

Programación de Robots móviles en institutos

Robocampeones 2006

Edición 2006

RoboCampeones es un proyecto de difusión de la ciencia y la tecnología financiado por el Ministerio de Educación y Ciencia dentro del programa nacional de divulgación científico técnica DIF...

Índice general

Índice de figuras	7
Índice de tablas	7
1. Objetivos	1
2. Participantes	3
2.1. Introducción	4
2.1.1. Equipos	4
2.1.2. Robots	4
2.2. Prueba A: El Transportista	5
2.2.1. Objetivo	5
2.2.2. Campo de juego	6
2.2.3. Robots	8
2.2.4. Desarrollo de la prueba	8
2.2.5. Puntuación	10
2.3. Prueba B: Fútbol 2+2	11
2.3.1. Objetivo	11
2.3.2. Campo de juego	11
2.3.3. Robots	14
2.3.4. Desarrollo del juego	15
2.3.5. Puntuación y juego limpio	20

Índice de figuras

2.1. Campo Transportista	6
2.2. Campo de fútbol	11
2.3. Campo Soccer	12
2.4. Bola emisora de infrarrojos para el futbol	12
2.5. Comprobación de los robots	15

Índice de cuadros

1

El Software Libre en la Robótica

El software es uno de los principales componentes de los robots. En particular, opinamos que es el principal problema para la extensión de la robótica en nuestra vida diaria. Uno de los factores que creemos que puede ayudar a paliar este problema es el fenómeno del software libre, que ha crecido de manera extraordinaria en el campo de la robótica. En este artículo analizamos la situación del software libre en tres subáreas de la robótica: industria, enseñanza e investigación.

Las herramientas libres han sido muy populares entre la comunidad docente e investigadora en el campo de la robótica. Incluso compañías cuyo principal modelo de negocio es la venta de robots, han creído conveniente compartir el software con el fin de promover comunidades alrededor de sus productos.

Todos estos casos muestran que el software libre puede actuar como un catalizador en la industria robótica, un sector aún en fase temprana de desarrollo, en el mundo investigador e industrial.

1.1. Introducción

El software libre ha alcanzado gran notoriedad gracias a proyectos muy difundidos como son el servidor Apache, el kernel de Linux o el paquete ofimático OpenOffice. Igualmente, disponemos de múltiples herramientas de programación libres para varios lenguajes

de programación como C Perl o Python. Pero más allá de esas aplicaciones, existe una amplia actividad en otros campos, quizá menos conocidos, pero también fuertemente influenciados por el software libre, sus modelos de desarrollo y herramientas. La robótica es claramente uno de estos campos.

Muchos de los kits robóticos que se venden actualmente incluyen algún tipo de software para controlar o interactuar con el robot. Estos programas puede adaptarse, mejorarse o incluso reemplazarse para añadir nuevas funciones que no fueron incluidas por el fabricante. Las comunidades que se forman alrededor de estos robots no comparten solo el interés por ellos, sino también mejoras, soluciones, código y nuevas ideas gracias a Internet. Los usuarios se han sentido atraídos por el modelo de desarrollo del software libre tanto por razones éticas [?] como practicas [?].

Como introducción a este libro sobre RoboCampeones 2005 nos ha parecido interesante describir la situación actual del software libre en el campo de la robótica. En particular, porque gran parte de los participantes han hecho uso de herramientas de software libre.

De esta forma, nuestro estudio ha dividido en tres partes el campo de la robótica: industria, educación e investigación. Para cada una hemos elegido algunos casos de estudio conocidos por los autores, donde el software libre ha sido la estrategia adoptada.

El resto del capítulo está organizado de la siguiente manera: en la siguiente sección analizaremos el estado de la robótica en la industria a través de tres ejemplos: el robot Cybe de Probotics, el robot Pioneer de Activmedia y el proyecto Orocos, financiado por la Comisión Europea. La tercera sección se centrará en la educación, donde encontramos el que es sin duda el kit robótico más extendido: el LEGO Mindstorms. La cuarta sección tratará sobre las iniciativas de investigación, mencionando la competición Robocup y las plataformas Player/Stage y JDE. Finalmente, se presentan una serie de conclusiones y algunas predicciones acerca de cuál creemos que será la evolución del fenómeno del software libre en el campo de la robótica.

1.2. Software libre en la industria robótica

Durante mucho tiempo se vienen utilizando robots en las aplicaciones industriales. Hoy en día no es raro encontrar robots en fábricas, soldando, transportando materiales, o en cualquier otra tarea repetitiva. En este mercado es difícil la entrada de nuevas ideas en él, y la cantidad de usuarios interesados en algo más que productos cerrados sigue siendo limitada.

A pesar de que las compañías que trabajan con robots caros todavía no han encontrado los alicientes suficientes para entrar a una dinámica dirigida por el uso de software libre, los nuevos mercados donde el número de robots vendidos es muy alto, se ha demostrado claramente que existen muchos buenos argumentos para impulsar a una compañía de robots a entrar en un modelo caracterizado por compartir el código fuente que hace funcionar a sus robots.

En esta sección veremos algunos ejemplos de estrategias de negocio sobre robots, poniendo atención en los robots Cy e y Pioneer. También hablaremos de la plataforma software Orocos, que pretende ser un estándar en las aplicaciones de la industria de la robótica.

1.2.1. El robot Cy e de Probotics

En los últimos años de la década de los 90, algunas compañías fabricantes de robots comenzaron a darse cuenta que había otras formas de gestionar el software que incluían con sus robots.

Un ejemplo es el robot Cy e¹ desarrollado y distribuido por Probotics Inc. Inicialmente esta compañía vendía su software *Map-N-Zap* independientemente de sus robots. Pronto se dieron cuenta que los robots no eran útiles sin ese software, así que decidieron incluirlo en el kit básico del robot. En enero del año 2000 finalmente decidieron distribuir el software bajo licencia GNU GPL (GNU General Public License). El anuncio de esta estrategia tenía como objetivo claro la comunidad de software libre²:

¹<http://personalrobots.com>

²el anuncio de la versión libre se puede encontrar en <http://slashdot.org/article.pl?sid=00/01/24/110219>

“Esta decisión fue tomada en parte por la numerosa cantidad de peticiones desde la comunidad Linux de nuestro código fuente, y el hecho de que ellos, y desarrolladores de otras plataformas, tienen mucho que ofrecer a nuestra misión, que es hacer verdaderamente atractiva la tecnología de los robots.”

Figura 1.1: Aspecto del robot Cye equipado con bolsa recoge polvo

1.2.2. El robot Pioneer robot de Activmedia

El robot Pioneer de Activmedia se vende como una herramienta de investigación, y se usa por una gran cantidad de universidades como principal plataforma de investigación en robótica móvil.

En el momento en que Activmedia comenzó a vender este robot existían otras alternativas en el mercado, las cuales tenían características similares y se había comprobado que eran lo suficientemente buenas para fines investigadores. Actualmente Activmedia es conocido como el mayor vendedor de robots para fines investigadores.

Probablemente la diferencia determinante para esta evolución fue que Activmedia comenzó a distribuir su software de manera abierta mientras que las otras empresas eran más restrictivas. El software de control del robot, Aria³, era una nueva versión de su anterior plataforma de desarrollo Saphira, esta vez publicado bajo la licencia GPL. Una de las metas de este proyecto era ofrecer soporte a la plataforma anterior y a los clientes que estaban aún usándola. Aria ha sido también mejorada por contribuciones de usuarios, repercutiendo en la calidad y funcionamiento del sistema.

1.2.3. El proyecto Orocós

En la situación actual, en la que se pueden encontrar robots desde en plantas industriales hasta en oficinas, es importante tener

³<http://www.activrobots.com/SOFTWARE/aria.html>

Figura 1.2: Robots pioneer equipados con un brazo robotizado, cámaras y sensores láser

un entorno común de desarrollo de aplicaciones para robots. Dichos entornos están presentes en otras tecnologías de software como el tratamiento de imágenes o la computación distribuida.

El proyecto Orocos [?] (Open RObot COntrol Software) presenta un entorno de desarrollo que ofrece funcionalidad genérica para los robots y sus aplicaciones. Este proyecto nació dentro de EURON (European Robotics Network) y fue patrocinado por la Comisión Europea (EC) en el año 2000. Actualmente el proyecto Orocos continúa con fondos del Centro de Tecnología Flanders Mechatronics, desarrollando el entorno de tiempo real y coordinando la integración de Orocos en máquinas industriales.

El objetivo principal del proyecto es desarrollar una plataforma software de control bajo un licencia libre, manteniendo la independencia sobre arquitecturas concretas, contribuir al desarrollo de interfaces de programación de arquitecturas de control y al desarrollo de material docente libre.

Además de sus raíces en el campo de la robótica, el entorno de desarrollo de aplicaciones en tiempo real de Orocos se ha centrado en el campo del control de maquinaria. Actualmente hay dos subproyectos principales dentro del marco de Orocos:

- **Servicios Abiertos de Control en Tiempo Real:** Este es un proyecto general, fuera del marco de la robótica, que desarrolla un kernel de tiempo real para todas las posibles arquitecturas de control.
- **Software Abierto de Control de Robots:** Este es un proyecto específico para robots, con la intención de desarrollar un entorno de desarrollo de aplicaciones, ofreciendo generación de movimientos e interpolación, cinemática y dinámica, algoritmos de control, estimación e identificación, etc.

Este proyecto puede ser considerado un hito en la industria robótica por su impacto en este campo: fue la primera vez que la

Unión Europea aportaba fondos a un desarrollo de software libre en la industria; y segundo, porque ha sido el punto de partida para otros proyectos como Orca⁴, Ocean⁵, etc.

1.3. Software libre en la enseñanza de la robótica

Además del uso práctico, en la industria, el software libre tradicionalmente ha sido muy empleado en la educación. En concreto, el uso del software libre es beneficioso para la enseñanza de ingenierías, como se ha reflejado ya [?]. Los cursos de robótica no son una excepción, como también han dejado claro [?] y [?].

En particular, el grupo de robótica de la URJC tiene experiencia en la enseñanza de los principios de la programación de control de robots a estudiantes de ingenierías informáticas. Además de buscar contenidos adecuados, es necesario elegir de manera adecuada el robot y las herramientas que deben usarse para su programación. En esta sección describiremos las herramientas que usamos para nuestros fines docentes, analizando las razones que hay detrás de nuestras decisiones.

1.3.1. LEGO Mindstorms

Los LEGO Mindstorms⁶ fueron originalmente concebidos como un producto para que los niños aprendieran, pero hoy en día es sin duda el kit para la construcción de robots más extendido en cualquier entorno docente.

Figura 1.3: Robot realizado con piezas de Lego procedentes de kits Mindstorms que resuelve el famoso cubo de Rubik

⁴<http://orca-robotics.sourceforge.net>

⁵http://www.fidia.it/english/research_ocean_fr.htm

⁶<http://www.legomindstorms.com>

El éxito de este producto es, en parte, debido a la comunidad que se ha formado alrededor de él y que, por ejemplo, ha desarrollado por completo un nuevo sistema operativo (BrickOS[?]) y un lenguaje de programación (NQC [?]), desplazando las herramientas cerradas proporcionadas por el fabricante. Los desarrollos de esta comunidad han sobrepasado los límites de los kits que vende LEGO, llevando Mindstorms a su posición privilegiada en el mercado.

1.3.2. El lenguaje Not Quite C (NQC)

NQC [?] son las siglas de Not Quite C, y es un lenguaje parecido a C que se puede usar para programar el ladrillo RCX de LEGO (del kit de Mindstorm). Es la alternativa textual sencilla al entorno visual que aporta Mindstorm.

NQC es software libre, publicado bajo la *Mozilla Public License (MPL)* pero, como usa el sistema operativo estándar desarrollado por LEGO, depende de un solución no libre.

1.3.3. El sistema operativo BrickOS

BrickOS⁷ es un sistema operativo embebido libre diseñado para el ladrillo de LEGO Mindstorms. Es una evolución de LegOS que fue principalmente diseñado por Markus Noga [?]. Comparado con el software estándar (el de LEGO), BrickOS ofrece extraordinarias mejoras de rendimiento y flexibilidad.

1.4. Software libre en la investigación en robótica

El software libre ofrece a la comunidad científica la posibilidad de compartir soluciones a diferentes problemas comunes. Esto, por supuesto, es básico para la evolución de la ciencia; pero el software libre va más allá y hace que el proceso suceda de una manera asequible [?].

⁷<http://brickos.sourceforge.net>

En esta sección hablaremos de cómo el software libre puede mejorar resultados de investigación en entornos competitivos, como es la Robocup. También veremos cómo los entornos de programación de software libre se usan para realizar tareas de programación de hardware heterogéneo usando plataformas abstractas de forma sencilla. Hay varias plataformas software y muchas de ellas son software libre, como por ejemplo Miro⁸, Marie⁹, Carmen¹⁰, Player/Stage¹¹ y JDE¹². Enfocaremos nuestra atención en las dos últimas. La primera se ha convertido en la plataforma de referencia en el mundo de la investigación y JDE es la plataforma desarrollada en nuestro grupo

1.4.1. La competición de la Robocup

Hemos decidido usar el caso de la competición de la Robocup [?], en el cual el grupo de robótica de la URJC lleva participando varios años, como ejemplo del uso del software libre en la investigación en robótica, por su popularidad y vistosidad y por haber sido utilizado como demostración en el intermedio de las pruebas de RoboCampeones 2005.

Robocup¹³ es el nombre “comercial” de una iniciativa internacional de investigación y divulgación. Su meta es fomentar la investigación en el campo de la inteligencia artificial y de la robótica, proporcionando un problema desafiante (principalmente jugar al fútbol), donde se pueden examinar e integrar una amplia gama de tecnologías.

La RoboCup está estructurada en varias categorías: los robots pequeños o F180 (180 mm de diámetro máximo) que usan una cámara cenital que envía imágenes a un PC que las analiza y envía las ordenes a los robots. Los robots medianos, donde los robots tienen que tener integrado todo su hardware, no hay visión, ni ordenador central. La liga de 4 patas donde el hardware es

⁸<http://smart.informatik.uni-ulm.de/MIRO/content.html>

⁹<http://marie.sourceforge.net/>

¹⁰<http://www-2.cs.cmu.edu/~carmen/>

¹¹<http://playerstage.sourceforge.net>

¹²<http://gsync.escet.urjc.es/~jmplaza/software.html>

¹³<http://www.robocup.org>

común, el robot Aibo de Sony y finalmente la liga de 2 patas (humanoides) que está todavía comenzando. Además existe una liga de robots simulados, una RoboCup Junior donde compiten alumnos de secundaria con robots Lego, y una liga de rescate.

Figura 1.4: Preparativos antes del comienzo de un partido de *small size* en Osaka 2005

Existen otras muchas competiciones internacionales entre robots, también nacionales, como el Hispabot (<http://www.hispabot.org>) para alumnos universitarios o RoboCampeones (<http://www.robocampeones.com>) para estudiantes de secundaria.

En el caso de la RoboCup se trata de una competición de carácter mundial y se celebra anualmente. Este año fue en Osaka (Japón) del 13 al 19 de Julio, el pasado año fue en Lisboa (Portugal) coincidiendo con el campeonato de fútbol entre humanos, y en el 2006 será en Alemania de nuevo coincidiendo con el campeonato de fútbol. Además existen unos eventos previos regionales: Open USA, Europeo, etc.

Alguno pensará que cuál es el objetivo de semejante ocurrencia. ¿Para qué queremos robots que jueguen al fútbol? Pues realmente para nada, es simplemente un dominio de prueba de las investigaciones en robótica y más concretamente en inteligencia artificial. Durante algún tiempo el problema estrella fue el ajedrez, sin embargo ya hace mucho que los ordenadores superan a los humanos sin problema. El ajedrez, aún siendo un dominio de búsqueda exponencial, es un dominio muy acotado, muy abstracto. El fútbol es un entorno mucho más realista: hay que tratar con un entorno muy dinámico, hay que lidiar con el mundo real, las luces, otros robots... Es decir, el objetivo no es el fútbol, es una excusa para promover el desarrollo de este campo.

Figura 1.5: Momento de proyección de un partido simulado durante la Robocup 2005

Lo que realmente se “investiga” aquí es en software, en crear comportamientos en los robots, o lo que es lo mismo, se trata de escribir software, y no pocas líneas. Por ejemplo, el equipo ganador en la categoría de las cuatro patas en el año 2004 fue el equipo alemán, mejor dicho, uno de los equipos alemanes llamado el *German-Team* y su código son unas 300.000 líneas de código. El de nuestro equipo, mucho más modesto, tiene unas 50.000.

¿Cómo sé el número de líneas de código del rival? Pues porque una regla de esta competición es que es obligatorio publicar el código así como una descripción del equipo. En concreto el caso del *German-Team* la descripción y documentación de su código es realmente ejemplar. Les obliga su naturaleza germánica supongo, y su propia organización. Este equipo está formado por 4 universidades distintas que compiten individualmente en el campeonato europeo (se ha llamado tradicionalmente German Open, aunque el año que viene se llamará European) y que luego integran las mejores ideas en un equipo común. Eso les obliga a realizar un código muy modular y muy bien estructurado.

A la hora de publicar el código cada equipo puede elegir que tipo de licencia le otorga, aunque lo usual es publicarlo con licencias libres tipo GPL. Esto entronca completamente con la tradición científica y tecnológica de la publicación y compartición de resultados. De hecho el progreso de la ciencia se basa en la comprobación de los mismos. No vale de nada asegurar haber realizado un descubrimiento si éste no es reproducible, es decir, hay que publicar los resultados y la forma en que se han obtenido para que otros puedan verificar que realmente los resultados son que se indican, que son inventados o resultado de una mala realización del experimento (el famoso caso de la fusión fría).

En el caso del campo de la informática la única forma de poder hacer esta comprobación a mi modo de ver es disponiendo del código fuente. ¿Cómo se pueden comprobar las cosas? Alguien podría decir que con el código propietario también se puede hacer, te dan el binario, lo pones en los robots y compruebas que te ganan, luego es mejor. Desde luego, esa aproximación es válida, pero repito que el objetivo no es ganar al fútbol, es demostrar la valía de los algoritmos propuestos, hay que poder comprobar que realmente se está usando lo que se dice.

Aceptando que la motivación científica fuese suficiente, aún nos queda otro escollo: a primera vista cuadra mal lo de competición y software libre. Si se trata de una competición nadie quiere que los rivales conozcan el software, ni desde luego que lo usen...

En principio ese razonamiento es muy plausible, por lo que se ha añadido una regla que es que los equipos liberan su código *después* de terminada la competición. Es decir, ahora mismo podemos ver el código del equipo alemán del año pasado, no el que competirá este año.

Eso soluciona el problema de no jugar contra tus propios algoritmos, contra tu propio software. No arregla el que otros equipos obtengan ventaja de tu trabajo. Efectivamente es así, hay equipos que literalmente son una copia con pocas modificaciones de otros. Por ejemplo, las 300.000 líneas del equipo alemán han sido el punto de partida de muchos equipos. De hecho es algo que se pretende desde la organización, no se trata de una competición pura, se trata de usar una competición para ayudar al avance de la ciencia y la tecnología.

Obviamente, un equipo basado totalmente en el equipo alemán no puede reclamar mucho crédito, aunque gane a otro, en especial si este otro ha sido realmente programado por ellos mismos. ¿Dónde estaría el mérito, en ser capaces de copiarlo? En cualquier caso será del equipo alemán. La garantía de que el mérito se adjudique a los autores originales es la posibilidad del escrutinio público. Si sólo se utilizase el resultado del partido sería mucho más difícil saber si un equipo simplemente usa el código de otro.

Si bien el usar “todo” el código de otro equipo no está bien visto, sí lo está el utilizar partes. En concreto partes completamente resueltas, o en las que un equipo determinado no está interesado pueden tomarse de otro equipo y obviamente atribuirles el crédito por esas partes. De eso se trata en el software libre.

Por ejemplo, nuestro equipo no está interesado en los *walking styles*, esto es en la locomoción del perro, trabajamos más en aspectos de localización, compartición de información, etc. Por ello usamos el código de locomoción del equipo francés, que sí está interesado en esos aspectos. Ellos por su parte utilizan nuestras rutinas de reconocimiento de la pelota, que todo hay que decirlo

son bastante buenas, en especial ante cambios en la iluminación ambiente.

En la liga de 4 patas (o de Aibo) como ya he comentado lo más relevante es el software, pues el hardware es igual en todos los equipos, pero lo es también en muchas otras ligas, en especial en la de simulación donde todo es software. En este caso los participantes tienen que programar 11 jugadores que se comunican mediante un protocolo estándar con un servidor que realiza la simulación del partido. El servidor les envía información de los elementos que “ve” el jugador (bola, líneas del suelo, porterías, jugadores, contrarios, etc.), con ruido (a mayor distancia mayor error); y el jugador tiene que enviar comandos (giro, avance, chut, mensaje a los compañeros, etc.).

Figura 1.6: Final entre los equipos *German Team* alemán y *Nubots* australianos durante la celebración de la Robocup 2005 en Osaka

En la liga de simulación se ha llegado a convertir código escrito por los participantes en código común. En concreto el cálculo de localización, es decir, el que permite a un jugador conocer su posición en el campo a partir de la información visual, ha sido incorporado a la distribución básica. Se trata de unos cálculos relativamente complicados, pero que bien formulados no aportan nada novedoso. En vez de re-escribir continuamente ese código se ha elegido una versión exhaustivamente probada y se ha liberado a los programadores de esa tarea.

Los programadores del código reutilizado pueden enorgullecerse de su trabajo y citar a los equipos que lo utilizan como parte de ese mérito. De nuevo es equivalente a lo que se hace en el campo científico, donde la métrica que se considera mejor actualmente es el número de otros investigadores que referencian los trabajos de un investigador (frente al número de artículos publicados que se usaba tradicionalmente). A la vez, los programadores interesados en otros aspectos (colaboración p.e.) pueden centrarse en su campo y olvidar aspectos de bajo nivel.

Además, los que trabajan en otros campos ahora no pueden

“excusarse” si el rendimiento de su equipo no es bueno, en que ellos no trabajan en esos campos de bajo nivel. Está resuelto.

Esta misma idea es la que en el fondo anima a muchas empresas a liberar su código incluso siendo líderes de su mercado: por una parte consiguen aún más visibilidad, para observadores externos “se pueden permitir el lujo de liberar su software”; y por otra se aseguran de liderar el mercado, pues muy probablemente muchas otras empresas usarán ese software de que son creadores y que por tanto conocen mejor que nadie. Es decir, están liderando su mercado desde el punto de vista de los clientes y de los competidores. Los competidores en muchos casos no pueden hacer más que seguir al líder.

En resumen, puede haber competición, en el sentido literal de la palabra, con software libre. El líder no pierde nada por liberar software, incluso gana porque marca tendencias que él lidera, además de conseguir mucha visibilidad.

1.4.2. La plataforma Player/Stage

Un componente importante del desarrollo con robots es la arquitectura software. Existen múltiples robots de diversos fabricantes que tienen hardware totalmente distinto y sus propios entornos del desarrollo. Con el fin de facilitar esta tarea han aparecido plataformas software que abstraen los detalles de más bajo nivel, proporcionando un API común para programar diferentes robots. El desarrollo de esta clase de herramientas no es la meta principal de la investigación, no obstante pueden mejorar la calidad y el coste de los resultados. El software libre presenta también algunas ventajas aquí. Compartiendo sus herramientas, diferentes grupos introducen cambios pequeños en el sistema completo, beneficiándose del trabajo hecho por el resto de la comunidad. Como un ejemplo tenemos el proyecto Player/Stage [?].

Figura 1.7: Captura de programa de stage mostrando el mapa de una oficina y la trayectoria seguida por varios robots

Desde el principio el desarrollo ha estado abierto a cualquier contribución, ya que proporcionaban su software bajo licencia GPL. Los autores originales sabían que los usuarios de su software eran también desarrolladores que podrían contribuir con su trabajo para conseguir un producto mejor. Debido a esto, el proyecto fue utilizado por más investigadores por todo el mundo, construyendo una gran comunidad. El soporte para diferentes modelos de robots y múltiples lenguajes de programación ha convertido a Player/Stage en la plataforma estándar *de facto* para desarrollos de investigación.

La difusión de Player/Stage ha aumentado de manera que se ha convertido en la plataforma de referencia para muchos equipos de investigación. Por lo tanto, aparecen nuevas herramientas, como un visualizador para Player y arquitecturas de control construidas sobre el sistema completo.

1.4.3. JDE

Sobre las plataformas citadas anteriormente (Aria y Player/Stage), se construyen arquitecturas de más alto nivel. Lo que se denominan arquitecturas *cognitivas*, que proporcionan las capacidades del robot. En el caso del grupo de robótica de la URJC se ha desarrollado la arquitectura JDE, iniciales de Jerarquía Dinámica de Esquemas, que también se distribuye como software libre.

JDE es una aproximación jerárquica a una arquitectura robótica de control, inspirada en la etología y basada en la activación selectiva de esquemas. Es decir, genera las jerarquías dinámicas que gobiernan el comportamiento del robot.

Figura 1.8: Interfaz gráfico de JDE mostrando las medidas sensoriales de un robot, las imágenes obtenidas por cámaras y los botones para interactuar con la aplicación

La implementación actual de JDE consiste en una infraestructura de software y una colección de esquemas específicos de comportamiento, que contienen

técnicas particulares de control o de percepción. La infraestructura incluye los esqueletos de esquemas, los esquemas del servicio según el robots soportado, los mecanismos del software para la selección distribuida de la acción y la activación selectiva de esquemas, así como su ajuste. Además, algunas bibliotecas también se han escrito para manipulación de rejillas y control borroso, con el fin de hacer más fácil la generación del comportamiento. Todo el código se ha desarrollado en el lenguaje C para máquinas GNU/Linux, y está liberado bajo licencia GPL.

1.5. Conclusiones

La robótica es un sector emergente de la industria de las computadoras que todavía se encuentra en etapas tempranas de investigación y uso. En este trabajo hemos presentado algunos ejemplos de porqué el software libre puede actuar como catalizador para el desarrollo de esta industria.

Hay muchas ventajas que se pueden derivar del uso de modelos y estructuras de desarrollo de software libre en el campo de la robótica, de las que, tanto usuarios como empresas, están comenzando a sacar partido. De hecho, algunas empresas han construido su modelo del negocio alrededor del uso del software del libre, como hemos visto en este trabajo.

Hemos mostrado ejemplos donde el uso del software libre ha sido el factor clave para hacer de un producto la referencia de un mercado completo, tanto en la investigación como en la educación. En ambos casos, el software libre es más fácil de adaptar a las necesidades particulares de usuarios y reutilizarlo en otras situaciones. Las comunidades del software libre han demostrado que herramientas libres pueden llegar a ser más populares incluso que las proporcionadas por los fabricantes originales.

Finalmente hemos presentado un ejemplo donde el uso del software libre ha permitido alcanzar progresos más rápidos y resultados mejores. Compartiendo las soluciones ya conocidas, los desarrolladores nuevos pueden centrarse en mejorar las partes del sistema que no están maduras.

2

Objetivos

El objetivo del concurso Robocampeones es fomentar la capacidad creativa y de innovación de los estudiantes, en especial en la rama científicotécnica.

El concurso se organiza en dos pruebas de competición entre robots autónomos.

Las pruebas persiguen potenciar el conocimiento y uso de las nuevas tecnologías de forma estimulante y atractiva para los estudiantes.

3

Participantes

El concurso va dirigido a estudiantes de secundaria y bachillerato de centros públicos. Los equipos formados en parte o en su totalidad por alumnos de cursos inferiores (primaria p.e.) son bienvenidos. No pueden formar parte alumnos de cursos superiores (universidad, titulados, etc.)

La participación de equipos de centros privados y concertados es también bienvenida pero no podrán optar a la financiación pública si existiese, aunque sí a los premios que aporten los patrocinadores privados.

Cada centro podrá presentar un máximo de dos grupos de alumnos, siempre que cada uno de ellos esté bajo la responsabilidad de un profesor diferente. Cada grupo de alumnos estará formado por un máximo de cuatro alumnos.

Durante la celebración de las pruebas se realizarán fotografías y grabaciones de vídeo con fines promocionales (prensa, Internet, etc.). Los profesores responsables de cada grupo son los encargados de obtener los permisos de los padres o tutores de los alumnos. La inscripción en el concurso presupone el permiso de los padres o tutores, tanto para la participación como para la grabación.

4

Descripción de las pruebas

Las pruebas del concurso en la edición del 2005 son: .^{E1} pañuelo Curvoz "Limpiadores de Latas". A continuación se da su descripción y reglamento.

4.1. Prueba 1: El pañuelo curvo

La primera prueba consiste en una versión simplificada del tradicional juego del pañuelo. En ella dos robots, que parten de extremos opuestos de una línea curva, deben seguir dicha línea para atrapar una lata de refresco que se encuentra sobre la misma, y llevarla a su lugar de origen. Se descalificará a los equipos que aprovechen un posible conocimiento de la pista (o que evidentemente ignoren la línea).

El ganador será el que consiga llevar la lata a su lugar de partida. El robot deberá detenerse una vez alcanzada la posición inicial (la lata deberá cruzar la línea de partida). Si los dos robots quedan bloqueados se considera empate y se repetirá la prueba, igualmente si un robot consigue coger la lata pero no dejarla en la posición de salida se considerará empate. Si se repitiese el empate por segunda vez se repetiría de nuevo, y de producirse un tercer el empate el árbitro decidirá el ganador.

La pista de competición consistirá en una línea de longitud desconocida, de trazado también desconocido (igual para ambos

participantes) de color negro sobre fondo blanco, y 1,5 centímetros de ancho. En los extremos y en el centro de la misma se colocarán líneas perpendiculares de igual anchura y 20 centímetros de largo que marcarán los lugares de partida de los robots y la posición de la lata. Los robots podrán tener cualquier tamaño, pero en la salida todo el robot deberá estar por detrás de la línea de partida.

La lata estará pintada de blanco y lastrada con arena de forma que el peso total de la misma sea de aproximadamente 150gr. La figura ?? ilustra una posible configuración de la prueba.

Figura 4.1: Pista ejemplo para la prueba del pañuelo

4.2. Prueba 2: Limpiadores de latas

La segunda prueba consiste en la expulsión de 3 latas de refresco, forradas de cartulina negra o blanca, de un área rectangular de dimensiones conocidas. En la prueba participarán simultáneamente 2 robots, cada uno de los cuales deberá expulsar las latas de un color (uno las blancas y el otro las negras). Al comienzo de la prueba se sorteará el color de las latas a extraer.

El ganador será el robot que consiga sacar en primer lugar las 3 latas de su color fuera del recinto. Existirá un tiempo límite de 3 minutos, pasado el cual si ninguno de los robots ha sacado las latas ganará el que más latas haya sacado. Si han sacado el mismo número se considerará ganador al que haya sacado antes la última lata expulsada de su color.

La pista de la prueba será un cuadrado de 2x2 metros, dentro del cual se delimita una zona de 1,5x1,5 metros mediante una línea negra de 1,5 centímetros de grosor. Las latas tienen que salir completamente del cuadrado para considerarse expulsada. Los robots pueden salir y entrar del cuadrado cuantas veces quieran, pero no podrán ser manipulados durante la duración de la prueba (p.e. recolocados)

La Figura ?? muestra un esquema de una posible configuración:

La disposición de las latas será desconocida a priori y puede variar entre la ejecución de las distintas fases. Se garantizará que la colocación de las latas será simétrica para que los equipos compitan en igualdad de condiciones. Los dos robots parten simultáneamente de lugares opuestos de la pista. Los lugares de partida de los robots estarán marcados con las líneas perpendiculares al borde, como muestra la figura ??.

Un robot no puede mover las latas que tiene que sacar su oponente. Alrededor de cada lata habrá un cuadrado que marcará el máximo desplazamiento que un robot puede hacer de las latas de su oponente. Si un robot saca alguna lata de su oponente de ese cuadrado perderá esa prueba. El cuadrado estará realizado con una línea fina, que se intentará que sea invisible para los sensores.

4.3. Introducción

El presente reglamento se aplicará a todas las pruebas de la competición RoboCampeones 2006.

4.3.1. Equipos

- a** Cada equipo estará formado por un mínimo de 2 alumnos y un máximo de cuatro.
- b** Los miembros del equipo, en el momento de celebrarse las pruebas, estarán cursando Educación Primaria o Educación Secundaria (incluyendo hasta Segundo de Bachiller y Formación Profesional Grado Medio).
- c** Un mismo equipo puede presentarse a una o a las dos pruebas.
- d** En caso de que un mismo equipo se presente a dos pruebas deberá estar formado obligatoriamente por cuatro alumnos.
- e** Todos los equipos estarán tutelados por un profesor/entrenador.
- f** Un mismo tutor sólo puede tener asignado un único equipo.

4.3.2. Robots

Construcción

- a Los robots estarán contruidos *únicamente por piezas de LEGO* de cualquier tipo.
- b Cada robot tendrá un único controlador RCX.
- c La alimentación del robot será las propias baterías o acumuladores del RCX. No están permitidos paquetes adicionales de baterías ni estar conectado a la red eléctrica durante la competición.
- d No está permitido sujetar las piezas por medios externos, como cinta adhesiva, pegamento, tornillos...
- e Sí se podrán utilizar materiales como cartulina, cinta adhesiva... con fines ornamentales. También se pueden usar gomitas, cinta adhesiva, etc. para sujetar o colocar *los cables*, o para adecuar el funcionamiento de los motores y sensores, pero nunca con fines estructurales.

Control

- a Los robots deben ser completamente autónomos.
- b La puesta en marcha de los robots se hará manualmente por uno de los miembros del equipo.
- c No está permitido el uso de controles remotos de ningún tipo.

Programación del RCX

- a Está permitido utilizar cualquier lenguaje, firmware y entorno de programación para generar el código del RCX (RCX-code, NQC, C, Java, Robolinux, Robolab, Labview...).
- b Cada robot podrá usar un único programa que estará *siempre en el bloque de memoria número 5*.

- c Cualquier robot que durante la competición use un programa que no esté en el bloque de memoria indicado en el apartado anterior, será automáticamente descalificado.
- d Sí está permitido el uso de otros programas fuera de la competición, por ejemplo para facilitar el proceso de calibración, sin que necesariamente se encuentren en el bloque de memoria 5.

4.4. Prueba A: El Transportista

4.4.1. Objetivo

El propósito de esta prueba es construir un robot capaz de seguir una línea de forma que pueda coger y transportar latas de refresco de un extremo a otro de la línea.

4.4.2. Campo de juego

En la figura 2.3 se muestra un ejemplo de posible pista para esta prueba.

Figura 4.2: Campo Transportista

Dimensiones

- a La longitud máxima del campo será de 244 centímetros medidos en línea recta desde el comienzo hasta el final del campo.
- b La anchura máxima del campo será de 122 centímetros, y habrá siempre una distancia mínima entre el recorrido y los bordes laterales del tablero de 15 centímetros.
- c Desde el comienzo hasta el primer giro habrá una distancia nunca inferior a 30 cm.

- d Desde el último giro hasta la primera lata la separación mínima será de 30 cm.
- e La separación mínima entre los centros de las latas es de 20 cm.

Suelo

- a El suelo será una superficie lisa y nivelada.
- b Puede estar formada por un papel impreso o por un tablero blanco de madera adecuadamente marcado.
- c Podrá colocarse directamente sobre el suelo o sobre una mesa o peana.

Línea de salida

- a La línea de salida será de color negro, perpendicular a la línea del recorrido y con una longitud mínima de 20 cm.
- b La anchura de la línea tendrá un mínimo de 1'5cm y un máximo de 2cm.

Línea de recorrido

- a La línea de recorrido será de color negro.
- b La anchura de la línea tendrá un mínimo de 1'5cm y un máximo de 2cm.
- c El trazado inicial y final sera recto durante un mínimo de 30 cm al comienzo y de 50 cm al final. El resto del trazado podrá tener cualquier curvatura y geometría.

Puntos de repostaje

- a Se definen dos puntos de repostaje en el circuito.
- b Los puntos de repostaje serán dos círculos de color verde de un diámetro de 3 cm.

- c Los puntos de repostaje podrán estar a cualquier lado de la línea de recorrido estando separada de ésta 5 cm (distancia entre bordes).

Latas

- a Se colocarán dos latas en posición vertical situadas en el medio de la línea, tal y como se muestra en la figura 2.1.
- b Se utilizarán latas de refresco vacías de 33cl.
- c Las latas estarán lastradas de forma que su masa total sea de aproximadamente 100 gr. En concreto se introducirán 3 pilas gastadas del tipo “AA” (como las que se usan en el RCX) en una lata vacía.
- d Las latas será de color blanco, bien pintadas o forradas con cartulina de dicho color. En el caso de estar forradas, la cartulina estará fija a la lata de forma que no se deslice cuando se levante.

4.4.3. Robots

- a Cada equipo estará compuesto por un único robot.
- b No hay limitaciones respecto de la forma o el tamaño del robot.

4.4.4. Desarrollo de la prueba

Preparativos

- a Los equipos podrán acceder al campo de juego antes de comenzar la competición para realizar labores de calibración y verificación.
- b En la medida de lo posible se intentará que también se puedan realizar estas labores en los intermedios entre las distintas fases de la competición.

Duración de la prueba

- a La prueba tendrá una duración máxima de 5 minutos.
- b Cuando el robot termine la prueba, o el tiempo para la misma se termine, el equipo participante recibirá la puntuación correspondiente según se indica en el apartado 2.2.5.

Inicio de la prueba

- a El robot se colocará *detrás de la línea de salida*.
- b Ninguna parte de la proyección vertical del robot podrá tocar la línea de salida.
- c Cuando el árbitro dé la señal, uno de los miembros del equipo pondrá en marcha el robot de forma manual. En ese momento también se pondrá en marcha el reloj de la prueba.
- d Una vez puesto en marcha, el robot comenzará a moverse de forma inmeditata, siguiendo la línea del recorrido.

Navegación

- a El comportamiento fundamental del robot es el de un "sigue líneas".
- b El robot deberá seguir la línea de recorrido en todo momento, no se permiten atajos entre las curvas del recorrido.
- c Si por alguna razón el robot perdiese la línea del recorrido, el robot deberá intentar buscarla.
- d Sólo se permite abandonar la línea intencionadamente en el caso de que fuese necesario realizar un cambio de sentido. Se deberá realizar la maniobra en el menor espacio posible para retomar la línea del recorrido.

Repostaje

- a** A lo largo del recorrido hay dos puntos de repostaje señalizados con un círculo verde, tal y como se indica en el apartado 2.2.2.
- b** Para efectuar el repostaje, el robot deberá detenerse durante un mínimo de un segundo a la altura del punto de repostaje.
- c** El robot podrá repostar en todos y cada uno de los puntos de repostaje por los que pase y tantas veces como pase por encima de ellos.
- d** Cada vez que el robot efectue un repostaje válido acumulará puntos según se indica en el apartado 2.2.5.

Transporte de las latas

- a** Una vez que el robot llegue siguiendo la línea de recorrido hasta la posición donde se encuentran las latas, deberá coger una de ellas.
- b** Las latas pueden ser levantadas, arrastradas o transportadas.
- c** Las latas no pueden ser tiradas ni rodadas.
- d** Si la lata cae al suelo, tanto al cogerla como al transportarla, el robot deberá comenzar de nuevo desde la línea de salida perdiendo los puntos de ese recorrido.
- e** La lata deberá ser depositada pasada la línea de salida y sin que toque a esa línea. Si la lata toca la línea de salida o se queda antes de llegar a esta, se perderán los puntos del transporte, pero no los de repostaje.
- f** Una vez depositada la primera lata, el robot deberá repetir el proceso para transportar la segunda lata.
- g** Para evitar que las dos latas interfieran en la zona de salida, una vez que el robot consiga transportar la primera y la suelte, ésta podrá ser retirada por uno de los miembros del equipo. No está permitido arrebatarse la lata al robot, sí retirarla una

vez que éste la ha depositado y ya no está en contacto con ella. Si el robot emprende la marcha hacia la segunda lata sin necesidad de que ninguna persona le retire la primera lata el robot acumula más puntos que si necesita que se la retiren.

Conclusión de la prueba

- a La prueba terminará cuando el robot haya conseguido transportar las dos latas a la línea de salida.
- b Una vez depositada la segunda lata, el robot se detendrá de forma automática.

4.4.5. Puntuación

- a Cada repostaje efectuado valdrá 5 puntos.
- b Cada lata transportada correctamente valdrá 30 puntos.
- c Si el robot se detiene él sólo al terminar la prueba, suma 20 puntos.
- d Si se termina la prueba antes del tiempo máximo estipulado en el apartado 2.2.4 se obtendrá una bonificación añadiendo tantos puntos como segundos resten a la diferencia entre el tiempo máximo y el tiempo empleado.
- e Si el robot no necesita que le retiren la primera lata para cambiar el sentido de la marcha y encaminarse hacia la segunda lata, suma 20 puntos.
- f Si el robot deposita las latas verticalmente, suma 5 puntos por cada lata.

4.5. Prueba B: Fútbol 2+2

4.5.1. Objetivo

El propósito de esta prueba es realizar un pequeño partido de fútbol entre dos equipos formados por dos robots móviles cada uno,

mediante el uso de un campo y pelota de juego especiales que permitan la orientación de los robots y la localización de la pelota en todo momento.

4.5.2. Campo de juego

Tamaño

- a Las dimensiones totales del campo de juego serán de 183 cm por 122 cm.

Figura 4.3: Campo de fútbol

- b Como se muestra en la figura 2.3 cada esquina estará formada por un triángulo rectángulo con una hipotenusa de 8cm de largo.

Suelo

- a El campo de juego estará cubierto mediante un papel en el que estará impreso un degradado de grises, del negro al blanco, en tonos mate.
- b El campo de juego estará situado de forma que se encuentre plano y nivelado, pudiendo estar encima de una mesa o en el suelo.
- c La organización de Robocampeones-2006 suministrará a cada centro participante un papel impreso con el degradado de grises para construir un campo de pruebas.

Figura 4.4: Campo Soccer

Paredes

- a El campo estará delimitado por paredes de madera (aglomerado, contrachapado...) a lo largo de todo su perímetro, incluso por detrás de las porterías.

- b La altura de las paredes será de 20 cm.
- c Las paredes estarán pintadas en su parte interior mediante pintura negra mate.

Porterías

- a El ancho de cada portería es de 45 cm y estará centrada en cada uno de los lados más cortos del campo
- b Las paredes y el suelo del interior de cada portería estarán pintadas con pintura gris mate (75 % de negro y 25 % de blanco).

Bola de juego

Figura 4.5: Bola emisora de infrarrojos para el fútbol

- a La bola de juego es una esfera de plástico de unos 7 cm de diámetro que emite luz infraroja de una forma más o menos uniforme (figura 2.4).
- b La bola se alimenta de forma autónoma mediante unas baterías recargables alojadas en su interior.
- c La bola dispone de un interruptor para su encendido y de un LED verde que nos indica el correcto estado de las baterías.
- d La organización suministrará a todos los participantes una bola de estas características para poder realizar las pruebas y ajustes necesarios de los robots.

Puntos muertos

- a Se situarán en el campo de juego un total de 5 puntos muertos.
- b Uno de ellos está en el centro del campo.

- c Los otros cuatro restantes estarán próximos a cada una de las esquinas, situados a 45 cm de cada portería y alineados con cada uno de los extremos de la misma.
- d Los puntos muertos o neutrales son posiciones del campo donde el árbitro situará los robots o la bola de juego en el caso de que se produzca una interrupción del juego” (Ver la sección 2.3.4).
- e Los puntos neutrales estarán marcados mediante pequeños círculos de color verde y de 1 cm de diámetro como máximo.
- f La bola de juego se situará en uno de los puntos neutros de la zona de gol (el más próximo) si sucede una interrupción del juego en el área de gol.
- g La bola de juego se situará en el punto neutro central si sucede una interrupción del juego en el área central.

Condiciones de iluminación

- a Los equipos deberán venir preparados para ajustar los robots en función de las condiciones de iluminación de la sala.
- b La organización procurará que los niveles de iluminación del campo sean lo más bajos y homogéneos posibles.
- c Resulta muy aconsejable que los participantes diseñen sus robots para que puedan funcionar en condiciones de iluminación variables.

4.5.3. Robots

Equipos

- a Un equipo estará formado por un total de 2 robots.
- b Uno de los robots ejercerá de portero y el otro de defensa/delantero.

Dimensiones y requisitos de construcción

- a Los robots serán medidos en su posición vertical y con todas sus partes móviles extendidas.
- b En esas condiciones, el robot debe poder introducirse en un cilindro de 22 cm de diámetro.
- c La altura máxima del robot será de 22 cm.
- d Los robots deberán estar marcados o decorados para que resulte fácil identificarlos como pertenecientes a un mismo equipo.
- e Los colores y materiales de la decoración de los robots no debe interferir en los sensores de lectura de luz de los otros robots. En particular, sus sensores de luz que no estén mirando al suelo deben tener el emisor tapado y configurado desde los programas como sólo lectura. De esta manera evita confundir a los otros robots cuando éstos buscan la luz de la pelota.
- f La pelota no podrá entrar más de 2cm en cualquier parte del robot que pueda ser usada para retener o pasar la pelota. Hay que tener especial cuidado con aquellas partes entrantes del robot en las que la pelota puede quedar retenida para avanzar con ella o para "chutar".
- g Cualquier incumplimiento de estos requisitos de construcción supondrá la descalificación automática del equipo.

4.5.4. Desarrollo del juego

Preparativos

- a Los equipos podrán acceder al campo de juego antes de comenzar la competición para realizar labores de calibración y verificación.
- b En la medida de lo posible se intentará que también se puedan realizar estas labores en los intermedios entre las distintas fases de la competición.

Comprobación de los robots

Figura 4.6: Comprobación de los robots

- a** Los robots serán comprobados por los jueces antes de tomar parte en la competición.
- b** Para comprobar correctamente que un robot cumple con las dimensiones especificadas en el apartado 2.3.3, el robot deberá permanecer en su posición de funcionamiento con las partes móviles extendidas a su máxima longitud.
- c** Se comprobará explícitamente que los sensores de luz susceptibles de confundir a otros robots tienen el emisor tapado y configurado desde los programas como sólo lectura. De esta manera se impide interferir con otros robots cuando éstos buscan la luz infrarroja de la pelota. No es necesario tapar el sensor de luz que apunta al suelo, porque no es susceptible de confundir a otros robots.
- d** Si los robots son modificados durante el desarrollo de la competición, es obligación del equipo solicitar una nueva verificación por parte de los jueces.
- e** Ningún robot que no haya sido verificado por los jueces podrá tomar parte en la competición. En caso contrario será descalificado de forma inmediata.

Duración del partido

- a** Cada partido constará de dos tiempos de 5 minutos cada uno.
- b** Se establecerá una pausa entre cada uno de los tiempos, lo justo para cambiar de campo y de programa (quien lo necesite).
- c** Los equipos que no estén presentes en el momento de comienzo del partido serán penalizados con un gol en contra por cada minuto de retraso.

- d Si un equipo no se presenta transcurridos 5 minutos de la hora de comienzo de juego, perderán el partido con un resultado en contra de 5-0.

Comienzo del partido

- a Al comienzo del primer tiempo se procederá al sorteo de los campos mediante el uso de una moneda.
- b El equipo ganador del sorteo elegirá el campo y comenzará en ataque.
- c En el segundo tiempo se intercambiarán los campos y el que comenzó el partido en ataque comenzará el segundo tiempo en defensa.

Saque inicial

- a Cada tiempo comienza con un saque inicial desde el centro del campo.
- b Todos los robots deberán colocarse en su lado correspondiente del campo.
- c Todos los robots deberán estar parados.
- d La bola será colocada por el árbitro en el centro del campo.
- e Los robots del equipo que comienza jugando en defensa deberá colocarse a un mínimo de 30 cm de la bola.
- f El equipo que comienza jugando en defensa colocará los robots en el campo en primer lugar. Uno de los robots se colocará en la zona de la portería. Una vez situados los robots, no podrán ser reposicionados.
- g El equipo que comienza jugando en ataque pondrá un robot cerca de la bola.
- h El árbitro podrá modificar la colocación de los robots si así lo estima oportuno.

- i** A una señal del árbitro todos los robots serán puestos en marcha por un miembro de cada equipo.
- j** Cualquier robot que comience a moverse antes de la señal del árbitro, será retirado del campo y tratado como un “robot estropeado” (ver apartado 2.3.4).

Miembros del equipo

- a** En general, no se aceptará el movimiento de los robots por ninguna persona miembro del equipo.
- b** Un miembro del equipo sólo puede mover un robot bajo petición directa del árbitro.
- c** Antes del comienzo de cada tiempo, cada equipo deberá designar a uno de sus componentes para que actúe como capitán. Es el capitán el único que podrá arrancar, colocar, quitar y reemplazar los robots durante el partido, según las normas y las instrucciones del árbitro.
- d** Los demás miembros del equipo deberán permanecer separados del terreno de juego.

Movimiento de la bola

- a** Ningún robot puede “sujetar la bola”. Se entiende por “sujetar la bola” tomar control de la misma impidiendo que se pueda mover libremente para ningún lado.
- b** La bola no puede ser mantenida debajo de un robot.
- c** La bola tiene que estar visible en todo momento.
- d** El resto de los robots jugadores deberán poder acceder a la bola.

Porteros

- a** El portero podrá desplazarse lateralmente y avanzar.

- b Se define “zona de portería” como el rectángulo entre los postes y los dos puntos muertos más próximos. Si el robot portero se queda detenido fuera de esa “zona de portería” el árbitro lo considerará como “robot dañado” y podrá ser recolocado en la zona de portería.
- c Quedarse detenido dentro de la “zona de portería” es legal. En ese caso no se considerará al portero como “robot dañado”.
- d El portero no podrá permanecer en la zona interior de la portería (zona pintada en gris) más que el tiempo necesario para situarse.

Falta de progreso

- a Esto ocurre si la pelota se atasca entre múltiples robots, o entre uno o varios robots y la pared, o cuando el árbitro considera que la bola no tiene posibilidad de moverse.
- b También se produce una “falta de progreso” cuando la bola no es tocada por ningún robot en al menos 20 segundos y parezca que ningún robot tenga posibilidades de tocar la pelota.
- c Cuando se produzca una “falta de progreso” la bola será movida al punto muerto libre más próximo a la posición de la pelota.
- d Cuando los robots sean liberados mediante el uso de pequeños movimientos por parte del árbitro, los porteros serán mantenidos en la misma posición.
- e Cuando se produzca una situación de falta de progreso, cualquier robot situado tras la línea de gol será movido fuera del área de gol.

Robots dañados

- a Si un robot no se mueve durante un periodo de al menos 20 segundos y/o no reacciona ante la pelota, será considerado por el árbitro como un “Robot Dañado”.

- b** El árbitro o los jugadores pueden retirar un “Robot Dañado” del campo.
- c** Un “Robot Dañado” puede regresar con el permiso del árbitro a un punto neutral que este próximo al lugar de donde fue retirado y que no le suponga ninguna ventaja.
- d** Los porteros “dañados” regresarán al area de la portería.
- e** El juego puede continuar en el proceso de retirada, reparación y colocación de un “Robot Dañado”. El árbitro puede interrumpir el juego si el daño del robot se produjo por una colisión contra otro robot.
- f** Si un robot vuelca por sí solo, será considerado como un “Robot Dañado”. Si el robot se cae como consecuencia de una colisión contra otro robot o contra la pared, éste será levantado por el árbitro mientras el juego continúa.

Faltas

- a** El árbitro puede pitar “Falta” si un robot ataca (golpea) de forma continuada a otro robot que no está en posesión de la pelota. El capitán del equipo retirará el robot del campo de juego durante 30 segundos y procederá a corregir el problema. El juego continuará igual que si se tratase de un Robot Dañado”.
- b** Si el robot reitera la falta, será retirado de forma permanente del terreno de juego.
- c** Si un robot resulta dañado por una falta de un contrario, el árbitro parará el juego y el reloj mientras se realizan las reparaciones.
- d** No hay tiros libres, ni penaltis, ni fueras de juego.

Interrupción del juego

- a** La situaciones contempladas anteriormente pueden causar una interrupción del juego, normalmente el resultado será llevar la pelota a un “punto muerto” y reanudar el juego.

- b El juego puede ser detenido por el árbitro mediante un toque de silbato, pero el reloj del partido sigue corriendo. Se pararán todos los robots inmediatamente y se colocarán en la posición que tenían cuando sonó el silbato.
- c Una vez que el juego se haya parado, el juego se reanudará mediante una indicación del árbitro y todos los robots serán puestos en marcha simultáneamente.

Sustitución

La sustitución de cualquier robot en cualquier momento de la competición está estrictamente prohibida. Cualquier equipo o equipos que de forma intencionada realice cualquier sustitución de robots, será descalificados de la competición.

4.5.5. Puntuación y juego limpio

Goles

- a Se realiza un gol cuando la pelota cruza completamente la línea de la portería.
- c El árbitro hará sonar el silbato cuando se marque un gol.
- d Una vez que se consiga un gol, se produce un nuevo saque desde el centro del campo. El equipo goleado tendrá la primera posesión de la pelota.
- e Se considerará como gol válido cuando una pelota que lleve la trayectoria de entrar en la portería choque contra un robot que este parcial o totalmente dentro de la línea de gol.
- f Los goles en propia meta serán considerados como goles conseguidos por el equipo contrario, incluso si la pelota fuese empujada dentro de la portería.

Resolución de conflictos

- a Durante el desarrollo del juego las decisiones del árbitro son inapelables y tienen carácter definitivo.

- b** El capitán de cada equipo firmará la hoja de la puntuación al final del partido.
- c** Cualquier protesta que se haga conluido el partido solo podrá estar relacionada con una incorrecta interpretación de la hoja de puntuación.

Juego limpio

- a** Los robots que causen deliberadamente daños a otros robots, al campo de juego o a la bola, serán descalificados de la competición.
- b** Los participantes que causen deliberadamente interferencias en el funcionamiento o daños a otros robots, al campo de juego o a la bola, serán descalificados junto al equipo al que pertenezcan.
- c** Se espera de todos los participantes un juego limpio en todo momento.