

on11Introducciónchapter.1 ótica1.11Robóticasection.1.1 ótica.1.23Brebe historia
 de la robóticasection.1.2 on de robots1.34Clasificación de robotssection.1.3
 óticos1.3.14Brazos Robóticossubsection.1.3.1 ótica Móvil1.3.28Robótica
 Móvilsubsection.1.3.2 ía y plan de trabajo2.318Metodología y
 plan de trabajosection.2.3 on informática427Descripción
 informáticachapter.4 no Global4.127Diseño Globalsection.4.1 on4.227Chut de
 precisiónsection.4.2



INGENIERÍA INFORMÁTICA EN INFORMÁTICA DE SISTEMAS

Escuela Técnica Superior de Ingeniería Informática

Curso académico 2009-2010

Proyecto Fin de Carrera

Componente BICA para la realización de chuts precisos en el
robot NAO

Tutor: Francisco Martín Rico

Autor: Eduardo Parada Pardo

Agradecimientos

Quiero dar las gracias a todos los miembros del Grupo de Robótica de la Universidad Rey Juan Carlos por su apoyo y colaboración.

También agradecer a Francisco Martín Rico por el trabajo, dedicación, horas y horas revisando la memoria y ayuda que me ha prestado durante todos estos meses de trabajo.

Agradecer mis amigos Tamy, Jas, Carlos, Fran y demás amigos que me han apoyado para acabar el proyecto.

Y por último, a mis padres y familiares que me han animado a continuar con la carrera y el proyecto.

A todos... ¡Muchas gracias!

Resumen

Mi proyecto se encuadra en el campo de la robótica , más concretamente en la robótica móvil. Dentro de la robótica móvil, el desarrollo de este proyecto viene enfocado en el fútbol robótico, y está enfocado en el control de los actuadores para producir una actuación precisa.

Actualmente el fútbol robótico cuenta con grandes deficiencias aun no programadas, entre ellas un chut que permita a los robot enviar el balón a un punto preciso del campo.

El objetivo es crear un chut de precisión controlando los actuadotes de una manera precisa para poder llevar el balón a un determinado lugar y con una fuerza determinada.

El proyecto comprende 2 grandes puntos: el primero es la integración del software creado en la arquitectura BICA. El segundo es la capacidad de crear un mecanismo que permita una precisión en el chut asegurado de antemano.

Índice general

1. Introducción	1
1.1. Robótica	1
1.2. Breve historia de la robótica.	3
1.3. Clasificación de robots	4
1.3.1. Brazos Robóticos	4
1.3.2. Robótica Móvil	8
1.4. Robocup	12
2. Objetivos y requisitos	17
2.1. Objetivos.	17
2.2. Requisitos	17
2.3. Metodología y plan de trabajo	18
3. Entorno y plataforma de desarrollo	21
3.1. Simulador Webots	21
3.2. Robot Nao	22
3.3. NaoQi	23
3.4. Arquitectura BICA	25
3.5. JManager	26
4. Descripción informática	27
4.1. Diseño Global	27
4.2. Chut de precisión	27
4.2.1. Algoritmo matemático	29
4.2.2. Implementación	34
4.2.3. Experimentación	35
5. Conclusiones y Trabajos futuros	40
5.1. Conclusiones	40
5.2. Trabajos futuros	41

Índice de figuras

1.1. Brazo robótica de la fabrica de coches Citroën	5
1.2. Brazo robótica de un hostipal	6
1.3. Rehabilitación con brazos robóticos	7
1.4. Dextre, brazo robótica de la ISS	7
1.5. Robot Lunar Spirit	8
1.6. Robots tipo Piernas/patas (BigDog)(Izq.) y UVG (Derecha)	10
1.7. Sky Warrior (Izq.) y Sentry robot de exploracion submarina (Derecha)	10
1.8. ASIMO	11
1.9. Robocup-2005	12
1.10. Robocup-2006	13
1.11. Robocup-2007	13
1.12. Robocup-2008	13
1.13. Robot durante una prueba de la competición	14
1.14. Partido de la Robocup Soccer	15
1.15. Campo de la Robocup Soccer (Datos en mm.)	16
2.1. Modelo en espiral basado en prototipos.	19
3.1. Entorno de Webots	22
3.2. Actuadores de Nao	23
3.3. Distribución de trabajo del MainBroker	24
3.4. Composición interna del MainBroker	24
3.5. Herramienta Jmanager	26
4.1. Diferentes articulaciones del pie.	28
4.2. Vista Cenital Nao	29
4.3. Vista Horizontal Nao	30
4.4. Recta de Chut	31
4.5. Circunferencia de golpeo	32
4.6. Intersección de rectas	33

4.7. Desplazamiento de pie	34
4.8. Chut de 90 grados	37
4.9. Chut de 90 grados completado	37
4.10. Chut de 45 grados inicio	38
4.11. Chut de 45 grados completado	38
4.12. Chut de 45 grados, vista cenital	39
4.13. Chut de 45 grados, vista cenital completado	39

Capítulo 1

Introducción

Mi proyecto trata de resolver unos de los grandes problemas de los robots bípedos, el equilibrio. Es en el campo de la robótica móvil donde se centra mi proyecto, dicho proyecto pretende solventar una de las deficiencias existentes en este área.

En el campo de la robótica se diferenciaran 2 grandes ramas, la robótica móvil y los brazos robóticos. En nuestro caso nos centraremos en la robótica móvil. Dentro de la robótica móvil se centra en el fútbol robótico, con el propósito de crear un software destinado a mejorar el funcionamiento del robot Nao.

Este proyecto está destinado a crear un sistema integrado dentro de la arquitectura software que permita realizar una acción (golpeo del balón) de la forma más precisa posible. Esto permitirá conseguir tácticas más eficientes durante el partido a la hora de crear jugadas o chutes más difíciles de parar.

Las principales dificultades de este proyecto son: En primer caso, que este nuevo componente creado se incorpore a la arquitectura BICA. Y en segundo caso, la complejidad de conseguir que los actuadores del robot trabajen con la máxima precisión para llevar a cabo un Chute de estas características.

1.1. Robótica

La robótica es un campo amplio de investigación. Existen robots que desarrollan labores de limpieza, salvamento, construcción, etc.

Existen numerosas definiciones de la palabra robótica:

- **RAE:** *La robótica es la ciencia y la tecnología de los robots. Se ocupa del diseño, manufactura y aplicaciones de los robots. La robótica combina diversas disciplinas como son: la mecánica, la electrónica, la informática, la inteligencia artificial y la ingeniería de control. Otras áreas importantes en robótica son el álgebra, los autómatas programables y las máquinas de estados.*
- **UMSAAC:** *La Robótica es una ciencia o rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas realizadas por el*

ser humano o que requieren del uso de inteligencia. Las ciencias y tecnologías de las que deriva podrían ser: el álgebra, los autómatas programables, las máquinas de estados, la mecánica o la informática.

Resumiendo, la Robótica es la ciencia que desarrolla desde el comportamiento del robot en el mundo real, hasta su diseño físico. El robot es la plataforma hardware sobre la que diseñaras el software del mismo.

Para comprender la robótica debemos entender las distintas partes de las que se compone dicho robot (ya sea de ruedas, humano, animal,). Todo robot, poseerá:

1. Un armazón donde descansara todo la estructura principal del robot.
2. Sensores o captadores de estímulos. Dichos sensores le servirán al robot para desenvolverse adecuadamente por el mundo y poder captar los estímulos de alrededor. Estos sensores son unas de las partes más importantes del robot, ya que gracias a ellos el robot podrá conocer el mundo que le rodea y poder usar estos datos para poder crear mapas, evadir obstáculos,.. entre otros.
3. Motores y actuadores. Permitirán al robot desplazarse por el terreno. La precisión de estos motores es crucial dependiendo del tipo de movimiento que se desee realizar, en mi caso, la precisión y fuerza con la que se mueven y ejercen fuerza estos motores es decisiva para poder realizar un chut perfecto.
4. Por último, la CPU del robot, con la que controlará todas las funciones del robot. Un buen software robótico debe ser capaz de procesar el mundo que le rodea, para así poder comprender el mundo que le rodea. Posteriormente, razonar, con el fin de utilizar la información guardada al responder preguntas y obtener nuevas conclusiones. Dicho software debe ser capaz de tener autoaprendizaje, para que se adapte a nuevas circunstancias.

Este capítulo se divide de la siguiente manera:

1. Introduciremos los antecedentes de la robótica. En la sección 1.2 explicaremos desde sus orígenes hasta la actualidad.
2. En la seccion 1.3 clisificaremos los robots dependiendo del grado de libertad de los mismos.

1.2. Breve historia de la robótica.

A lo largo de la historia, el hombre se ha sentido fascinado por las máquinas y dispositivos capaces de imitar las funciones y movimientos de los seres vivos. Los griegos tenían una palabra específica para denominar las máquinas: autómatas, (esta palabra derivó de lo que actualmente se conoce como autómeta). Durante toda la historia se han ido desarrollando cada vez más complejos autómatas, desde el León mecánico de Leonardo da Vinci 1499, el jugador de ajedrez *w. Von Kempelen 1769*, la muñeca mecánica capaz de dibujar (*H. Maillander, 1805*), hasta el considerado padre de la robótica industrial (*George Devol*).

George Devol fue un inventor estadounidense que buscó diseñar una máquina flexible, adaptable al entorno y de fácil manejo. De esta forma *George Devol*, patentó en 1948 un manipulador programable que a posteriori fue el prototipo del primer robot industrial.

En 1954 *George Devol* tubo la idea de crear el primer robot programable. En 1956, *Joseph Engelberger*, director de ingeniería aeroespacial de la empresa *Manning Maxwell y Moore*, coincide con *George Devol* en una cena. Ambos deciden crear la primera compañía fabricante de robots, fundando la *Consolidated Controls Corporation*, que más tarde se convierte en *Unimation (Universal Automation)*.

La palabra robot fue usada por primera vez en el año 1921, cuando el escritor checo *Karel Capek* estrena en el teatro nacional de Praga su obra *Rossums Universal robot*. Esta palabra posiblemente hubiera acabado en desuso, si no hubiese sido por los escritores de ciencia ficción, que usaron dicha palabra para referirse a los autómatas. Pero sin duda el autor que hizo que la palabra robot llegara a consolidarse fue el escritor americano de origen ruso *Isaac Asimov* donde en uno de sus más famosas obras *Yo Robot*, enunció las 3 leyes de la robótica:

1. *Un robot no debe dañar a un ser humano o, por su inacción, dejar que un ser humano sufra daño.*
2. *Un robot debe obedecer las órdenes que le son dadas por un ser humano, excepto si estas órdenes entran en conflicto con la Primera Ley.*
3. *Un robot debe proteger su propia existencia, hasta donde esta protección no entre en conflicto con la Primera o la Segunda Ley.*

1.3. Clasificación de robots

La robótica se puede dividir, atendiendo a las capacidades de motrices, en dos tipos:

- Robótica móvil.
- Brazos robóticos.

Se presenta de esta manera diferenciando principalmente que en los brazos robóticos, el área de trabajo esta delimitado por el alcance de los brazos y en la robótica móvil su área de trabajo es hasta donde le permita desplazarse el método de locomoción que posea.

1.3.1. Brazos Robóticos

En un principio, donde más se aplicó la robótica fue en la industria, donde se crearon los primeros brazos robóticos. Dichos brazos fueron creados para manipular elementos peligrosos sin riesgo para el operador. Este sistema consistía en un dispositivo maestro-esclavo donde el operador estaba situado tras un grueso cristal desde donde controlaba el brazo mecánico.

Existe una importante diferencia entre los brazos robóticas y la robótica móvil. Los brazos robóticas, en su mayoría, presentan una mayor precisión en las labores que desempeñan, pero tienen en el problema que su radio de acción es fijo. Mientras que para los robots móviles, carecen de este espacio y pueden desplazarse por el mundo para realizar las acciones para las que haya sido programado. Existen diversos robots, con forma de vehículo, animal, humano pero todos tienen una característica en común, su CPU.



Figura 1.1: Brazo robótica de la fabrica de coches Citroën

En 1954 hizo uso de la tecnología electrónica y el servo control sustituyendo la transmisión mecánica por otra eléctrica y desarrollando así el primer telemanipulador con servo control bilateral. Posteriormente, con la creciente evolución de los computadores, los operarios fueron sustituido por un ordenador, consiguiendo una precisión perfecta para realizar trabajos. Fue en el 1972 cuando Japón creó el JIRA (Asociación de robótica industrial de Japón), significando un adelanto enorme en este campo. Uno de los múltiples brazos robóticos que se usan en las fábricas de coches (Figura 1.1) es capaz de desplazar una tonelada de carga hasta más de seis metros de distancia. Pero no solo posee potencia, también posee una gran precisión: fue pensado para la colocación y manejo de piezas tan delicadas como bloques de motores íntegros, secciones de metal y cristalería.

Tras la creación del primer robot orientado a ensamblar piezas de forma automática (SCARA), se inició una carrera por la evolución de la robótica. Tras 30 años, la robótica ha llegado a un punto donde los robots se usan para toda clase de aplicaciones, tanto terrestres como submarinas y subterráneas.

Continuando con la evolución progresiva de los robots, se fue consiguiendo poco a poco una mayor precisión y fiabilidad en el movimiento de los brazos robóticos hasta el punto de ser usados en la medicina.

Es una herramienta más, pero inteligente, ya que trata de compensar las deficiencias y limitaciones que pueda tener el cirujano para realizar ciertas actuaciones. De este modo, se hace posible la implantación de algunas técnicas de cirugía mínimamente invasiva gracias a la utilización de ayudas de soportes robotizados, consiguiendo

minimizar la herida, reducir el tiempo de intervención y el de posterior recuperación.

La idea de usar **robots en el campo de la cirugía**, surgió hace muchos años en el Departamento de Defensa de los E.E.U.U., que buscaba desarrollar máquinas que pudiesen transmitir movimientos y conocimientos a distancia para utilizarlas en el campo de batalla.



Figura 1.2: Brazo robótica de un hospital

Hoy en día la cirugía (Figura 1.2) robótica es una realidad, que permite al médico operar desde la habitación contigua o bien a muchos kilómetros del paciente. Así se habla del telecirugía y del tele diagnóstico.

La Robótica no solo se usa en los quirófanos, sino que se está también implantado en proyectos muchos menos complejos y caros como la rehabilitación de pacientes con ingenios robóticas (miembros artificiales, robots de soporte a las terapias de rehabilitación (Figura 1.2) o robots para proveer asistencia personal en hospitales).

La robótica en la medicina no solo ha quedado anclada en los hospitales con grandes presupuestos, sino que ha llegado hasta las propias farmacias, donde algunas llegan a tener un complejo sistema robótica que funciona a modo de dispensador para las personas que atienden las farmacias.



Figura 1.3: Rehabilitación con brazos robóticos

Otro importante campo donde esta evolucionando a gran velocidad es biónica, donde se están diseñando brazos y piernas robóticas con las que un humano puede llegar, desde poder recuperar la movilidad en un miembro perdido hasta poder ejercer una presión casi exacta sobre un objeto.

La robótica de brazos fijos alcanzó uno de sus momentos más altos con la creación del brazo robótico **Dextre** (Figura 1.4), anclado en la Estación Espacial Internacional, que reúne las capacidades más avanzadas de la robótica fija con las características de poder aguantar en una situación tan adversa como el espacio.



Figura 1.4: Dextre, brazo robótica de la ISS

La clave de la robótica móvil para alcanzar el mercado de la robótica industrial no

es que estos tengan la capacidad de realizar operaciones de elevada potencia, sino que posean alta movilidad.

En este proyecto nos centramos en la robótica móvil, pero dentro de la robótica móvil, como ya he explicado anteriormente, habrá varios tipos, en nuestro caso nos centraremos en la robótica humana.

Actualmente la robótica empieza a impartirse en el campo de educación, cada vez de una manera mas temprana, ya que, ya no solo disponen de robótica en las instituciones de mas nivel educacional como las universidades, sino que en institutos se dispone de pequeños robots programables para empezar a desarrollar las primeras inteligencias artificiales.

1.3.2. Robótica Móvil

Un robot móvil es aquel robot que es capaz de trasladarse por el entorno para poder realizar una acción. Dentro de los robots que componen la robótica móvil, una de las áreas donde los robots son imprescindibles (por la imposibilidad de los humanos de poder permanecer muchos meses en otros planetas) son los robots lunares. Como el Spirit (Imag. 1.5), este robot actualmente esta operativo en al superficie marciana y recoge muestras, que el propio robot analiza.



Figura 1.5: Robot Lunar Spirit

La robótica ha evolucionado a un punto donde es obligatorio su presencia por las características que poseen los robots, como por ejemplo: en la actualidad los robots bombero extienden el fuego en lugares donde las temperaturas impiden el paso de humanos. Otro caso donde es necesaria la robótica es el anteriormente mencionado robot marciano Spirit (Figura 1.5). Un último ejemplo donde la robótica es imprescindible es en caso de los robots submarinos que reparan cables o prospecciones petrolíferas a alta profundidad. Por último, otro caso en el que los humanos requieren de estos robots es las operaciones de reparación de tuberías de petróleo a grandes profundidades marinas.

La robótica móvil plantea una serie de problemas en comparación con los brazos robóticos:

- **La localización:** Es la capacidad de un robot de posicionarse en lugar. Para los robots móviles, la localización es compleja, ya que están obligados a constantemente buscar referencias de donde se encuentran para posicionarse en un mapa.
- **La navegación:** Es la capacidad de un robot de ir desplazándose por el terreno conociendo en todo momento el recorrido que está realizando. Este es uno de los grandes problemas que tiene la robótica móvil, aunque los robots móviles de tipo ruedas poseen una odometría bastante fiable, en el caso de las piernas es mucho más compleja e inexacta, por lo cual es más difícil para el robot localizarse en un mapa.
- **Interacción:** Ya que el robot móvil comparte el entorno con los seres humanos, ha de convivir y relacionarse con ellos. Por ello el robot móvil debe interactuar con el mundo para solventar diversas situaciones, para ello debe ser capaz de usar todos sus actuadores y sensores para poder interactuar con el entorno que le rodea. Ya sean puertas, escaleras, etc.
- **Control:** Es la capacidad del robot de manejar con precisión todos sus sensores y actuadores. Aunque este área debe ser muy precisa para los 2 tipos de robótica, se ve más acentuada en la robótica móvil, ya que no solo deberá controlar con precisión el actuador en cuestión, sino toda la estabilidad del cuerpo con todos los actuadores trabajando simultáneamente.

Atendiendo a los actuadores que un robot móvil posee. La movilidad en la robótica se puede dividir en 4 grandes bloques.

- A través de ruedas, orugas o sistemas similares. UGV (Unmanned Ground Vehicle) Fig.1.6
- Piernas y patas. Fig.1.6
- Robótica Subacuática. AUV (Autonomous Underwater Vehicle) Fig.1.7
- Robótica aérea/espacial. UAV (Unmanned Aerial Vehicle) Fig.1.7

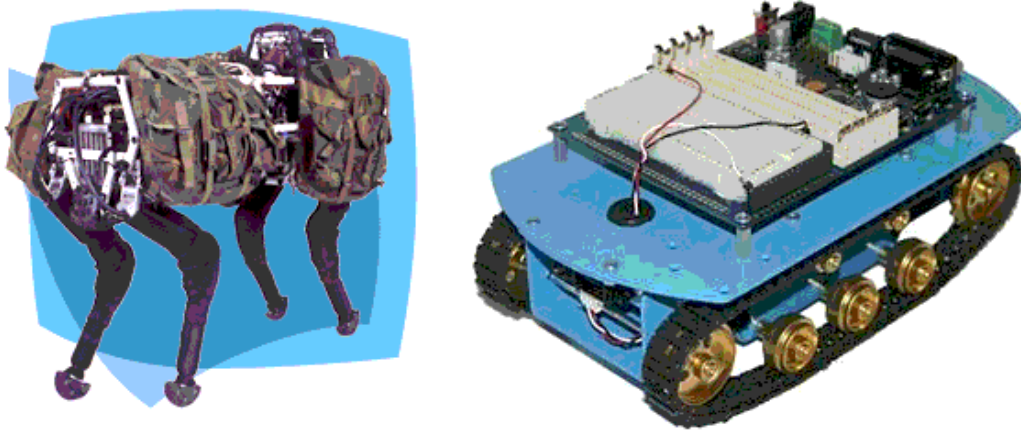


Figura 1.6: Robots tipo Piernas/patas (BigDog)(Izq.) y UGV (Derecha)

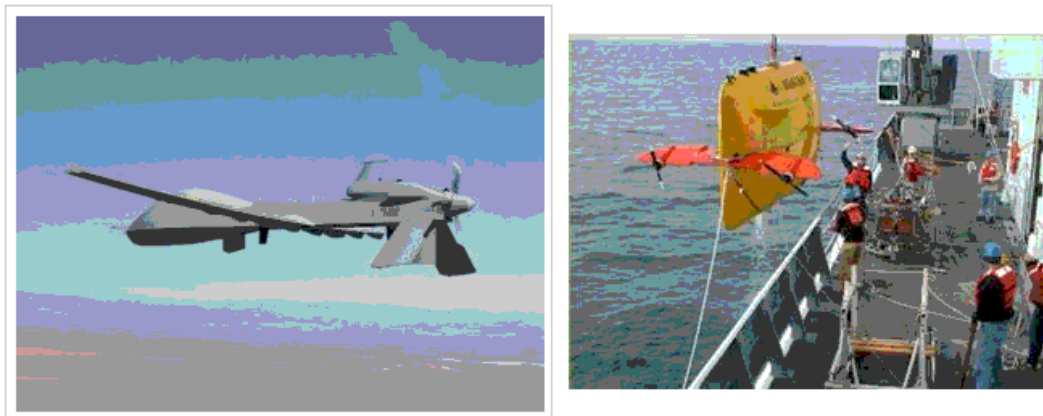


Figura 1.7: Sky Warrior (Izq.) y Sentry robot de exploracion submarina (Derecha)

Dentro de estas 4 grandes ramas, nosotros nos centraremos en la robótica terrestre. La robótica terrestre existen 2 grandes ramas: *robots con ruedas (UGV)* y con *piernas ó patas*.

1. *Piernas ó patas*: La capacidad para poder transportar al robot por cualquier superficie, ya sea lisa ó abrupta. Pudiendo subir escalones o pasar por encima de obstáculos de grandes dimensiones.

Los robots equipados con piernas ó patas, pueden llegar desde realizar saltos hasta poder transportar al robot por tierra o agua, siendo imposible esto en un sistema de ruedas a no ser que no sean ruedas normales que le permitieran navegar.

2. *Ruedas*: El sistema de odometria es mucho más fiable y precisión en un sistema de ruedas que en uno piernas.

La velocidad a la que puede llegar un sistema de ruedas es mucho más rápida. Además la estabilidad que posee un sistema de ruedas es muy superior al que se pueden conseguir con piernas.

Los primeros diseños de robot móviles fueron de ruedas, ya que conseguir una estabilidad en piernas/patas es extremadamente difícil.

En este proyecto nos centraremos en la robótica de piernas en particular, en el control de los actuadores, ya que son estos los que marcan la diferencia a la hora de realizar un movimiento complejo de una forma óptima.

Actualmente existen grandes robots en el campo de la robótica terrestre (más concretamente en la de piernas) como ASIMO (Figura 1.8). Estos robots integran, no solo una capacidad motriz ejemplar, sino una compenetración con otros sensores y actuadores que le permiten realizar un movimiento similar al de un humano (en el caso de los robots bípedos).

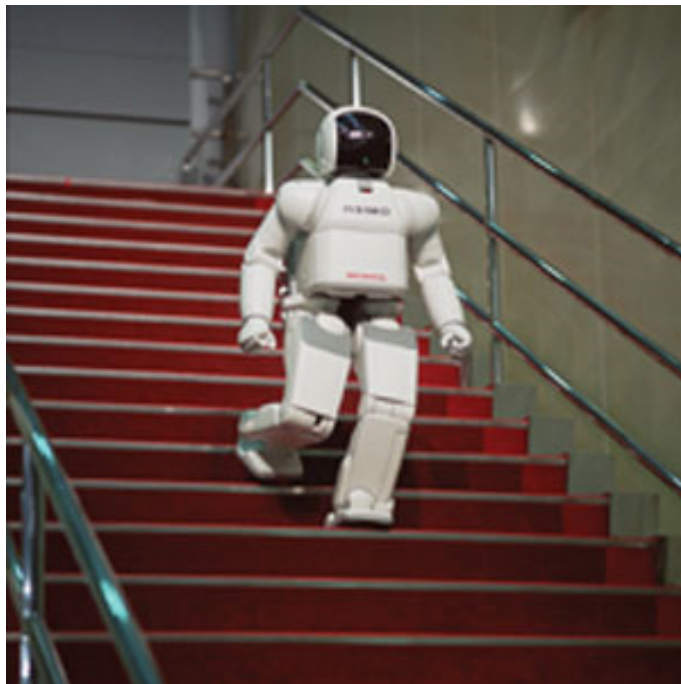


Figura 1.8: ASIMO

1.4. Robocup

Existen diversos robots humanoides actualmente destinados a la investigación, uno de ellos es Nao. Este robot es la plataforma estándar que se presenta para poder participar en la robocup.

Robocup es un proyecto internacional para promover, a través de competencias integradas por robots autónomos, la investigación y educación sobre inteligencia artificial.

La Robocup se creó para fomentar el desarrollo de la robótica a través de una serie de competiciones que llevarán al desarrollo de robots con una fuerte inteligencia artificial. Dicho proyecto se decidió dividir entre varias secciones según el campo que se quiera potenciar.

Una de las metas de impulso la Robocup fue que para el año 2050, desarrollar un equipo de robots humanoides completamente autónomos que puedan ganar, de acuerdo con las reglas oficiales de la FIFA, contra el ganador de la copa mundial de fútbol.

Durante los últimos 13 años, se han ido llevando a cabo las Robocup en diferentes países. Desde su inauguración en Nagoya en el año 1997, se han ido sumando cada año más equipos en la Robocup.

Este evento se inició en 1995 el nacimiento de la Robocup que 2 años después abrió sus puertas en Japón en 1997.

1993: Estudio de factibilidad 1995: Anuncio oficial de Robocup

1997: Robocup-97; Nagoya 1998: Robocup-98; Paris

1999: Robocup-99; Stockholm 2000: Robocup-2000; Melbourne

2001: Robocup-2001; Seattle 2002: Robocup-2002; Fukuoka/Busan

2003: Robocup-2003; Padua 2004: Robocup-2004; Lisbon

2005: Robocup-2005; Osaka



Figura 1.9: Robocup-2005

2006: Robocup-2006; Bremen



Figura 1.10: Robocup-2006

2007: Robocup-2007; Atlanta



Figura 1.11: Robocup-2007

2008: Robocup-2008; Suzhou



Figura 1.12: Robocup-2008

2009: Robocup-2008; Austria

La Robocup se divide en varias ligas:

Estándar League.

1. **Robocup Rescue** En esta sección los robots deberán completar una serie de pruebas de rescate (Figura 1.13). Dichas pruebas podrán ser llevadas a cabo de manera completamente autónoma o teledirigida.

Durante las pruebas los robots deberán llegar a diferentes secciones del terreno, de difícil acceso, y realizar tareas de salvamento o llegar hasta otra zona, completando así una ruta.



Figura 1.13: Robot durante una prueba de la competición

2. **Robocup Home** Esta competición se centra en la robótica orientada a realizar tareas hogareñas y de ayuda a facilitar la vida a los humanos. La competición se centrará en un conjunto de pruebas que evaluarán las habilidades de los robots para realizar tareas de un entorno realista.

Pero esta competición no solo se centra en ámbitos hogareños, sino que también se enfoca a interacción robot-humano, navegación, cartografía en entornos dinámicos, visión computacional, reconocimiento de objetos en condiciones naturales, manipulación de objetos, integración de comportamiento e inteligencia artificial.

3. **Humanoide**: Esta liga se introduce en el 2002 y se participa con robots bípedos autónomos. Se subdivide en 2: Kid-size y Teen-size.
4. **Tamaño medio**: Son robots con ruedas, con un máximo de tamaño de 50 cm y los robots que participan no son autónomos.
5. **Tamaño pequeño**: Esta liga se juega con un máximo de hasta 5 robots con un tamaño máximo de 18 cm. Todo el procesamiento es efectuado en un ordenador central, basado en información (posición de los jugadores, del balón, etc.). Dicha información es recogida desde una cámara alojada en el techo del campo.

6. Simulador: Se juega en un campo virtual. Esta liga se subdivide en : 2D,3D y realidad mixta.

7. Standar League

Hasta el 2009 se había usado como plataforma estándar el Sony AIBO (Robot con aspecto de perro). La interrupción de la fabricación en el 2006 de estos robots, provocó un cambio de robot. Se decidió pasar a una plataforma humana, denominada Nao de la compañía Aldebaran. Gracias a tener una plataforma estándar se consigue que el equipo que juegue mejor sea porque posee un software para llevar a cabo las tareas necesarias para jugar al fútbol, superior al de los demás, sin tener en cuenta el hardware.

Esto consigue que no ocurra como en otras competiciones, donde al no haber un estándar de plataforma, era muy probable que el que ganara fuera porque dispusiera de mayor presupuesto. Como por ejemplo si un equipo poseyera un gran presupuesto y pudiera usar cámaras mucho más precisas o actuadores más potentes o rápidos, lo cual desequilibraría los partidos de la Standar League (Figura1.14).

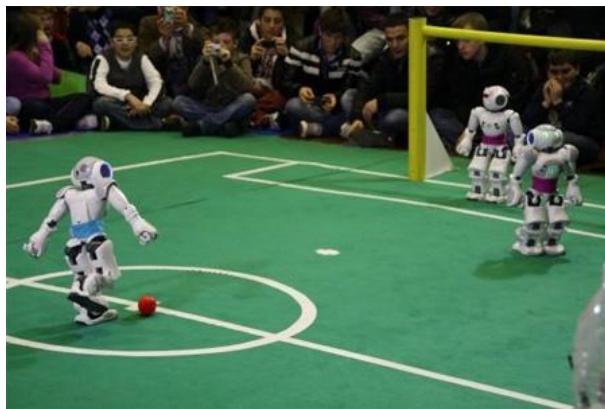


Figura 1.14: Partido de la Robocup Soccer

Actualmente la compañía Aldebaran continúa el desarrollo de este robot para conseguir una plataforma más robusta y fiable, por lo cual las instituciones que compiten en la Robocup Soccer no deben preocuparse de tener un hardware obsoleto o con posibles fallos.

El hecho de ser una competición de fútbol presenta muchos de los problemas que los robots tienen actualmente para poder desenvolverse en la vida real. Como por ejemplo la navegación localizada, la coordinación de las articulaciones del robot,

la capacidad de realizar varios comportamientos simultáneamente, poder percibir las diferentes partes y componentes del mapa.

Ya que poder jugar un partido requiere que los robots no solo consigan chutar y meter el balón en la portería, sino que requiere un equilibrio a una sola pierna, lo cual quiere un control del centro de masas. Además necesita poder comunicarse entre ellos a tiempo real mientras realizan otras operaciones. También es importante mencionar que los robots se pueden posicionar gracias a lo que ven entre ellos, lo que implica no solo comunicación entre ellos sino posicionamiento de los robots entre varios robots.

Las características del campo (figura 1.15), son las siguientes:

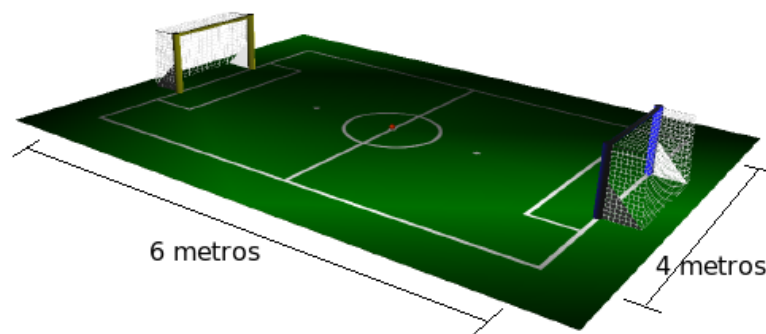


Figura 1.15: Campo de la Robocup Soccer (Datos en mm.)

El campo cuenta con una serie de marcas visuales con la que los robots deberán ser capaces de localizarse, este será uno de los principales problemas de los robots, ya que si los robots salen del campo serán penalizados.

Al inicio del partido los robots serán colocados en sus diferentes posiciones, una vez colocados los robots deberán asumir diferentes roles (portero, defensa, delantero). Si un portero sale de su área será penalizado. El hecho que un robot golpee a otro, se penalizará.

Durante el partido una de las causas que puede decidir el partido es la capacidad de estar compenetrados y poder los robots de cada equipo de interactuar con ellos, por ello, el hecho que un robot pueda pasar el balón a otro con la mayor precisión posible, puede marcar la diferencia en un partido. Este Chut de precisión, es lo que solucionara este importante punto a la hora de realizar tácticas avanzadas.

Esta liga es el área donde está centrado mi proyecto, ya que es la plataforma en la cual se ha realizado, es el robot Nao.

Capítulo 2

Objetivos y requisitos

El objetivo de este proyecto es conseguir una mayor eficiencia de los robots durante el partido. Para ello los robots contarán con un nuevo comportamiento que les permitirá llevar el balón a un punto concreto del campo.

2.1. Objetivos.

El sistema de chut de alta precisión ha sido creado para aportar una funcionalidad a la arquitectura BICA en el campo de la movilidad del robot. Este Chut esta pensado para conseguir marcar una diferencia entre robots, al conseguir una coordinación entre estos por medio de pases. Existe una gran importancia a la hora de controlar el comportamiento del robot, ya que en la mayor parte de los casos, los robots se centraban en alejar el balón lo más lejos posible de su portería. Esto simplemente conseguía alejar el balón de tu portería para que no marcaran gol. Para resolverlo, el robot ahora decide a donde quiere enviar dicho balón y conseguir así crear jugadas complejas y eficientes.

El proyecto se divide en varios subobjetivos:

1) El cuerpo del proyecto se centra en el chut, este buscará llevar el balón desde un punto del espacio (cuyas coordenadas 3D se han obtenido a través de la cámara del robot) hasta otro punto del espacio que se habrá marcado como destino.

2) El chut deberá ser capaz de controlar el ángulo con el que se debe golpear y la fuerza con la que se debe aplicar el chut para llegar a su posición final.

2.2. Requisitos

Los requisitos que debe cumplir este proyecto vienen marcados principalmente por la plataforma en la que se ha desarrollado:

- El sistema operativo GNU Linux (La distribución Ubuntu) la plataforma de trabajo webots.
- API de aldebaran (NaoQi)

- El propio Nao
- La arquitectura BICA, creada por el equipo de robótica de URJC con la cual mi software debe cumplir una serie de pre-requisitos para poder convivir con los demás componentes.

Todo el software se ha realizado en C++ y Java. El software creado tiene un importante componente matemático, con los que se han conseguido la precisión necesaria para que el software creado fuera óptimo.

En un principio se desarrollo la interfaz con la plataforma WxGlade, esta nos permitía desarrollar una interfaz que a través de un entorno sencillo, que permitía crear un entorno visual para poder darle ordenes al robot desde el entorno de simulación Webots. Dicho entorno usaba el lenguaje XML.

Otro requisito inicial fue aprender Phyton, ya que para poder lanzar los sistemas de prueba se usaba este lenguaje, que posteriormente se dejo de usar para usar otros sistemas para realizar pruebas.

Es importante decir que dicho proyecto se ha creado como un comportamiento que permite añadirse al software ya existen creado por el equipo de la URJC, para conseguir una potente evolución del software ya existente. La tarea de integrar un comportamiento creado por mi y enlazarlo con BICA ha sido de gran dificultad ya que este es un software de gran complejidad y evolución constante.

BICA, es una arquitectura en constante crecimiento. Esto conlleva que constantemente el software deba continuamente ser ajustado de nuevo a esta arquitectura, ya que muchas funcionalidades cambian y provocan errores de calculo el comportamiento que había creado.

No solo BICA esta constantemente cambiando, sino el propio NaoQi. Este cambia funcionalidades que afectan directamente a tu comportamiento por lo que debes amoldarte a cada cambio que se realiza. Por ejemplo, en últimas versiones varias funcionalidades relacionadas con el equilibrio han sido eliminadas, con lo que conlleva la creación de un comportamiento que ya existía.

2.3. Metodología y plan de trabajo

El desarrollo del proyecto se ha basado en un modelo de desarrollo basado en prototipos (Como se muestra en la figura 2.1).

Este sistema te permite dividir el proyecto en ciclos, todos ellos con un desarrollo similar y con un número máximo de ciclos. Durante todos estos ciclos el proyecto ha sido supervisado y controlado semanalmente por el tutor.

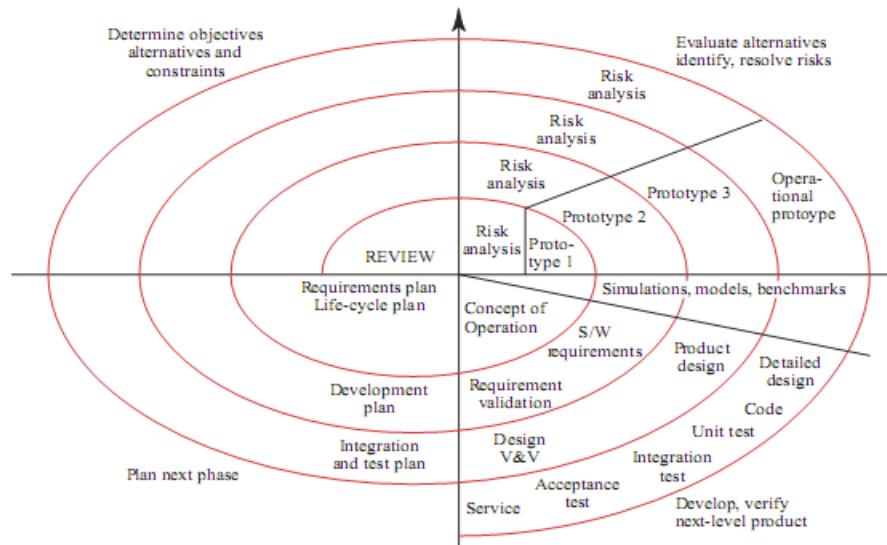


Figura 2.1: Modelo en espiral basado en prototipos.

1. Para realizar el proyecto se ha seguido un plan de trabajo.
 - Se inicio una formación de todos los lenguajes necesarios. Primero se estudio el lenguaje de programación C++ (que sería usado en el 90% del proyecto, ya que es el lenguaje elegido para programar el robot). También se necesitó aprender a programar en Phython, ya que era el lenguaje usado para crear los sistemas de pruebas. En un principio se empezó usando Ruby on Rails, en este lenguaje sólo se realizó una formación, pero posteriormente no se llegó a usar para el proyecto.

También se necesito una formación en Java, ya que el sistema de interfaz se creaba con Java para la plataforma NetBeans. Se requirió también una pequeña introducción al lenguaje XML para la creación del interfaz del Wxglade, Wxglade fue el sistema de interfaz de comunicación con Webots en un principio.

La plataforma usada para realizar el proyecto fue la distribución de Linux Ubuntu.

El API con el que tuve que familiarizarme para trabajar correctamente con el robot fueron las obtenidas a través del API de NaoQi.

- Interfaz del robot. La segunda parte del robot tras la formación y familiarización de entorno de Nao, fue el diseño de la interfaz para poder realizar los ejemplos simples con el simulador (Webots). Dicho entorno en un inicio fue realizado con el Wxglade y posteriormente con NetBeans

- Aprendizaje de moviendo de articulaciones y control del centro de masas del robot. Ya que el control del robot era de gran importancia para realizar correctamente el chut, se realizaron gran cantidad de pruebas en este punto.
- Búsqueda del algoritmo matemático que realizara el chut. Esta parte del proyecto engloba gran cantidad del tiempo del proyecto, ya que hasta que matemáticamente no fuera correcto el chut no seria llevado a código. De esta forma me aseguraba en gran medida que los errores no vinieran de un error en un ciclo del proyecto anterior.
- Implementación del algoritmo matemático a código. Durante esta fase, se realizaron ligeras modificaciones por las diversas limitaciones del robot.
- Ampliación del proyecto a su fase 2. Donde no solo realizaría el chut sino que además informaría al otro robot del pase y este otro robot debería prepararse para recibirlo.

Capítulo 3

Entorno y plataforma de desarrollo

En este capítulo explicare las plataformas usadas, lenguajes y programas usados para llevar acabo el proyecto de chut/pase de precisión.

Las plataformas software usadas son: El simulador físico webots, el modelador de interfaces Wxglade (que usa el lenguaje XML), posteriormente se uso el modelador de interfaces NetBeans (que usa el lenguaje Java).

3.1. Simulador Webots

Webots es un potente simulador de diferentes plataformas robóticas. Dicho simulador contiene un simulador físico que te permite realizar pruebas lo más parecido posible a el uso del robot en la realidad. Webots (Figura 3.1) es una plataforma que permite simular varios tipos de robots, entre ellos Nao.

El lenguaje elegido para trabajar sobre Webots es el C++, dicho lenguaje permite realizar potentes programas, ya que te permite crear todos los comportamientos en diferentes clases con lo que la legibilidad y potencia están más que asegurados.

El simulador contempla las medidas exactas (en proporción con el escenario) del robot real. Además también contempla los tamaños reales del campo de fútbol e intentan simular la física real de los robots, ya sea su peso o la fuerza con la que golpea la pierna al balón.

El simulador esta preparado para soportar partidos reales con gran cantidad de robots simultáneos en el campo, con lo que se podrá simular un partido real antes de llevarlo al mundo físico.



Figura 3.1: Entorno de Webots

Una vez lanzado el simulador Webots, se carga BICA sobre el Nao simulador y posteriormente a través del NetBeans, se lanza el interfaz que permite activar los distintos comportamientos del robot.

3.2. Robot Nao

En este punto explicamos las diferentes partes de las que se compone el robot Nao (Figura 3.2). Uno de las grandes cualidades de Nao es la gran cantidad de servos que posee y de la buena precisión de los mismos. Gracias a esto se ha podido llevar a cabo este proyecto, cuya finalidad es usar al máximo el potencial de los actuadores del robot.

Nao (Figura ??) tiene 25 grados de libertad, diversos sensores para interactuar con el mundo, disponemos de 2 cámaras situadas en la cabeza (con una resolución de 640x480 pixeles y de una frecuencia máxima de 30 FPS), 2 sensores de ultrasonidos en el pecho, sensores de presión (en la planta del pie) y pulsadores en la punta de los pies, 2 giroscopios, 3 acelerómetros, leds, 4 micrófonos, 2 altavoces, una conexión inalámbrica vía WIFI , un puerto USB y un puerto Ethernet (RJ45).

El robot es capaz de expresarse a través de los leds que tiene por su cuerpo y a través de sonido con el que puede reproducir palabras. Nao es capaz de entender palabras y reconocer caras.

La red wifi de la que dispone le permite comunicarse fácilmente entre los demás robots o con el mismo, esto es muy útil e importante a la hora de jugar un partido, ya que para informar donde se encuentra la pelota o otros jugadores, dará una gran ventaja a tu equipo.

Nao mide 57 cm. de altura y pesa 45 kg. de peso. Nao está presente actualmente en una versión para investigación pero se prevé que a principios del 2011, se presente una versión para la venta al público. La empresa que se ocupa del robot Nao es Aldebaran.

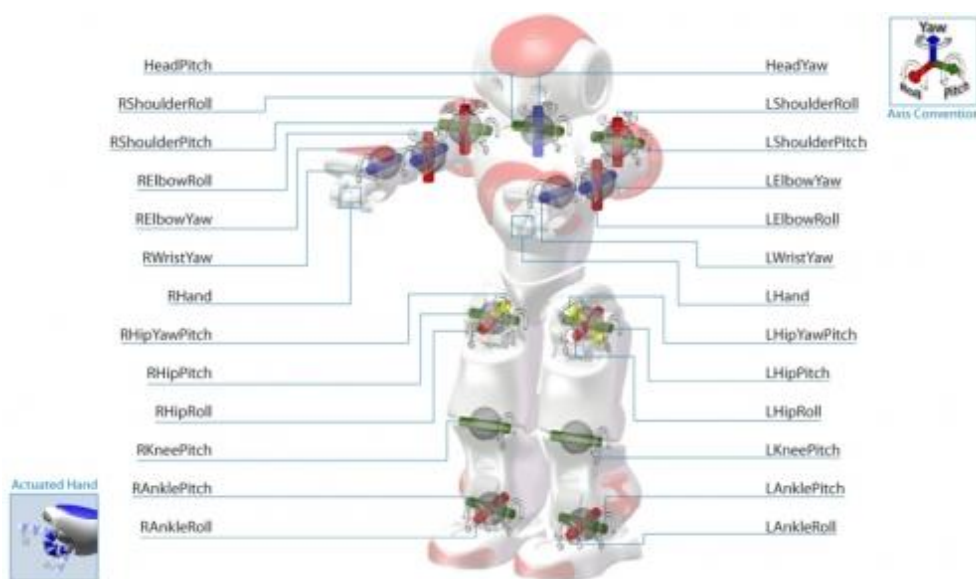


Figura 3.2: Actuadores de Nao

Por último, como hardware interno, Nao lleva integrado un microprocesador AMD Geode x86 a 500 MHz y 256 MB de SDRAM. Además cuenta con una memoria externa de 1 GB flash. En nuestro Nao hemos instalado en esta memoria externa un sistema operativo Linux (32 bits) de la distribución OpenEmbedded.

3.3. NaoQi

NaoQi es una capa software que nos proporciona Aldebaran Robotics ¹ para la comunicación y desarrollo del software de Nao.

Se basa en la arquitectura de cliente-servidor donde los módulos actúan como servidores. Los módulos pueden crearse como Brokers (ejecutables) o como bibliotecas. Dichos brokers permiten ejecutarse independiente al MainBroker principal, permitiendo

¹<http://www.aldebaran-robotics.com/>

de esta forma que si tu broker queda bloqueado, el sistema principal que controla todos los brokers no se colapse y por ejemplo se caiga el robot.

NaoQi soporta la programación en C++ o Python, y puede trabajar sobre diferentes plataformas como Windows, Linux o MacOS.

En la figura 3.3 y 3.4 se muestra como trabaja el MainBroker y de lo que esta compuesto el MainBroker.

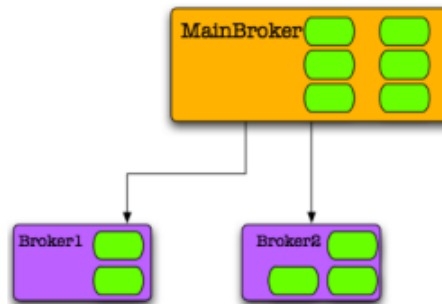


Figura 3.3: Distribución de trabajo del MainBroker



Figura 3.4: Composición interna del MainBroker

1. En mi proyecto principalmente se habrán usado ALMemory, DCM, NaoCam y sobre todo ALMotion.
 - **ALMemory:** es la pizarra compartida para la comunicación entre módulos. Garantizando la exclusión mutua. Este permite leer los sensores.
 - **ALMotion:** es el modulo principal encargado de la locomoción. Nos permite manejar todos los actuadores del robot, tanto su posición, ángulo y fuerza. Permite obtener la posición de la odometría del robot.
 - **NaoCam:** es el modulo encargado de comunicar con las cámaras del robots. Actualmente Nao posee 2 cámaras con una visión no superpuesta.

Nao puede ser programado con varios lenguajes de programación como C++, Python o Urbi. También está preparado para ser programable de una manera mucho más sencilla, a través del interfaz de programación Choregraphe.

Choregraphe, es un sencillo interfaz de programación con el que se puede crear y editar movimientos e incluso comportamientos de una forma sencilla.

Dentro de la creación de comportamientos, se puede programar para conseguir un sistema mucho más avanzado y difícil, para poder satisfacer las necesidades desde el nivel más básico al más experto.

Choregraphe acepta Urbi y Python, por lo que se pueden realizar llamadas a los módulos hechos con C++.

3.4. Arquitectura BICA

La arquitectura BICA ha sido creada por el grupo de robótica de la universidad Rey Juan Carlos, para poder tener un software robusto y completo. Y así poder competir con el en la robocup, aunque se ha pensado para poder ser extensible al mundo fuera de la robocup.

La arquitectura BICA está basada en un comportamiento cooperante a partir de la ejecución interactiva de unas unidades software llamadas componentes.

Estos componentes pueden ser de alto nivel (activan a otros), perceptivos o de actuación. La arquitectura se dividió en varios niveles. El nivel superior se establece lo que se desea conseguir y en los niveles inferiores se consigue la información generada por el conjunto de los sensores virtuales (los cuales son una abstracción de los sensores reales).

Cada funcionalidad del robot está modularizada en un componente. Dicho componente es llamado por un módulo principal. Este módulo principal llama a todos los componentes a la misma función (`Step()`), con lo que permite abstraerse de las funciones internas del componente.

El comportamiento del robot puede estar basado en una máquina de estados que elige su actividad según acción y reacción, o en un esquema reactivo.

Los principales componentes usados para mi proyecto, son `Faceball`, `Kinematics` y `FollowBall`.

- **Kinematics:** Se encarga de proporcionarme las coordenadas en 3 dimensiones de la posición del balón. Dicho comportamiento localiza la posición en 3D a través de una sola cámara.

- **FaceBall**: Realiza un tracking de la pelota.
- **FollowBall**: Se encarga que el robot dirija la cabeza a la pelota y una vez localizada vaya hasta donde se encuentra. Este componente integra FaceBall y Turn.

3.5. JManager

Para poder visualizar y lanzar el componente creado se ha diseñado desde el programa externo NetBeans². Dicho programa te permite programar un interfaz con el cual depurar tu software antes de usarlo en el propio robot.

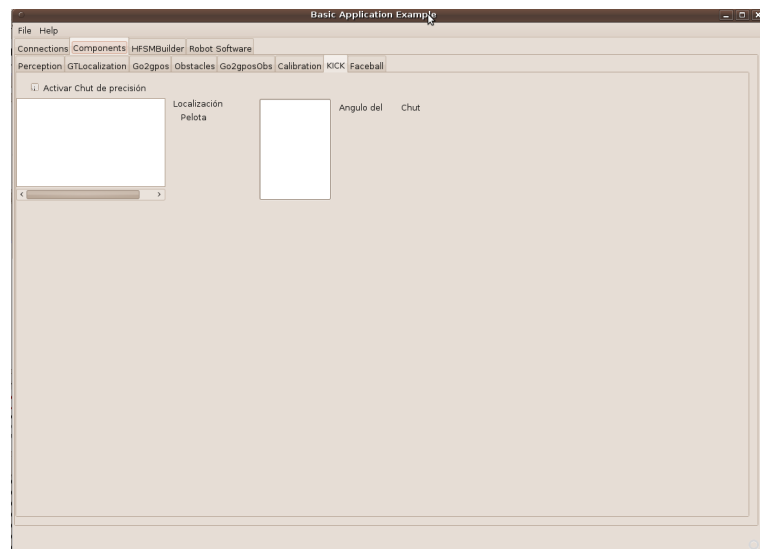


Figura 3.5: Herramienta Jmanager

Esta herramienta está escrita en el lenguaje de programación Java³. Dicho lenguaje es de tipo orientado a objetos, esto permite abstraerse de gran cantidad de código y centrarse en el software final.

La comunicación con la arquitectura BICA se lleva a cabo por TCP. Una vez activado el componente se iniciará la comunicación a nuestra arquitectura BICA, por medio del módulo coach, que analizará qué petición de componente llega y realizará las operaciones necesarias.

Nuestra herramienta creada es bastante simple (figura 3.5), ya que en un principio fue creada para analizar el comportamiento de los chuts según un ángulo pedido. Posteriormente solo se hizo necesario activar o desactivar el comportamiento de chut.

²<http://netbeans.org/>

³<http://www.java.com/es>

Capítulo 4

Descripción informática

Una vez presentado todos los aspectos del proyecto, los objetivos que se desean cumplir y la plataforma que usare para desarrollarlo, en este capitulo presentare el proyecto en sí.

Como ya he descrito, este proyecto se basa en realizar chuts de alta precisión. En este capitulo, explicare paso a paso como se fue desarrollando.

4.1. Diseño Global

El diseño del proyecto ha ido evolucionando según avanzaba. El planteamiento inicial del proyecto estaba enfocado a obtener un algoritmo a través de pruebas de fallo-error. Una vez realizados estas pruebas, y a través de tablas, se buscaba encontrar un algoritmo que, según la distancia desde el punto de apoyo hasta cierta posición del eje Y, realizaba el chut en un ángulo pedido. Posteriormente volvería a su posición inicial (desplazado una distancia en el eje X). Este diseño no era perfecto y no matemáticamente probado, por lo que se busco un enfoque más preciso , fiable y riguroso.

4.2. Chut de precisión

Esta parte del proyecto engloba el 90 % del mismo, ya que es aquí donde se generan la mayor parte de los cálculos matemáticos que le permite llevar a cabo el chut.

El algoritmo que se ha usado para conseguir el chut ha ido evolucionando según las necesidades del funcionamiento global del robot. Como he explicado anteriormente, el robot necesitaba realizar gran cantidad de operaciones mientras realizaba el chut. Un requisito de este componente es que el comportamiento no fuera bloqueante. Esto es obligatorio porque durante el chut, el robot debe poder comunicarse con otros robots o poder mover la cabeza o realizar cualquier operación que posea la arquitectura BICA.

Otra importante restricción, viene en la propia fisiología del robot, ya que sus dimensiones y sus proporciones no son como las de un ser humano, existía el problema

de que por el peso del robot y la longitud de sus piernas no se pudiera conseguir un comportamiento similar al del ser humano.

El primer problema era que al poner en posición de equilibrio al robot, éste perdía el equilibrio al preparar la posición de chut. Otro problema eran las piernas, al ser extremadamente cortas, hacían que el robot no fuera capaz de alargar mucho la pierna sin perder el equilibrio, por lo que estaba obligado a controlar la distancia máxima a la que el robot podía chutar sin caer. Por último un gran problema vino para conseguir que el balón fuera en la dirección correcta.

Estos problemas se fueron corriendo usando las capacidades motrices del robot y ajustando al máximo lo que el robot era capaz de hacer.

Para el problema inicial del equilibrio se optó por usar los sistemas de equilibrio que poseía el API de NaoQi, pero obligaba a realizar el movimiento mucho mas lento de lo que es normal para que el robot le diera tiempo a ir corrigiendo el centro de masas según levantaba la pierna.

El problema más grande era cómo conseguir que el balón no solo fuera en la dirección correcta sino además con la fuerza exacta. Para ello se decidió no realizar un chut estándar como el que realizaría un humano, sino manteniendo la suela del robot constantemente paralela al suelo, esto se consiguió corrigiendo constamente que dicha suela estuviera paralela al suelo.

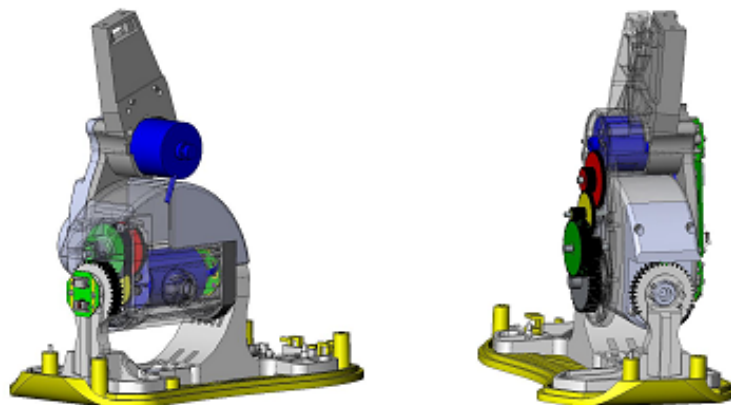


Figura 4.1: Diferentes articulaciones del pie.

Como se puede ver en la figura 4.1, el robot posee 3 motores que le permiten controlar no solo la posición del tobillo sino también la inclinación, con lo que el problema quedaba solventado.

4.2.1. Algoritmo matemático

Esta fue la etapa mas larga del proyecto, donde se realizó el trabajo matemático para obtener el algoritmo para el chut. Todo el algoritmo se desarrollo inicialmente en lenguaje matemático, para que una vez que fuera llevado a código, hubiera el menor número de errores posibles en cuanto al chut.

El algoritmo se ha dividido en varias fases:

- Equilibrar el robot a una pierna y datos iniciales. Antes de empezar el algoritmo se calcula las distancias en las que parte el robot justo en el momento anterior del chut. De esta forma se definieron los datos más significativos del robot:

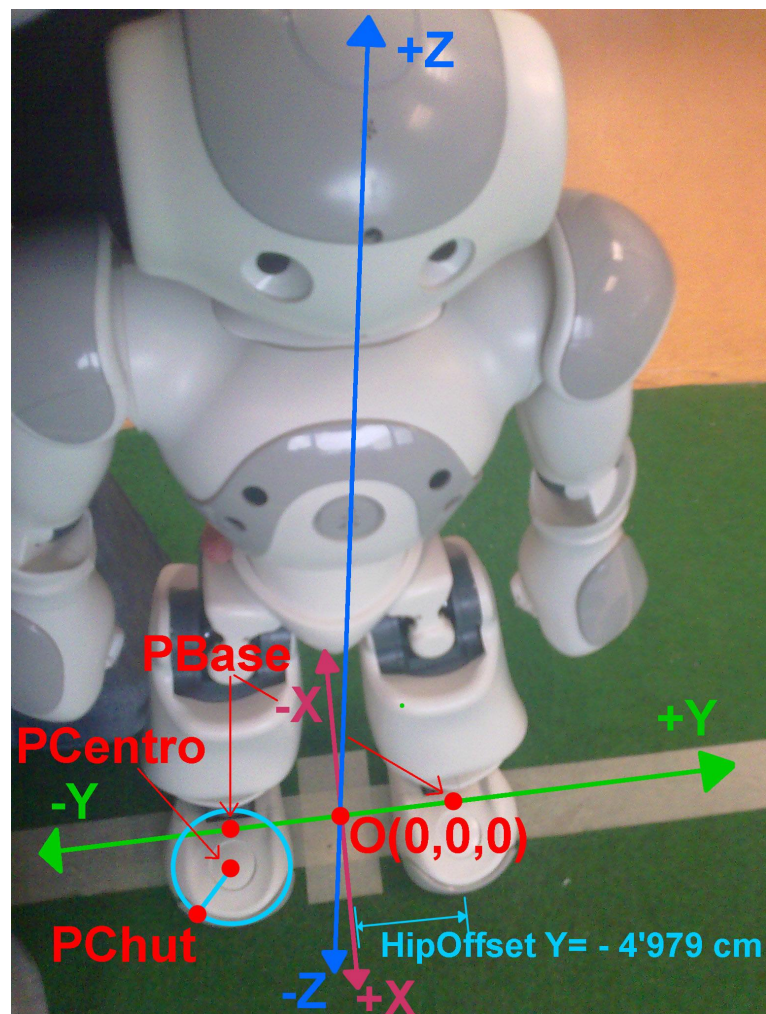


Figura 4.2: Vista Cenital Nao

En la primera imagen el pie esta apoyado en el suelo y obtendríamos los puntos

más importantes iniciales, que serían:

$P_{Base} (0, 4'979, 0)$ $P_{Centro} (5'6, 4'979, 0)$

Es importante mencionar que estos valores han sido tomados de manera precisa para que el error final del chut no viniera dado por pérdida de precisión. Como se puede ver en la foto, los ejes de coordenadas no son los comunes, en el sistema de coordenadas de Nao, la Y positiva esta a nuestra izquierda, la X positiva esta hacia delante y la Z positiva esta hacia arriba.

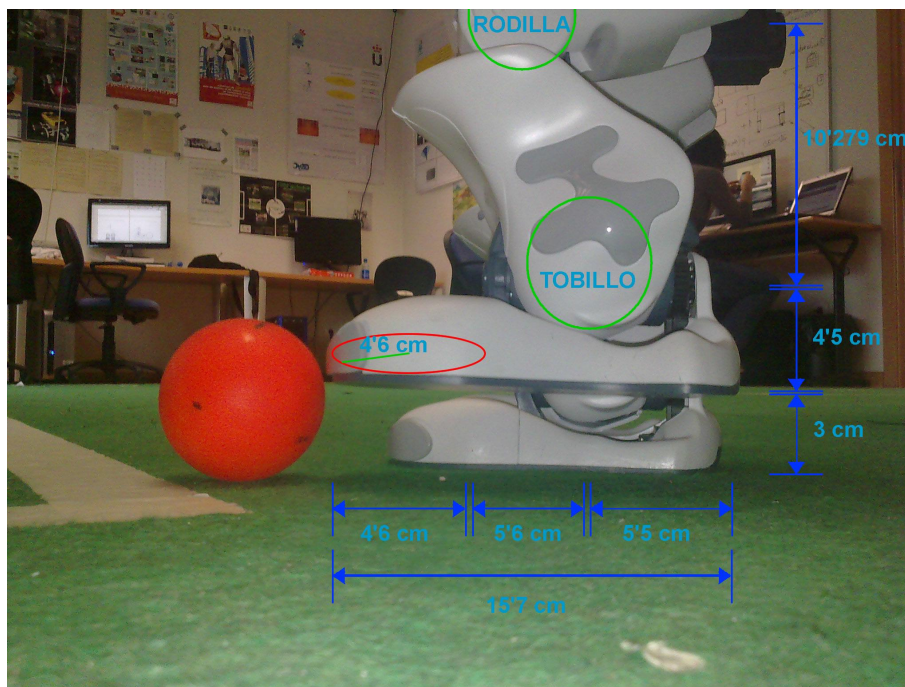


Figura 4.3: Vista Horizontal Nao

En la figura 4.3, definimos los datos desde una visión lateral, para así conocer la distancia desde el tobillo hasta el P_{Centro} que será nuestro punto de referencia para luego calcular el punto de golpeo y también conocer la altura a la que debe elevar el pie para el golpeo.

- Calcular punto de golpeo y rectas de golpeo. Se genera una recta en el espacio dado un punto inicial (donde esta situada la pelota) y un punto final (donde quieres que llegue) y una vez obtenida la recta, es necesario calcular el ángulo respecto del eje Y. Este ángulo permite calcular el ángulo donde debería golpear el pie del robot y una vez obtenido este punto, se debe mover la pierna del robot

este punto y hacerlo secante a la recta que se ha creado de los 2 puntos iniciales. Obtenido el punto secante, se desplaza la pierna en el eje Y tantos centímetros hasta llegar al punto dado.

Por último, el robot mueve la pierna en la misma trayectoria que la recta dada para que golpee la pelota en el ángulo que hemos calculado.

El pie en todo momento se mantiene paralelo al suelo para que no haya error con el ángulo de chut, esto se ha conseguido como se ha explicado anteriormente.

Se ha hecho una aproximación para la cual la puntera del pie del robot sea como una circunferencia y poder calcular así el ángulo de chut. Esto provocara posteriormente un error de precisión que será explicado detenidamente en la sección de experimentación

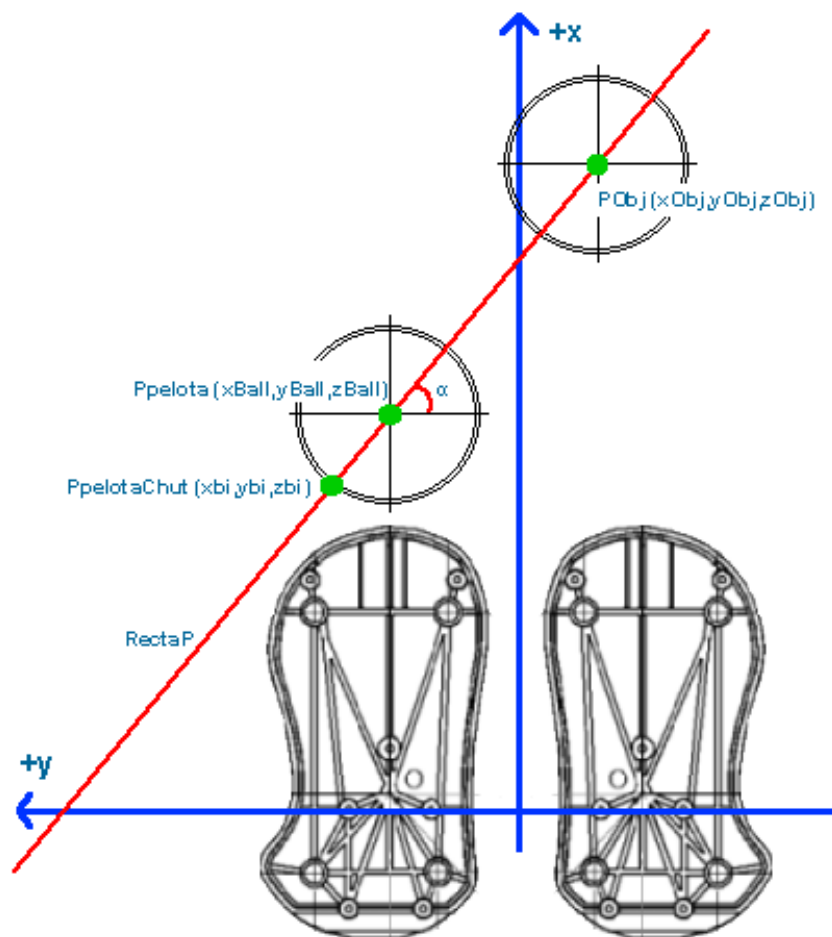


Figura 4.4: Recta de Chut

$$\alpha = \arctang \frac{(xObj * xBall)}{(yObj * yBall)}$$

Para realizar el chut de precisión, solo es necesario conocer 1 dato, que es la posición final (en coordenadas cartesianas: x,y,z) de la pelota: $PObj (xObj, yObj, zObj)$.

Una vez conocido la posición final, obtenemos la posición donde se encuentra la pelota respecto del eje central del robot: $P_{pelota} (x_{Ball}, y_{Ball}, z_{Ball})$.

Dados 2 puntos calcularemos la recta resultante que pasan por estos 2 puntos ($RectaP$). A través de trigonometría sacaremos el ángulo que forma la recta respecto del eje horizontal (obteniendo así α).

$P_{pelotaChut}$ es el punto donde el robot debe golpear a la pelota para que realice la trayectoria deseada. $P_{pelotaChut} (x_{bi}, y_{bi}, z_{bi})$, estos valores cartesianos incluyen el calculo para colocarlo espacialmente.

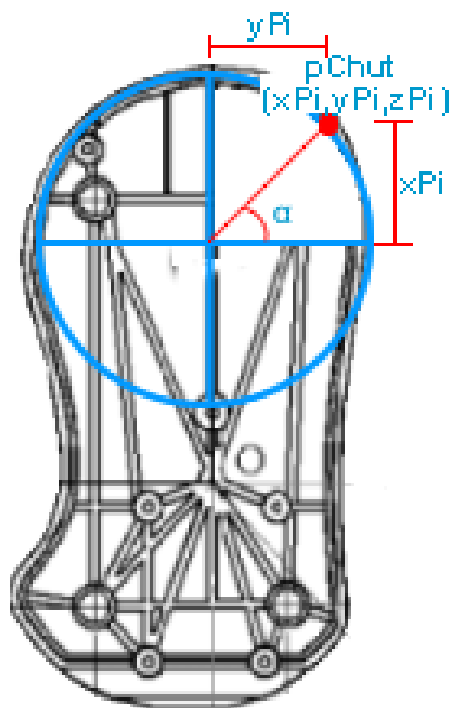


Figura 4.5: Circunferencia de golpeo

$$P_{Chut} = ((\text{sen } \alpha * \text{distanciaCentroPie}) + \text{distanciaCentroPutapie}, (\text{cosa} * \text{distanciaCentroPie}) + \text{distanciaCentroPieY})$$

Una vez que tenemos el ángulo, aplicaremos el mismo cálculo que hemos realizado para saber la posición con la que golpearemos la pelota y así calcular el punto del pie con el que hay que golpear la pelota.

Este punto lo definiremos como: $P_{chut} (x_{Pi}, y_{Pi}, z_{Pi})$

Una vez calculado el punto $P_{chut} (x_{Pi}, y_{Pi}, z_{Pi})$, obtendremos la recta paralela al eje horizontal pero que pase por el punto P_{chut} ($RectaChut$). Esta recta será la recta que definirá por donde debe desplazarse el pie del robot lateralmente.

Punto de intersección de las rectas de golpeo y desplazamiento. A continuación, se calcula el punto resultante de la intersección de las 2 rectas (*RectaChut* y *RectaP*). Dicha recta dará como resultado el punto con el que el pie debe golpear a la pelota. Una vez que coincida el *Pchut* con este nuevo punto de intersección (*PChutFinal*), tendremos la posición final de golpeo. (Tal y como se puede apreciar en la Figura 4.6).

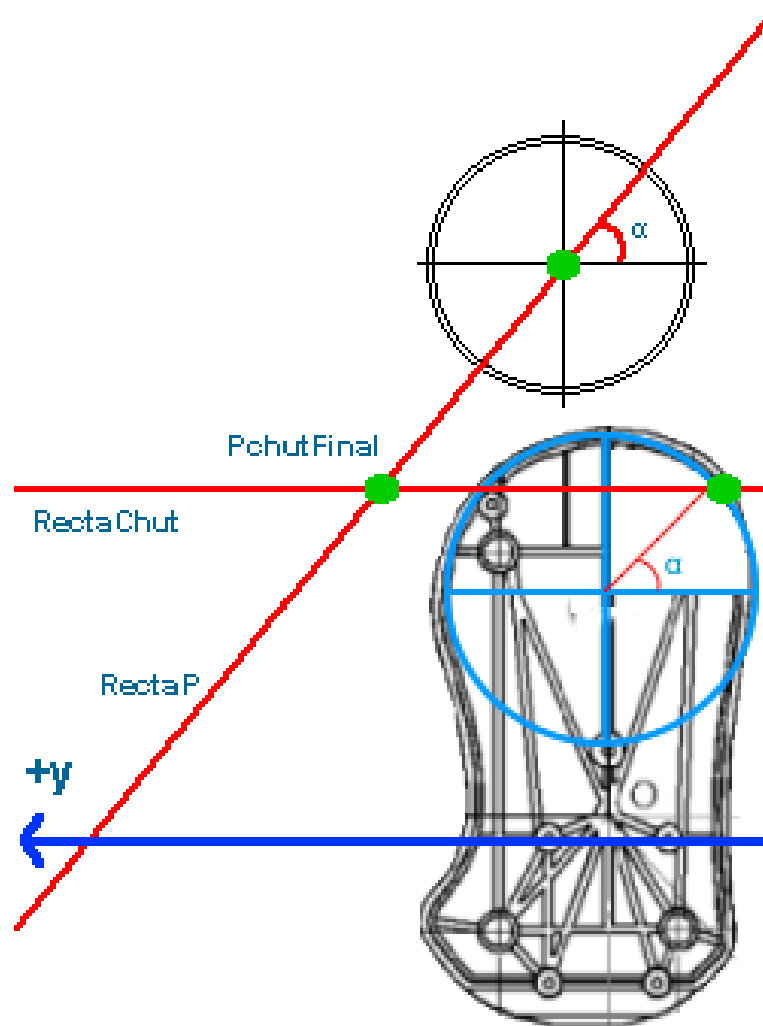


Figura 4.6: Intersección de rectas

$$PChutFinal(x_{pi}, \frac{(RectaP_{independ}) - (RectaPx * x_{pi})}{RectaPy, 3})$$

El siguiente paso es calcular la posición que debe desplazar el pie para llevarlo a esa posición (*PChutFinal*) a partir del punto de golpeo. Para ello se calcula el centro de golpeo de la nueva posición y una vez obtenido, se calcula el punto a donde debe llevarse la última articulación del tobillo.

Para conseguir este punto donde debo calcular el desplazamiento del el tobillo, debe volver ha hacer la operación de calcular el punto de golpeo con el ángulo, pero a la inversa, es decir calcular el punto central del pie. Una vez obtenido nuestro antiguo PCentro (pero en la nueva posición de golpeo), calculare la posición donde debe llevar la ultima articulación del tobillo.

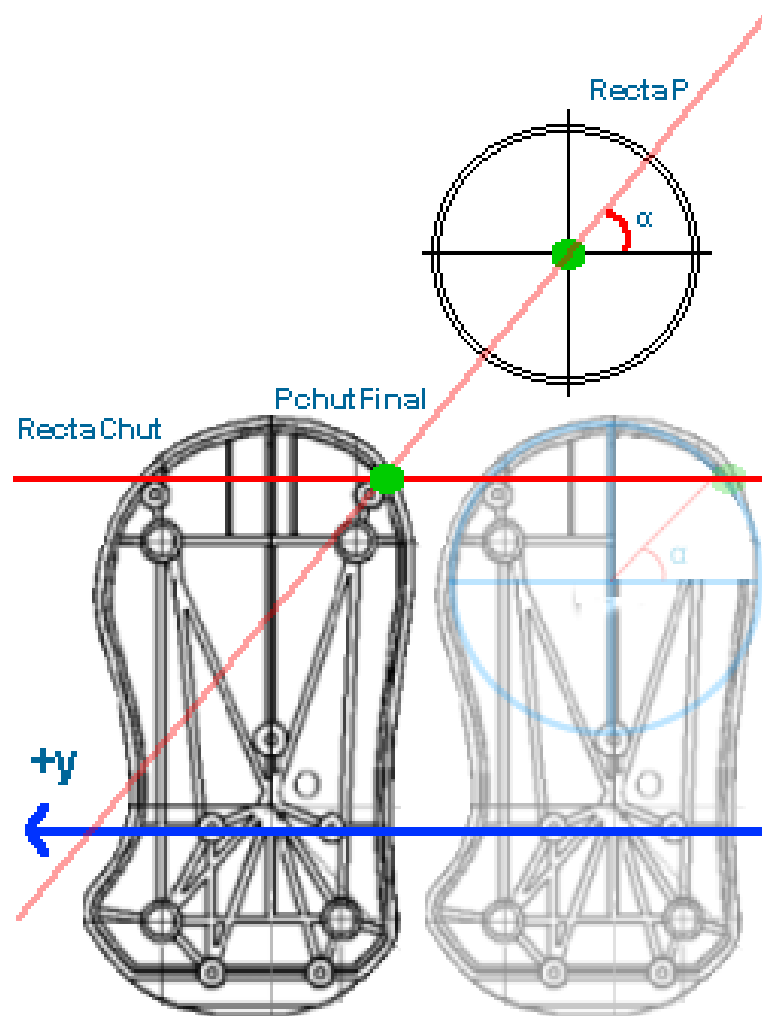


Figura 4.7: Desplazamiento de pie

Por último, se calcula el punto donde debo golpear a la pelota (dentro de la *rectaP*). Una vez obtenido este punto calculare como he hecho anteriormente el eje de ese punto y por ultimo el punto de desplazamiento.

4.2.2. Implementación

Para la implementación, antes de poder realizar el algoritmo de chut, debe conseguir mantener un equilibrio a una pierna de forma estable incluso desplazando la pierna que

esta en el aire a cualquier punto. Esto se consigue controlando constantemente el centro de masas.

Para realizar esto, el robot primero se establecía un posición de equilibrio de la pierna que se deseaba quedar fija. Una vez preparado el robot, lo siguiente es lanzar la instrucción de ponerse a una sola pierna. Esta función consiste en mantener el eje de masas siempre lo mas cerca de la rodilla del robot que va a quedar fija al suelo.

Este sistema solo tenía una carencia, es que cuando el robot se prepara para entrar en equilibrio, toda la operación es un poco lenta y ralentiza todo el chut. Para solventarlo con la nueva versión de NaoQi.

4.2.3. Experimentación

Para demostrar la fiabilidad del algoritmo, se han realizado una serie de pruebas de caja negra.

Se han introducido diferentes posiciones finales del objetivo, manteniendo en la misma posición la pelota, para comprobar si la dirección era correcta. Como se puede apreciar en los datos mostrados a continuación, el error cometido por el chut, es bastante bajo. Este error puede por varios factores:

- El primer factor se debe a la propia precisión de los servos para mover las articulaciones, ya que todo motor posee un error al calcular ángulos.
- El segundo factor se debe a por el error al calcular la posición inicial de la pelota que te proporciona el componente de cinemática (explicado anteriormente).
- El tercer factor se debe a que ya que se define la zona de chut del pie como una circunferencia. Esto al no ser 100 % real, es posible encontrar algún ligero error a la hora de encontrar el punto exacto y el ángulo real que debiera darse.

A pesar de estos factores de error, el error final es casi despreciable comparado el potencial uso que se le puede dar a este algoritmo. Los beneficios que se obtienen en un partido de la Robocup pudiendo dirigir la pelota a una posición final controlada son extremadamente altos.

Pruebas: Se han realizado pruebas para 6 ángulos diferentes. Según el se realizaba mayor ángulo de disparo mayor era el error del chut. Los ángulos que se muestran empiezan a crecer desde la horizontal.

- **90 grados:** El chut es aproximadamente del 100 % de precisión al realizarse.
- **70 grados:** El chut es aproximadamente del 85 % de precisión al realizarse.
- **49 grados:** El chut es aproximadamente del 80 % de precisión al realizarse.
- **45 grados:** El chut es aproximadamente del 95 % de precisión al realizarse. Realizar chuts para este ángulo, es relativamente más preciso por la posición del robots antes del chut. Dicha posición es mas propicia para realizar un chut casi perfecto a 45 grados.
- **41 grados:** El chut es aproximadamente del 85 % de precisión al realizarse.
- **23 grados:** El chut el error empieza a ser relativamente alto, aproximadamente de 10 grados en ambos sentidos.

Para calcular el error realizado se han realizado 10 muestras de cada ángulo de chut.

Uno de los problemas que existen a la hora de calcular la recta de chut desde la posición inicial y la posición final, es que si están en línea recta, se provoca una división por cero, lo que provoca una indeterminación. Para evitarlo se comprueba antes de realizar la operación y si esta va a tomar 0 en la divisor se da un valor muy próximo a cero pero sin provocar la indeterminación. Esto no provoca un error de ángulo final, ya que el chut envía el balón con una precisión cercana al 100 % a 90 %. Como vemos en la figura 4.8 y figura 4.9, el robot es capaz de realizar con la mayor precisión dicho chut.

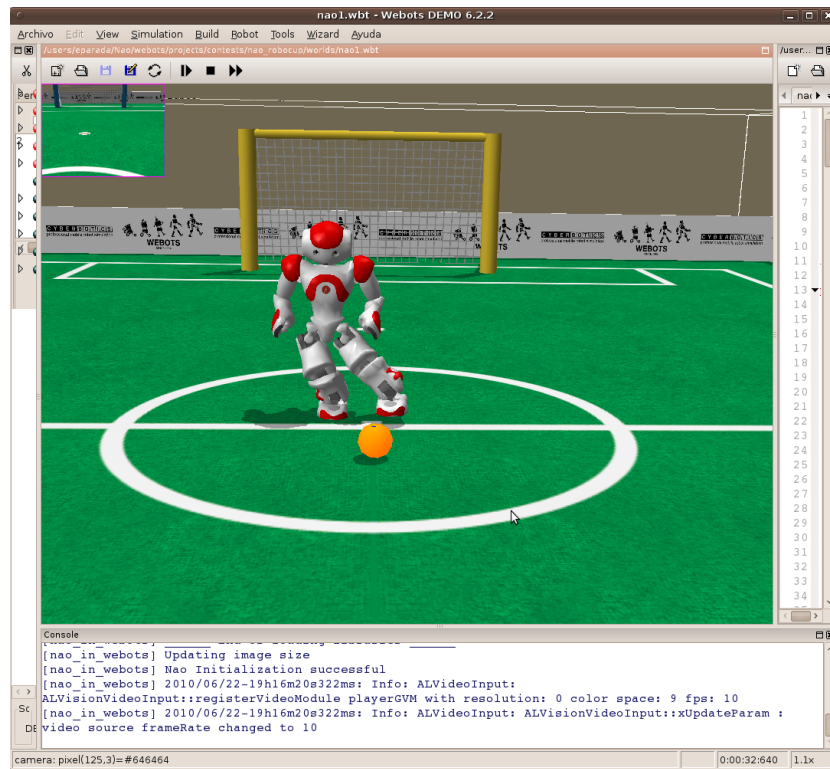


Figura 4.8: Chut de 90 grados

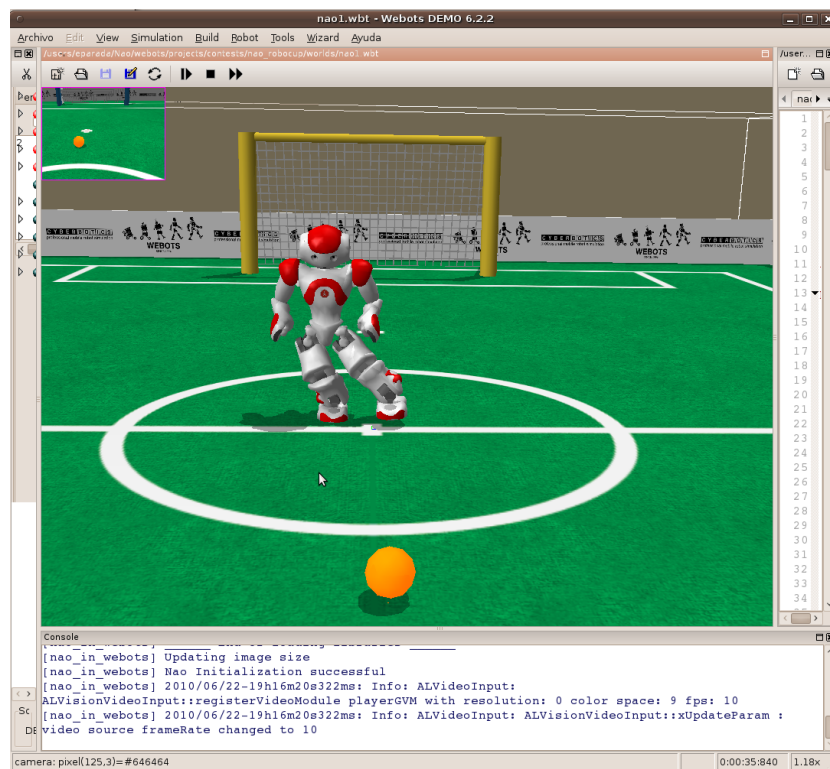


Figura 4.9: Chut de 90 grados completado

En las siguientes imágenes (figura 4.10, 4.11, 4.12 y 4.13) se puede apreciar con claridad que el chut a 45 grados en las 4 figuras que hay a continuación, posee una alta precisión en el disparo.



Figura 4.10: Chut de 45 grados inicio



Figura 4.11: Chut de 45 grados completado



Figura 4.12: Chut de 45 grados, vista cenital

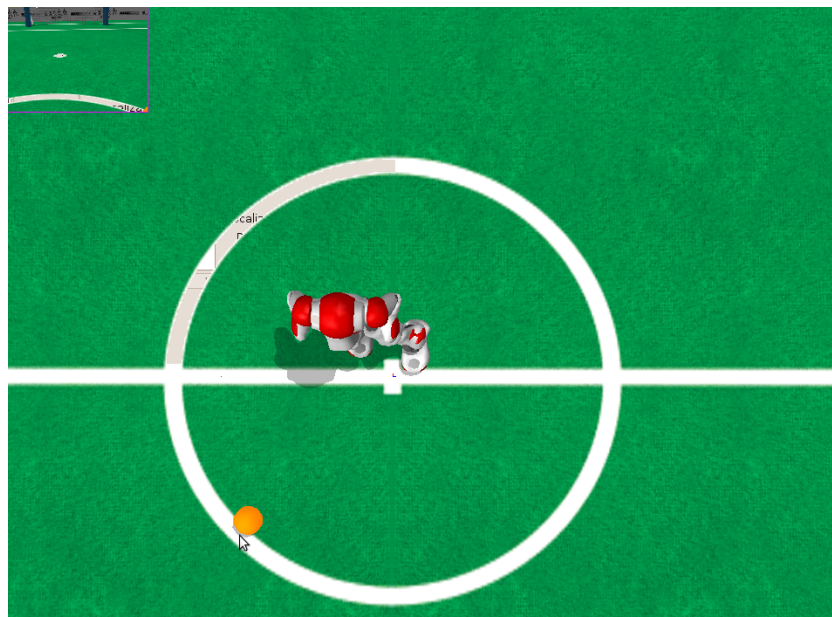


Figura 4.13: Chut de 45 grados, vista cenital completado

En el siguiente link, se puede apreciar el video de unas de las pruebas a 45 grados:
<http://robotica-urjc.es/eparada/Videos/chut45grados.ogv>

Capítulo 5

Conclusiones y Trabajos futuros

Una vez presentados los algoritmos desarrollados en mi proyecto, y los experimentos llevados a cabo para validarlo, en este capítulo vamos a presentar las principales conclusiones que hemos obtenido, así como comentar los posibles trabajos futuros.

5.1. Conclusiones

1. Tras completar un chut de precisión a través de prueba error, se demostró que crear un segundo algoritmo basado en principios matemáticos conseguía un chut mas fiable y preciso.
2. Las pruebas realizadas demuestran que el chut de precisión consigue que los robots se comporten de una manera mas eficiente durante un partido entre robots Nao.

Este algoritmo se puede comparar a partir de los anteriores partidos realizados en la robocup. Los videos demuestran que si los Nao en esos partidos, hubieran poseído dicho comportamiento a la hora de realizar chuts a puntos dados, la eficiencia de los mismos hubieran sido altamente superior, ya que los comportamientos de esas anteriores de chut, se basaban en alejar al máximo el balón de su propia portería.

Requisitos

Los requisitos que ha tenido que cumplir el comportamiento de chut de precisión se han cumplido. Entre ellos es el correcto funcionamiento junto con la arquitectura BICA y una precisión del chut en diversos ángulos superior al 70%.

La robustez del algoritmo es muy alta, ya que al haberse realizado una potente batería de pruebas matemáticamente de diferentes ángulos antes de llevarse a código, se ha conseguido que el algoritmo fuera lo más eficiente posible.

5.2. Trabajos futuros

El trabajar sobre el robot Nao en el campo de la movilidad, solo ha sido el primer acercamiento habia la investigación de control de precisión de actuadores robóticos. En futuros proyectos deseo realizar una investigación más detallada en el campo de los brazos robóticos para conseguir una precisión a la hora de mover y coger objetos similar a la humana en el campo de la robótica humanoide.

Otros proyectos futuros que deseo conseguir realizar, son los comportamientos en movilidad similares a los de los humanos, todo ello englobado en el campo de la robótica humana. Tanto desplazamientos de robots bípedos con la máxima estabilidad y sistemas de recuperación de trayectorias, hasta interactuar con los actuadores de los que disponga el robot con el mundo que le rodea.

Bibliografía
