mismos al tiempo que una comparación de los mismos sobre la misma base hardware. Estos factores han facilitado la evolución notable de sistemas de locomoción diversos al margen del proporcionado por el fabricante del robot. Vamos en esta sección a describir los más notables de aquellos que han

participado en la competición de 2011.

Una característica común de la mayoría de los sistemas de locomoción es que se basan en el concepto de ZMP (Zero Moment Point [22]) el cual es el punto de contacto con el suelo en el que la resultante de las fuerzas dinámicas que actúan sobre el robot se anulan. Con el objeto de simplificar el cálculo del ZMP suele usarse una aproximación de control basada en el péndulo invertido. De esta forma desde el punto de vista dinámico el robot se comporta como un único punto localizado en el centro de masas del mismo (CoM o Center of Mass).

Uno de los sistemas de locomoción más utilizados en el robot Nao es el proporcionado por el fabricante del mismo [23]. En un primer estado los comandos de velocidad recibidos por el sistema de locomoción se usan para definir los puntos de paso de los pies. Estos a su vez permiten determinar una trayectoria del ZMP del robot que garantice la estabilidad del mismo. El estado del robot como sistema dinámico es representado por la posición, velocidad y aceleración del CoM, mientras que el jerk del mismo es utilizado como señal de control. Se utiliza un controlador predictivo para obtener dichos valores minimizando el error en el seguimiento de la trayectoria del ZMP. Finalmente, técnicas cuadráticas de programación son utilizadas para resolver dicho problema.

Habría que remarcar que en la solución obtenida por el fabricante, con el fin de evitar el desplome de la cadera en la etapa de apoyo simple, se añade una señal trapezoidal a la referencia de la articulación de cadera, con lo que se evita este inconveniente. La velocidad alcanzada con esta solución es mejorable, pero cabría remarcar la gran estabilidad lograda.

Del mismo modo el equipo NaoDevils utiliza el mismo modelo predictivo para calcular el CoM en el seguimiento del ZMP [5][24]. Aunque a diferencia de en el método anterior, en la representación dinámica del estado del robot, se utiliza el ZMP en lugar de la aceleración del CoM, y como señal de control se utiliza la velocidad del ZMP, que se integra a fin de realimentar el controlador. Utilizando dicha configuración se alcanza la mayor velocidad de entre los equipos de la SPL, según sus TDPs. Un resumen de estas velocidades puede ser observado en el tabla 1.

	Forwards (cm/s)	Backwards (cm/s)	Lateral (cm/s)	Rotational (deg/s)
Aldebaran	9.52	9.52	9.52	42
Nao Devils	44	*	*	*
B-Human	32	22	12	92
Wright Eagle	33	20	11	90

Tabla 1: velocidades máximas documentadas en los TDP (Team Description Paper). \* no proporcionada.

En la competición, muchos de los equipos han optado por utilizar el sistema de locomoción desarrollado por el equipo B-Human. Este sistema que ofrece muy buenas prestaciones es lanzado como código libre tras cada una de

las competiciones. Dicha locomoción también se basa en el concepto del ZMP, pero en esta ocasión no se utiliza el Preview Controller [25][15]. En dicha propuesta, la posición del ZMP es situado en un punto fijo del pie y la duración del paso es determinada por las ecuaciones del péndulo invertido. Mientras que en el plano sagital, la velocidad del robot es controlada desplazando la posición de la base del péndulo a lo largo del pie. La posición y velocidad del CoM es estimada a partir de la información de los sensores. Para simplificar estos cálculos, la rotación del robot en el plano horizontal se desestima. Una de las peculiaridades de este sistema de locomoción es que carece de fase de soporte doble en contraposición con las expuestas anteriormente, esto posibilita que sea una de las más veloces a la par que robusta de la liga SPL.

Otra solución basada en ZMP ha sido desarrollada por el equipo WrightEagle, con resultados excelentes [21]. Lo más destacable de dicha implementación es que calcula la compensación de la caída de la cadera de forma teórica, modelando dichas articulaciones como uniones elásticas. Como contrapartida, este sistema de locomoción necesita de un estado de transición entre movimientos de diferente tipo.

Otros equipos en la SPL han contribuido con trabajos originales a la vez que interesantes. A modo de ejemplo, El equipo NUBots ha modificado la locomoción proporcionada por el fabricante incrementando su velocidad y mejorando el consumo de batería todo ello modificando los valores de par máximo de los motores ([26][27]), estos fueron seleccionados mediante un proceso iterativo que maximiza el coste de transporte.

Así mismo, el equipo HTWK ha propuesto una versión diferente basada en algoritmos genéticos que resultó en un conjunto de movimientos tan antinaturales como rápidos y robustos. Desafortunadamente este sistema no era omnidireccional. Actualmente están trabajando en una versión con esta característica basada también en algoritmos genéticos [4].

## V. COMUNICACIONES

Las comunicaciones en el robot Nao son soportadas por medio de las interfaces Ethernet y WiFi que se encuentran en el módulo principal. Mientras la primera es usada principalmente para la conexión a aplicaciones de monitorización y calibración, la segunda es la que permite conectar los robots con la aplicación de arbitraje (GameController) y con el resto de robots del equipo. Es este último caso de aplicación el que adquiere especial relevancia ya que permite implementar sistemas de coordinación de equipo y de esta forma estrategias de grupo.

Cabe destacar como detalle que hasta la versión actual de

hardware (V3.3 con chip WiFi Ralink) el dispositivo WiFi estaba basado en un chip Zydas el cual adolecía de un bajo rendimiento (con pérdida de señal y por tanto de comunicación) proporcional a la temperatura del mismo. Esto implica que el soporte de comunicaciones y las capas que sobre el mismo se deben construir asumiendo dicho comportamiento de pérdidas temporales de paquetes.

Si bien la gran mayoría de los equipos dan soporte a esa capacidad de comunicación inter-robot de una u otra forma (desde el uso de middleware de terceros como es el caso de ROS [14] hasta sistemas propios de suscripción-publicación [10]) el aspecto diferencial a este nivel es la implementación de una percepción colectiva y estrategias de grupo. Cabe destacar que algunos equipos que participan también en la liga de simulación, intentan trasladar sus algoritmos de coordinación a la liga SPL ([12]) sin que de momento sus resultados hagan pensar en que es viable ese salto del mundo virtual al real.

El control del equipo es un aspecto de la arquitectura global de control que cada vez se hace más indispensable, ya que el número de robots en el campo se va incrementando en cada edición (desde los 2 jugadores en 2008 hasta los 4 de 2011). No abordar estos aspectos de coordinación entre jugadores desencadena habitualmente un deficiente control del espacio de juego, concentrándose los robots jugadores entorno al objeto de atención (la pelota) en forma de melés de rugby.

Por otro lado implementar estrategias de grupo requiere un soporte de información resultado de la fusión sensorial que nos permita localizar con un cierto grado de fiabilidad los objetos en el campo (pelota, jugadores del mismo equipo y si es posible oponentes y porterías) así como la capacidad de regenerarlas cuando el robot cae y momentáneamente pierde puntos de referencia durante unos ciclos de ejecución de los bucles de control.

Equipos como B-Human [15] implementan un sistema de mundo global común para todos los jugadores (calculado de forma cooperativa) en el cual la posición y velocidad de la pelota es conocida por todos ellos. Esta es la base para la selección del rol a ejecutar por cada uno de los jugadores. Esto les permite ahorrarse comportamientos de búsqueda de la pelota ya que tan solo con que uno de los jugadores la vea, en poco tiempo (dependiendo del tiempo empleado en el algoritmo de decisión y los retardos en las comunicaciones) esta información es compartida con el resto de compañeros de juego. De forma similar NaoDevils [5] implementa un modelo global de referencia de los objetos de interés (multi-robot SLAM) sobre el cual se construye el sistema de táctica de equipo, sacrificando la velocidad de la locomoción (de hasta 44 cm/s) a favor de una mayor precisión a la hora de construir dicho modelo global.

Sin embargo lo habitual es implementar sistemas de asignación de roles de forma dinámica a los jugadores

basándose en información local ([14]).

Uno de las aproximaciones más utilizadas para implementar el control de grupo es el paradigma basado en agentes ([11], [13]), bajo el cual se modelan diferentes roles (atacante, portero, defensa etc) así como las interacciones entre ellos.

## VI. ARQUITECTURA HIDALGOS

Este capítulo describe la arquitectura utilizada en los robots del equipo Hídalgos. Algunos de sus módulos tienen por origen el sistema utilizado por el equipo TeamChaos tanto para robots humanoides (Nao) como para robots cuadrúpedos (Aibo). A su vez esta, estaba basada en la arquitectura ThinkingCap aunque en la actualidad la arquitectura dista mucho de esta filosofía inicial.

La arquitectura se ha dividido en 5 módulos desde el punto de vista de diseño lógico, funcionamiento e implementación. Estos módulos se han dividido atendiendo a las funciones específicas que realizan dentro del sistema. Así los módulos son:

- PAM: Módulo de percepción. Aporta información de sensorización interna y externa al sistema.
- CTRL: Módulo de control. Gestiona el comportamiento del robot y la estrategia del equipo.
- GM: Módulo de localización. Realiza la localización del robot y los objetos de interés en el mundo.
- CMD: Módulo de interfaz y locomoción. Implementa la interfaz con el middleware de gestión del robot (Naoqi) y contiene los diferentes motores de locomoción.
- COMMS: Módulo de comunicaciones. Gestiona la distribución de la información entre robots así como las comunicaciones con el exterior (GameController - Usuario).

A continuación se realizará una descripción de cada uno de los módulos desde el punto de vista funcional así como una descripción de cómo se sincronizan y comparten información en la ejecución.

### A. Módulo PAM.

El módulo de percepción es el encargado de obtener la información del robot y de su entorno a partir de la información sensorial.

La principal fuente de información externa proviene del análisis de las imágenes procedentes de la cámara. Por las características prefijadas del entorno, los objetos se obtienen inicialmente a partir de una diferenciación por color, a posteriori se le aplican un conjunto de algoritmos de forma, posición, etc. para aumentar la confianza en la detección del objeto, incluso analizando la disposición de grupos de objetos con este fin. Una vez detectado, se caracteriza y localiza. Adicionalmente el módulo es capaz de realizar el seguimiento visual de los objetos de interés,

con el fin de que la información aportada por los mismos se refresque constantemente, así como realizar maniobras que localicen los mismos. Esto se conoce como visión activa. El objeto/s de interés es indicado por el módulo CTRL y viene definido por el comportamiento en ejecución.

Los objetos de interés que el robot debe ser capaz de identificar son:

- Pelota,
- Portería (Propia y Oponente),
- Robots / Bandas
- Terreno de juego,
- Líneas y marcas de interés en el campo.

Así mismo, el módulo es el encargado de obtener la información del CoP (Center of Pressure), ZMP (Zero Moment Point) y la inclinación del torso, a partir de la información procedente de la unidad inercial, los sensores de presión de la planta del pie y la posición leída de las articulaciones.

Se puede destacar que el módulo se complementa con dos herramientas software para uso fuera de línea que permiten calibrar los objetos de interés y depurar los algoritmos de visión desarrolladas en Java y Qt respectivamente. Mientras la primera es usada para la calibración de los robots y obtención de las tablas LUT características de cada uno, la segunda permite ejecutar el mismo código C++ de los algoritmos de visión que se ejecuta en los bípedos. De esta forma usando imágenes previamente adquiridas, el procesamiento puede ser validado en sus distintas etapas, permitiendo este sistema compilar el mismo código fuente para la versión a empotrar en el robot.

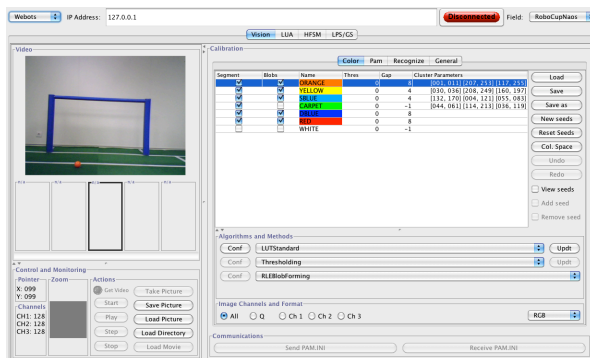


Fig 1: Herramienta de calibración

### B. Módulo CTRL.

El módulo de comportamiento es el que contiene la unidad de toma de decisiones del robot. Esta es jerárquica y rige también el comportamiento del grupo en el caso de que se trabaje en equipo.

Se divide en tres capas como se puede ver en la siguiente figura.

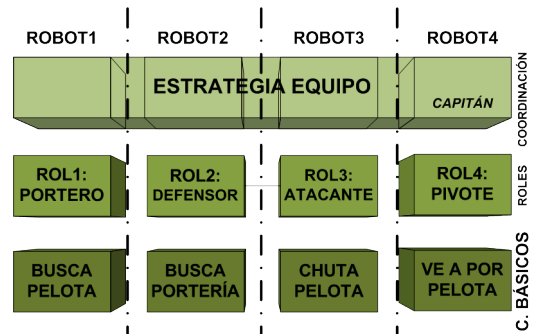


Fig 2: Sistema jerárquico de control

- La capa de Estrategia, dirige el comportamiento del equipo y asigna un rol para cada uno de los robots. A modo de ejemplo, al inicio de un partido, se asignaría el rol de atacante al robot más cercano a la pelota, de pivote al siguiente y defensor y portero a los otros dos. Cuando el equipo pierda la pelota, esta capa reasignaría roles de defensor a cada uno de los robots, o aquellos roles que favorezcan el estado del juego según lo programado.
- La capa de Roles, desarrolla la conducta del propio robot para ello va ejecutando un conjunto de comportamientos simples para conseguir un objetivo. Como ejemplo, en el rol de atacante, el robot deberá de ejecutar los siguientes comportamientos simples: localizar la pelota, encaminarse hacia ella, localizar la portería, localizarse adecuadamente respecto a pelota y portería, y chutar la pelota.
- La capa de comportamientos simples, es la encargada de desarrollar estos y para ello realiza movimientos o aquello que sea conveniente. A modo de ejemplo, para realizar el comportamiento de aproximación a la pelota, el robot se debe de desplazar a una velocidad ( $V_x$ ,  $V_y$ ,  $\omega$ ) obteniéndose estos valores como función de la posición de la pelota respecto del robot. Su diseño se facilita mediante una herramienta gráfica (figura 3).

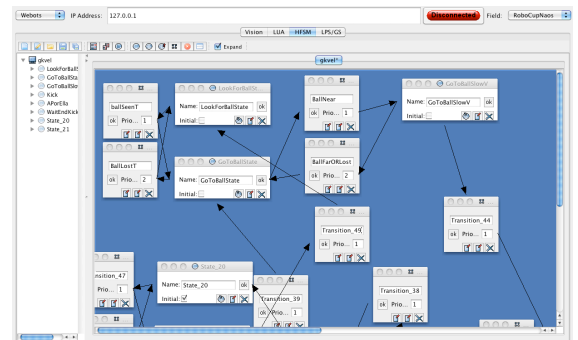


Fig 3: Herramienta gráfica de diseño de comportamientos

### C. Módulo GM.

El módulo de localización obtiene la posición del robot en el campo utilizando la información de posición relativa frente a objetos conocidos y fijos en el campo. En ausencia de los mismos el robot actualiza su posición utilizando la odometría. Actualmente este módulo tan solo actúa con una localización robo-céntrica de los objetos, la cual tiene corta duración en el tiempo. Es objetivo del equipo de desarrollo abordar la localización global de objetos, siendo requisito previo alcanzar una mayor precisión en la localización local y la fusión de datos entre robots.

### D. Módulo CMD.

El acceso al hardware del robot se realiza a través de un middleware del fabricante, por ello este módulo realiza la función de interfaz hacia el mismo garantizando que esto se realiza de modo adecuado y seguro. Adicionalmente el módulo da soporte y gestiona los diferentes motores de locomoción.

El sistema dispone de dos tipos diferentes de motores de generación de movimientos. El proporcionado por el fabricante (básicamente traslaciones y rotaciones en modo control de velocidad) y aquellos generados por un motor propio ajeno a Naoqi (chuts, levantarse del suelo y otros específicos). Ambos son combinables durante la ejecución de comportamientos debiéndose realizar una sincronización cuidadosa entre ambos.

El módulo CMD realiza la gestión de las trayectorias enviando las ordenes de modo que todos los motores de movimientos se sincronicen adecuadamente y asegurando que todos los movimientos se realizan de forma segura.

### E. Módulo COMMS.

El módulo de comunicaciones se encarga de intercambiar información con el exterior, siendo los posibles interlocutores otro robot, la herramienta de calibración del sistema de visión (HidalgosManager) y/o el Ordenador que indica las penalizaciones en los partidos GameController.

Desde el punto de vista de arquitectura su función principal es permitir la comunicación inter-robot. Para ello se usan los mismos puertos UDP en cada robot de envío y recepción gestionados por threads. Las funciones a realizar por el módulo serían:

- Ofrecer objetos de sincronización al resto de módulos del sistema para el envío y recepción de mensajes.
- Gestión de la lista de jugadores activos: mediante un sistema de mensajes periódicos los robots establecen los robots que se encuentran activos (en juego) y con los cuales pueden coordinarse. La ausencia de una cantidad (P) determinada de los mismos durante ciclos consecutivos (C) hacen que el robot en cuestión sea considerado

como no activo. Los parámetros P y C se varían en función de la fiabilidad de la red.

- Gestión de jugador capitán: el robot ejerciendo de capitán será aquel que ejecutará una parte de la estrategia de equipo vinculada a la selección de los roles para cada uno de los jugadores. Para evitar que el equipo deje de tener capitán en el módulo toman decisiones que conllevan al cambio de capitán cuando no se reciben mensajes del actual.

### F. Ejecución del sistema.

Los módulos se ejecutan de modo concurrente para ganar en reactividad y aprovechar mejor los recursos del sistema. Por ello debe haber unos mecanismos de sincronización y protección en la ejecución y en la compartición de información.

El sistema se ejecuta utilizando el soporte de threads POSIX. Los threads se ejecutan sobre el planificador FIFO de alta prioridad que dispone el núcleo Linux que da soporte al sistema. El middleware que soporta funciones propias del robot dispone de prioridad 0 la más alta de dicho planificador y del sistema y un periodo de 10ms. La distribución y configuración de threads es la siguiente:

- PAM: 1 hilo periódico que realiza la adquisición y procesado de la información, con un periodo de 180ms y una prioridad de 14.
- CTRL: 1 hilo periódico que desarrolla el comportamiento del robot, con un periodo de 45ms y una prioridad de 12. 1 hilo periódico que desarrolla el comportamiento del equipo, con un periodo de 500ms y una prioridad de 13.
- GM: 1 hilo periódico que realiza la localización del robot, con un periodo de 250ms y una prioridad de 15.
- CMD: 1 hilo que se activa como call-back en cada ejecución del middleware por tanto con un periodo de 10ms y una prioridad de 10. Se encarga de realizar las tareas de interfaz con el middleware y gestiona la locomoción. 1 hilo aperiódico de prioridad 11 que gestiona las llamadas al middleware bloqueantes a fin de evitar bloqueos.
- COMMS: consta de varios hilos periódicos y aperiódicos de prioridad por debajo de aquellos de control y percepción.

Entre los diferentes módulos la información se distribuye a partir de objetos compartidos cuyo acceso se encuentra sincronizado mediante mutex POSIX.

## VII. CONCLUSIONES

El desarrollo de un sistema para el control de los robots Nao es un proyecto muy complejo y multidisciplinar, el cual hace necesario para obtener buenos resultados de grandes equipos o bien aprovechar desarrollos liberados

([3],[20]).

A la vista de la revisión realizada podemos concluir que:

- Hasta el momento gran parte del esfuerzo de los equipos se ha centrado en el desarrollo de sistemas de percepción que trabajando con recursos de cómputo muy limitados proporcionen precisión en la información generada. Para ello la clave ha sido los sistemas de calibración on-line y el submuestreo de las imágenes.
- Otro de los aspectos diferenciadores ha sido el dotar de sistemas de locomoción más rápidos que el proporcionado por el fabricante. Esto se ha convertido en clave y es un punto fuerte común de los equipos mejor clasificados.
- Los aspectos de coordinación y control de grupo van emergiendo como otro de los puntos fundamentales a desarrollar. Estrategias de equipo, asignación de roles dinámica, percepción global son algunos de los aspectos que poco a poco deberán ser abordados por los equipos en un escenario en el que el número de jugadores crece con las ediciones.
- El diseño de una arquitectura desde sus aspectos de módulos, modelo de referencia conceptual y soporte de ejecución no parece ser un aspecto especialmente relevante entre los grupos. Más allá de la adopción de una tecnología para la implementación de los sistemas de control (comportamientos como agentes, reguladores clásicos, modelos fuzzy etc) no existe de forma generalizada una aproximación común.

En el futuro las arquitecturas podrán crecer en la complejidad de los sistemas que implementan así como en abstracciones dado el salto en hardware que se va a hacer (pasando de un empotrado basado en AMD Geode 500 MHz a Intel Atom 1.6 GHz). Estos avances permitirán la definición de sistemas más complejos los cuales requerirán de mejores modelos de desarrollo.

## REFERENCES

- [1] H. Levent Akin, Baris, Gökçe, Ergin Özkucur, M. Murat Sevim, Okan Asik, Tuğçe Ayar. "Cerberus'11 RoboCup SPL Team Description", Bogazici University, Department of Computer Engineering Bebek, Istanbul, TURKEY
- [2] Dutch Nao Team. "Team Description for RoboCup 2011 – Istanbul". Faculty of Science, Universiteit van Amsterdam, The Netherlands
- [3] Subramanian Ramamoorthy, Aris Valtzanos, Efstathios Vafeias, Christopher Towell, Majd Hawasly, Ioannis Havoutis, Thomas McGuire, Seyed Behzad Tabibian, Sethu Vijayakumar, Taku Komura. "Team Edinferno Description Paper for RoboCup 2011 SPL". School of Informatics, The University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, United Kingdom.
- [4] Rico Tilgner Thomas Reinhardt Daniel Borkmann, Stefan Seering Tobias Kalbitz Robert Fritzsche, Katja Zeil Christoph Vitz Sandra Unger, Manuel Bellersen Hannah Muller Samuel Eckermann, "Nao-Team HTWK TDP". Gustav-Freytag-Strae 42A, 04277 Leipzig, Germany.
- [5] Matthias Hofmann, Soren Kerner, Ingmar Schwarz, Stefan Tasse, and Oliver Urbann. "Nao Devils Dortmund, Team Description for RoboCup 2011". Robotics Research Institute. Section Information Technology. TU Dortmund University. Dortmund, Germany
- [6] Eric Chown, Jack Morrison, Nathan Merritt, Octavian Neamtu, Wils Dawson, Dani McAvoy, Elizabeth Mamantov, Edward Googins, Josh Zallinger, Ellis Ratner. "The Northern Bites 2011 Standard Platform League Team" Department of Computer Science. Bowdoin College, College Station, Brunswick, ME, 04011-8486, USA
- [7] RoboEireann, Prof Rick Middleton, "NUIM Team Description for 2011 RoboCup. Standard Platform League – Nao". The Hamilton Institute, The National University of Ireland Maynooth, Co Kildare, Ireland
- [8] TJArk, Robot & Intelligent System Laboratory "The TJArk' Team Description for 2011". Tongji University, Shanghai 201804, P.R. China
- [9] Samuel Barrett, Katie Genter, Todd Hester, Piyush Khandelwal, Michael Quinlan, Mohan Sridharan and Peter Stone. "TT-UT Austin Villa 2011 Team Description Paper for the Standard Platform League". Department of Computer Science. The University of Texas at Austin. University Station C0500. Austin, Texas 78712-1188
- [10] E. Chatzilaris, I. Kyranou, J. Ma, E. Orfanoudakis, A. Panakos, A. Paraschos, G. Pierris, N. Spanoudakis, J. Threlfall, A. Topalidou, D. Tzanetidou, E. Vazaios, S. Cameron, T. Dahl, M. G. Lagoudakis "Noxious-Kouretes 2011 SPL Team Description Paper" Cognitive Robotics Research Center (CRRC), University of Wales, Newport, UK
- [11] Somchaya Liemhetcharat, Brian Coltin, Junyun Tay, and Manuela Veloso. "CMurfs '11: Carnegie Mellon United Robots for Soccer". School of Computer Science, Carnegie Mellon University, Pittsburgh, USA
- [12] A. J. R. Neves, N. Lau, Luis Paulo Reis, Antonio P. Moreir, Alina Trifan, Bruno Pimentel, Carlos Sobrinho, Edgar Domingues. "SPL Portuguese Team: Team Description Paper for RoboCup 2011" IIEETA/DETI – University of Aveiro, 3810-193 Aveiro, Portugal 2LIACC/FEUP – Faculty of Engineering of the University of Porto, Portugal 3INESC-P – University of Porto, Portugal
- [13] David Claridge, Brock White, Carl Chat\_eld, Sean Harris, Belinda Teh, Jimmy, Kurniawan, Yongki Yushmanthia, Youssef Hunter, Yiming Deng, Manuel Lange, Hung Nguyen, Jarupat Jisarojito, Benjamin Vance, Jayen Ashar, Brad Hall, Bernhard Hengst, Claude Sammut, and Maurice Pagnucco. "Team rUNSWift, University of New South Wales, Australia, RoboCup 2011, Standard Platform League". Computer Science and Engineering, University of New South Wales, Sydney 2052 Australia.
- [14] Markus Bader, Alexander Hofmann, Jens Knoop, Bernhard Miller, Dietmar Schreiner, and Markus Vincze. "Austrian-Kangaroos 2011-Team Description Paper (TDP)", Vienna University of Technology, University of Applied Sciences Technikum Vienna, Vienna, Austria.
- [15] Thomas Röfer, Tim Laue, Judith Müller, Alexander Fabisch, Katharina Gillmann, Colin Graf, Alexander Härtl, Arne Humann, Felix Wenk. "B-Human, Team Description for RoboCup 2011". Universität Bremen, Fachbereich 3 - Mathematik und Informatik, Postfach 330 440, 28334 Bremen, Germany
- [16] Vincent Hugel, Juan Francisco Blandes Noguera, Nicolas Jouandeau, Juan José Alcaraz Jiménez, Humberto Martínez Barberá, Zeki Y. Bayraktaroglu, and Pinar Boyraz. "L3M SPL Team Description". Univ. Versailles, Univ. Politécnica de Valencia, Univ. Paris 8, Univ. Murcia, Istanbul Teknik Üniversitesi.
- [17] J.M.Cañas, F.Martín, C.Aguero, E. Perdices, V.Matellán, Juan García, Francisco Rodríguez, D.Puig, Tomás Gonzalez, and Julián Cristiano. "SPITeam's Team Description Paper for RoboCup-2011". U.Rey Juan Carlos, U. Len, U. Rovira i Virgili (Spain).
- [18] Jordan Brindza, Levi Cai, Ashleigh Thomas, Ross Boczar, Alyin Caliskan, Alexandra Lee, Anirudha Majumdar, Roman Shor, Barry Scharfman, Dan Lee. "UPennalizers, RoboCup Standard Platform League, Team Report 2010". General Robotics Automation Sensing Perception (GRASP) Laboratory of Robotics Research and Education University of Pennsylvania

- [19] Hans-Dieter Burkhard, Thomas Krause, Heinrich Mellmann, Claas-Norman, Ritter, Yuan Xu, Marcus Scheunemann, Martin Schneider, and Florian Holzhauer. "*NaoTH 2011, The RoboCup Team of Humboldt-Universität zu Berlin*". Institut für Informatik, LFG Kunstliche Intelligenz, Humboldt-Universität zu Berlin, Rudower Chaussee 25, 12489 Berlin, Germany
- [20] Fredrik Clinckemaillie, Christopher Conley, Runzi Gao, David Kent, Benjamin Leone, William Mulligan, Khan-Nhan Nguyen, Quinten Palmer, and Sonia Chernova. "*WPI Warriors: Team Description, Standard Platform League 2011*". Worcester Polytechnic Institute. 100 Institute Rd. Worcester, MA, 01609 USA
- [21] James Hornitzky, Benjamin Johnston, Alankar Karol, Rony Novianto, Gavin Spargo, Wei Wang, Xun Wang, Glenn Wightwick, Mary-Anne Williams. "*Wrighteagle Unleashed. RoboCup2010 Team Report, Standard Platform League*". University of Technology, Sydney.
- [22] M. Vukobratovic and B. Borovac. "*Zero-moment point - thirty five years of its life*". International Journal of Humanoid Robotics 1(1), 157-173 (2004)
- [23] D. Gouaillier, C. Collette, and C. Kilner. "*Omni-directional closed-loop walk for NAO*". In: Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on. pp. 448-454 (2010).
- [24] S. Czarnetzki, S. Kerner and O. Urbann. "*Observer-based dynamic walking control for biped robots*". Robotics and Autonomous Systems 57(8), 839-845 (2009).
- [25] C. Graf and T. Rofer. "*A closed-loop 3D-LIPM gait for the RoboCup Standard Platform League*". In Proceedings of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the 2010 IEEE-RAS International Conference on Humanoid Robots. (2010)
- [26] J.A. Kulk and J.S. Welsh. "*A low power walk for the NAO robot*". In Proceedings of the 2008 Australasian Conference on Robotics & Automation (ACRA-2008) (2008).  
S.P. Nicklin, S. Bhatia, D. Budden, R.A.R. King, J. Kulk, J. Walker, A.S.W. Wong, and S.K. Chalup. "*NUbots Team Description for RoboCup 2011*". RoboCup 2011: Robot Soccer World Cup XV Preproceedings (2011).