

VOLUMEN I

Memoria del proyecto

**DISEÑO Y CONSTRUCCIÓN DE UN ROBOT
HEXÁPODO**
Hardware y Software de Control

REALIZADO POR
Juan Antonio Fernández Madrigal

DIRIGIDO POR
Jesús Manuel Gómez de Gabriel

DEPARTAMENTO
Ingeniería de Sistemas y Automática

FACULTAD DE INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

UNIVERSIDAD DE MÁLAGA
Facultad de Informática

Reunido el Tribunal examinador en el día de la fecha, constituido por:

D. _____

D. _____

D. _____

para juzgar el Proyecto Fin de Carrera titulado **DISEÑO Y CONSTRUCCIÓN DE UN ROBOT HEXÁPODO. HARDWARE Y SOFTWARE DE CONTROL** del alumno D. Juan Antonio Fernández Madrigal, dirigido por D. Jesús Manuel Gómez de Gabriel,

ACORDÓ POR _____ OTORGAR LA CALIFICACIÓN DE

Y PARA QUE CONSTE, SE EXTIENDE FIRMADA POR LOS COMPONENTES DEL TRIBUNAL, LA PRESENTE DILIGENCIA.

Málaga, a de de

El Presidente

El Secretario

El Vocal

Fdo.:

Fdo.:

Fdo.:

INTRODUCCIÓN

Se ha llevado a cabo el diseño y construcción de un robot móvil articulado con patas controlado por ordenador. Se partió de un objetivo primordial: realizar el diseño de un robot experimental con presupuesto y complejidad reducidos. El prototipo construido, denominado RHEX-I, cumple este objetivo y sirve de precedente para futuras investigaciones en este campo.

RHEX es el primer robot de este tipo construido en esta Universidad, aunque existen antecedentes en otras Universidades y Departamentos de Investigación, tanto en España como fuera de nuestro país.

No se partió de ningún trabajo previo, salvo la documentación citada en la bibliografía, diseñándose todos los sistemas mecánicos, electrónicos e informáticos que se consideraron necesarios. Se experimentó con diversas estructuras mecánicas, servomecanismos de posicionamiento, electrónica de generación de señales y software jerarquizado y modular. Finalmente, se implantó e integró todo ello, logrando un sistema completo y preparado para futuras extensiones.

Todo el trabajo ha sido completamente especificado en el presente proyecto, dividiéndolo de forma lógica en dos volúmenes, con un total de nueve capítulos y diez apéndices.

En el primer volumen están los nueve capítulos y los primeros nueve apéndices.

En el primer capítulo se exponen los objetivos que llevaron a la construcción del robot. El segundo capítulo contempla los antecedentes y explica la situación de RHEX dentro del amplio espectro de los robots existentes en la actualidad. El tercer capítulo trata el diseño

preliminar, subdividiéndolo en las diversas fases que éste abarcó. Los capítulos cuarto, quinto y sexto desarrollan con más profundidad cada una de estas fases: diseño mecánico, electrónico y software de control. El capítulo séptimo es un manual de usuario del entorno que se implantó para el manejo cómodo y eficiente de todos los aspectos de RHEX, desde la calibración de servos individuales hasta la realización de movimientos complejos tales como andar. El capítulo octavo enumera las pruebas que se realizaron durante el proyecto y describe los resultados obtenidos. El capítulo noveno expone las conclusiones a que ha dado lugar este proyecto, así como las futuras líneas de ampliación del mismo.

El contenido de cada apéndice es el siguiente:

El apéndice A muestra un estudio de los servomecanismos utilizados para el posicionamiento de las articulaciones.

El apéndice B contiene los planos de las estructuras mecánicas que forman una pata del robot, y el apéndice D contiene los planos del cuerpo.

El apéndice C estudia el modelo cinemático de RHEX.

El apéndice E da información detallada sobre el integrado i8253.

El apéndice F contiene los planos y documentación sobre la electrónica de control.

Los apéndices G y H exponen las características principales de las dos tarjetas de entrada y salida utilizadas para el envío de órdenes desde el ordenador.

El apéndice I es un desarrollo matemático de las fórmulas para la cinemática directa e inversa de una pata del robot.

El segundo volumen incluye el décimo apéndice, que contiene todos los listados del software implantado.

1. Objetivos y justificación del proyecto

El principal objetivo de este proyecto es el diseño y construcción de un robot móvil con patas, con presupuesto y complejidad reducidos.

Este objetivo plantea esfuerzos de investigación en los siguientes campos:

-Diseño de estructuras mecánicas poliarticuladas, capaces de posicionar de manera eficiente el sistema robot (maximización del espacio alcanzable frente a minimización de colisiones entre los elementos).

-Estudio y búsqueda de sistemas mecánicos de servocontrol (minimización del tamaño, peso y coste frente a optimización del control en el sentido de fuerza ejercida y reducción del error).

-Arquitecturas de control (autonomía frente a control remoto).

-Diseño de la electrónica de control y de los interfaces (estructuras digitales para el mantenimiento de las señales de posicionamiento y elección de dispositivos de entrada/salida para el control desde el software).

-Sistemas software estructurados en niveles de abstracción, capaces de planificar eficientemente movimientos complejos del robot (arquitecturas reactivas frente a arquitecturas jerarquizadas).

-Interfaz para el manejo del robot por el usuario (facilidad de uso frente a potencia. Acceso a todos los niveles de control del robot, desde configuración del sistema a la simulación de los futuros movimientos y del comportamiento previsto).

En la implantación de todos estos sistemas se han encontrado los problemas y restricciones enumerados a continuación:

-Limitación de presupuesto.

-Limitada disponibilidad de materiales y de medios de mecanización de los mismos.

-Escasez de documentación sobre los dispositivos de servocontrol utilizados, y en general, sobre los robots de este tipo realizados en otros departamentos.

-Limitación de tiempo y de personal para un proyecto individual, debida principalmente a la extensión y la diversidad de los campos abarcados.

-Espacio de trabajo reducido.

En general, estos problemas no se han podido obviar. El resultado final se ha adaptado a las restricciones expuestas, consiguiendo pese a ello sus objetivos.

Estos objetivos incluyen, aparte del ya mencionado criterio de bajo coste, otros más generales:

-Facilidad de reemplazo de componentes.

-Modularidad y reusabilidad del software.

-Sencillez de diseño.

-Preparación para futuras ampliaciones y/o mejoras.

Por último, el desarrollo de este proyecto supone la apertura de una nueva línea de investigación en esta Facultad, y concretamente en el departamento de Ingeniería de Sistemas y Automática. Esta línea está destinada a una rápida expansión que estaría limitada únicamente por la capacidad e inventiva que se vuelque en ella y no por la falta de medios.

2. Antecedentes

La documentación existente en el campo de los robots móviles con patas es relativamente escasa, y se limita al desarrollo de sistemas dirigidos a objetivos muy concretos. Por ello, durante la realización de este proyecto se pusieron en marcha ideas, que si no son nuevas en el campo de la computación, si lo son aplicadas a este tipo de robots. También es a causa de esto por lo que el diseño tuvo una gran componente de invención y experimentación, apartándose en muchos casos de lo ya aplicado en este campo.

Existen varias formas de clasificar robots. Desde un primer punto de vista, podemos dividir éstos en *experimentales* y *aplicados*. La primera categoría incluye aquellos robots de propósito general diseñados para realizar un conjunto bastante amplio de experiencias, es decir, aquellos pensados con un estricto enfoque de investigación. La segunda categoría abarca los robots construidos con algún propósito específico (industrial, de exploración, etc...). En esta clasificación RHEX entraría en el primer grupo, debido a que su construcción persigue la creación de una línea de investigación en principio sólo limitada por la estructura física del robot, estando abierta a cualquier tipo de experiencias que se adapten a esta restricción.

Desde otro punto de vista, se pueden encontrar robots *móviles* o *estáticos*. Los robots industriales son fundamentalmente estáticos, es decir, incapaces de desplazarse libremente por un entorno no limitado. RHEX es un robot móvil, por lo que a partir de ahora la clasificación se centrará en este tipo de dispositivos.

Dentro de los robots móviles, se encuentra una primera división en robots *autónomos* y *no-autónomos*. Los primeros portan todo el software y hardware de control sobre la

estructura mecánica. Esto les da un rango de alcance limitado únicamente por la duración de las fuentes de alimentación que utilicen, pero encarece y produce una mayor complejidad en el sistema. RHEX es un robot *no-autónomo*. Es gobernado por un ordenador externo al que se comunica a través de un bus de señales de datos y control. Las fuentes de alimentación son así mismo externas.

Desde un segundo punto de vista, los robots móviles pueden clasificarse atendiendo al medio de locomoción que utilicen. Los robots con patas permiten desplazamientos más eficientes sobre terrenos de cualquier tipo (rugosos, con obstáculos o desniveles,...), además de ofrecer un control de estabilidad más completo y requerir menor potencia. Los robots con otro tipo de locomoción (ruedas, orugas, ...) simplifican el posicionamiento y los cálculos necesarios para el mismo.

Existen diversos sistemas con patas desarrollados por Universidades u otras entidades en los años precedentes. A continuación se comentan las características más notables de algunos de ellos.

ROBOTS DE UNA PATA. El ejemplo más representativo es un robot desarrollado por el Instituto de Robótica y el Departamento de Informática de la Universidad americana Carnegie-Mellon. Este dispositivo salta sobre su única extremidad, buscando continuamente la estabilidad dinámica, ya que la estática es imposible en general sobre menos de cuatro patas. Supone un gran esfuerzo en el desarrollo del sistema de control.

ROBOTS BÍPEDOS. Los robots de la serie *BIPER*, diseñados en la Universidad de Tokyo, pueden caminar lateralmente, avanzar y retroceder, simulando más o menos aproximadamente el modo de andar humano.

ROBOTS CUADRÚPEDOS. Este tipo tiene más representantes, tanto fuera de España como en nuestro país. En el Instituto Tecnológico de Tokyo fue construido un vehículo de cuatro patas dotado de sensores táctiles y detector de posturas. Cada pata tiene 3 grados de libertad. El control se realiza desde un microordenador que asegura la existencia de un triángulo de apoyo sobre 3 de las patas continuamente.

Por otra parte, en España, el *RIMHO* (Robot de Intervención en Medios HOstiles), diseñado conjuntamente por el Instituto de Automática Industrial (*IAI*) y el Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (*CIEMAT*) soporta nuevos algoritmos para conseguir que el robot conserve la estabilidad mientras camina, así como para la detección de inestabilidades y la planificación de modos de caminar.

ROBOTS HEXÁPODOS. En la Universidad Estatal de Ohio ha sido desarrollado un robot con 6 patas que permite al operador tomar decisiones estratégicas, independientemente del posicionado particular de cada extremidad. Está basado en un sistema de control consistente en 13 computadores de a bordo Intel 86/30. En el Instituto de Robótica de la Carnegie-Mellon (EEUU) ha sido desarrollado el *Ambler* (*Autonomous mobile exploratory robot*), un robot hexápodo dirigido por programas específicamente diseñados para optimizar el movimiento de avance sobre terrenos abruptos.

Una última clasificación distinguiría entre robots con control *jerárquico* y robots con control *reactivo*. Los primeros implican un tiempo de proceso mayor, pues se basan en un sistema fuertemente jerarquizado en el que los niveles de control más altos generan trayectorias que son expandidas y planificadas por los niveles intermedios, y ejecutadas por los niveles inferiores. Los robots con control *reactivo* tienen implantados comportamientos más o menos simples que exhiben reaccionando a la influencia del medio. Carecen de una

planificación detallada a priori de sus trayectorias. Esto les da más flexibilidad y les evita tiempo de cálculo, pero hace que no puedan encontrar solución para algunas órdenes de alto nivel.

Ésta última es una tendencia seguida por los conocidos *MOBOTS* del MIT: la distribución del sistema total en pequeños autómatas capaces de funciones sencillas de control, interconectados entre sí, y que en su conjunto permiten funciones mucho más complejas e incluso similares a las de algunos insectos.

En la categoría de robots *jerárquicos* entraría RHEX: su sistema de control software consiste en un único programa dividido en módulos de distinto nivel de abstracción que gobierna todas las funciones del robot, desde las más básicas hasta las puramente estratégicas.

3. Diseño preliminar

En este capítulo se detalla el diseño que se llevó a cabo previamente a la construcción de RHEX. En el mismo se exponen los problemas encontrados en la elección de la configuración mecánica y electrónica del robot, y las soluciones a las que dieron lugar.

3.1. Configuración y distribución de las patas

Existen varias configuraciones posibles para un robot con patas:

-Aquellas que contemplan menos de cuatro patas presentan problemas más propios de la Ingeniería de Control que de la Computación. Dentro de los robots de más de tres patas, los que nos interesan adoptan la forma de cuadrúpedos o hexápodos.

-Los robots cuadrúpedos y hexápodos permiten un mayor desarrollo de los aspectos de planificación del movimiento. En ellos los problemas derivados de la estabilidad están relativamente simplificados o son susceptibles de simplificación, con lo que el diseño del hardware de control se hace más sencillo. Esto añade cierta complejidad al software, pero esa es la orientación que se pretendía dar al proyecto.

-No hay ninguna ventaja en usar una configuración pentagonal (en todo caso puede presentar inconvenientes debido a la más compleja simetría).

-Robots de más de seis patas sólo son útiles en campos mucho más específicos, además de presentar un mayor coste.

Un concepto importante a la hora de elegir el número de patas del robot es el de *estabilidad estática*. Este término se refiere a la capacidad del robot para permanecer estable (sin caerse) cuando no está en movimiento. Es más fácil mantener la estabilidad estática en un robot hexápodo que en uno cuadrúpedo por un motivo muy sencillo: hay más patas libres para reposicionar el cuerpo del robot mientras éste se apoya en tres de ellas. En un robot cuadrúpedo, sin embargo, estamos obligados a utilizar un algoritmo de avance en el que las patas vayan alternándose, puesto que sólo una está disponible una vez apoyado el robot en las otras tres.

Por todos estos motivos, se escogió la configuración hexápoda.

Dentro de esta configuración, surgieron de forma natural dos distribuciones de las patas alrededor del cuerpo del robot.

La primera de ellas, la denominada *bilateral*, se puede observar en la figura 3.1.a. Esta distribución presenta una simetría a lo largo del eje longitudinal del robot. Tiene la ventaja de una mayor simplicidad a la hora de programar los movimientos, debido a que el robot está preparado físicamente para facilitar el avance en

direcciones paralelas al eje de simetría. El inconveniente fundamental es que el sistema ofrece una mayor limitación en el movimiento en otras direcciones. Esto se puede obviar implantando un nuevo movimiento, el de **giro**, para cambiar de orientación explícitamente, pero ello complica más el software.

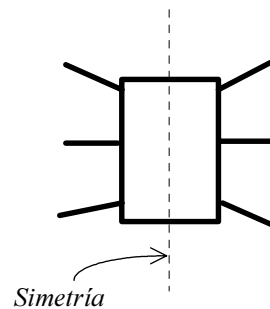


Figura 3.1.a.: Configuración bilateral de las patas de un robot hexápodo.

La distribución *radial*, que aparece en la figura 3.1.b. fue la finalmente escogida para RHEX. No sufre el problema comentado, puesto que cualquier dirección es exactamente igual a cualquier otra. En dos dimensiones, esta distribución es

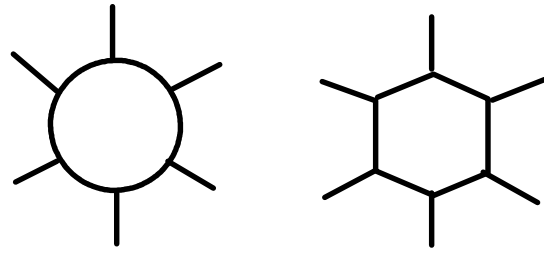


Figura 3.1.b.: Configuración radial para las patas de un robot hexápodo. La figura de la derecha es la configuración escogida finalmente para RHEX.

totalmente simétrica, y eso lleva a un software más genérico y más simple. No necesita movimientos específicos de **giro** puesto que es capaz de desplazarse en cualquier dirección sin tener previamente que cambiar de orientación (el robot es omnidireccional), aunque este tipo de movimiento puede implantarse para casos específicos si es necesario.

3.2. Estructura de las patas

Condicionadas por la elección de configuración hexápoda y distribución radial, las

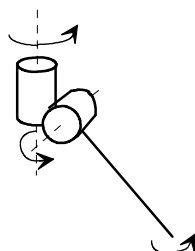


Figura 3.2: Una pata con dos grados de libertad que presenta problemas de deslizamiento

patas debían contar con la mayor movilidad y accesibilidad dentro de sus espacios de trabajo.

Desde el primer momento se pensó en utilizar servos para las articulaciones, lo cual llevó a estudiar diseños que poseyeran únicamente articulaciones de rotación.

En la figura 3.2 aparece el esquema de una pata con dos grados de libertad. Esta estructura permite posicionar el extremo de la pata en cualquier punto de la superficie de una esfera cuyo centro está en el interior de la primera articulación, pero provoca deslizamientos indeseables.

Para posicionar completamente el extremo de una pata hacen falta seis grados de libertad: tres para especificar la *posición* y tres para especificar la *orientación*. Sin embargo, puesto que el extremo de la pata se considerará puntual, no hará falta especificar su orientación. Por consiguiente, son necesarios solamente tres grados de libertad.

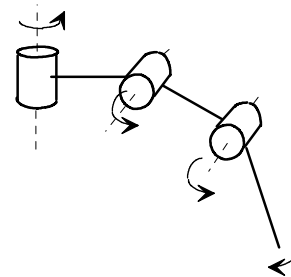


Figura 3.3.: Esquema de los tres grados de libertad de una pata de RHEX.

El esquema de una pata con esta característica aparece en la figura 3.3.

Estos tres grados de libertad aparecen en los dos tipos de estructuras mostradas en la figura 3.4 (además de otras como la de *pantógrafo*, no considerada aquí). De ellas se escogió

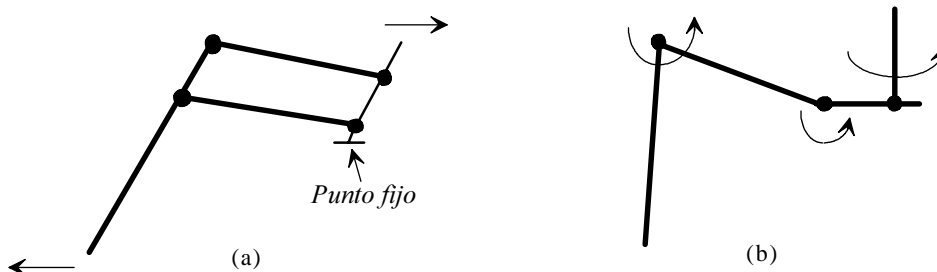


Figura 3.4.: Dos posibles estructuras para las patas de un robot.
(a) paralelogramo deformable, y (b) actuación in-situ

la estructura de *actuación in-situ*, en la cual los motores se sitúan directamente en las articulaciones. Esto permite una mayor simplicidad en la función que relaciona posición del servo con posición de la articulación (de hecho ésta es inmediata salvo algún factor de offset en ciertas articulaciones). Presenta sin embargo el inconveniente de un mayor esfuerzo de la estructura, aunque éste es perfectamente soportable con los materiales escogidos.

El otro tipo (*paralelogramo deformable*) consiste en un paralelogramo cuyos ángulos pueden variar mediante servos situados lejos de la articulación. A pesar de evitar un peso importante en la misma situando el centro de gravedad de la pata más cercano al robot, este diseño acentúa las posibles holguras que con toda seguridad aparecerán en la estructura. Además genera una mayor complejidad en la función anteriormente indicada.

3.3. Arquitectura de control

La figura 3.5 muestra el sistema de control del robot.

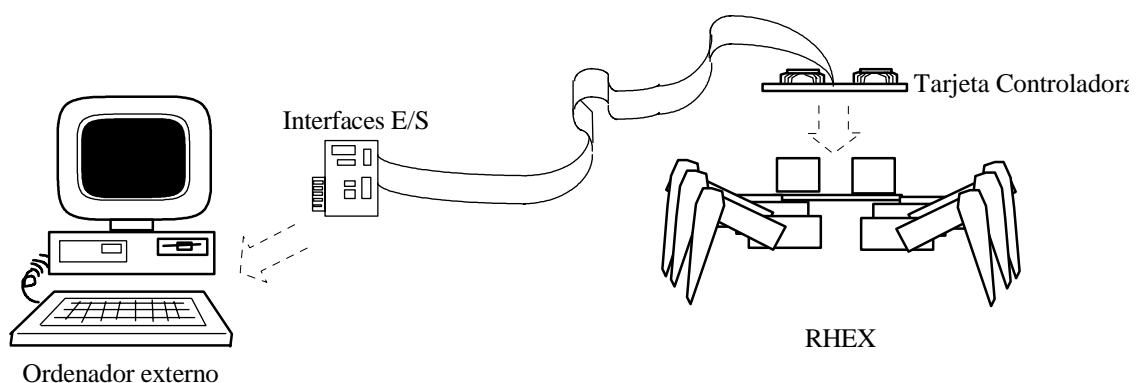


Figura 3.5.: Sistema de control de RHEX

Los componentes principales se describen a continuación:

-Ordenador externo: ejecuta una capa de software de control jerarquizado desde el más bajo nivel (envío de órdenes simples de posicionado de servos particulares) hasta el más complejo (planificación de trayectorias de todas las articulaciones para que el robot ande).

-Interfaces de Entrada/Salida: permiten conducir las órdenes desde el ordenador hasta la placa controladora del robot. Proporcionan las señales de reloj apropiadas para que ésta última funcione correctamente.

-Tarjeta Controladora: situada sobre el robot, genera las señales de posicionamiento de los servos (y las mantiene) a partir de los datos numéricos enviados desde el ordenador.

-Robot: los servos actúan de acuerdo con las señales de la controladora, movilizandoo la estructura.

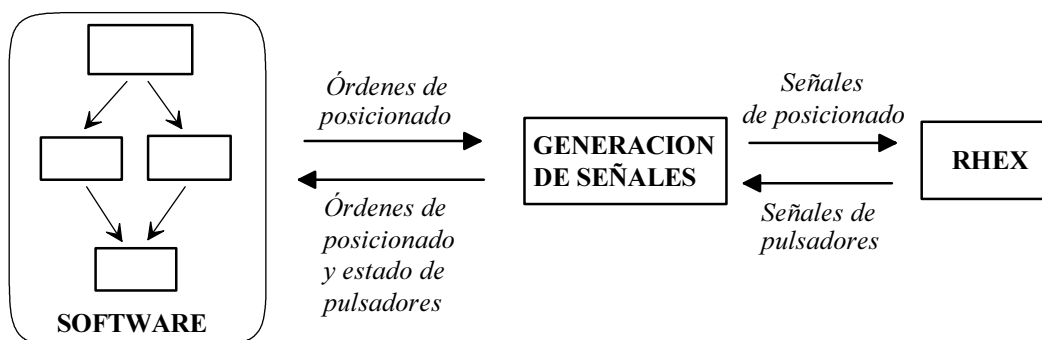


Figura 3.6.: Esquema de bloques del sistema de control de RHEX.

Aunque en principio todas las señales circulan unidireccionalmente (desde el ordenador hasta el robot), se ha considerado el envío de cierta información de regreso al ordenador (pulsadores situados en los extremos de las patas y verificación de datos).

El esquema lógico del sistema de control aparece en la figura 3.6. En él se aprecia que el control se realiza básicamente en bucle abierto, debido a que no se realimentan las posiciones de los servos (por motivos de coste y complejidad), sino sólo los datos enviados (*bucle de verificación de señales*) y el estado de los pulsadores situados en los extremos de las patas (*bucle de pulsadores*).

Todo lo que es de más alto nivel que las operaciones realizadas por la tarjeta *controladora* lo lleva a cabo el software en el ordenador externo. Éste, como se ha mencionado, está jerarquizado y distribuido en módulos, de tal forma que los de más alto nivel mandan órdenes a los inferiores. Éstos las traducen o interpretan, y las propagan hacia abajo o las ejecutan si son ellos los destinos. Esta modularidad está totalmente contrapuesta a la arquitectura "subsumption" empleada en ciertos robots de este tipo. Las ventajas que aporta son:

- Mayor claridad en el diseño del software.
- Facilidad de modificación.
- Facilidad de ampliación por niveles.
- Utilización de las técnicas del software modular.
- Menor ocurrencia de errores.

La estratificación en capas está más detallada en el capítulo 6.

3.4. Diseño definitivo

RHEX es un dispositivo con seis patas cada una de las cuales consta de tres articulaciones de rotación capaces de posicionar el extremo de la pata en cualquier punto dentro de su espacio de alcanzabilidad, aunque no con cualquier orientación. Los servos responsables del movimiento de las articulaciones están situados en las mismas, evitando de esta forma el uso de mecanismos para la propagación de sus actuaciones y simplificando el envío de órdenes.

Los extremos de las patas poseen pequeños pulsadores para detectar su posicionamiento sobre el suelo.

Sobre RHEX está la placa *controladora*, encargada de generar y mantener las consignas necesarias para posicionar los 18 servos. También contiene la lógica necesaria para leer los pulsadores y verificar que los datos enviados a la tarjeta han llegado correctamente.

RHEX es gobernado mediante un ordenador externo tipo PC debido a su baja relación precio/prestaciones. En el PC se dispone de una tarjeta de E/S PC-LabCard modelo PCL-812PG que contiene los puertos necesarios para enviar los datos a la controladora y la señal de reloj de 2MHz, imprescindible para que ésta actúe correctamente.

El software de control está jerarquizado y dividido en niveles. Éstos son tres, de más alto a más bajo: el PLANIFICADOR DE MOVIMIENTOS, el PLANIFICADOR DE TRAYECTORIAS y el EJECUTOR. Los modos de andar del robot se programan en el primero de ellos. Éste los envía al nivel intermedio, quien los descompone en las trayectorias articulares que deben seguir los servos. Estas listas de coordenadas articulares pasan a su vez al nivel inferior para su proceso y envío a la controladora.

4. Diseño mecánico

En este capítulo se detallan los componentes mecánicos que dan lugar a la estructura del robot. Éstos son tres fundamentales: los *actuadores*, las *patas* y el *cuerpo*.

Los actuadores implementan las articulaciones. Son servomecanismos de rotación que permiten el posicionado de cada articulación de forma absoluta y evitan los errores cometidos (por esfuerzos o falta de tiempo) mediante un bucle interno de control.

Las patas se diseñaron para soportar tres de estos actuadores en el menor espacio posible (ya que a mayor tamaño las holguras propias de estos servos se amplificaban, aparte del mayor peso) y permitir que la fuerza ejercida por éstos se propagara de la forma más eficiente hacia el extremo.

El cuerpo, finalmente, fue construido pensando en minimizar las colisiones entre patas sin evitar por completo el solapamiento de las zonas alcanzables por cada una.

Seguidamente se explican con más detalle estos componentes estructurales.

4.1. Los actuadores

Fueron escogidos para mover las articulaciones de RHEX servos de radiocontrol. Éstos son muy apropiados por lo siguiente:

-Al contrario que los motores paso a paso, los servomecanismos son reguladores que fuerzan una posición. Así, si se les desvía de ella, tienden siempre a volver,

minimizando el error. El control lo hacen típicamente proporcional, absorbiendo más corriente de la fuente de alimentación cuanto más distancia los separa de la posición deseada.

- La relación fuerza ejercida/consumo es relativamente alta.
- Los hay de diversos tamaños y pesos.
- Las órdenes que reciben siguen un formato PWM estándar, fácilmente generable.
- Son de relativamente bajo coste.
- Permiten la lectura de la posición a través de sus potenciómetros. Aunque esto no se ha implantado en el prototipo, es una dirección importante de mejora del mismo.

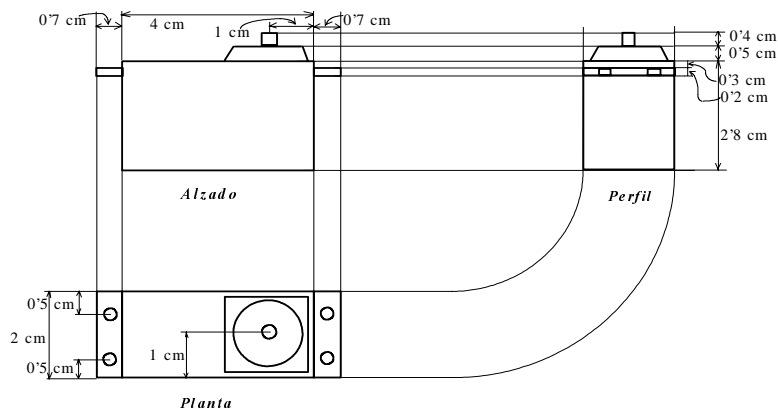


Figura A.1: Dimensiones de los servos de RHEX.

Para más detalles sobre el envío de órdenes y comportamiento de los servos véase el apéndice A.

4.2. Las patas

El dibujo simplificado de una pata aparece en la figura 4.1. Las tres piezas metálicas que la componen se han diseñado ajustando sus dimensiones para conseguir el mejor compromiso entre dos objetivos contrapuestos:

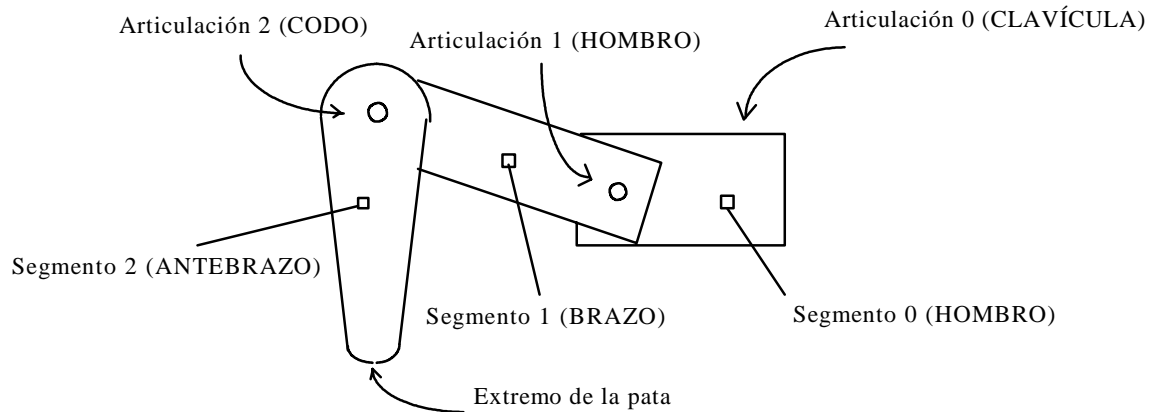


Figura 4.3: Etiquetado de elementos en una pata.

nombre del segmento 0 y el de la articulación 1 será convenientemente evitada cuando se considere necesario.

4.3. El cuerpo

El diseño del cuerpo se ha hecho teniendo en cuenta el diseño previamente realizado de las patas. Basándose en la distribución radial, se ha optimizado en varios sentidos:

1º) Sus dimensiones no deben ser demasiado grandes, para evitar futuras diferencias entre la realidad y el modelo cinemático del robot debidas a holguras e imprecisiones. Tampoco deben ser tan pequeñas como para permitir un gran solapamiento en los espacios de trabajo de cada pata, aunque una cierta compartición de espacio sí es deseable.

2º) Se ha preparado para contemplar la posibilidad de un estado especial de RHEX en el que éste tenga todas sus patas recogidas.

El resultado es un hexágono con "muescas" donde las patas pueden recogerse, ahorrando gran cantidad de espacio. En el mismo hay lugar para alojar la tarjeta controladora.

El diseño final detallado, incluyendo las dimensiones exactas, puede consultarse en el apéndice D. Un esquema del cuerpo puede verse en la figura 4.4.

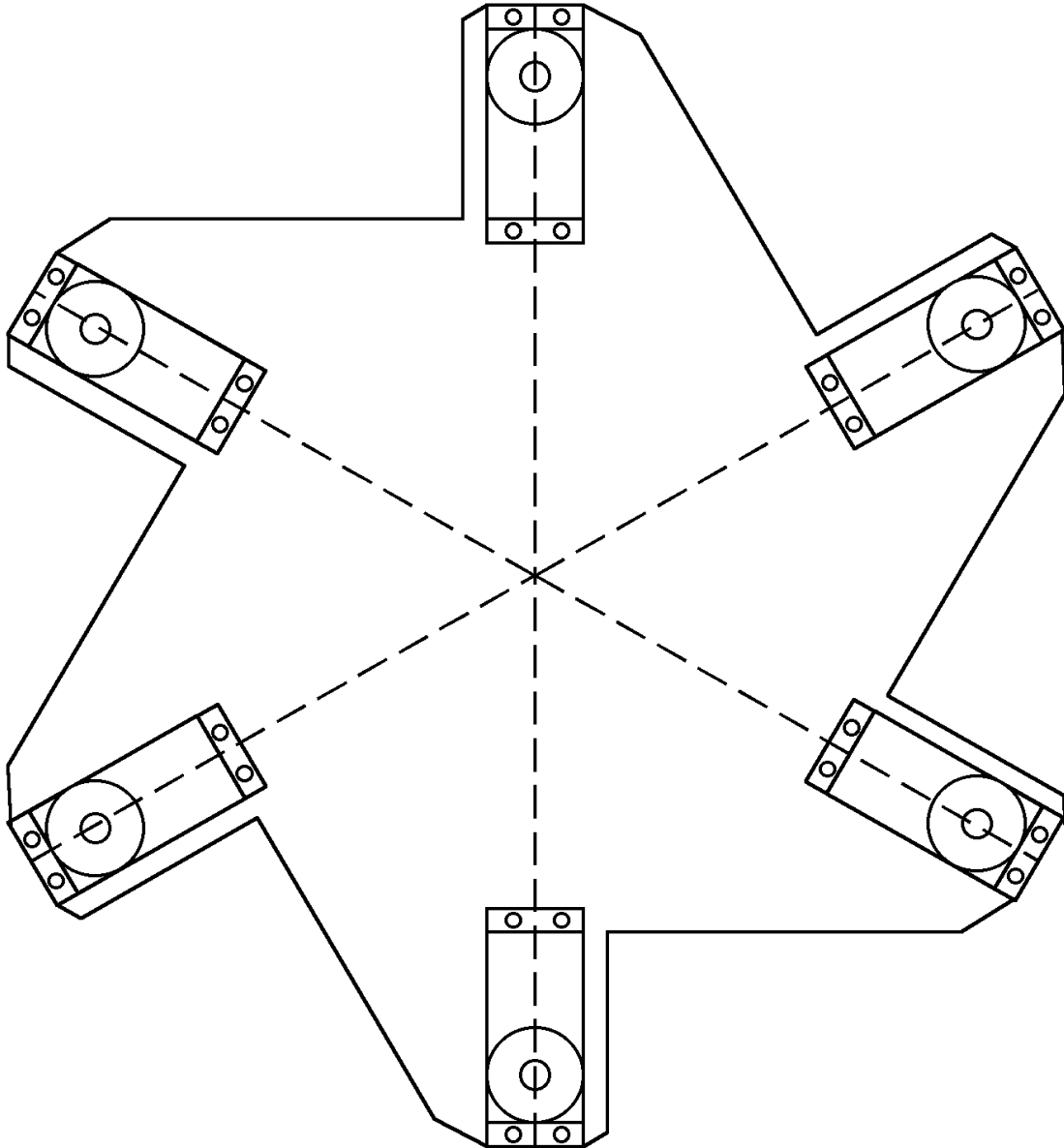


Figura 4.4.: Planta del cuerpo de RHEX, fabricado de una sola pieza, mostrando los servos de las clavículas.

5. Electrónica de control

En este capítulo se describe el aspecto electrónico del control del robot RHEX. Se diseñaron varios prototipos de circuitos capaces de generar y mantener las ondas que los servos aceptan como señales de posicionamiento. Estas ondas, llamadas PWM, codifican un dato numérico que indica al servo la posición absoluta a la que debe desplazarse y en la que debe permanecer mientras la onda no cambie. Se escogió el integrado i8253 para generar estas ondas, debido a su versatilidad y facilidad de programación. Finalmente se diseñó la llamada *tarjeta CONTROLADORA*, que incluye toda la lógica de selección de servos y generación de ondas PWM.

También se detallan en este capítulo los problemas encontrados con la alimentación de servos y controladora, y cómo se solucionaron.

Por último se presenta un estudio con el que se enumeran las características de dos tarjetas de entrada/salida con las que se hicieron pruebas de envío y recepción de órdenes entre el ordenador y la controladora. Finalmente se escogió la tarjeta PC-LabCard PCL-812PG por las razones que posteriormente se indican.

5.1. La tarjeta controladora

La controladora, situada sobre RHEX, recibe órdenes de posicionado individual de servos que transforma en ondas PWM gracias a los integrados i8253. Se basa en la señal de reloj proveniente del ordenador de 2MHz.

Dentro de este subcapítulo se explican los fundamentos de las ondas PWM, la lógica de la tarjeta controladora y los distintos bucles de control electrónico implantados en ésta.

5.1.1. Generación de ondas PWM

La señal PWM de posicionamiento de un servo es generable por un integrado i8253. Este dispositivo programable puede dar como salida hasta 3 ondas independientes de muy diversos tipos. Una descripción completa se da en el apéndice E.

Para generar una onda PWM hará falta utilizar sus tres contadores conectados en cascada. En la figura 5.1 se aprecia esta conexión.

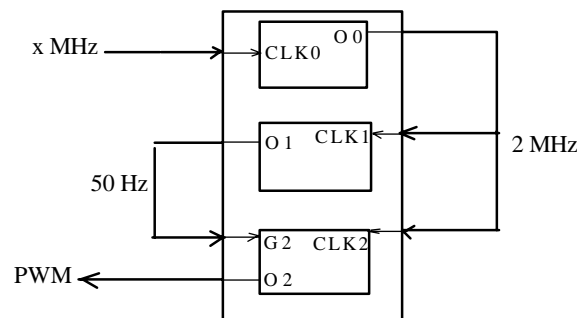


Figura 5.1: Generación de una onda PWM

-El primer contador (contador 0) se encarga de convertir la onda cuadrada del reloj externo (x MHz) a 2 MHz, que será la frecuencia base por la que se regirán los otros dos contadores. Para ello será programado en modo 3 (*Square Wave Mode*) o *modo divisor de frecuencia*. El valor de su palabra de cuenta será $x/2e6$.

-El segundo contador (contador 1) será el encargado de generar la señal **gate** del tercero. Esta señal deberá ser cuadrada, con una frecuencia de 50 Hz. Ésta es la frecuencia de

la onda PWM. Para que cumpla esta misión, se programará también en modo 3, con una palabra de cuenta de $2e6/50 = 40000$.

-Por último, el tercer contador (contador 2) generará la onda PWM. Se programará en modo 1 (*Hardware Retriggerable One-Shot*). En este modo, cada vez que se reciba un pulso alto en el **gate** se recargará la palabra de cuenta y se la irá decrementando en 1 por cada pulso de reloj. Durante este tiempo, la salida estará baja. Al terminar la cuenta, la salida pasará a estado alto hasta volver a recibir un pulso en el **gate**. De esta forma, cambiando la palabra de cuenta se puede reprogramar el ancho del pulso positivo de la PWM.

El valor que hay que cargar en el tercer contador será el nº de pulsos de longitud 0.5 microsegundos ($1/2e6$) que permanezca la onda en estado bajo. Es decir, en realidad no se programa la duración del pulso positivo, sino la del negativo. Sin embargo esto no presenta ningún problema, ya que tras cargar la cuenta, a partir del siguiente flanco ascendente en la

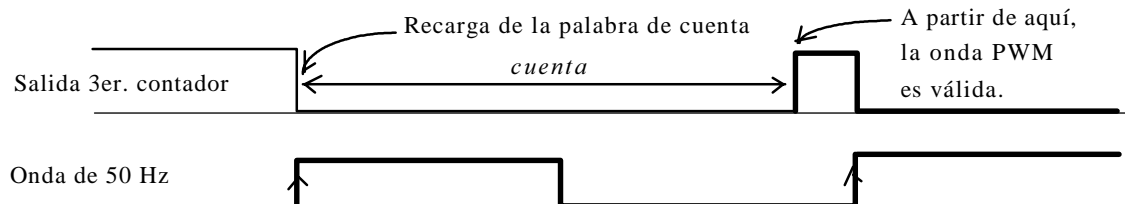


Figura 5.2: La onda PWM generada por nuestro i8253.

salida se obtendrá la onda PWM deseada. Con ello se ahorra un inversor. Este "truco" se puede ver en la figura 5.2.

Dado que habrá que programar 18 servos (6 patas con 3 servos por pata) harán falta 7 integrados i8253. Uno de ellos generará la señal PWM de 50 Hz, ya que el reloj de 2 MHz se

obtendrá directamente desde el PC. Posteriormente se utilizarán sus otros contadores para funciones auxiliares.

Cada uno de los 6 integrados restantes estará asociado a una pata. Y cada uno de sus contadores, a una articulación de esa pata (coincidentes numéricamente, es decir, el contador 0 manejará la clavícula y así sucesivamente).

Por supuesto, la tarjeta controladora deberá disponer así mismo de una lógica de

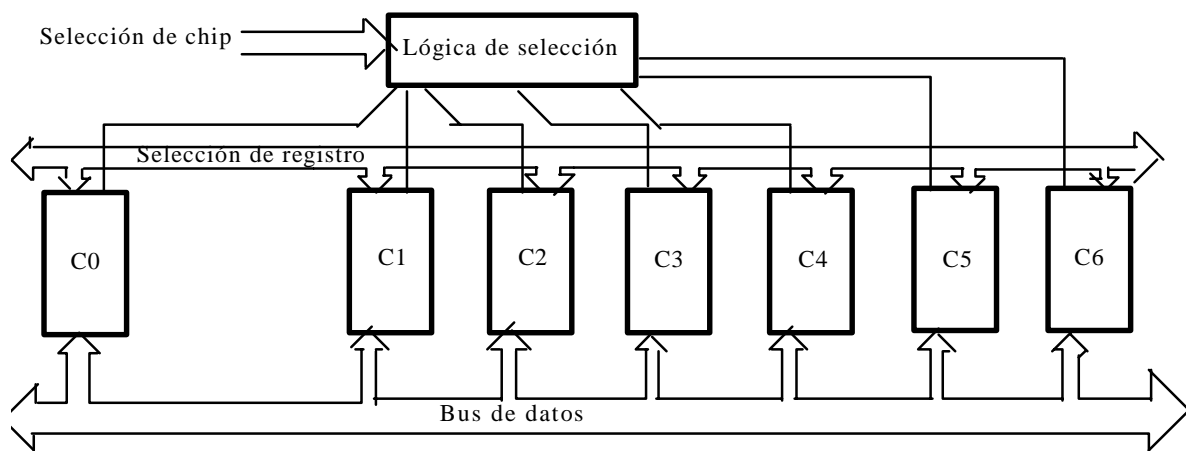


Figura 5.3: Lógica de selección de chips y registros.

selección de integrado. Ésta se puede ver esquemáticamente en la figura 5.3, y más detallada en el apéndice F.

La lógica de selección recibe un n° binario que le indica el chip a activar. Decodifica este número y activa el **chip select** de ese chip, únicamente. El bus de datos y el bus que lleva el n° de registro al que se quiere acceder de ese chip actúan siempre sobre todos los integrados. El **chip select** inhibido de aquellos no seleccionados impide que las operaciones posteriores afecten a los mismos.

La resolución obtenida en la longitud del ancho positivo de la PWM depende del reloj base de los contadores. Con el que se ha utilizado, de 2 MHz, se pueden enviar pulsos positivos de anchura comprendida entre 0.5 microsegundos (para un valor de la palabra de cuenta de 39999) y 19999.5 microsegundos (para una palabra de cuenta de 1), a incrementos de 0.5 microsegundos (correspondientes a decrementos de 1 unidad en la palabra de cuenta).

Las duraciones típicas del pulso positivo de la PWM que admiten los servos utilizados están entre 600 y 2600 microsegundos aproximadamente, valores a los cuales los cálculos anteriores se adecúan perfectamente.

5.1.2. El bucle de control

El bucle de control está muy simplificado. No se pueden obtener las verdaderas posiciones de los servos, ya que eso supondría una electrónica más compleja y la inclusión de conversores A/D y/o lógica de multiplexión analógica. Ello conlleva un control en bucle abierto. Dadas las características de los servos en cuanto a corrección automática del error, esto no supone un problema demasiado acuciante, aunque lo tocante al software de control se complica debido a la falta de información real sobre las posiciones de las articulaciones. Esta información se suple con un modelo interno del robot almacenado en la memoria del ordenador que se mantiene sincronizado en todo momento con el robot físico.

A pesar de todo ello, se pretendió suplir esta carencia de realimentación proporcionando formas de recuperar información desde el robot. Así, RHEX dispone de un doble bucle que implanta algunas características adicionales que permiten verificar de forma

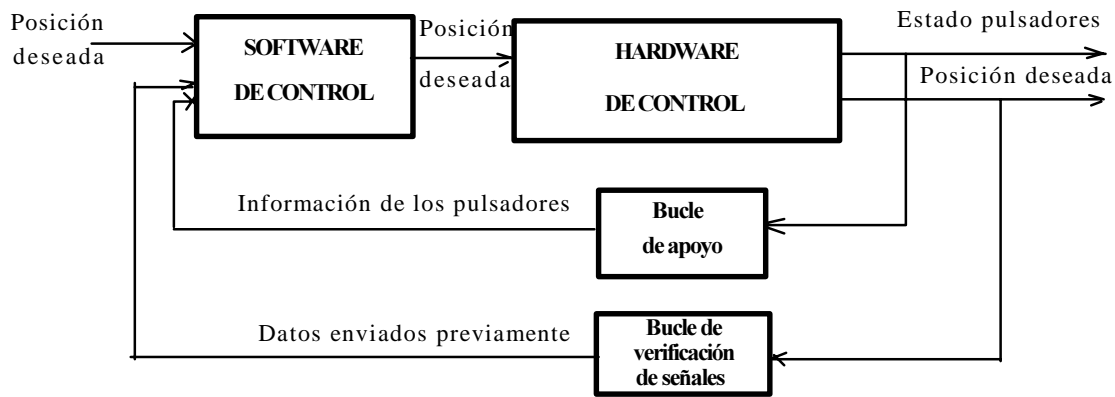


Figura 5.3: Los dos bucles de control implantados en RHEX.

limitada el cumplimiento de las órdenes enviadas. Esta información es fundamentalmente digital, por lo que no necesita una electrónica demasiado compleja que añadir a la ya comentada.

Este doble bucle se descompone en los denominados *bucle de apoyo* y *bucle de verificación de señales*, esquematizados en la figura 5.3.

El **bucle de apoyo** consiste en la realimentación del estado de los pulsadores situados en los extremos de las patas. Éstos indican al programa de control qué patas están realmente apoyadas en el suelo y cuales no, permitiendo realizar ajustes de las posiciones de las mismas. Sin embargo, en la implantación de este sistema se han encontrado varios inconvenientes:

-La baja calidad de los pulsadores existentes en el mercado, unida a los errores cometidos en las dimensiones de las patas durante su fabricación manual hacen que los datos obtenidos no sean suficientemente fiables.

-El problema básico encontrado es que un pulsador no esté activado aún estando la pata apoyada en el suelo (debido a una posición no vertical y/o a su baja sensibilidad). La

solución adoptada para evitarlo fue seguir bajando la pata hasta que el pulsador diera señal. Pero en muchas ocasiones, éste seguía inactivo cuando la pata alcanzaba el límite de su espacio de trabajo, con lo cual esta solución quedaba invalidada.

Por tanto, este bucle de control solamente se aprovecha en la búsqueda de triángulos de apoyo del robot, tal y como se explica más en profundidad en el capítulo 6.

El **bucle de verificación de señales** es de suma utilidad para evitar movimientos descontrolados del robot, debidos principalmente a la fuga o desvirtuación de las señales enviadas a la tarjeta controladora desde el ordenador. Se basa en la monitorización del contenido de los buses de datos y de selección de registro de la tarjeta controladora. Así, el software es capaz de detectar cuándo un dato enviado no ha sido recibido correctamente y en ese caso, de corregirlo o advertir al operario. Este bucle no es completo, es decir, los datos que vuelven desde la tarjeta controladora hacia el PC para su monitorización no están verificados y pueden llevar errores inducidos.

En la primera versión realizada de la tarjeta controladora no se implementaban las funciones relativas a estos dos bucles. Esta tarjeta era simplemente el extremo del sistema de control en bucle abierto. Posteriormente se extendió esta tarjeta añadiéndole otra que implantaba el bucle de apoyo, lo que permitió considerar los problemas expuestos anteriormente respecto a la baja calidad de este sistema.

Finalmente, en la tarjeta controladora definitiva, se incluyen los dos bucles (apoyo y verificación de señales) en la misma placa. Esto lleva a un diseño más complejo de la misma, pero la calidad conseguida (la placa se construyó como circuito impreso) hace que este problema se obvie.

5.1.3. Las fuentes de alimentación

Se han utilizado dos fuentes de alimentación lineales para la electrónica externa al ordenador. Una de ellas alimenta los 18 servos. El consumo de éstos es variable dependiendo de los errores encontrados entre sus posiciones ideales y las reales. Aún así, no supera en ningún momento los 4 amperios. Esta fuente proporciona 7 voltios con un amperaje máximo de 4 amperios. Es de vital importancia que la fuente utilizada para los servos tenga potencia suficiente para proporcionar este amperaje sin perder el nivel de tensión, ya que con tensiones fuera de los límites necesarios para la alimentación de los servos, el comportamiento de los mismos se hace impredecible.

La otra fuente es necesaria para aislar a la tarjeta controladora de las influencias (picos de corriente) de la alimentación de los servos. Proporciona 5 voltios y 1 amperio, energía más que suficiente para los 7 i8253 y la lógica de selección.

Debido al cambio de carga a la salida de ambas fuentes, éstas no pueden ser conmutadas, pues las fuentes de éste tipo se apagan cuando la carga varía.

Se experimentó también con alimentación en base a baterías de Níquel-Cadmio recargables, pero no proporcionaron suficiente autonomía para los 18 servos (4 baterías de 1.5 voltios conectadas en serie alimentando a 3 servos tenían una duración de 30 minutos, aproximadamente).

Otra posibilidad no ha entrado en el ámbito de este proyecto: situar las fuentes o baterías sobre el mismo robot. Ello es debido a su influencia en el sobrepeso del robot y a su excesivo tamaño en el caso de las fuentes, añadido a que éstas necesitan de todas formas una conexión a la red eléctrica.

5.2. La tarjeta de E/S

Se han hecho pruebas de control del robot con dos tarjetas de entrada y salida conectadas al PC: la I/O8255 y la PC-LabCard modelo PCL-812PG. Los mejores resultados se han obtenido con la segunda, que es la utilizada finalmente.

En la tabla que se muestra a continuación, se enumeran las ventajas e inconvenientes de ambas. Para una descripción más detallada, consúltense así mismo los apéndices G y H.

TARJETA I/O 8255:

VENTAJAS	INCONVENIENTES
-Bajo coste	-Reloj dependiente del resto del hardware de ordenador
-Menor tamaño	-Mapeado inadecuado de los pines de los conectores
-48 bits de E/S reconfigurables	-No dispone de conversores A/D, aunque éstos no son indispensables
-Los 3 contadores de un Intel 8253 disponibles	-Distinto comportamiento del reloj de la tarjeta dependiendo del modelo de ordenador en el que se instala

Tarjeta PC-LABCARD:

VENTAJAS	INCONVENIENTES
-Mayor facilidad de programación	-Mayor tamaño y coste
-Disponibilidad de conversores A/D y D/A	- 16 bits de salida digital y 16 bits de entrada digital no reconfigurables
-Reloj interno basado en un cristal de cuarzo, que proporciona una frecuencia fija y exacta de 2 MHz	-Sólo un contador de un 8253 libre para el programador
-Distribución de los pines en los conectores de forma modular	

5.3. Comunicación entre tarjetas

Se ha experimentado con cables de tipo plano de 40 hilos soportando señales digitales en paralelo. A pesar de la longitud de los mismos (1'5 metros aproximadamente) no se ha observado pérdida alguna en las señales, concretamente en la del reloj de 2 MHz, que es la de mayor frecuencia.

Así mismo, las señales no provocan interferencias sobre otras próximas.

El esquema de los distintos buses de datos que comunican la tarjeta de E/S y la controladora puede observarse en la figura 5.4.

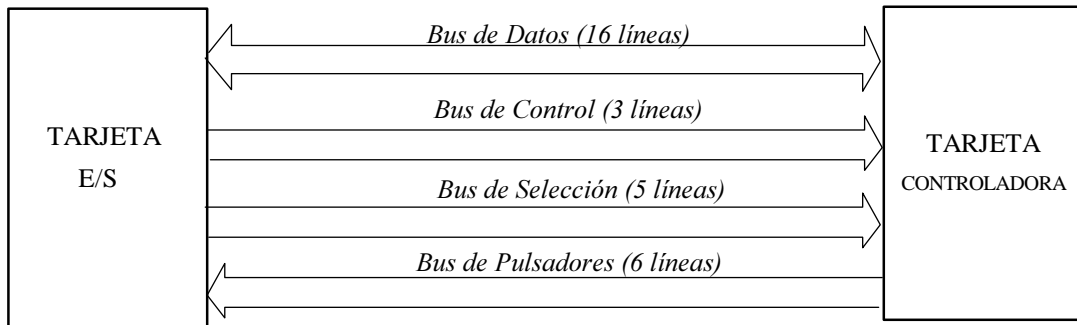


Fig. 5.4.: Buses de comunicación entre el ordenador y RHEX.

Los buses son cuatro:

-BUS DE DATOS (16 líneas): envía datos de 8 bits hacia la controladora para programar los i8253 de la misma, y recibe el último dato de 8 bits enviado.

-BUS DE CONTROL (3 líneas): contiene las señales que controlan el funcionamiento de la controladora. Las líneas son:

-Escritura (WR): un pulso bajo en esta señal hace que el último dato enviado a la controladora re programe un registro de uno de los i8253.

-Reloj (CLK): señal cuadrada periódica de 2MHz indispensable para el funcionamiento de los i8253.

-Habilitación del reloj (E_CLK): esta señal permite eliminar la señal CLK de la entrada de todos los i8253 de la controladora. Es útil para hacer que los servos dejen de ejercer control en situaciones de sobre esfuerzo.

-BUS DE SELECCIÓN (5 líneas): tres de ellas seleccionan el integrado i8253 al que afectará la próxima señal de escritura. Las dos restantes seleccionan un registro dentro del mismo.

-BUS DE PULSADORES (6 líneas): este bus unidireccional envía el estado de los pulsadores (pulsados o no) hacia el ordenador en todo momento. Una señal por cada pata.

6. Estructura del Software

El software de control se ha distribuido jerárquicamente en diversos módulos. Cada uno de ellos implementa funciones con objetivos coincidentes dentro de su nivel de abstracción. Esta orientación diverge del enfoque *reactivo* con el que se han diseñado muchos de los robots con patas actuales, entre ellos los robots tipo insecto.

Dentro de la jerarquía del software se han distinguido tres niveles de abstracción:

-*Nivel EJECUTOR*, el más cercano al hardware. Es el encargado de traducir órdenes básicas de posicionado de servos o de los extremos de las patas a señales utilizadas para programar los i8253 de la tarjeta controladora.

-*Nivel PLANIFICADOR DE TRAYECTORIAS*, o nivel intermedio. Planifica las órdenes enviadas desde el nivel más alto y las traduce a comandos básicos del nivel Ejecutor. Implementa pseudoconurrencia y detecta posibles colisiones sobre un modelo simplificado del robot.

-*Nivel PLANIFICADOR DE MOVIMIENTOS*, el nivel más alto en la pirámide de control. Genera las órdenes adecuadas para que el robot se levante, avance o se recoja, enviándolas a los niveles inferiores para su descomposición en trayectorias de articulaciones individuales y su ejecución.

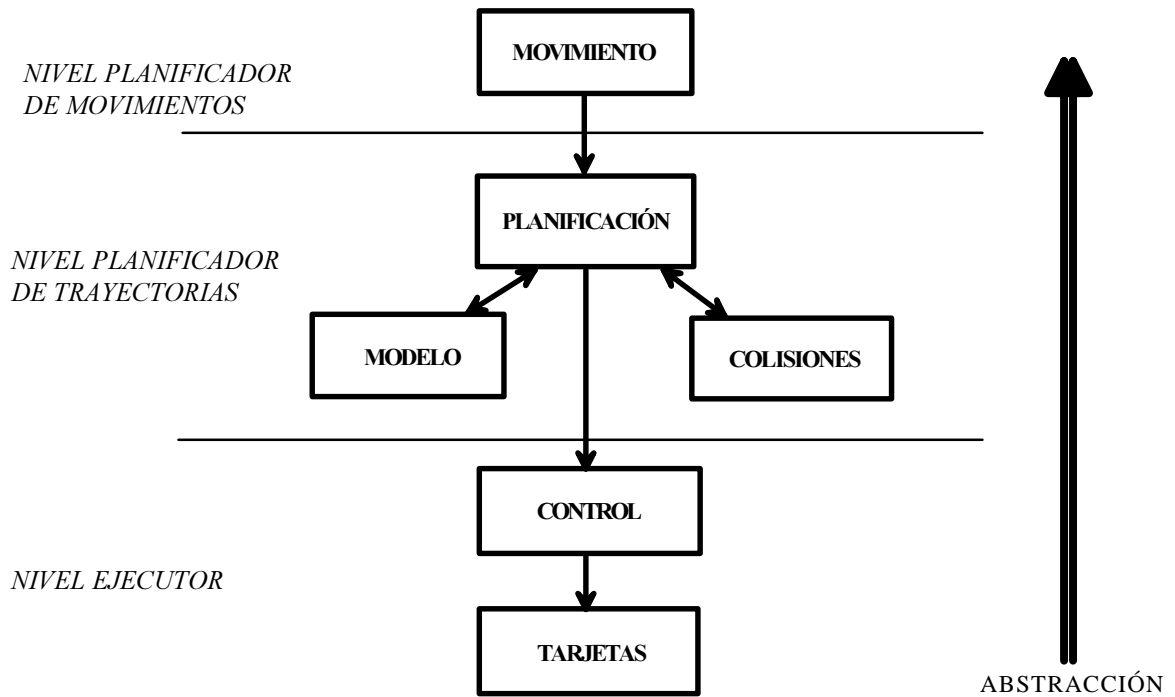


Figura 6.1: Distribución jerárquica del software de control

Esta distribución se puede ver esquemáticamente en la figura 6.1. En la misma se observan los módulos software más importantes que residen en cada nivel. Las flechas entre módulos indican el flujo de información entre los mismos para su proceso.

Seguidamente se explican con detalle estos módulos y otros que no aparecen en la figura anterior.

6.1. Nivel EJECUTOR

Este nivel contiene los módulos que permiten la traducción de órdenes elementales de posicionamiento de articulaciones a comandos para la programación de los servos de la tarjeta controladora. No se ocupa de la detección de colisiones entre las patas al efectuar estos

movimientos. Permite sin embargo la manipulación a bajo nivel de los buses descritos en el apartado 5.3. y operaciones sencillas sobre los datos de los mismos.

Este nivel se compone de dos módulos básicos: el de CONTROL proporciona el verdadero núcleo del nivel Ejecutor. El de TARJETAS es un interfaz que hace transparente al módulo de CONTROL el tipo de tarjeta de E/S que se está utilizando.

6.1.1. Módulo de TARJETAS

Tal como se ha explicado en el epígrafe 5.2 el software está preparado para trabajar con dos tarjetas de E/S comerciales. Este módulo hace transparente a los demás el manejo de cualquiera de ellas. Una vez iniciado indicando qué tarjeta se utilizará, el envío de órdenes se hará con las mismas funciones, independientemente de cuál esté activada. Internamente, el módulo ejecuta las órdenes de una forma u otra según qué tarjeta sea el destino.

Proporciona funciones para el envío de un dato de 8 bits a uno de los integrados i8253 de la tarjeta controladora, para la iniciación del módulo y para la desactivación de la controladora. Ésta última función simplemente elimina el último dato que se envió a la misma y deselecciona todos los integrados.

Además, dispone de una función que devuelve el estado de los 6 pulsadores del robot comprimidos en un octeto.

6.1.2. Módulo de CONTROL

Éste es el más importante del nivel Ejecutor. Proporciona varios grupos de funciones:

- De iniciación de la tarjeta controladora.
- De calibrado de los servos.
- De mapeado de los integrados i8253.
- De posicionado básico de las articulaciones (articular y cartesiano), incluyendo interpolaciones lineales muy simples.
- De retardo, con resolución de milisegundos, independiente del ordenador en el que se ejecute.
- De lectura de pulsadores, a más alto nivel que la proporcionada por el módulo de TARJETAS.

Seguidamente se explican con más detalle estos grupos.

6.1.2.1. Funciones de iniciación

Sólo una función pertenece a este grupo. Es de ejecución obligatoria antes de cualquier operación efectuada en cualquier otro módulo superior a éste. Inicia el reloj de 50 Hz de la tarjeta controladora, carga en memoria los datos sobre el calibrado de los servos (ver 6.1.2.2.), programa inicialmente los i8253 para situar las patas en posición de reposo (recogidas), y por último, mapea los integrados (ver 6.1.2.3.) según su posición por defecto sobre la tarjeta.

6.1.2.2. Funciones relativas al calibrado de los servos

Es necesario almacenar en alguna parte la información que permita saber qué ancho de pulso positivo de la PWM es necesario para llevar cada servo a una posición determinada. Puesto que el control del movimiento se hace básicamente en bucle abierto, se presenta un problema: la generación de esta información.

El problema se ha resuelto teniendo en cuenta una de las conclusiones que se obtuvieron tras realizar varias pruebas en servos individuales de distintas marcas: su posicionamiento es lineal. Es decir, hay una relación lineal entre ancho de pulso positivo de la onda PWM y posición del servo. Así pues, si conseguimos saber qué ancho de pulso es necesario para situar el servo a 0 grados, qué ancho de pulso es necesario para situarlo en el tope contrario y qué ancho es necesario para situarlo a 90 grados, podremos saber qué ancho es necesario para situarlo en cualquier posición, y qué rango de movimiento tiene (en grados). Las posiciones de 0 y 90 grados se escogieron por su facilidad de comprobación visual, ya que el calibrado (la recogida de estos datos para cada servo) se realiza "a mano". Es decir, una

PATA 0		
CLAVICULA		[834,2238,150]
HOMBRO		[642,2226,172]
CODO		[654,2514,191]

Figura 6.2.: Porción de un fichero de calibrado.

El primer valor de cada articulación es el ancho de pulso (en microsegundos) necesario para situar el servo a cero grados. El segundo valor es el rango en microsegundos que tiene ese mismo pulso. Y el tercer valor es el rango en grados del servo.

persona es la encargada de ir variando manualmente el ancho del pulso hasta que estime que el servo se ha situado en estas posiciones. Hay cierta imprecisión implícita en el método, pero

la exactitud del posicionado del servo no es demasiado grande (de unos 3 ó 4 grados, aproximadamente). De ahí que el robot se comporte razonablemente bien con estos datos.

En este grupo hay también funciones que permiten guardar en y recuperar de disco ficheros con toda la información necesaria para el calibrado (fig. 6.2).

Otras funciones permiten:

- Consultar datos concretos sobre la calibración de un servo.
- Modificarlos.
- Convertir de posición a microsegundos de pulso positivo y viceversa.

Se pueden ver los resultados que se pueden obtener con estas funciones en el entorno RHEX (capítulo 7).

6.1.2.3. Funciones para el mapeado de los integrados

La tarjeta controladora se sitúa físicamente sobre el cuerpo del robot, intentando minimizar la distancia desde los conectores hasta los servos. A pesar de ello, puede haber problemas con estas conexiones.

Para evitarlos en lo posible, estas funciones permiten asociar a una pata concreta uno de los integrados i8253, de forma que las órdenes enviadas al mismo se refieran a esa pata. Esto no cambia el cableado de la tarjeta (como es obvio). Solamente hace que al especificar un nº de chip para referirse a una pata, éste sea traducido al verdadero nº que está asociado a esa pata. Es decir, hay un nº de chip lógico (el que se especifica al mapear los chips sobre las

patas) y otro físico (el chip 1 siempre estará asociado a la pata 2, por ejemplo). Una vez realizado un mapeado, todas las referencias al chip lógico serán "desviadas" hacia el correspondiente chip físico.

6.1.2.4. Funciones de posicionamiento básico de articulaciones

Permiten posicionar articulaciones individuales (sin control sobre si la posición es alcanzable o no) o el extremo de una pata (en coordenadas cartesianas, con la misma salvedad). Tampoco esperan a que las articulaciones lleguen efectivamente a sus posiciones de destino. Sin embargo, existen funciones auxiliares de este grupo que calculan el tiempo estimado de llegada, basándose en las posiciones de inicio, final y en la velocidad media de un servo, que se ha comprobado experimentalmente que es suficientemente lineal.

Otras funciones de este grupo permiten generar posiciones intermedias entre origen y destino (interpolando linealmente) y ejecutar los movimientos en tres velocidades distintas. Esto es una aproximación bastante burda a las tareas realizadas por el módulo de PLANIFICACIÓN, y son sustituidas por ellas en la mayor parte de los casos.

Además, existe la posibilidad de recuperar las últimas posiciones enviadas a los servos, ya que éstas se guardan internamente.

Y por último, hay funciones auxiliares para acotar una posición dentro de los valores permitidos por la calibración de un servo, así como para calcular una medida positiva de la distancia entre dos posiciones de una pata (que será mayor cuanto más distancia deban recorrer los servos para pasar de una a otra).

6.1.2.5. Funciones para la generación de retardos

Sólo hay una función en este grupo, y es muy simple: mediante el reloj interno del ordenador, y por interrupciones, realiza retardos con resolución de milisegundos. Esta función se diseñó debido a que los retardos generables mediante funciones del compilador de C eran dependientes del ordenador.

6.1.2.6. Funciones relacionadas con los pulsadores

Basadas en la análoga del módulo de TARJETAS, pueden realizar esperas activas hasta que una pata pise el suelo (hasta que su pulsador se active) o hasta que deje de pisarlo. Así mismo, permiten de forma cómoda saber si un pulsador concreto está activado.

6.2. Nivel PLANIFICADOR DE TRAYECTORIAS

Este nivel de la jerarquía descompone los movimientos generados por el nivel superior (PLANIFICADOR DE MOVIMIENTOS) en trayectorias individuales asociadas a cada articulación, y las ejecuta en pseudoconurrencia enviando las órdenes de posicionado oportunas al nivel inferior (EJECUTOR).

Así mismo, dentro de este nivel se sitúan los módulos que implementan el modelo interno del robot, con las funciones que permiten acceder al mismo.

Los módulos más importantes que lo componen son el del MODELO (contiene el modelo de RHEX), el de COLISIONES (trata las colisiones entre patas del modelo y/o de RHEX) y el de PLANIFICACIÓN (descompone los movimientos en trayectorias articulares).

6.2.1. Módulo del MODELO SIMPLIFICADO

Se hicieron multitud de pruebas para preveer el futuro comportamiento de RHEX, cuando éste aún no estaba construido. Todas ellas desembocaron en la programación de un simulador del robot simplificado, pero capaz de ejecutar cualquier orden que pudiera enviarse al mismo. Este módulo implementa las estructuras básicas y las órdenes alternativas a las del módulo de CONTROL para el movimiento del modelo.

En el apéndice C pueden encontrarse un estudio detallado del modelo simplificado del robot.

Al igual que en el módulo de CONTROL, las funciones del modelo pueden agruparse. Estos son los grupos más importantes:

- De iniciación y calibrado del modelo.
- De tratamiento del modelo como un todo.
- De posicionado básico, articular y cartesiano, de articulaciones o patas, y de consulta de la situación actual de las mismas.

-De obtención de datos de un "esqueleto" que represente al modelo. Esto es útil para el módulo del MODELO GRÁFICO, comentado en 6.4.

-De establecimiento y consulta de la altura del robot.

-De consulta de los "ángulos de apoyo" y de cálculo y operaciones básicas del "área de barrido".

A continuación se explica detalladamente cada grupo.

6.2.1.1. Funciones de iniciación y calibrado del modelo

Existe un modelo interno en la memoria del ordenador gestionado por este módulo. La única función perteneciente a este grupo rellena ese modelo situando todas las patas en posición recogida. A esta función se le especifica también un fichero de calibración como el comentado en 6.1.2.2.

6.2.1.2. Funciones que actúan sobre el modelo como un todo

Hay tres funciones en este grupo:

-Para el almacenamiento del estado del modelo.

-Para la recuperación de ese modelo previamente almacenado.

-Para situar cada articulación del modelo tal y como esté en el robot. Esta función consulta el módulo CONTROL para obtener las últimas posiciones enviadas a los servos. La utilidad de esta función se explica mejor cuando se comenta el entorno de usuario (capítulo

7), pero básicamente sirve para sincronizar modelo y robot si se activa o desactiva alguno de ellos.

6.2.1.3. Funciones para el posicionamiento básico

Son idénticas a las del módulo de CONTROL (ver 6.1.2.4.) salvo que dirigen sus datos hacia el modelo simplificado. No existen aquí las funciones de interpolación lineal entre dos posiciones, ya que éstas fueron pensadas para realizar pequeñas pruebas de movimiento con los servos, y en el módulo de PLANIFICACIÓN hay otras análogas que lo hacen de forma más potente.

Tampoco están en este módulo las funciones para el cálculo del tiempo de movimiento, la de acotación de posiciones ni la de cálculo de una medida positiva de la distancia a recorrer entre dos posiciones. Esto es porque son perfectamente generales y válidas las que se dan en el módulo de control.

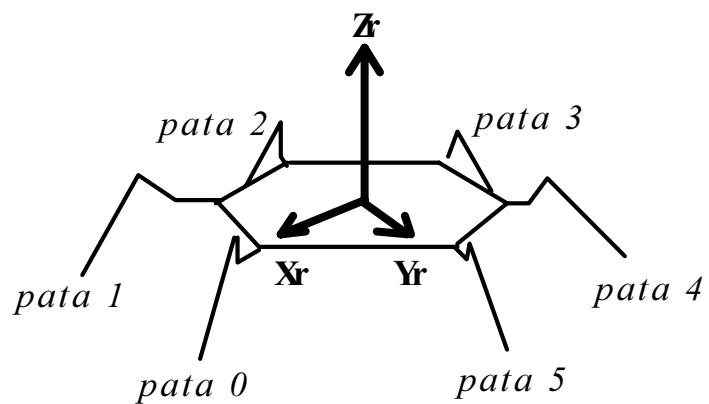


Figura 6.3.: Sistema de referencias "R"

6.2.1.4. Funciones relativas al "esqueleto" del modelo

Éstas permiten obtener las posiciones en el espacio (relativas a un sistema de coordenadas centrado en el cuerpo del robot) de los vértices de un esqueleto que representa gráficamente al modelo de RHEX. Estos datos son actualizados lo más eficientemente posible cada vez que se efectúa un movimiento en el modelo. Posteriormente, el módulo de



Figura 6.4.: Medida de la altura del robot al suelo.

MODELO GRÁFICO las proyecta en pantalla en perspectiva. De esta forma puede visualizarse cómodamente la situación del modelo y su respuesta a las órdenes enviadas.

El sistema de coordenadas referenciado será usado en todos los cálculos posteriores. A partir de ahora será denominado *sistema R*, y puede verse su definición en la figura 6.3.

6.2.1.5. Funciones relativas a la altura del robot

Esta altura es la distancia en centímetros que el robot (y el modelo) deberán mantener entre el suelo (o los extremos de las patas) y el plano que contiene a los ejes de giro de todos los hombros (articulaciones nº 1). En la práctica, será la medida que se utilizará de la distancia del robot al suelo. La figura 6.4 muestra la altura del robot gráficamente.

Como se verá en el próximo apartado, a mayor altura, menor libertad de movimiento tendrán las patas (menor espacio), pero harán menos esfuerzo. Cuando la altura es menor, el "área de barrido" (ver próximo apartado) es mayor, pero el esfuerzo de los servos para mantener el robot a esa altura, y de ahí, el consumo de los mismos, aumentan.

6.2.1.6. Funciones relativas al área de barrido y ángulos de apoyo

Los ángulos de apoyo no son otra cosa que los valores que deben mandarse a los servos de las articulaciones 1 y 2 de una pata (hombro y codo) para hacer que ésta sostenga al robot a la altura establecida sobre el suelo. Existe una función que calcula estos dos ángulos (el de la clavícula es indiferente dada la configuración física del robot).

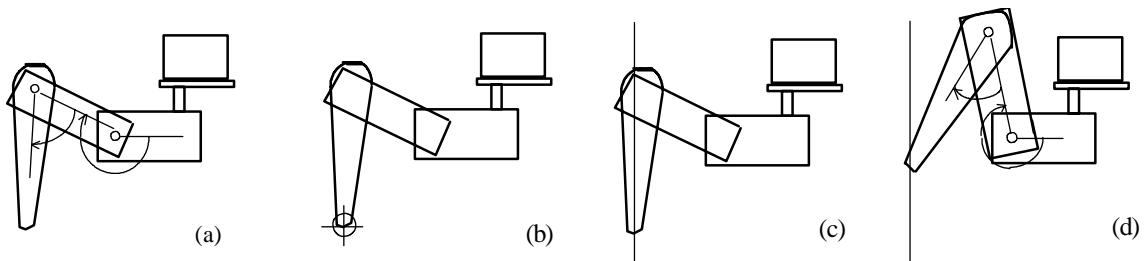


Figura 6.5: Conceptos relacionados con los ángulos de apoyo

- (a) Ángulos de apoyo.
- (b) Punto de apoyo.
- (c) Vertical de apoyo.
- (d) Ángulos de sobreapoyo.

Hay otros conceptos que se emplean relacionados con estos ángulos:

Punto de apoyo de una pata: punto del espacio en el que se situará el extremo de la misma cuando su hombro y codo se hayan situado en las posiciones indicadas por los ángulos de apoyo, y la clavícula a 90 grados.

Vertical de apoyo de una pata: la recta paralela al eje OZ del sistema R y que pasa por el punto de apoyo de esa pata.

Ángulos de sobreapoyo de una pata: análogos a los ángulos de apoyo, hacen que la pata se sitúe en la vertical de apoyo, con el extremo a la misma altura que el borde inferior del segmento 0 de la pata (hombro).

En la figura 6.5 se pueden ver estos conceptos representados gráficamente.

Existe otro subgrupo de funciones dentro de este grupo. Están relacionadas con el cálculo del área de barrido de una pata. Para mayor claridad en la exposición, se exponen a continuación varios conceptos previos:

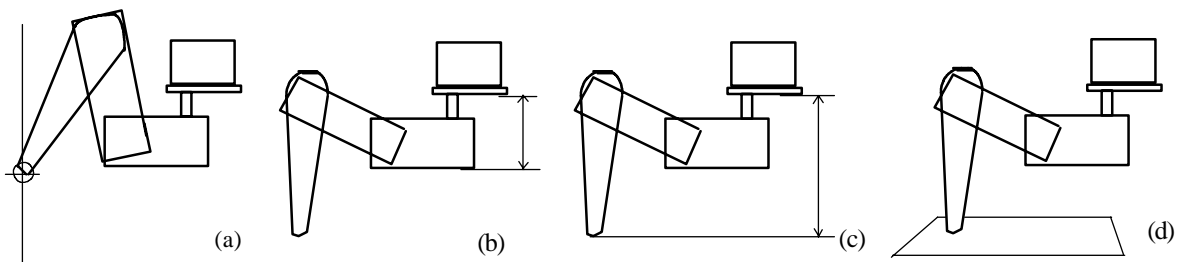


Figura 6.6: Conceptos relativos al área de barrido.

- (a) Punto de sobreapoyo.
- (b) Altura de la estructura del robot.
- (c) Altura efectiva del robot.
- (d) Plano de apoyo.

Punto de sobreapoyo de una pata: punto en el que se sitúa el extremo de la misma cuando el hombro y el codo están en las posiciones indicadas por los ángulos de sobreapoyo y la clavícula a 90 grados.

Altura de la estructura del robot: distancia positiva entre el punto de apoyo de cualquier pata y su punto de sobreapoyo.

Altura efectiva del robot: resultado de sumar la altura del robot y la altura de su estructura.

Plano de apoyo: plano paralelo al plano OXY del sistema R, situado por debajo del

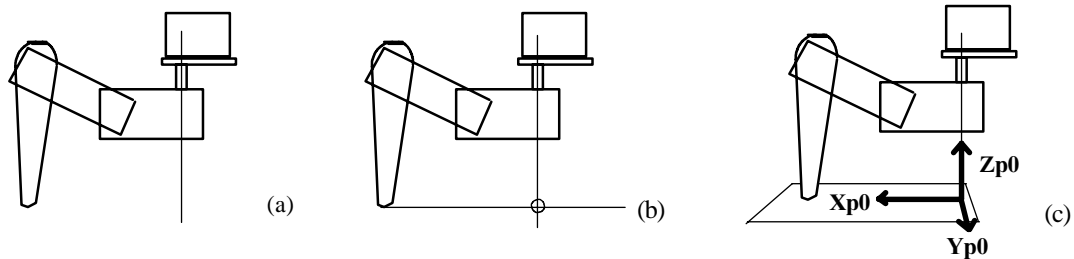


Figura 6.7: Más conceptos previos al área de barrido.

- (a) Vertical de clavícula.
- (b) Punto intersección de clavícula.
- (c) Sistema de referencias P_0 .

mismo, a una distancia negativa igual en módulo a la altura efectiva del robot.

Estos conceptos previos se muestran gráficamente en la figura 6.6.

Vertical de clavícula de una pata: recta paralela al eje OZ del sistema R, y que contiene al eje de la articulación de la clavícula de la pata.

Punto intersección de clavícula (abrev. PIC): punto en el que corta la vertical de clavícula de una pata al plano de apoyo del robot.

Sistema de referencias P_n : sistema de coordenadas asociado a la pata n , cuyo origen está en el punto intersección de clavícula de esa pata, cuyo eje OZ es paralelo y tiene el mismo sentido que el del sistema R, y cuyo eje OX es paralelo al segmento 0 (hombro) de esa pata cuando la articulación 0 se sitúa a 90 grados.

De nuevo pueden observarse estos conceptos en la figura 6.7.

Y al fin se explican los dos conceptos más importantes:

Área total de barrido de una pata: zona cerrada y en general no convexa, contenida en el plano de apoyo de la pata, y que contiene todos los puntos alcanzables por el extremo de la misma al mismo tiempo que mantiene la altura del robot constante.

Área efectiva de barrido de una pata, o Área de barrido: zona cerrada y convexa contenida en el área total de barrido, que simplifica los cálculos realizados para el movimiento del robot y que permite que éste se lleve a cabo sin que la pata pase por puntos no alcanzables.

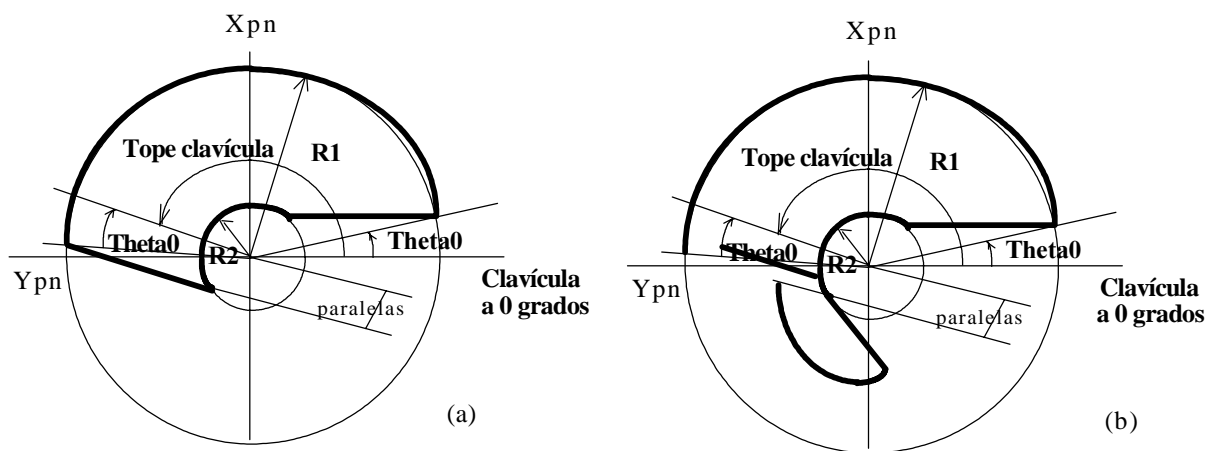


Figura 6.8: los dos tipos de áreas totales de barrido.

Se pueden distinguir dos tipos de áreas totales de barrido, dependiendo de la altura del robot:

Tipo 1: el área tomará la forma indicada en la figura 6.8.a si, desplazando el extremo de la pata desde el punto de apoyo hacia el punto intersección de clavícula de la misma siguiendo la recta que los une, y manteniendo en el movimiento la altura del robot, llegamos a un punto en el que el eje del codo se sitúa por debajo del plano de apoyo. Es decir, la articulación del codo colisiona con el suelo antes de llegar al PIC de la pata.

Tipo 2: el área total de barrido tomará la forma mostrada en la figura 6.8.b si el

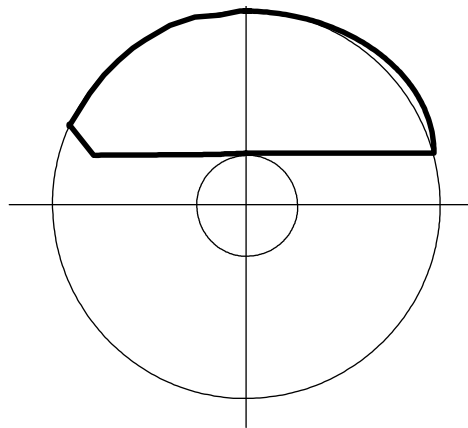


Figura 6.9: Área efectiva de barrido de una pata.

extremo de la pata puede llegar o superar al PIC de la misma, siguiendo el mismo movimiento que el descrito en el párrafo anterior, sin que la articulación del codo colisione con el suelo.

Las áreas de tipo 1 se consiguen con una altura del robot pequeña, mientras que las de tipo 2 son típicas de alturas grandes, tales que permiten a las patas alcanzar puntos debajo del cuerpo de RHEX sin que los codos colisionen con el suelo.

Como se ve en la figura 6.8, el tipo 2 no es de ninguna utilidad, ya que si bien permite una zona extra de movimiento debajo del cuerpo del robot, la mayor parte de las trayectorias rectas que pasen por ella intersectan los límites del área de barrido, es decir, pasan por posiciones no alcanzables.

De ahí que al final se use un área efectiva de barrido resultado de simplificar las fronteras del área total de tipo 1. Éste área efectiva se muestra en la figura 6.9.

Por último, existen otras funciones en este subgrupo. Una de ellas, dado un punto en el interior del área efectiva de barrido y una dirección, calcula el punto más lejano del punto dado, en esa dirección, que esté dentro del área. Otra calcula la distancia entre esos dos puntos, y otra calcula las coordenadas en el sistema R de cualquier punto perteneciente a la recta que une a esos dos puntos. Todo esto se usa para el movimiento de avance del robot (ver el subcapítulo 6.3).

6.2.2. Módulo de COLISIONES

En este módulo se encuentran las funciones necesarias para la detección eficiente de colisiones entre las patas del robot. En una primera fase de diseño, las colisiones entre patas se calculaban mediante intersecciones de prismas: uno por cada segmento de cada pata. Esto, si bien daba una gran precisión, conllevaba un tiempo de cálculo excesivo ($(3*3*2)*6 = 108$ cálculos de intersección

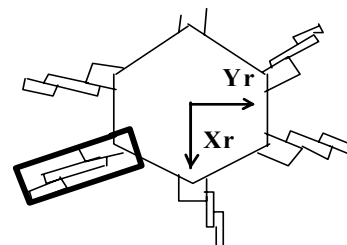


Figura 6.10: Rectángulo límite de la pata 1.

prisma a prisma). Puesto que los espacios de trabajo de las patas permitían una facilidad de posicionamiento sin colisiones relativamente grande, se optó por simplificar el cálculo a costa de una menor precisión. En la versión definitiva, las colisiones se detectan basándose en los "rectángulos límites", explicados en 6.2.2.1.

Las funciones de este módulo pertenecen a tres grupos diferenciados:

- De "rectángulos límite".
- De análisis de futuros movimientos.
- De optimización de la detección de colisiones.

6.2.2.1. Funciones relativas a los "rectángulos límite"

El rectángulo límite de una pata está contenido en el plano OXY del sistema R, y es el rectángulo de menor área que contiene en su interior la proyección ortogonal en el plano OXY del sistema R de todos los segmentos de la pata.

Se puede observar un ejemplo de rectángulo límite en la figura 6.10.

Como se ve fácilmente, el chequeo de colisiones entre dos patas consecutivas (no existe otro caso posible de colisión entre dos patas dadas las dimensiones del robot) se reduce al cálculo de intersecciones entre dos rectángulos contenidos en el mismo plano. Esto conlleva $4*6 = 24$ cálculos de intersección semirrecta-rectángulo, que se pueden simplificar si se usa un sistema de coordenadas en el que uno de los rectángulos límite tenga los lados paralelos a los ejes. Así, el cálculo de la intersección de las semirrectas que forman los lados del otro rectángulo límite con este rectángulo se reducen en gran medida.

La función perteneciente a este grupo devuelve las coordenadas del rectángulo límite de una pata.

6.2.2.2. Funciones para el análisis de movimientos

Hay una única función en este grupo. Se encarga de devolver información sobre las colisiones que ocurrirían si una articulación de una pata efectuara cierto movimiento. Uno de sus parámetros le indica si la detección de colisiones se realiza sobre el modelo o sobre el robot.

Esta función calcula las intersecciones entre los rectángulos límite de la pata que efectúa el movimiento y los de las dos patas adyacentes. Informa a su regreso de con qué patas de estas dos colisionaría (y si llegaría a hacerlo), además de indicar si el movimiento provocaría que la articulación que lo va a efectuar se saliera del rango posible de movimiento, contemplado en los datos de calibración del servo. Toda esta información ocupa una tripleta de enteros: dos para indicar con qué patas colisionaría y otro que indica si la posición está fuera de rango.

6.2.2.3. Funciones para la optimización de los cálculos

Debido a que incluso la simplificación que supone el uso de rectángulos límite no es capaz de evitar la relativamente grande cantidad de cálculos de intersección (además supone la aplicación de una matriz homogénea para rotar los rectángulos hasta ser paralelos a un sistema de referencias), se ha previsto la posibilidad de optimizar algunas partes del cálculo. Esto se puede llevar a cabo siempre que haya memoria disponible para ello, puesto que la

optimización consiste en almacenar en memoria los resultados de ciertos cálculos, previamente generados. Concretamente, se optimiza el cálculo de la dimensión longitudinal del rectángulo límite (el más largo de los dos lados, el único que es variable: depende de los valores que tengan los ángulos de las articulaciones de hombro y codo de la pata). Para cada pareja de valores de articulación (hombro,codo) se almacena en una gran tabla la longitud del rectángulo límite.

6.2.3. Módulo de PLANIFICACIÓN

Este es, con diferencia, el módulo más complejo y potente de todo el software de RHEX. En esencia, realiza una planificación de varias trayectorias (secuencias de posiciones asociadas a una articulación de una pata) simulando concurrencia, con posibilidad de sincronizarlas en distintos instantes de tiempo.

El módulo está jerarquizado internamente. Esto es debido en principio a que se realizó una versión preliminar (la que ahora implanta los *planes*) para realizar diversas pruebas. Posteriormente se le añadió un refinamiento (que implanta los *superplanes*).

La primera capa, más básica, permite:

- Diseñar y ejecutar en pseudoconurrencia planes (conjuntos de varias trayectorias)
- Sincronizar las trayectorias de un plan, pero sólo en su finalización. Es decir, hacer que finalicen al mismo tiempo.
- Realizar interpolaciones lineales entre dos posiciones cartesianas del extremo de una pata.

-Especificar los puntos de las trayectorias en coordenadas articulares o cartesianas (en este caso, del extremo de una pata).

-Especificar las trayectorias mediante un pseudolenguaje muy simple en el que cada punto de una de ellas va acompañado de un código que le dice al programa si éste es cartesiano, articular, etc.

-Consultar el estado de cada trayectoria una vez finalizado el plan (información sobre colisiones y ajustes de posiciones fuera de rango).

La segunda capa es una extensión de la anterior. Añade las siguientes características:

-Manejo de *superplanes*. Éstos son una ampliación de los planes del nivel anterior, e incorporan:

-Sincronización en cualquier punto de una trayectoria.

-Especificación de posiciones relativas, tanto en coordenadas articulares como cartesianas.

-Realización de interpolaciones entre posiciones articulares o cartesianas.

-Incorporación de un pseudolenguaje más potente para la especificación de trayectorias, que contempla las nuevas posibilidades de los superplanes.

-Información posterior a la ejecución del superplan mucho más completa y específica que la proporcionada en la capa básica.

Las funciones que componen este módulo son muchas y muy diversas, por lo que a continuación se explican los conceptos fundamentales en que se basan y no cada función en particular.

6.2.3.1. Planes

Un plan consta de 18 trayectorias, una por cada articulación de cada pata, que pueden estar definidas o no. A la hora de ejecutarlo, debe haber al menos una definida.

Hay varias funciones que permiten la definición completa de una trayectoria. Se puede crear una trayectoria mediante la llamada a funciones que añaden puntos, entre los cuales puede haber una interpolación lineal o no. Esta forma supone la adición de puntos uno a uno. O bien puede crearse directamente una trayectoria mediante un pseudolenguaje básico. Este lenguaje no incorpora palabras clave. Es simplemente una lista de puntos. Estos puntos son conjuntos de 4 valores: los 3 primeros especifican unas coordenadas. El cuarto, si es 0, indica que las coordenadas son articulares. Si es mayor que 0, por ejemplo n , indica que las coordenadas son cartesianas y se refieren al extremo de la pata. Además, entre el punto anterior de la trayectoria y ese punto, habrá que realizar una interpolación lineal de n pasos. De esta forma, se definen 6 posibles trayectorias, que internamente se expanden a las 18 (ya que los puntos de la lista contienen los 3 valores de las articulaciones de la pata).

En la ejecución de un plan, puede especificarse que se sincronice la finalización de todas las trayectorias. También se especifica en dónde se ejecutará el plan: en el modelo, en el robot o en ambos. Además, se establece el nivel de detección de colisiones. Éste puede ser de 4 tipos:

Nivel 0: se ignorarán las colisiones entre patas y las posiciones fuera de rango. La ejecución del plan continuará a pesar de las mismas. Hay que destacar la diferencia existente entre *posiciones fuera de rango* y *posiciones inalcanzables*. Las primeras son acotadas al tope

más cercano del servo. Las segundas sí provocarán error, ya que suponen situar la pata en una posición imposible.

Nivel 1: si hay alguna colisión, detiene la trayectoria asociada a la articulación que la ha provocado. Las demás trayectorias continúan en ejecución, y los ajustes de posiciones fuera de rango siguen siendo ignorados (acotados).

Nivel 2: sigue ignorando las posiciones fuera de rango, pero cualquier colisión provocará que el plan se detenga por completo.

Nivel 3: el más restrictivo. Cualquier colisión o acotación de posición hará que el plan se detenga.

Tras la detención (con éxito o no) de un plan, se puede consultar el estado en que se quedó cualquiera de las trayectorias. La información proporcionada es idéntica a la de la función de análisis de movimientos, explicada en 6.2.2.2. De este modo pueden averiguarse las causas que provocaron la detención.

Otra facilidad de que se dispone a la hora de ejecutar un plan es la especificación de una función de tratamiento del teclado. Cada vez que se pulse una tecla durante la ejecución del plan, será llamada esta función. Lo que ésta realice no influirá en la reanudación de la ejecución, salvo en que se le permite indicarle al planificador que se debe terminar la misma. Esto es útil si queremos disponer, por ejemplo, de una tecla que aborte inmediatamente la ejecución de un plan.

6.2.3.2. Pseudoconurrencia

La ejecución en paralelo de las distintas trayectorias de un plan se realiza mediante un algoritmo de *rebanadas de tiempo con asignación de prioridades dinámica*. Previamente a la ejecución del plan, todos los puntos de las trayectorias han sido expandidos y/o traducidos a coordenadas articulares.

El algoritmo es como sigue:

- 1.- Enviar a los servos (o al modelo, o a ambos) todas las posiciones de las trayectorias que no estén en espera. Al comienzo del plan, ninguna trayectoria está en espera, con lo que se lanzarán al mismo tiempo todas las posiciones iniciales. Si previamente al envío de alguna posición a una articulación se detecta que habrá una colisión o acotación, se actúa en consecuencia con el nivel establecido de tratamiento de colisiones o ajustes.

- 2.-Tras enviar una posición a un servo, esa trayectoria se queda en estado de espera, y almacena el tiempo estimado que tardará el servo en situarse en la posición especificada. Éste será denominado *tiempo de espera* de la trayectoria.

- 3.-Se busca el menor tiempo de espera que se haya calculado.

- 4.-Se resta ese tiempo de todos los tiempos de espera de todas las trayectorias en espera, quedando aquellas que tuvieran ese tiempo mínimo preparadas para seguir ejecutándose.

- 5.-Se llama a la función de tratamiento del teclado si se detecta alguna tecla pulsada.

- 6.-Se vuelve a 1 a menos que todas las trayectorias hayan terminado o se haya recibido orden de terminar la ejecución desde la función de tratamiento del teclado.

6.2.3.3. Superplanos

Dotan al módulo de planificación de una mayor potencia. Básicamente son la ejecución consecutiva en el tiempo de varios planes. Los puntos de unión de éstos constituyen los puntos de sincronización del superplan.

Se puede especificar un superplan completo con sólo una lista de valores, relativamente pequeña. Este pseudolenguaje tiene una potencia muy superior a la del de especificación de trayectorias de planes, estando pensado para trabajar a muy alto nivel.

Cada punto del superplan contiene varios valores:

-Un octeto de control. Especifica si el punto contiene coordenadas articulares o cartesianas, relativas o absolutas, interpoladas o no (pudiendo hacer cualquier combinación de estas tres características). Y si tras este punto del superplan debe haber una sincronización de todas las trayectorias. Las coordenadas *relativas* son añadidas a la posición anterior de la trayectoria para obtener las verdaderas coordenadas. Esta facilidad permite desplazar una pata del robot, por ejemplo, relativa a su posición actual, y sin saber cuál es ésta.

-Un entero que sólo será considerado en caso de que el octeto de control indique que hay que hacer una interpolación. Este valor da el nº de pasos de la misma.

-Una tría de valores que especifican las coordenadas. Pueden ser articulares (en grados) o cartesianas (en centímetros).

-Un octeto que indica a qué patas se les aplica éste punto del superplan. De esta forma se puede hacer fácilmente que las patas impares bajen 2 cm (por ejemplo) con sólo un punto del superplan.

En los listados que acompañan a esta memoria, se puede apreciar la potencia de esta especificación. Basta ver en la función encargada de hacer que el robot se levante, programada en el módulo de MOVIMIENTO, la especificación del superplan asociado.

Dentro de los superplanes, cabe destacar también la información que puede obtenerse sobre el estado final de la ejecución de un superplan. Ésta incluye lo siguiente:

-Tipo del error que provocó la finalización de la ejecución: o no hubo error, o hubo alguna colisión o ajuste, o no hubo memoria suficiente para expandir los datos del superplan, o se llegó a una posición inalcanzable por alguna pata, o el usuario abortó la ejecución por teclado.

-Mapa de colisiones: una tabla bidimensional. Si la celda (i,j) contiene un valor distinto de cero, la pata i colisionó con la j durante la ejecución del superplan.

-Mapa de ajustes de posiciones fuera de rango: una tabla bidimensional. Si la celda (p,s) contiene un valor distinto de cero, es porque la articulación s de la pata p recibió una posición fuera de rango.

Por último, todas las demás características de la ejecución de planes (especificación de nivel de detección de errores, direccionamiento de la salida al robot o al simulador, especificación de la función de tratamiento del teclado) se conservan en la función de ejecución de superplanes.

6.3. Nivel PLANIFICADOR DE MOVIMIENTOS

Este nivel contiene los módulos necesarios para implantar el modo de andar (*gait* en inglés) de RHEX, así como el resto de movimientos de alto nivel (levantamiento, reposo, cambio de altura). Todo ello lo ofrece el módulo de MOVIMIENTO, descrito a continuación.

6.3.1. Módulo de MOVIMIENTO

En este módulo se dan las funciones necesarias para realizar los movimientos de más alto nivel: levantamiento, avance, reposo y cambio de altura. Además, incluye todo lo relacionado con los "triángulos de apoyo".

6.3.1.1. Funciones relativas a los triángulos de apoyo

Un triángulo de apoyo está formado por los extremos de tres patas del robot (o del modelo) que lo mantienen estable sin necesidad de apoyar las otras patas. Puede haber varios triángulos de apoyo distintos para una misma posición global de RHEX. En el caso del robot, el que una pata esté apoyada en el suelo se chequea verificando su pulsador. En el caso del modelo, se ordenan las patas por orden creciente de sus coordenadas Z en el sistema R (coordenadas Z de sus extremos), y las primeras 3 son las que se consideran apoyadas en el suelo.

Existe una función que chequea si tres patas dadas forman triángulo de apoyo. Esto se hace comprobando si la proyección vertical (siguiendo el eje OZ del sistema R) del centro de

masas del robot sobre el plano que forma el triángulo está en el interior del mismo. El centro de masas se considera coincidente con el origen del sistema R, pero la función que devuelve sus coordenadas puede modificarse en diseños futuros para contemplar un cálculo más preciso de éste (dependiente de los pesos de las patas y su distribución en cada momento).

6.3.1.2. Funciones para el movimiento de alto nivel

Como ya se ha explicado, existen cuatro movimientos de alto nivel:

-Levantamiento: partiendo de una postura recogida (todas las articulaciones con valores de reposo, idénticos a los que tienen justo después de iniciar los módulos de CONTROL y/o de MODELO SIMPLIFICADO), se pretende que el robot se levante verticalmente hasta quedar a la distancia del suelo especificada por el operario. Esto es un movimiento que se realiza con las 6 patas al mismo tiempo (para minimizar el esfuerzo y de ahí el consumo de los servos). Éstas empiezan situándose en sus respectivos puntos de sobreapoyo. Una vez todas estén ahí, comienzan a descender siguiendo la vertical de apoyo (en una trayectoria cartesiana relativa e interpolada) hasta llegar a los puntos de apoyo. La postura alcanzada finalmente se denominará *robot en pie*.

-Reposo: partiendo del robot en pie, las patas comienzan a subir de nuevo por la vertical de apoyo hasta llegar a sus puntos de sobreapoyo. De allí, se recogen hasta sus posiciones de reposo a la vez, sin interpolar. Si al principio del movimiento de reposo el robot no estaba en pie, se hace una llamada a la función de levantamiento.

-Avance: se desarrolla en un bucle. Antes y después de cada paso, el robot deberá encontrarse en pie. Durante cada paso, se hará que avance cierta distancia en la dirección

especificada. El bucle se repetirá hasta agotar la distancia que se le ordenó avanzar o hasta que haya algún error, lo que antes suceda. El algoritmo que sigue en su avance es el de *trípode alternante*, que se muestra a continuación a grandes rasgos:

- 1.-Calcular la mínima distancia que se pueden desplazar las patas en la dirección de avance. Si ésta es mayor que la distancia que el robot debe avanzar, escoger ésta última como mínima distancia.

- 2.-Levantar las patas pares y situarlas a esa distancia. Mientras, el robot se apoya en las patas impares.

- 3.-Desplazar todas las patas en la dirección contraria esa distancia. Como todas se encuentran en el suelo, conseguirán desplazar el cuerpo del robot en la dirección de avance, puesto que empujarán contra el suelo.

- 4.-Levantar las patas impares y situarlas de forma que el robot de nuevamente en pie. Las patas pares no deben realizar este movimiento puesto que tras los pasos 2 y 3 ya están en esa posición.

- 5.-Sumar a la distancia total recorrida, esa mínima distancia.

- 6.-Si aún no se ha recorrido toda la distancia especificada inicialmente, volver al primer paso del algoritmo.

Este algoritmo fue el inicialmente implantado en el robot. Sin embargo, es susceptible de modificaciones y optimizaciones, como por ejemplo, alternar el uso de las patas pares con las impares para apoyarse. Así se distribuye el trabajo más uniformemente entre todos los servos.

6.4. Módulos Auxiliares

En este subcapítulo se explican el resto de módulos que componen la capa de software desarrollada para RHEX. Estos módulos no pertenecen a la jerarquía anteriormente descrita, pero son usados por casi todos los módulos de la misma.

6.4.1. Módulo de VECTORES 3D

Contiene todas las operaciones necesarias sobre vectores de tres componentes: escalado, suma, resta, módulo, módulo al cuadrado, productos escalar y vectorial, copia, normalización y vector nulo.

6.4.2. Módulo de TRANSFORMACIONES

Realiza funciones de definición y manejo de transformaciones homogéneas (4x4). Tiene varios tipos de ellas predefinidos: identidad, rotaciones, escalado y perspectiva. Las operaciones básicas son la concatenación de matrices y la aplicación de una matriz a un vector 3D.

6.4.3. Módulo de optimización MATEMÁTICA

Una vez iniciado, rellena tablas internas de senos y cosenos. Así se acelera el cálculo posterior de los mismos.

6.4.4. Módulo de CINEMÁTICA

Este módulo consta sólo de dos funciones: cinemática directa de una pata y cinemática inversa de una pata. Las dos trabajan con coordenadas en centímetros relativas al sistema R del robot.

La cinemática directa de una pata obtiene, dados los valores articulares de la misma, la posición en el espacio de su extremo.

La cinemática inversa de una pata obtiene hasta dos soluciones para los valores articulares, dadas las coordenadas que debe tener el extremo de la pata. En caso de que haya dos soluciones, las ordena, devolviendo en primer lugar la de menor coste a partir de la posición actual.

Este y otros módulos utilizan un fichero donde residen todas las medidas reales del robot. Éste fichero se incluye en los fuentes antes de compilarlos.

El desarrollo de las fórmulas utilizadas en ambas funciones puede estudiarse en el apéndice I.

6.4.5. Módulo de CÁMARAS

Este módulo contiene funciones para la definición de cámaras en el espacio tridimensional. Permite posteriormente asociarles porciones de la pantalla gráfica, para así proyectar puntos y segmentos en perspectiva. Es usado por el modelo gráfico para mostrar en pantalla una proyección tridimensional en perspectiva del modelo del robot.

6.4.6. Módulo del MODELO GRÁFICO

Es una capa software que se sitúa envolviendo completamente al módulo del MODELO SIMPLIFICADO (ver 6.2.1). Ofrece funciones análogas a éste para mover articulaciones, pero las órdenes que recibe las envía previamente a un modelo gráfico interno (*wireframe*) para actualizar la proyección en pantalla del modelo del robot.

Tiene así mismo varias cámaras predefinidas, intercambiables: vista aérea y vista frente a cada una de las patas. Puesto que utiliza el módulo de CÁMARAS, permite todas las posibilidades de éste, entre ellas la manipulación de una cámara para rotarla, desplazarla, etc.

6.4.7. Módulo de RATÓN

Permite manejar el dispositivo apuntador dentro del entorno de usuario.

Ofrece funciones para la detección de su posición y el diseño de su representación en pantalla gráfica.

6.4.8. Módulo de ENTORNO gráfico

Contiene todas las funciones necesarias para definir y utilizar un entorno gráfico con menús "pull-down" manejado con ratón. Se usa para integrar todas las funciones de los módulos en un entorno de usuario (ver capítulo 7).

6.4.9. Módulo de VENTANAS

Añade al entorno gráfico la posibilidad de abrir ventanas compuestas por varios campos. También contiene funciones para la definición y manejo de ventanas de edición de texto en modo gráfico.

6.4.10. Módulo principal del entorno de usuario

Contiene al programa que genera el entorno de usuario RHEX. Es descrito ampliamente en el capítulo 7.

7. Entorno de usuario RHEX

El entorno gráfico se diseñó para integrar todas las funciones descritas en el capítulo 6 en un sólo programa. De esta forma, el usuario puede fácilmente cambiar la altura del robot, calibrar servos, o visualizar los movimientos ejecutándolos sobre el modelo gráfico, sobre el robot, o sobre ambos, por poner sólo los ejemplos más característicos de uso del entorno.

Tal y como se explica en 6.4.9, el entorno está basado en menús pull-down. Existe una barra en la parte superior de la pantalla que contiene los grupos de operaciones ejecutables. Al pulsar con el ratón en una de ellas, se despliega un menú con esas operaciones, cada una de las cuales puede a su vez desplegar un nuevo submenú.

Info Principal Calibrado Ejecución

PATA 0	CLAVICULA	[834,2238,150]
	HOMBRO	[642,2226,172]
	CODO	[654,2514,191]
PATA 1	CLAVICULA	[798,2202,155]
	HOMBRO	[654,2214,167]
	CODO	[606,2370,195]
PATA 2	CLAVICULA	[702,2010,158]
	HOMBRO	[760,2240,162]
	CODO	[720,2580,181]
PATA 3	CLAVICULA	[834,2178,150]
	HOMBRO	[582,2034,168]
	CODO	[486,2418,198]
PATA 4	CLAVICULA	[894,2226,154]
	HOMBRO	[534,2190,175]
	CODO	[582,2466,211]
PATA 5	CLAVICULA	[786,2178,154]
	HOMBRO	[630,2142,164]
	CODO	[522,2418,198]

En la zona izquierda de la pantalla se pueden editar por teclado los ficheros de calibrado del robot. Además en ella se mostrarán instrucciones, informes, etc.

En la zona superior derecha hay un esquema que representa al robot y que se utilizará para seleccionar patas, articulaciones e integrados, simplemente pulsando sobre ellos. Bajo esta zona hay un botón deslizante que sirve para calibrar servos y variar la altura del robot.

En la zona derecha inferior se proyecta la visión en perspectiva del *wireframe* que representa al modelo del robot. Esta vista puede cambiarse entre siete vistas posibles, y cada una de ellas puede variarse manualmente.

Entre estas dos últimas zonas existe una línea de mensajes que da instrucciones breves sobre las operaciones a llevar a cabo para realizar ciertas funciones.

En la figura de la página anterior se muestra una pantalla típica del entorno RHEX.

A continuación se explica cada una de las opciones de los menuses del entorno.

7.1. Info

Este menú contiene una única opción que da información acerca de la versión del entorno y los créditos del mismo.

7.2. Principal

Iniciar tarjeta: llama a las funciones de inicialización de los módulos de CONTROL y MODELO GRÁFICO. El robot volverá a la posición de *recogido*.

Mapear chips: permite cambiar el mapeado de los i8253 de la tarjeta controladora. Una vez pulsada esta opción, se deberán seleccionar las parejas chip-pata deseadas, terminando al pulsar en el rectángulo rojo adyacente a la tarjeta. Para seleccionar una asociación chip-pata se pulsa sobre el chip y luego sobre la pata.

Altura robot...: Esta opción permite especificar la altura del robot de tres formas:

Ajuste manual: moviendo el botón deslizador mediante el ratón, se varía la altura en centímetros. El botón se puede mover de forma rápida (pulsando en su centro y moviendo el ratón lateralmente) o de forma más precisa (pulsando repetidamente en ambos extremos del botón). Para terminar la operación, se pulsa sobre el botón azul del deslizador, que pasará a rojo para indicar desactivación.

Ajuste numérico: Se puede introducir una altura en décimas de centímetro, siempre que esté entre los dos límites físicos (3'5 y 12'5 cm).

Altura por defecto: Establece la altura por defecto, de 7'4 cm. A esta altura, las patas sitúan hombros, codos y clavículas a 90 grados.

Nivel de error...: mediante esta opción se especifica el nivel en la detección de colisiones o ajustes de posiciones fuera de rango (ver 6.2.3.1). Este nivel aparecerá en forma de "led" en la parte derecha de la barra superior de menú, con un número en su interior que indica el nivel.

Ni colisiones ni ajustes: nivel 0 de detección de errores.

Detención individual por colisión: nivel 1.

Detención total por colisión: nivel 2.

Detención total por colisión o ajuste: nivel 3.

Ver últimos errores: da un listado explicativo de las causas que motivaron el cese en la ejecución del último movimiento. Si hubo colisión entre patas, indica cuáles estuvieron implicadas. Si hubo ajustes de posiciones fuera de rango, indica en qué articulaciones de qué patas se produjeron. En esencia, muestra la información que devuelve la función de ejecución de un superplan (ver 6.2.3.3)

Activar optimizaciones: llama a la función de activar optimizaciones en la detección de colisiones (ver 6.2.2.3). Si no hay memoria, no se activan.

Desactivar optimizaciones: libera la memoria ocupada por las tablas de optimización de la detección de colisiones.

Salir del programa: termina la ejecución del entorno. Resetea la tarjeta controladora (sitúa el robot en modo *recogido* y deselecciona todos los i8253) y vuelve al DOS.

7.3. Calibrado

Calibrar...: este menú permite calibrar manualmente servos individuales o varios servos. Depende de la subopción que se escoja:

Articulación: para calibrar un servo solamente. Una vez seleccionada esta opción, se debe pulsar sobre una pata del esquema del robot de la zona superior derecha de la pantalla, y luego sobre el segmento que es movido por la articulación a calibrar. A continuación se mueve el botón deslizador siguiendo las instrucciones que aparecerán en la línea de mensajes. Bajo la ranura del deslizador, a ambos extremos, aparecen

los dos valores límite para el ancho del pulso positivo de la onda PWM que se está mandando al servo. Éstos son 600 y 2600 microsegundos.

Pata: pide que se seleccione una pata y luego entra en un bucle para calibrar (de la misma forma que la opción anterior) cada articulación de esa pata.

Robot: comienza directamente un bucle en el que llama a la opción anterior para cada pata.

Editar calibrado...: este submenú permite modificar mediante teclado los valores de calibración de los servos.

Actual: para editar el calibrado que haya en memoria en ese momento.

De fichero: para editar el calibrado que haya grabado en un fichero en disco.

No afecta al calibrado cargado en memoria.

Grabar calibrado: permite grabar en fichero el calibrado que actualmente tiene el robot.

Cargar calibrado: carga un calibrado desde disco e inicia el robot con ese calibrado.

Es de notar que el calibrado activo en cada momento es el mismo para el robot que para el modelo, por lo que cualquier modificación del mismo afectará a ambos.

7.4. Ejecución

Cámara...: mediante este menú se puede variar la vista del modelo del robot. Las opciones que tiene son:

Frente pata 0, ..., Frente pata 5: cambia la posición y orientación de la cámara para situarla frente a la pata deseada.

Vista superior: sitúa la cámara sobre el eje OZ del sistema R, mirando hacia su sentido negativo. Es decir, se verá al modelo desde arriba. Es la vista por defecto.

Ajuste manual: al activar esta opción aparecerán unas breves instrucciones en la zona izquierda de la pantalla, que explican cómo mover la cámara mediante el teclado. Se puede desplazar y cambiar de orientación en el espacio. El desplazamiento se realiza a lo largo de la línea de visión (la línea en la que estamos mirando), hacia adelante o hacia atrás. La orientación se cambia modificando los tres ángulos que la definen: el *giro* hará rotar la cámara alrededor de su línea de visión, el *cabeceo* hará mirar hacia arriba y hacia abajo, y la *panorámica* es como el homónimo cinematográfico: moverá la imagen hacia ambos lados. Aparte de la modificación del estado de la cámara, mediante esta opción se pueden escoger los elementos que se mostrarán del modelo: el esqueleto (siempre aparece), los rectángulos límite y un triángulo de apoyo.

Ver posiciones...: este submenú permite ver las coordenadas articulares y cartesianas del robot o del modelo.

De RHEX: muestra en la zona izquierda las del robot.

Del Simulador: muestra las del modelo.

Movimientos...: este submenú contempla los movimientos predefinidos por el software:

Situar articulación: para especificar numéricamente (en grados) la posición que se desea que tome una articulación concreta, que previamente habrá que seleccionar.

Levantamiento: pone al robot en pie. (ver 6.3.1.2)

Reposo: lo vuelve a poner *recogido*.

Cambio de altura: permite cambiar la altura del robot directamente, haciendo que éste descienda o ascienda hasta alcanzarla partiendo de la posición *en pie*.

Avance: hace que el robot ande en cierta dirección una distancia especificada.

La dirección y la distancia se dan moviendo el ratón hasta obtener los valores adecuados. En la zona de proyección del modelo aparecerá una vista superior del mismo (muy simplificada) sobre la que se situará el vector cuya

orientación equivalga a la dirección de avance y cuya longitud sea proporcional a la distancia a recorrer.

Conexión con RHEX...: este submenú realiza el direccionamiento de órdenes hacia modelo y/o robot. El destino que se seleccione aparecerá en la barra superior como un "led",

de color verde si las órdenes van al modelo, de color rojo si van al robot y de color amarillo si van a ambos. En su interior aparecerá un signo de admiración si las órdenes afectan al robot. En cualquier caso, cualquier movimiento puede ser interrumpido pulsando la tecla ESCAPE. Al finalizar el movimiento aparecerá a la izquierda la información referente a las causas de su terminación.

Ejecutar en RHEX: los movimientos se efectuarán sólo en el robot.

Ejecutar en Simulador: los movimientos se efectuarán en el modelo solamente.

Ejecutar en ambos: los movimientos serán enviados a ambos. Esto enlentece el movimiento del robot, ya que intercala los cálculos del modelo gráfico.

Visualizar pulsadores: esto es un test de prueba que se detendrá pulsando cualquier tecla. Durante el mismo, las patas del esquema de selección de patas (zona superior derecha de la pantalla) se iluminarán si sus respectivos pulsadores están activos (pulsados).

Grabar estado simulador: almacena la situación actual del modelo en memoria (ver 6.2.1.2).

Recuperar estado simulador: recupera la situación que se guardó por última vez.

8. Pruebas y Resultados

Se hicieron distintas pruebas desde el mismo comienzo del proyecto, hasta culminar en el movimiento de avance del robot ya construido. A continuación se explica cada una de ellas.

8.1. Pruebas de utilización de servos

Una vez conocida la forma de la onda que haría posicionarse al servo, se diseñó una pequeña tarjeta que, conectada a la tarjeta I/O 8255, generaría la PWM programable. Esto dio como resultados:

- Constatación de la linealidad en el posicionamiento.
- Constatación de la linealidad en la velocidad y de la magnitud de la misma.
- Diseño del método de calibración de servos y de un primer prototipo software para el mismo.
- Diseño del primer núcleo software, capaz de enviar a un servo la onda que lo haría mantenerse en la posición deseada.

Esta tarjeta, que sólo contenía un i8253 y una sencilla lógica que permitía seleccionar uno de los registros del mismo, fue la que evolucionó durante el diseño posterior hacia lo que es la actual *tarjeta controladora*.

8.2. Primeros diseños de pata

En la figura 4.2 se puede ver la primera pata que se construyó. Puesto que la tarjeta de control que se montó para hacer pruebas con los servos era capaz de enviar órdenes a tres a la vez (los relojes de 2 MHz y 50 Hz los tomaba de la tarjeta de E/S) se usó para hacer pruebas con este primer diseño de pata.

Se hizo un pequeño programa que movía la pata simulando que caminaba. De estas pruebas se sacaron la siguientes conclusiones:

- Este diseño de pata sufría holguras excesivas.
- La fuerza que eran capaces de desarrollar los servos podría sostener a un robot completo.
- La velocidad de movimiento de los servos era aceptable para ser utilizados en movimientos más complejos, siempre que se utilizaran en la mayor parte de los mismos posicionamientos articulares, y no interpolaciones cartesianas.
- El material empleado para la pata (aluminio de 1 mm de espesor recortado manualmente) resistía perfectamente los esfuerzos a que era sometido.

Posteriormente se mejoró este diseño de pata hasta llegar a la pata definitiva, mucho más compacta pero sin una pérdida demasiado grande de espacio de trabajo ni de fuerza.

8.3. Pruebas con el modelo del robot

Se hizo un programa que contenía el núcleo básico de lo que después serían los módulos del MODELO SIMPLIFICADO y MODELO GRÁFICO. Mediante el mismo se obtuvieron los siguientes resultados y conclusiones:

- Necesidad de un módulo de planificación para poder ejecutar en pseudoconurrencia tareas complejas del robot.
- Diseño de métodos de simplificación de los cálculos en la proyección en perspectiva del esqueleto del modelo.
- Diseño del método definitivo de detección de colisiones. Inicialmente se utilizó un método basado en prismas, para más tarde llegar al método de los rectángulos límite.
- Cálculo y representación gráfica de las fronteras de las áreas totales de barrido de las patas, lo que permitió a su vez diseñar el método de avance del robot.

Posteriormente, y una vez implantado el núcleo del entorno RHEX, se diseñaron los movimientos estratégicos del robot, que no han sufrido modificaciones importantes desde entonces.

8.4. Pruebas con las tarjetas de E/S

Con el robot ya construido, aunque todavía alimentado desde el ordenador y con pilas de NiCd recargables, se intentó programarlo a través de la tarjeta I/O 8255. Ésta dio los problemas señalados en la tabla 5.1. Por ello se creó el módulo de TARJETAS, que es capaz de manejar ésta y la PC-LabCard, que es la que se usó en el diseño final. Hasta ese momento, todas las pruebas se habían realizado con la I/O 8255.

La programación de la PC-LabCard no dio mayores problemas.

8.5. Las fuentes de alimentación

En las primeras pruebas de movimiento con todas las patas (ya se habían implantado las funciones de movimiento estratégico y se habían probado sobre el modelo) se alimentaron los servos con una fuente conmutada, extraída de un ordenador PC. El cambio en la carga hacía que ésta se apagara continuamente, con lo que se optó por una fuente lineal. También se usó una de este tipo para alimentar la tarjeta controladora, que ya tenía conectados los 7 i8253.

Las pruebas realizadas de movimiento llevaron a varias conclusiones y resultados:

- Obtención del consumo de los servos (no más de 3'5 Amperios con los 18 servos activos).
- Obtención del pico de corriente de los mismos al efectuar movimientos con distinta duración (0'75 A aprox.)

Así mismo, se perfeccionaron ciertos detalles en los algoritmos de movimiento. Por ejemplo, al principio, el movimiento de levantamiento se realizaba con 3 de las patas (las pares), pero el consumo (= esfuerzo) de los servos medido por la propia fuente que los alimentaba era demasiado grande. Ello llevó a modificar el movimiento para que actuaran a la vez las 6 patas del robot.

9. Mejoras y futuras líneas de desarrollo

Debido a la amplitud del proyecto, muchas ideas que han surgido durante su desarrollo no se han podido llevar a cabo. Algunas de ellas son de tal extensión que podrían dar lugar a otros proyectos completos, tan amplios como el presente.

A continuación se enumeran las más obvias:

-CIERRE DEL BUCLE DE CONTROL. El efectuar lecturas de las posiciones de los servos provocaría una simplificación enorme en la programación de todos los módulos, sobre todo los del nivel EJECUTIVO. El movimiento sería mucho más exacto y más flexible, y todo con un menor esfuerzo del programador.

-ADICIÓN DE SISTEMAS DE PERCEPCIÓN. Posibles opciones son: pulsadores más precisos y sensibles, buscador de norte, sensores de fuerza en los extremos de las patas, "bigotes de gato" (para la detección de choques), sensores infrarrojos (para la detección del movimiento) e incluso cámaras miniatura, aunque esto último supondría un mayor esfuerzo de programación, no sólo en la interpretación de las imágenes, sino en la necesidad de mantener la orientación del cuerpo del robot constante mientras anda. Ello llevaría a la inclusión de giróscopos.

-PROGRAMACIÓN DE OTROS MODOS DE CAMINAR. Sería interesante implantar un módulo que sustituyera al de MOVIMIENTO y que permitiera moverse al robot

según distintos "*gaits*", así como algoritmos para moverse por entornos no uniformes (rugosos, con obstáculos, blandos o deformables, inclinados, etc).

-DESARROLLO DE AUTONOMÍA. El hacer a RHEX autónomo le daría una nueva dimensión a todo el software. Se necesitarían módulos de más alto nivel que el de MOVIMIENTO, encargados de planificar trayectorias que posiblemente contemplarían la detección de obstáculos. Además, el hardware debería ser apropiadamente adaptado para este cometido: la estructura del robot debería ser reforzada y redistribuidos sus componentes, ya que habría que situar fuentes de alimentación (preferentemente baterías) y tarjetas más complejas sobre ella (por ejemplo: una placa madre de pequeño tamaño, existente actualmente en el mercado, junto a una disquetera para cargar los programas, si no se utilizan EPROMs).

APÉNDICE A

Servomecanismos de Radiocontrol

En este apéndice se muestra una descripción completa de los servos utilizados en la construcción de RHEX.

A.1. Aspectos mecánicos

Existen multitud de modelos de servos para radiocontrol. Las dimensiones de los escogidos se pueden ver en la figura A.1.

Estos servos ejercen una fuerza de 3'0 Kg/cm y tardan 0'22 seg. en moverse 60°.

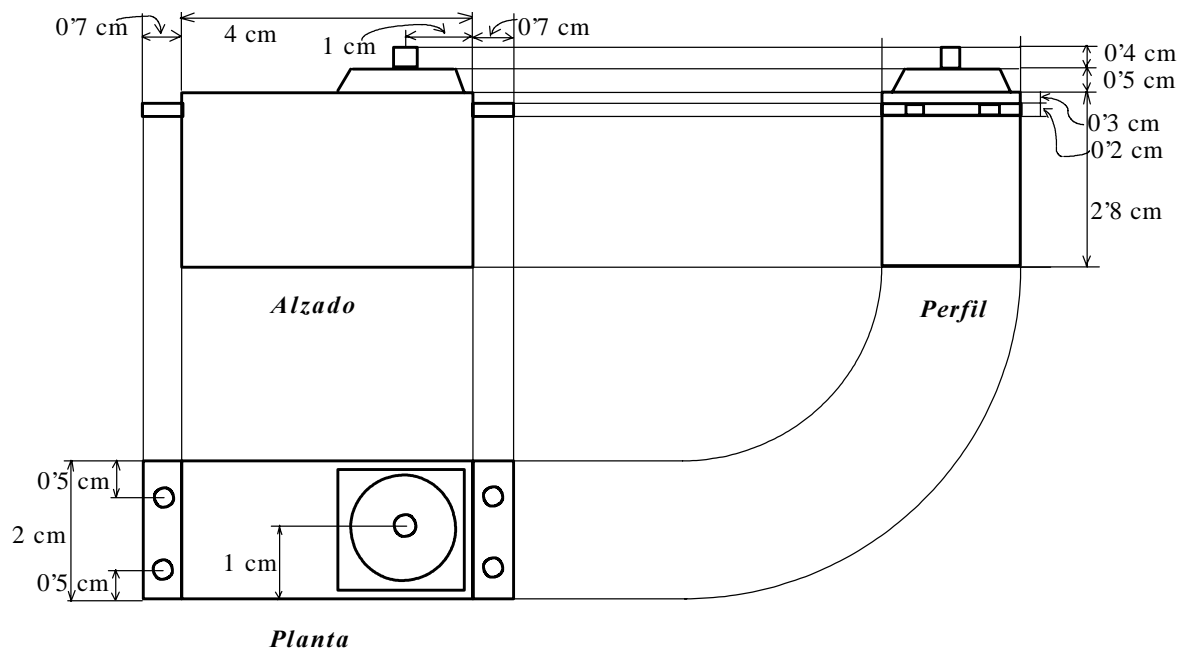


Figura A.1: Dimensiones de los servos de RHEX.

A.2. Aspectos electrónicos

Los servos trabajan con un rango de alimentación relativamente amplio (desde unos 5 voltios hasta 8 voltios, aproximadamente). A mayor voltaje de alimentación, mayor velocidad y fuerza en la respuesta.

El posicionamiento se realiza enviando al servo una onda digital con formato PWM (*Pulse Width Modulation* o *modulación del ancho del pulso*). Este formato es una variante de una onda cuadrada que se diferencia de ésta en que la duración del pulso positivo (o negativo) es variable, manteniendo siempre la misma frecuencia. Esta duración codifica el dato que porta la onda. En el caso de los servos, la duración es proporcional a la posición absoluta que deberá tomar el eje.

La resolución obtenida en la longitud del ancho positivo de la PWM depende del reloj base de los contadores. Con el que se ha utilizado, de 2 MHz, se pueden enviar pulsos positivos de anchura comprendida entre 0.5 microsegundos (para un valor de la palabra de cuenta de 39999) y 19999.5 microsegundos (para una palabra de cuenta de 1), a incrementos de 0.5 microsegundos (correspondientes a decrementos de 1 unidad en la palabra de cuenta). No es recomendable utilizar valores de la palabra de cuenta superiores a 39999 (ni 0, que es equivalente a 65536), ya que hacen que la duración del pulso negativo alcance o supere los 20 milisegundos, que es el período básico de la onda PWM de 50 Hz. Dado que la señal **gate** no afecta al contador mientras éste no termine su cuenta, estos casos "degenerados" provocan pulsos positivos anormales, cuya longitud es de $40000 - c * 0.5$ microsegundos (donde **c** es la palabra de cuenta, $c \geq 40000$). Esto no es de ninguna utilidad, ya que esta longitud estará comprendida entre 20000 ($c=40000$) y 7232 ($c=0$, que en realidad es 65536) microsegundos,

valores abarcados por los casos "no degenerados"; y además hacen que la onda PWM cambie su período por otro mayor, de 40 milisegundos (25 Hz).

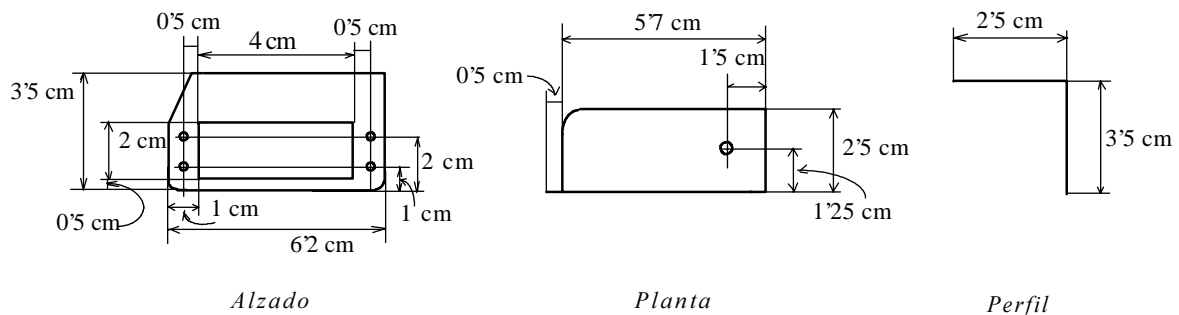
Las duraciones típicas del pulso positivo de la PWM que admiten los servos utilizados están entre 600 y 2600 microsegundos aproximadamente, valores a los cuales los cálculos anteriores se adecúan perfectamente.

Estructura mecánica de RHEX (I): patas

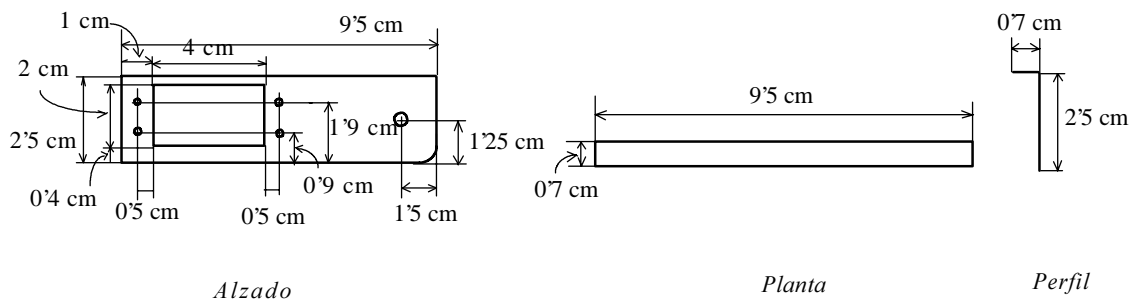
Las tres piezas que se ensamblan para formar una pata de RHEX se han fabricado en aluminio de 1 mm de espesor. Se han recortado manualmente de ángulos en "L" de este metal para dotarlas de mayor resistencia.

Los tres segmentos y sus medidas exactas son mostrados en las figuras siguientes.

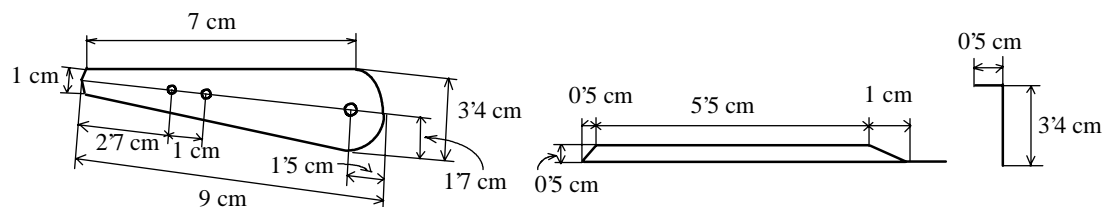
SEGMENTO 0 (Hombro)



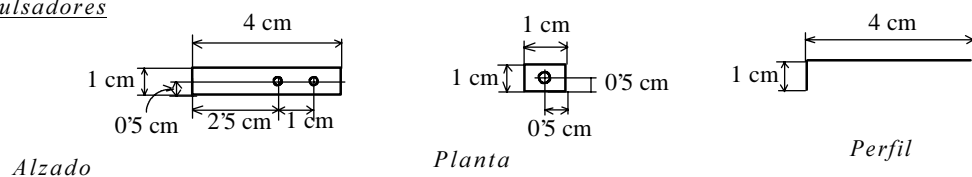
SEGMENTO 1 (Brazo)



SEGMENTO 2 (Antebrazo)



Soporte de pulsadores



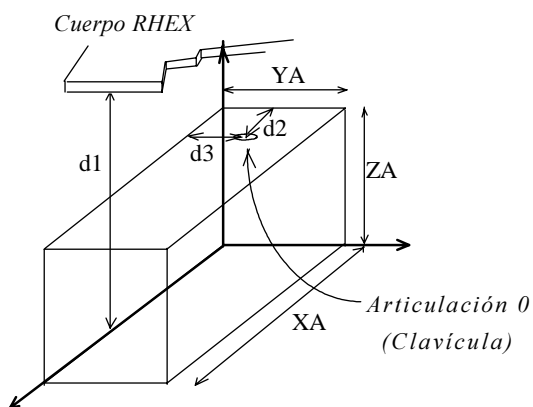
Nota: taladros de sujeción de 4 mm de diámetro; taladros para paso de ejes de 4 mm.

APÉNDICE C

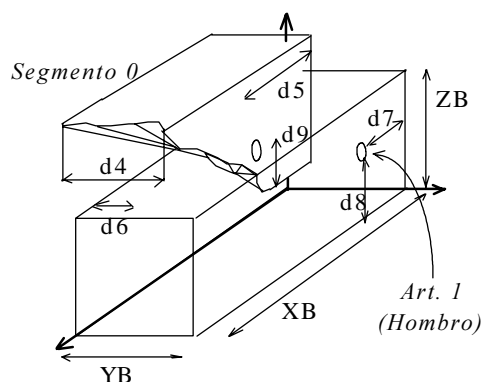
Modelo simplificado de RHEX

A continuación se muestran los sistemas de referencia y las medidas más importantes del modelo de RHEX basado en prismas (uno por cada segmento). El modelo que realmente

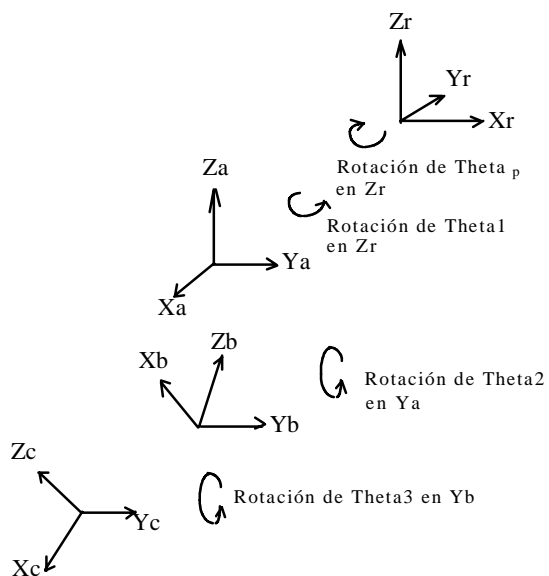
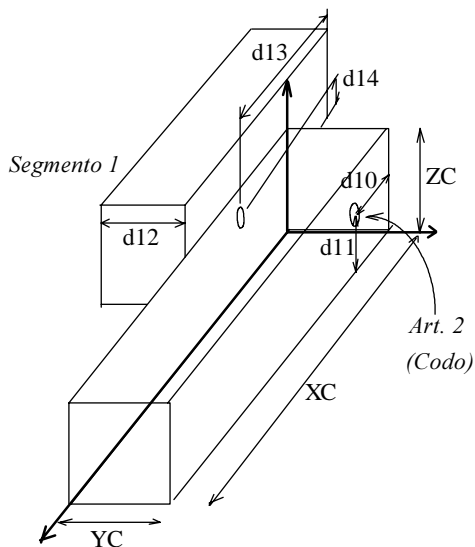
Segmento 0 (HOMBRO)



Segmento 1 (BRAZO)



Segmento 2 (ANTEBRAZO)



se usa es una simplificación de éste, en el que cada segmento se representa por un eje correspondiente a la articulación que lo mueve y una línea que representa el segmento en sí.

En la figura anterior se pueden ver los prismas que representan a cada segmento. Las medidas especificadas junto a los mismos se hallan en el fichero MEDIDAS.DEF, que es incluido en los programas que lo requieran previamente a la compilación.

El cuarto diagrama muestra los ángulos necesarios para pasar de un sistema de referencias a otro (cada segmento tiene asociado un sistema de referencias: el segmento 0 el "A", el segmento 1 el "B" y el segmento 2 el "C"), además del ángulo θ_p que hay que rotar el sistema R para alinear su eje X con el eje X del sistema A.

Las transformaciones homogéneas que pasan de un sistema de referencias a otro se dan a continuación, siguiendo la notación de Craig.

Paso del sistema A al R:

$${}^R_A T = \begin{vmatrix} c(t_p-t_1) & s(t_p-t_1) & 0 & -d_2c(t_p-t_1)-d_3s(t_p-t_1)+R.c_p \\ -s(t_p-t_1) & c(t_p-t_1) & 0 & d_2s(t_p-t_1)-d_3c(t_p-t_1)-R.c_p \\ 0 & 1 & 1 & -d_1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Paso del sistema B al A:

$${}^A_B T = \begin{vmatrix} c_2 & 0 & s_2 & -d_7c_2-d_8s_2+d_5 \\ 0 & 1 & 0 & -d_6+d_4 \\ -s_2 & 0 & c_2 & d_7s_2-d_8c_2+d_9 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

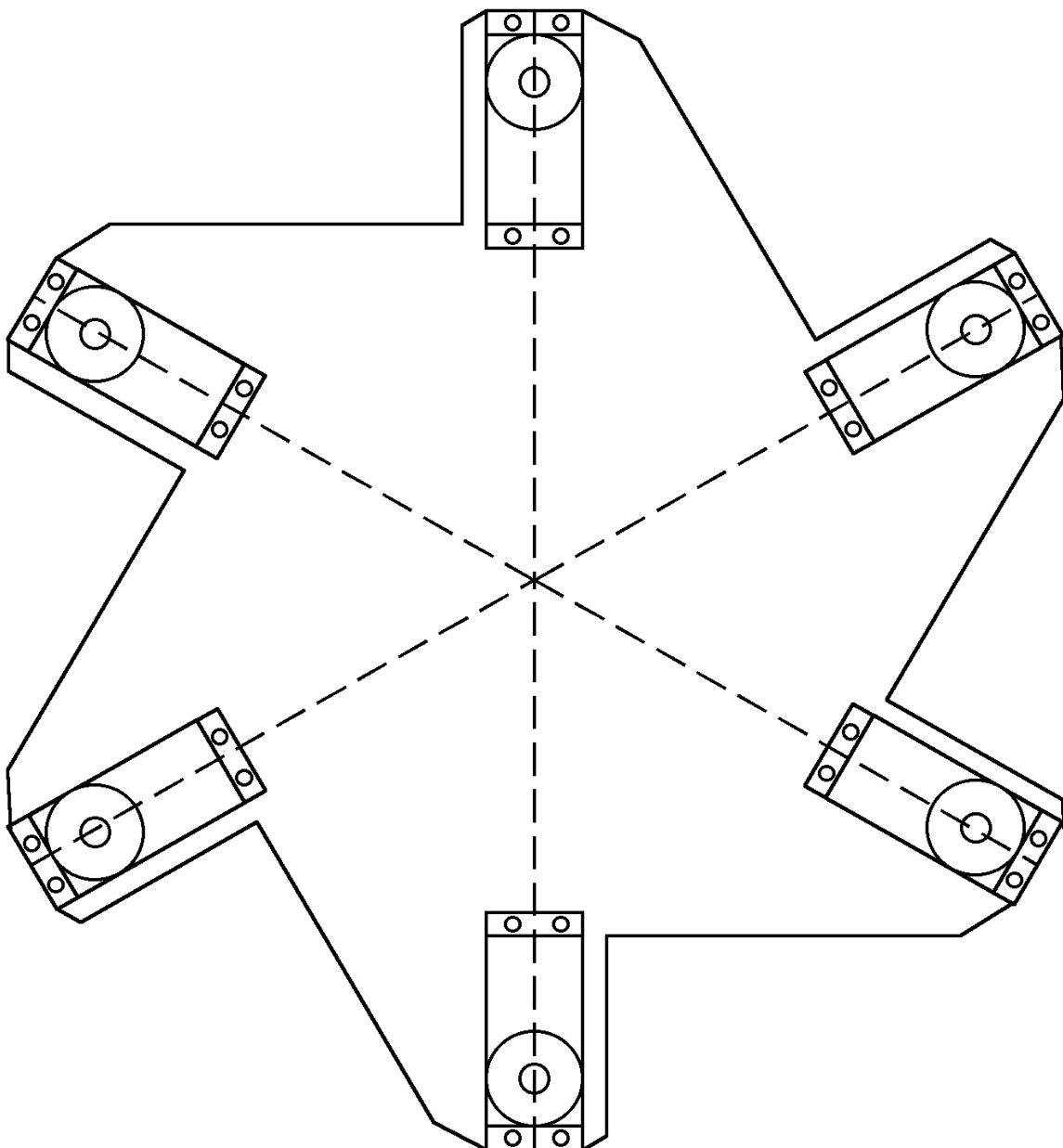
Paso del sistema C al B:

$${}^B_C T = \begin{vmatrix} c_3 & 0 & s_3 & -d_{10}c_3 - d_{11}s_3 + d_{13} \\ 0 & 1 & 0 & d_{12} \\ -s_3 & 0 & c_3 & s_3d_{10} - c_3d_{11} + d_{14} \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Las letras "s" y "c" indican senos y cosenos de los ángulos Theta correspondientes a sus subíndices. "t" equivale a "Theta" y "R" es el radio del cuerpo de RHEX.

Estructura mecánica de RHEX (y II): el cuerpo

En el esquema inferior se muestra el plano de la planta del cuerpo de RHEX. Éste se hace de una sólo pieza sobre una plancha de aluminio de 1 a 2 mm de espesor, recortando las ventanas apropiadas para insertar los servos de las clavículas.



APÉNDICE E

Descripción del i8253

El integrado i8253 tiene las siguientes características generales:

- Tres contadores decrementales independientes de 16 bits.
- Entrada de reloj de frecuencia comprendida entre 0 y 2'6 MHz.
- Varios modos de programación de los contadores.
- Cuentas binarias o en BCD.
- Alimentación de 5 v.
- Señales compatibles TTL.

A continuación se ofrece un resumen de la hoja de datos del integrado, en el que se destacan los puntos más interesantes.

Funciones

El i8253 es un elemento multi-temporizador de propósito general que puede ser tratado como una "caja negra" desde el software del sistema.

Algunos de los usos más típicos del i8253 son los siguientes:

- Generador de ondas de frecuencia programable.
- Contador de eventos.
- Divisor de frecuencias.
- Reloj de tiempo real.

- Generador de pulsos.
- Controlador complejo de motores.

Patillas

Bus de datos: son 8 líneas bidireccionales. Pueden estar en tres estados: entrada (escritura o programación del chip), salida (latcheada) y en alta impedancia (chip deshabilitado).

Lectura: Un nivel bajo en esta patilla hará aparecer en el bus el dato contenido en uno de los buffers de uno de los contadores, previamente seleccionado. No se permite leer el registro de control.

Escritura: Un nivel bajo en esta patilla hará que el dato del bus sea escrito en uno de los registros internos (3 buffers para los contadores, cada uno de 16 bits, y el registro de control, de 8 bits).

A0,A1 (nº de registro): Especifican (tanto en lectura como escritura) el registro al que se accederá:

A0	A1	Registro
0	0	Buffer contador 0
0	1	Buffer contador 1
1	0	Buffer contador 2
1	1	Registro de control

Chip select: Mientras esta patilla permanezca a nivel bajo, las operaciones de E/S con el i8253 estarán bloqueadas (bus de datos en alta impedancia).

Componentes

Registro de control: Almacena la configuración del chip: define el modo de operación de cada contador, el modo de lectura del mismo y si éste es binario o BCD. Internamente contiene 18 bits, pero éstos sólo se pueden escribir mediante un dato de 8 bits. Este dato tendrá el siguiente formato:

SC1	SC0	RL1	RL0	M2	M1	M0	BCD
7	6	5	4	3	2	1	0

SC1 y **SC0** especifican el nº de contador al que se refiere la información restante:

SC1	SC0	nº cont.
0	0	0
0	1	1
1	0	2
1	1	no usado

RL1 y **RL0** especifican el modo de lectura/escritura de los buffers de los contadores:

RL1	RL0	Modo de lectura/escritura
0	0	La palabra de cuenta será latched para que las operaciones de lectura o escritura no influyan en el funcionamiento del contador.
0	1	Sólo se enviará o recibirá el byte menos significativo de la palabra de cuenta.
1	0	Idem para el byte más significativo.

1	1	Primero deberá estar en el bus el byte menos significativo de la palabra de cuenta, y en la siguiente operación, el más significativo.
---	---	--

M2, M1 y M0 especifican el modo del contador (ver siguiente apartado). Cualquier valor binario de estos 3 bits superior a 5 (101) no será válido.

BCD indica si está a 0 que la palabra de cuenta estará en binario (de 1 a 65536 (=0)).

En caso de que contenga 1, la palabra de cuenta estará en BCD sin compactar (4 dígitos, uno por byte. Esto da un rango de cuenta entre 1 y 10000 (=0)).

Modos de programación de los contadores

Modo 0 (*Interrupción al final de la cuenta*): Al programar un contador en este modo o al recargar su palabra de cuenta, la salida del mismo será baja y comenzará a contar (decrementará en 1 su palabra de cuenta a cada pulso del reloj de entrada, que está en la patilla CLK del contador). Al llegar al final de la cuenta la salida se cambiará a alta y permanecerá así hasta una nueva reprogramación o recarga.

Modo 1 (*Pulso único programable*): Igual al anterior, excepto que el evento que provoca el inicio de la cuenta es un flanco de subida en la señal *gate* del contador. Así, éste puede redispararse con otro flanco similar, sin necesidad de recargar la cuenta.

Modo 2 (*Divisor de frecuencia*): Generará a la salida una onda de frecuencia divisora de la frecuencia de la onda de entrada. Al comienzo de cada período, dará señal baja durante tantos pulsos del reloj de entrada como indique la palabra de cuenta, y luego dará salida alta

durante el mismo tiempo. Mandando un flanco de subida por la señal *gate* del contador se obligará al contador a recomenzar la cuenta.

Modo 3 (*Generador de onda cuadrada*): Igual al anterior, salvo que la onda de salida permanecerá en estado bajo sólo la mitad de la palabra de cuenta, y el resto en estado alto.

Modo 4 (*Pulso disparado por software*): Al cargar la palabra de cuenta o al reprogramar el contador, éste comenzará dando salida alta y permanecerá así mientras decrementa la palabra de cuenta. Cuando ésta llegue a 0, dará salida baja durante un pulso del reloj de entrada y luego volverá a dar señal alta hasta una nueva reprogramación o recarga de la palabra de cuenta.

Modo 5 (*Pulso disparado por hardware*): Igual al anterior, pero el evento que dispara la cuenta es un flanco de subida en la señal *gate*. Por ello, es redisparable.

APÉNDICE F

Electrónica de control

En este apéndice se dan los esquemas eléctricos de las tarjetas diseñadas definitivamente para RHEX (hubo versiones previas como se ha comentado anteriormente en la memoria). La principal es la tarjeta *controladora*, responsable de generar y mantener las consignas a los servos. Para conectar ésta a la PC-LabCard, se diseñó la tarjeta *mapeadora*. La misión de ésta es mapear los pines de los conectores de la LabCard a los pines del conector de la controladora, además de regenerar el reloj de la misma a través de un integrado 74LS00. Ambos circuitos se diseñaron con el Orcad/SDT y Orcad/PCB.

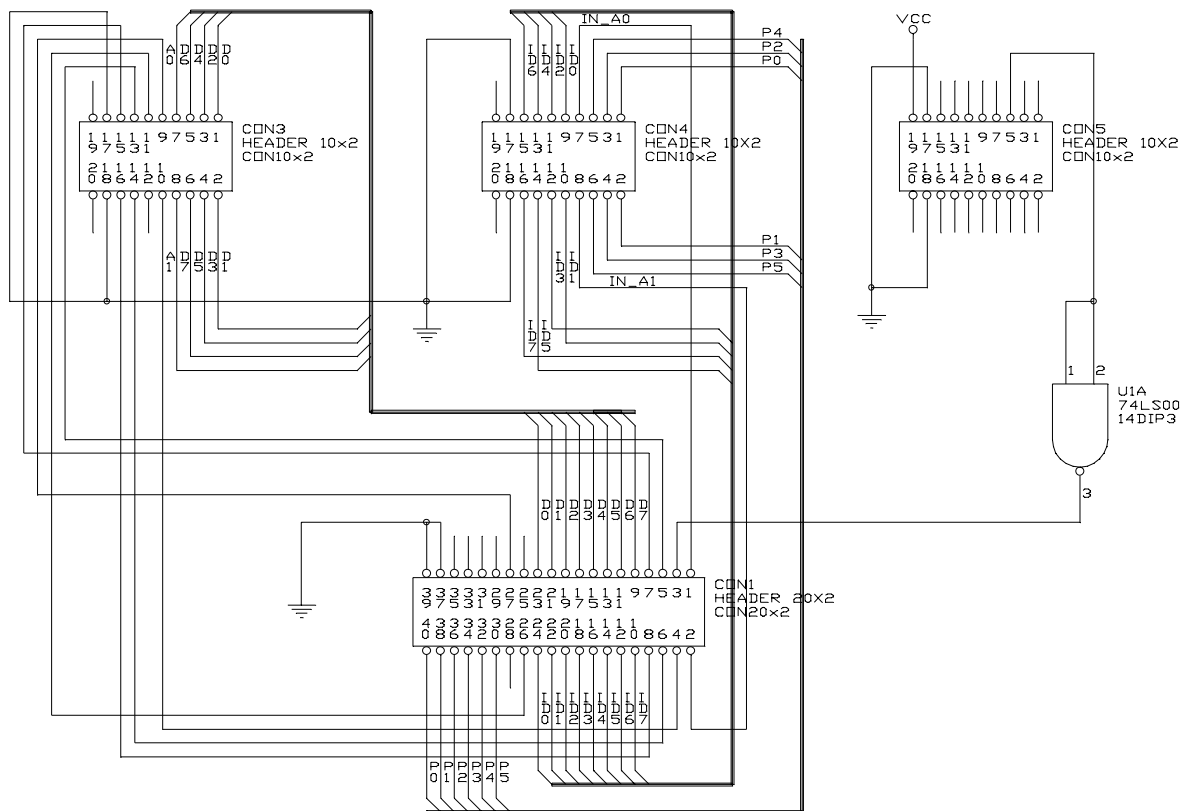


Fig. F.1.: Esquema eléctrico de la tarjeta *mapeadora*.

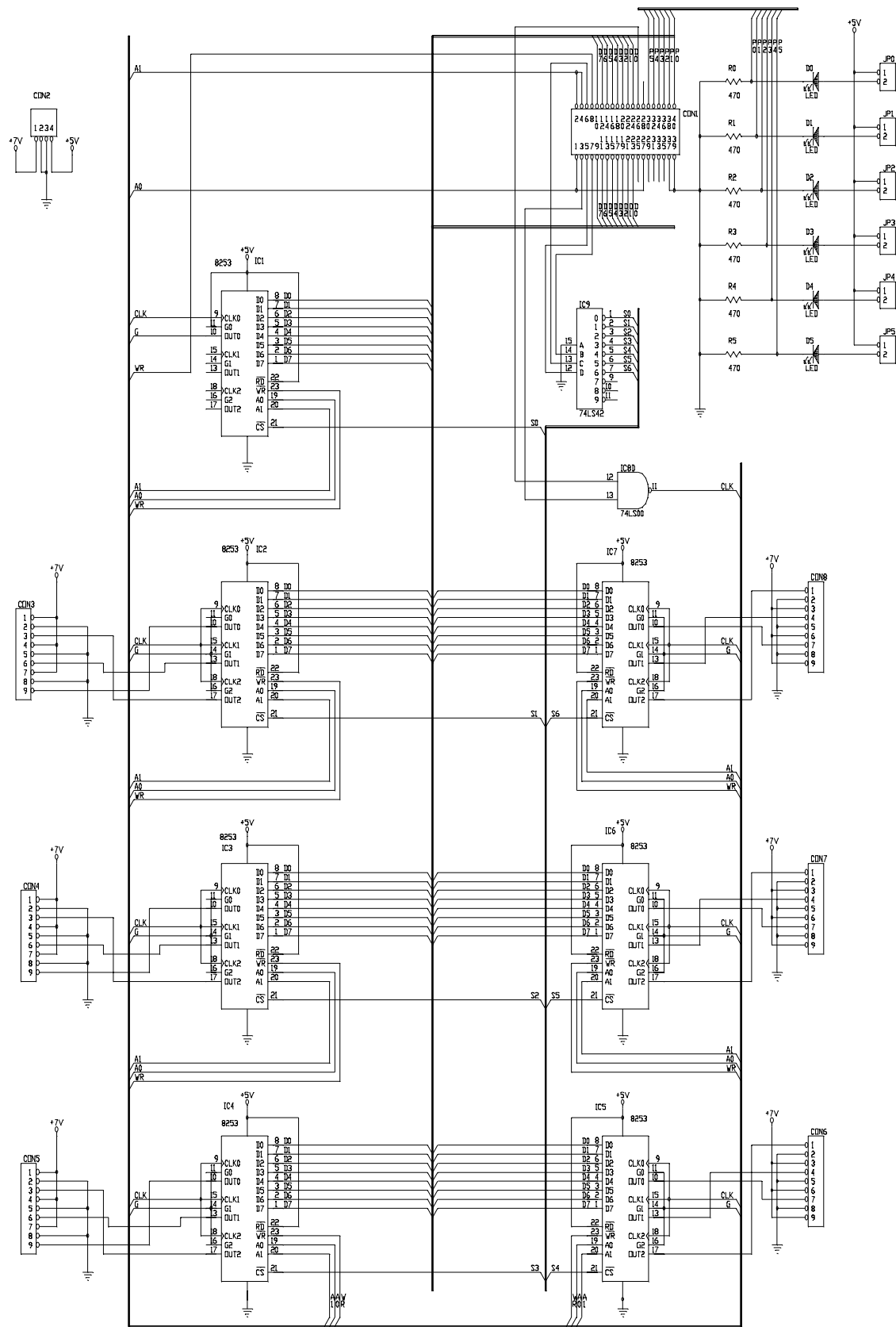


Fig. F.2: Esquema eléctrico de la tarjeta controladora.

El conector **CN1** de la controladora tiene el siguiente patillaje:

D0..D7: bus de datos. Está duplicado, existiendo un bus en los pines {24, 22, 20, 18, 16, 14, 12, 10} y otro en {23, 21, 19, 17, 15, 13, 11, 9}. El primero de ellos se utiliza como *bucle de verificación de señales* (ver 5.1.2) y el segundo para el envío de los datos desde el ordenador.

S0..S2: bus de selección de integrado. El valor binario que contenga es decodificado para seleccionar (enviar una señal CS baja) el i8253 deseado (valores válidos desde 0 a 6, 0 es el integrado generador de la señal cuadrada de 50 Hz y 1..6 son los integrados asociados a cada una de las patas).

A0..A1: bus de selección de registro. Especifica un nº de registro dentro del i8253 escogido con el bus de selección de integrado. Está duplicado como el bus de datos. Las patillas {1 -> A0, 2 -> A1} pertenecen al bucle de verificación de señales, mientras que las patillas {27 -> A0, 4 -> A1} son el propio bus de selección de registro.

WR: señal de escritura. Debe generarse un pulso bajo por esta patilla con una duración mínima de unos 200 ns para escribir el contenido del bus de datos en el registro seleccionado con **A0..A1** del integrado seleccionado con **S0..S2**.

CLK: reloj de 2MHz generado externamente.

E_CLK: Un cero en esta patilla inhabilita el reloj de 2MHz de la tarjeta, con lo que los servos dejan de recibir ondas PWM y no ejercen fuerza.

P0..P5: bus de pulsadores.

GND: las patillas 37 y 39 deben estar conectadas a la tierra de la tarjeta.

El conector **CN2** da la alimentación a la tarjeta y a los servos del robot. El pin marcado con **+7v** proporciona el voltaje de los servos (y un amperaje recomendado mínimo de 5 A) y el marcado con **+5v** proporciona alimentación a los integrados (amperaje recomendado de 1 A).

Los conectores **CN3** a **CN8** son para los servos de cada pata (cada uno para los tres servos). Los conectores hembra de los servos deben seguir el esquema siguiente:

+7v	GND	Señal PWM
-----	-----	-----------

Los jumpers **JP0** a **JP5** son para conectar los pulsadores (un pin a cada extremo del pulsador), independientemente de su polaridad.

Descripción de la tarjeta de E/S I/O 8255

Esta tarjeta comercial de E/S tiene las siguientes características generales:

- Un integrado i8253 cuyos tres contadores están disponibles para su programación.
- Dos integrados i8255 que proporcionan entre ambos 48 bits de E/S, divididos en grupos (A y B) y registros (A,B y C) para cada integrado.
- Una señal de reloj teórica de 2'386333 MHz, aunque en realidad no es estable en ordenadores PC de gama alta (386 y superiores).
- Display interno compuesto de 16 leds indicadores del estado de algunas líneas de E/S digitales.

Debido sin embargo a los problemas que se enumeran a continuación, ha sido preferible utilizar la tarjeta PCLabCard PCL-812PG:

- Reloj dependiente del resto del hardware del ordenador.
- Mapeado inadecuado de los pines de los conectores.
- No dispone de conversores A/D, aunque éstos no son indispensables.
- Distinto comportamiento del reloj de la tarjeta dependiendo del modelo de ordenador en el que se instala.

APÉNDICE H

Descripción de la tarjeta de E/S PC-LabCard

A continuación se ofrece una descripción general de las funciones de esta tarjeta utilizadas por el software de comunicación con RHEX.

Características principales

- 16 entradas analógicas con conversores A/D de resolución 12 bits.
- 2 salidas analógicas con conversores D/A de resolución 12 bits.
- Rango de las señales analógicas entre -5v y +5v.
- Conversores A/D por aproximaciones sucesivas y D/A por multiplicador monolítico.

Velocidad de conversión A/D máxima de 30 KHz. Tiempo de establecimiento del dato del conversor D/A de 30 μ seg.

- Transferencia de datos analógicos por espera activa, interrupción o DMA.
- 16 bits de entrada digital compatible TTL.
- 16 bits de salida digital compatible TTL.
- Un dispositivo i8253 con 1 contador libre para el usuario.
- Reloj interno accesible de 2 MHz.

Captura y envío de datos digitales

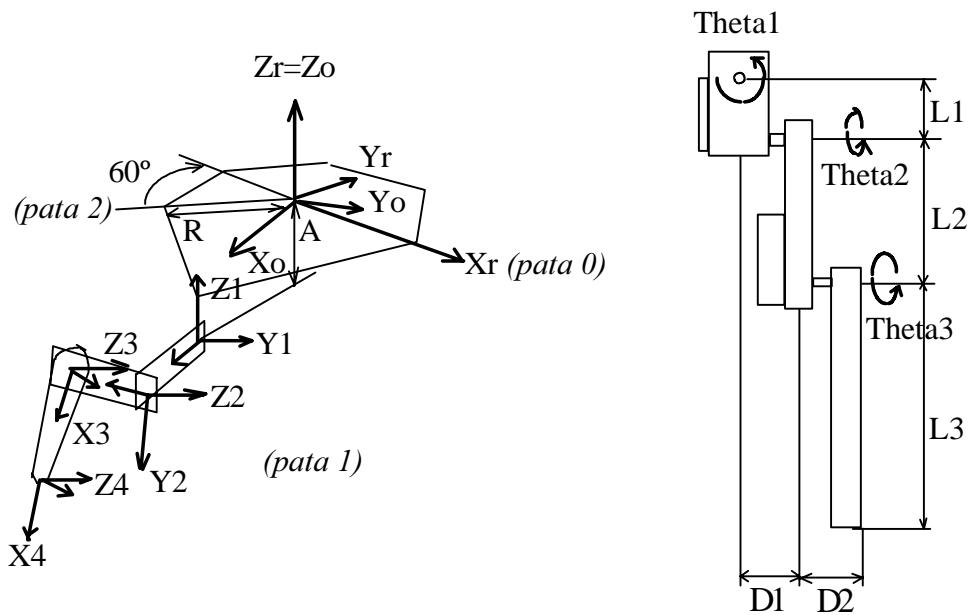
Los 16 bits de salida digital se envían a los puertos BASE+13 y BASE+14, donde BASE es por defecto 0x220. El primero de ellos recibirá el byte bajo (bits de salida 0 al 7) y el segundo el alto (bits de salida 8 al 15).

Los 16 bits de entrada digital se leen de los puertos BASE+8 y BASE+9, análogamente al caso anterior.

APÉNDICE I

Desarrollo de las fórmulas cinemáticas para una pata

La notación, sistemas de referencia, numeraje de las patas y ángulos de las articulaciones, se especifican gráficamente en la figura.



A la izquierda, se ven los sistemas de referencias asociados a cada articulación de una pata. A la derecha se ven los ángulos de las articulaciones (en sus sentidos positivos) y las dimensiones de cada una.

Siguiendo la forma de describir articulaciones dada por Craig, obtenemos la siguiente tabla, que modela las de una pata de RHEX:

	α_{i-1}	a_{i-1}	d_i	θ_i
art. 1	0	R	-A	t_1
art. 2	-90°	L_1	D_1	t_2
art. 3	0	L_2	D_2	t_3
art. 4	0	L_3	0	0

Mediante una observación cuidadosa de los sistemas de referencias, podemos obtener las siguientes matrices de transformación homogéneas que pasan de uno a otro:

Paso del sistema 0 al R:

$${}^R_0T = \begin{vmatrix} c\alpha & -s\alpha & 0 & 0 \\ s\alpha & c\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Donde $\alpha = (\text{n}^\circ \text{ de pata, de 0 a 5}) * (-60^\circ)$ grados.

Paso del sistema 1 al sistema 0:

$${}^0_1T = \begin{vmatrix} ct_1 & -st_1 & 0 & R \\ st_1 & ct_1 & 0 & 0 \\ 0 & 0 & 1 & -A \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Paso del sistema 2 al sistema 1:

$${}^1_2T = \begin{vmatrix} ct_2 & -st_2 & 0 & L_1 \\ 0 & 0 & 1 & D_1 \\ -st_2 & -ct_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Paso del sistema 3 al sistema 2:

$${}^2_3\mathbf{T} = \begin{vmatrix} ct_3 & -st_3 & 0 & L_2 \\ st_3 & ct_3 & 0 & 0 \\ 0 & 0 & 1 & D_2 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Paso del sistema 4 al sistema 3:

$${}^3_4\mathbf{T} = \begin{vmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Componiendo todas estas matrices, se obtiene la matriz de cinemática directa, que pasa del sistema 4 al sistema R:

$${}^R_4\mathbf{T} = \begin{vmatrix} c\alpha \cdot c_1 c_{23} - s\alpha \cdot s_1 c_{23} & -c\alpha \cdot c_1 s_{23} + s\alpha \cdot s_1 s_{23} & -c\alpha \cdot s_1 - s\alpha \cdot c_1 & c\alpha \cdot m_1 - s\alpha \cdot m_2 \\ s\alpha \cdot c_1 c_{23} + c\alpha \cdot s_1 c_{23} & s\alpha \cdot c_1 s_{23} - c\alpha \cdot s_1 s_{23} & s\alpha \cdot s_1 + c\alpha \cdot c_1 & s\alpha \cdot m_1 + c\alpha \cdot m_2 \\ -s_{23} & -c_{23} & 0 & -s_{23} L_3 - s_2 L_2 - A \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Donde:

$$m_1 = c_1(c_{23}L_3 + c_2L_2 + L_1) - s_1(D_2 + D_1) + R$$

$$m_2 = s_1(c_{23}L_3 + c_2L_2 + L_1) + c_1(D_2 + D_1)$$

Partiendo de esta matriz, se pueden deducir las ecuaciones de la cinemática inversa de una pata.

Para simplificar cálculos, se puede calcular la cinemática inversa hasta el sistema 0 y no el R. La matriz que pasa del sistema 4 al 0 es:

$${}^0_4\mathbf{T} = \begin{vmatrix} c_1c_{23} & -c_1s_{23} & -s_1 & m_1 \\ s_1c_{23} & -s_1s_{23} & c_1 & m_2 \\ -s_{23} & -c_{23} & 0 & -s_{23}L_3-s_2L_2-A \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

La cinemática inversa consistirá en, a partir de la posición del extremo de la pata, calcular los valores que deben mandarse a las tres articulaciones de la misma para que ésta se sitúe en esa posición.

Cálculo del valor de la clavícula

De la última matriz se deduce que, dado un vector con las coordenadas del extremo de la pata en el sistema 0 (si se tienen en el sistema R, pueden pasarse al 0 con ${}^0_R\mathbf{T}$, que es ${}^R_0\mathbf{T}^{-1}$), estas coordenadas deben cumplir las igualdades:

$$x = c_1a_1 - s_1a_2 + R$$

$$y = s_1a_1 + c_1a_2 \quad (\text{I})$$

$$z = a_3 - A$$

donde se conocen x, y, z, A y R , y además, $a_2 = D_1 + D_2$ y $a_3 = z + A$.

Se puede hacer la simplificación: $x' = x - R$, que junto con (I) da el sistema:

$$x' = c_1 a_1 - s_1 a_2 \quad (\text{II})$$

$$y = s_1 a_1 + c_1 a_2 \quad (\text{III})$$

que, elevando al cuadrado y sumando, lleva a:

$$x'^2 + y^2 = a_1^2 + a_2^2$$

de lo que se deduce:

$$a_1 = \sqrt{x'^2 + y^2 - a_2^2} \quad \text{que permite calcular } a_1.$$

Si se hace:
$$n = \sqrt{a_1^2 + a_2^2}$$

se puede suponer que: $a_1/n = cb$

$$a_2/n = sb$$

y de ahí:

$$b = \text{Atan2}(a_2/n, a_1/n)$$

ya que se han normalizado a_1 y a_2 . Una vez hecho esto, se pueden dividir (II) y (III) por n , con lo que queda:

$$x'' = x'/n = c_1 a_1/n - s_1 a_2/n = c_1 cb - s_1 sb = c_{1b}$$

$$y'' = y/n = s_1 a_1/n + c_1 a_2/n = s_1 cb + c_1 sb = s_{1b}$$

Según la notación de Craig, $c_{1b} = \cos(t_1+b)$ y $s_{1b} = \sin(t_1+b)$. La solución a este sistema es:

$$t_1 + b = \text{Atan2}(y'', x''), \text{ y de ahí}$$

$$t_1 = \text{Atan2}(y'', x'') - \text{Atan2}(a_2/n, a_1/n) \quad \text{(IV)}$$

que da el valor que debe tener la primera articulación de la pata (clavícula) para situar el extremo de la misma en la posición (x, y, z) .

Cálculo de los valores del hombro y codo

Se partirá de las dos ecuaciones:

$$a_3 = -s_{23}L_3 - s_2L_2$$

$$a_1 = c_{23}L_3 + c_2L_2 + L_1$$

donde las incógnitas son t_2 y t_3 , ya que los valores de a_1 y a_3 han sido calculados anteriormente y las L_i son constantes.

Se puede transformar este sistema en el siguiente:

$$a_1' = a_1 - L_1 = c_{23}L_3 + c_2L_2$$

$$a_3' = -a_3 = s_{23}L_3 + s_2L_2$$

Expandiendo este sistema, queda:

$$a_1' = c_2 c_3 L_3 - s_2 s_3 L_3 + c_2 L_2 = (c_2 c_3 - s_2 s_3) L_3 + c_2 L_2 \quad (\text{V})$$

$$a_3' = s_2 c_3 L_3 - s_3 c_2 L_3 + s_2 L_2 = (s_2 c_3 + s_3 c_2) L_3 + s_2 L_2 \quad (\text{VI})$$

Elevando al cuadrado y sumando ambas ecuaciones se obtiene:

$$a_1'^2 + a_3'^2 = L_3^2 + L_2^2 + 2c_3 L_3 L_2$$

de la que se puede despejar t_3 :

$$t_3 = \cos^{-1} [(a_1'^2 + a_3'^2 - L_3^2 - L_2^2) / (2L_3 L_2)] \quad (\text{VII})$$

Las ecuaciones (V) y (VI) pueden reestructurarse:

$$a_1' = (L_3 c_3 + L_2) c_2 + c_2 L_2 - s_2 s_3 L_3$$

$$a_3' = (L_3 c_3 + L_2) s_2 + s_3 c_2 L_3$$

Para solucionar este sistema (la incógnita es ahora t_2) se pueden hacer las siguientes equivalencias:

$$K_1 = L_3 c_3 + L_2$$

$$K_2 = s_3 L_3 = \sqrt{1 - c_3^2} L_3$$

con lo que el sistema quedaría:

$$a_1' = c_2 K_1 - s_2 K_2$$

$$a_3' = s_2 K_1 + c_2 K_2$$

y si se hace: $n = \sqrt{K_1^2 + K_2^2}$

la solución es:

$$t_2 = \text{Atan2}(a_3'/n, a_1'/n) - \text{Atan2}(K_2/n, K_1/n) \quad \text{(VIII)}$$

Así, mediante las fórmulas (IV), (VII) y (VIII) se pueden obtener los valores de los ángulos de las articulaciones de una pata para situar su extremo en unas coordenadas determinadas.

BIBLIOGRAFÍA

John J. Craig: **"Introduction to Robotics, Mechanics & Control"**

Addison-Wesley, 1989

K. S. Fu, R.C. González, C. S. G. Lee: **"Robótica, control, detección e inteligencia"**

McGraw Hill, 1988

Sandler: **"Robotics: designing the mechanisms for automated machinery"**

Prentice Hall, 1991

S. Bennet: **"Real-Time computer control: an introduction"**

Prentice Hall, 1988

A. Burus, A. Welling: **"Real-Time systems and their programming languages"**

Addison-Wesley, 1990

D. Wettergreen y otros: **"Planning strategies for the Ambler walking robot"**

Procs. of the IEEE International Conference on Systems Engineering, 1990

P.V. Nagy: **"Attitude and altitude control for a novel 6-legged robot"**

Part of the Doctoral Qualifying Examination.

D.N. Karastoyanov: **"Control of a quadruped walking robot"**

INCOM'89