

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Departamento de Computación

**Diseño y control de Kokone, un
pequeño robot humanoide**

Tesis que presenta:

Víctor Enrique González Hernández

Para obtener el grado de:

**Maestro en Ciencias
en Computación**

Director de la Tesis:
Dr. José Gabriel Ramírez Torres

© Derechos reservados por
V́ctor Enrique Gonźlez Herńndez
2009

Esta investigación fue financiada mediante el proyecto No. 231722 del Consejo Nacional de Ciencia y Tecnología, Conacyt.

This research was funded by project number 231722 from Consejo Nacional de Ciencia y Tecnología, Conacyt.

La tesis presentada por Víctor Enrique González Hernández fue aprobada por:

Dr. Luis Enrique Sucar Succar

Dr. Gregorio Toscano Pulido

Dr. José Gabriel Ramírez Torres, Director

Cd. Victoria Tamaulipas, México, 26 de Noviembre de 2009

A mi familia, soporte de mi vida

Agradecimientos

- Gracias Dios, en ti, todo es posible.
- Gracias a mi padres Víctor y Marlene, y a mis hermanos Oscar y Mary, por creer en mi y su gran apoyo incondicional en mi vida.
- Gracias a mi Dey, por estar siempre conmigo y brindarme momentos muy felices.
- Quiero agradecer a mi asesor, Dr. José Gabriel Ramírez Torres, por darme la oportunidad de realizar este trabajo y permitirme hacer lo que me apasiona.
- Quiero agradecer a los profesores del Departamento de Computación y del Laboratorio de Tecnologías de Información, por sus grandes enseñanzas y marcar el camino de la investigación.
- Gracias a mis amigos del Departamento de Computación y del Laboratorio de Tecnologías de Información, por su compañía y momentos alegres, los estimo mucho.
- Gracias al Consejo Nacional de Ciencia y Tecnología, Conacyt, por la beca de manutención durante mis estudios realizados.

Índice General

Índice General	I
Índice de Figuras	V
Índice de Tablas	VII
Índice de Algoritmos	IX
Publicaciones	XI
Resumen	XIII
Abstract	XV
1. Introducción	1
1.1. Antecedentes históricos	1
1.2. Desarrollo de los robots	4
1.3. Robots humanoides	7
1.4. Descripción del proyecto tesis	9
1.5. Organización de la tesis	10
2. Estado del arte en Robots humanoides	13
2.1. Introducción	13
2.2. Robots humanoides de tamaño real	14
2.2.1. Robot ASIMO	15
2.2.2. Robot KHR-3: HUBO	15
2.2.3. Robot REEM-B	16
2.2.4. Robot HRP-3 Promet Mk-II	17
2.3. Robots humanoides medianos	18
2.3.1. Robot SDR-4X: Qrio	18
2.3.2. Robot HOAP-3	19
2.4. Robots humanoides pequeños	19
2.4.1. Robot Robonova	20
2.4.2. Robot KHR-2	20
2.4.3. Robot HR-2	21
2.4.4. Robot TWNHR-IV	22
2.4.5. Robot Tao-PIE-PIE	22
2.4.6. Robot ZORC	23
2.5. Robots humanoides en México	24

3. Diseño de Kokone	25
3.1. Introducción	25
3.2. Esquema general del sistema	26
3.3. Motorización	27
3.4. Electrónica y potencia	28
3.5. Estructura mecánica	30
3.5.1. Grados de libertad	31
3.5.2. Diseño Asistido por Computadora, (Computer Aided Desing, CAD)	32
3.5.2.1. Diseño de las piernas	35
3.5.2.2. Diseño de los brazos	38
3.5.2.3. Diseño de la cabeza	39
3.5.2.4. Diseño del cuerpo	39
3.5.3. Construcción de <i>Kokone</i>	40
4. Cinemática de Kokone	43
4.1. Introducción	43
4.2. Representación de la posición y orientación	44
4.2.1. Posición y desplazamiento	45
4.2.2. Orientación y rotación	46
4.3. Transformaciones homogéneas	49
4.4. Representación geométrica	52
4.4.1. Representación geométrica de <i>Kokone</i>	55
4.5. Cinemática directa	59
4.5.1. Cinemática de <i>Kokone</i>	60
4.6. Cinemática inversa	64
4.6.1. Planteamiento de la cinemática inversa	65
4.6.2. AG para resolver la cinemática inversa de <i>Kokone</i>	68
5. Primeros pasos de Kokone	75
5.1. Introducción	75
5.2. Control de <i>Kokone</i>	76
5.3. Caminado bípedo	76
5.4. Aplicación de caminado	79
6. Conclusiones	83
6.1. <i>Kokone</i> : plataforma robótica de investigación	84
6.2. Futuras capacidades y aplicaciones	85
A. Biblioteca de control	87
A.1. Funciones para resolver la cinemática	87
A.2. Funciones para la configuración del robot	88
A.3. Aplicación	89

Índice de Figuras

1.1. Autómatas famosos	3
1.2. Unimate, primer robot industrial	5
1.3. Dos robots de servicio en la actualidad	8
2.1. Robots ASIMO y HUBO	16
2.2. Robot REEM-B y HRP-3	17
2.3. Robot Qrio y HOAP-3	19
2.4. Robonova y Kondo KHR-2	21
2.5. HR-2 y TWNHR-IV	22
2.6. Tao-PIE-PIE y ZORC	23
3.1. Esquema general del sistema	27
3.2. Servomotor BMS-380MAX	28
3.3. Tarjeta de control de servos SSC-32 Lynxmotion	30
3.4. GDL del robot	31
3.5. Diseño en CAD de <i>Kokone</i>	34
3.6. Servomotores virtual y real	34
3.7. Ensamble E1	35
3.8. Diseño de la pierna	36
3.9. Ensamble E2	36
3.10. sub-ensambles E2.1 y E2.2	37
3.11. Ensamble E3	37
3.12. Ensamble E4	38
3.13. Brazo derecho	39
3.14. Ensamble E5 y E6	40
3.15. Diseño de la cabeza	41
3.16. Cuerpo	41
3.17. Realización de <i>Kokone</i>	42
4.1. Ejemplos de cadena abierta y cerrada	44
4.2. Sistema de referencia y coordenadas Cartesianas	46
4.3. Rotaciones en los ejes OX , OY y OZ	48
4.4. Figura del ejemplo 1 y 2	53
4.5. Sistemas de referencia asignados a la pierna derecha de <i>Kokone</i>	57
4.6. Representación de las traslaciones	61
4.7. Representación de las traslaciones	62
4.8. Figura del ejemplo 3	64
4.9. Figura del ejemplo 4	74

5.1. Jerarquía de control	76
5.2. CM, PCM, PS, y ZMP	79
5.3. Inicialización	80
5.4. Ciclo de caminado	80
5.5. Metodología de control	81
5.6. Ciclo de caminado implementado en <i>Kokone</i>	82

Índice de Tablas

1.1. Clasificación de los robots	7
3.1. Características del servomotor BMS-380MAX	28
3.2. Configuración del puerto serie	29
4.1. Parámetros DH de ambas piernas	56
4.2. Parámetros DH del ejemplo 3	63
4.3. Rango de las variables	69
4.4. Resultados del AG sin reinicio	71
4.5. Resultados del AG con reinicio	72
4.6. Tabla del ejemplo 4	73

Índice de Algoritmos

1.	Enfoque general de un Algoritmo Genético	67
2.	Selección proporcional por ruleta	69
3.	Mutación Gaussiana	70

Publicaciones

V. González Hernández and G. Ramírez Torres. *Design of Kokone, a small humanoid robot*, in 6th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2009), an IEEE Press, Toluca City Mexico state, Mexico, November 2009.

Diseño y control de *Kokone*, un pequeño robot humanoide

por

Víctor Enrique González Hernández

Maestro en Ciencias del Departamento de Computación

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2009

Dr. José Gabriel Ramírez Torres, Director

Los avances en ciencia y tecnología han permitido el desarrollo de la robótica. En los últimos años, la construcción de robots se ha incrementado gracias a las necesidades de la industria, la milicia y el entretenimiento. Hoy en día, la academia y las empresas se esfuerzan por ofrecer robots que asistan en hogares, oficinas y talleres pequeños, nos referimos a los robots de servicio. Para su fácil adaptación al entorno productivo, es imprescindible que los robots cuenten con morfología humana, *i.e.*, robots humanoides. Su integración completa a la sociedad depende de la capacidad humana que posean: comportamiento, apariencia física y habilidad motriz. Las instituciones realizan investigación en prototipos robóticos, sin embargo, la mayoría de estos robots son de muy alto costo. Una alternativa, es la construcción de robots humanoides pequeños, diseñados de acuerdo a las necesidades de la experimentación.

En este proyecto de tesis, proponemos el diseño y control de *Kokone* (niños en náhuatl), un pequeño robot humanoide. *Kokone* tiene como objetivos asistir en el desarrollo de estrategias de control y fomentar el uso de métodos computacionales en la solución de problemas de robótica humanoide. El esquema de la plataforma robótica facilita las pruebas de laboratorio, pues se aprovecha la flexibilidad del uso de una PC como unidad de control. La distribución de sus 22 grados de libertad, permiten realizar la mayoría de los movimientos del cuerpo humano, y su diseño mecánico, realizado en Solid Edge, requiere un mínimo de piezas y un fácil ensamblaje. Con base en el diseño final, se resolvió la cinemática directa mediante el método de Denavit-Hartenberg, y se propuso un algoritmo genético para resolver la cinemática inversa. La biblioteca de control realizada (escrita en lenguaje

C) permite a las aplicaciones controlar los movimientos de las piernas especificando la localización de los pies y la cadera. Como primera aplicación, se propuso una estrategia de caminado basado en los conceptos de la caminata bípeda estática. Los desplazamientos obtenidos se verificaron en Matlab, para después ser implementados satisfactoriamente en el robot *Kokone*.

Design and control of Kokone, a small humanoid robot

by

Víctor Enrique González Hernández

Master of Science from the Computer Science Department

Research Center for Advanced Study from the National Polytechnic Institute, 2009

Dr. José Gabriel Ramírez Torres, Advisor

The advances in science and technology have allowed the development of robotics. In recent years, the construction of robots has increased due to the needs of industry, military and entertainment. Nowadays, academia and companies strives to provide robots to assist at homes, offices and small workshops, we refer to the service robots. For an easy adaptation to the production human environment, it is essential that robots have a human morphology, *i.e.*, humanoid robots. Their full integration into society depends on their human capacity: behavior, physical appearance and motor skills. Research centers perform experiments in robotics prototypes, however, most of these robots are expensive. An alternative of this problem are small humanoid robots, designed to the needs of experimentation.

In this document we propose the design and control of *Kokone*, a small humanoid robot. The main objectives of *Kokone* (“kid” in náhuatl, a mexican dialect) are to assist in the development of control strategies and to promote the use of computational methods for solving humanoid robots problems. The robotic platform scheme allows an easy laboratory testing using a PC as a control unit. Most of the human body movements can be done by the 22 degrees of freedom distribution, and mechanical desing, done in Solid Edge, it requires a minimum number of pieces and it is easy to assemble. The forward kinematics problem is solve using the Denavit-Hartenberg method. Furthermore, we propose a specific genetic algorithm in order to solve the inverse kinematics problem. The proposed functions library (written in C) provides methods for an easy control of the leg movements by specifying the locations of the feet and hip. As a first application, we proposed a walking strategy based on the

concepts of static bipedal walking. The displacements obtained are verified in Matlab and then they were successfully implemented on *Kokone* robot.

1

Introducción

La robótica tiene curiosos y particulares antecedentes que valen la pena mencionar. Desde los primeros autómatas hasta los avanzados robots industriales, el hombre ha construido máquinas para realizar distintos procesos de manera automática. Hace varios años, los robots mostraron su capacidad para realizar tareas repetitivas o peligrosas para los humanos y hoy en día, asisten en hogares y oficinas como robots de servicio. Estos ambientes, con toda la complejidad que implica el adaptar máquinas a entornos diseñados para personas, motivaron la construcción de robots humanoides.

1.1 Antecedentes históricos

Desde la revolución industrial los seres humanos empezaron el desarrollo de máquinas complejas para facilitar los trabajos arduos, difíciles y peligrosos. Sus primeros usos fueron en tareas que requerían esfuerzo físico y mucho tiempo para completarse. La industria de la construcción, la manipulación de objetos de gran tamaño y la milicia son algunos ejemplos de estas primeras aplicaciones. Estas máquinas requieren personas con habilidad para operarlas, con una experiencia adquirida a

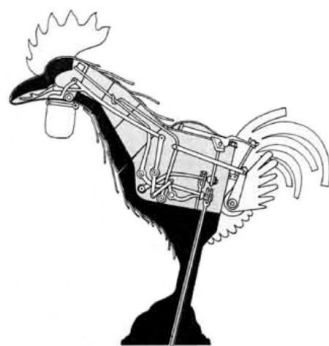
través de los años.

Conforme la demanda de servicios y productos crecía, otras áreas productivas necesitaron de máquinas dentro de los procesos de fabricación para aumentar la rapidez y la precisión, como por ejemplo las industrias automotriz y aeronáutica. Con la llegada del transistor, a finales de la década de los cuarenta, se desarrollaron dispositivos electrónicos capaces de procesar información más rápido que las personas. Estos dispositivos, que ahora conocemos como computadoras, marcaron un parteaguas en la forma de realizar los procesos. Surgen así, las industrias basadas en la ciencia y comienza la era tecnológica moderna. La sustitución del operador de las máquinas por un programa de computadora permitió la automatización parcial o completa de la producción, logrando aumentar la oferta de productos y bajar sus costos; características como rapidez, precisión y eficacia se volvieron imprescindibles en los procesos industriales. Las computadoras trajeron consigo ventajas importantes a la industria, de las cuales podemos mencionar dos de sumo interés: la abstracción del control de las máquinas a través de programas de computadora y la asistencia durante el desarrollo de mejores prototipos, promoviendo avances en el procesamiento de la información, en el desarrollo de dispositivos electrónicos y de artefactos mecánicos. Todos estos avances han llevado naturalmente al surgimiento de nuevas ramas en la investigación tecnológica, como la robótica.

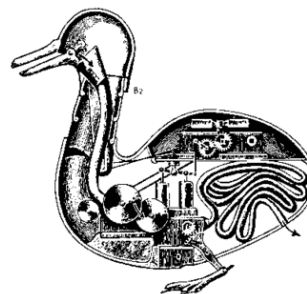
La robótica es la rama de la tecnología que estudia todos los aspectos involucrados en la construcción y operación de los robots. La palabra “robot” tiene un origen antiguo: proviene de las palabras checas “robota” que significa labor o trabajo y de “robotnik” que quiere decir trabajador. Este término fue empleado por el escritor Karel Capek (1890 - 1938) en su obra de teatro “Rossum’s Universal Robot”. La historia de los robots se remonta a las primeras civilizaciones que construyeron máquinas para realizar movimientos de manera autónoma, semejantes a los de un ser vivo. Los griegos las denominaron *automatos*, término del que proviene la palabra autómatas: máquina que imita la figura y movimientos de un ser animado. Ejemplos de antiguos autómatas son los mecanismos de Herón de Alejandría (85 d.C.) que eran movidos por poleas y palancas, y los sistemas dispensadores de agua de la cultura árabe construidos para la realeza (siglos VIII a XV).

De algunos autómatas sólo se tienen vagas referencias como el *Hombre de hierro* de Alejandro

Magno (1204 - 1282) y la *Cabeza parlante* de Roger Bacon (1214 - 1294). Sin duda el autómata más famoso que aún se conserva es el *Gallo de Estrasburgo* (1352) que formaba parte del reloj de la catedral de Estrasburgo y al dar las horas movía las alas y el pico (Figura 1.1.a). En la época del renacimiento también se construyeron autómatas novedosos como el *León mecánico* de Leonardo Da Vinci (1492 - 1519), que enseñaba el escudo de armas del rey, y el *Hombre de palo* construido por Juanelo Turriano en el siglo XVI para el rey Carlos V de España. Otros autómatas avanzados para su época fueron los construidos durante los siglos XVII y XVIII. Jacques Vaucanson (1709 - 1782) construyó diversas marionetas animadas, que van desde un flautista hasta un pato (1738), que era capaz de graznar, comer e incluso evacuar (Figura 1.1.b). Algunos hombres de oficio utilizaban sus conocimientos para la construcción de muñecos autónomos que imitaban algunas actividades de los seres humanos. El relojero suizo Pierre Jaquet Droz (1721 - 1790) y sus hijos fabricaron muñecos capaces de escribir (1770), dibujar (1772) e interpretar algunas piezas de música en un piano (1773). Al mismo tiempo Henri Maillardet (1745 - ?), mecánico suizo, construyó una muñeca capaz de dibujar (Actualmente se encuentra en el instituto Franklin de Filadelfia).



(a) Gallo de Estrasburgo



(b) Pato de Jacques Vaucanson

Figura 1.1: Autómatas famosos

1.2 Desarrollo de los robots

La ciencia ficción ha envuelto el concepto de robot en mitos e ideas un poco distintas de la realidad. Es común ver y recordar películas donde robots antropomorfos confrontan a la raza humana, siendo máquinas hostiles y con conciencia propia. A pesar de su aspecto negativo, esta percepción ha aumentado la popularidad y el interés hacia los robots. En la actualidad existen muchos tipos de robots en todo el mundo, muy diferentes a los mostrados en las películas futuristas, pero que desde el punto de vista tecnológico representan máquinas realmente sorprendentes. Hoy en día los robots se encuentran en las fábricas automotrices como brazos robóticos para soldar y ensamblar, en el espacio exterior explorando los confines del sistema solar, en las calles de las ciudades ayudando en la desactivación de bombas, en conflictos bélicos asistiendo a los soldados en el campo de batalla, y actualmente en las casas y oficinas podando el césped y aspirando las alfombras.

Los autómatas descritos en el párrafo §1.1 son los antepasados de los primeros robots, después, con las mejoras en dispositivos mecánicos y electrónicos se construyeron los telemanipuladores, considerados los parientes más cercanos de los robots actuales. R.C. Goertz desarrolla en 1948 el primer telemanipulador en el Argonne National Laboratory. Su objetivo era manipular objetos radioactivos sin ningún riesgo para el operador. Más tarde, en 1954, gracias a los avances en la electrónica, Goertz reemplaza los dispositivos mecánicos por mecanismos servocontrolados. Salvo por las aportaciones en el control supervisado y la telepresencia, los telemanipuladores no han cambiado mucho desde los primeros prototipos, además de pertenecer a un mercado selecto como la industria nuclear, espacial y militar. Posteriormente los avances en computación permitieron controlar a los manipuladores por medio de computadoras, dando como resultado el concepto de robot actual. En 1954 el inventor británico C.W. Kenward solicita la primera patente para un dispositivo robótico, la cual fue concedida con el número 781465 en el Reino Unido en 1957. El ingeniero norteamericano George C. Devol concibió la idea de un dispositivo programado para la transferencia de artículos, patentado en los Estados Unidos en 1961 con el número 2988237.

Devol y su amigo Joseph F. Engelberg (1925 -), director de ingeniería de la división aeroespacial de la compañía Manning Maxwell, comenzaron a trabajar en 1956 sobre los primeros robots

manipuladores fundando la Consolidated Controls Corporation, que más tarde se convertiría en Unimation (Universal Automation). En 1960 instalan el primer robot industrial, denominado Unimate (Figura 1.2), en la fábrica General Motors de Trenton, Nueva Jersey, dentro de una aplicación de fundición por inyección. Así, los robots industriales han existido desde hace más de 40 años.



Figura 1.2: Unimate, primer robot industrial

De acuerdo con el reporte del WTEC¹ (World Technology Evaluation Center, Centro de Evaluación de Tecnología Mundial) sobre el estado de la robótica [2], los robots industriales cuentan con un mercado estimado en 4 mil millones de dólares, y con una tasa de crecimiento del 4% anual. La mayoría de las aplicaciones son en el manejo de materiales y en la soldadura.

Actualmente la IFR² (International Federation of Robotics, Federación Internacional de Robótica) brinda una definición actual y formal para la palabra robot:

Un robot es un manipulador programable y multifunción, diseñado para mover materiales, partes, herramientas o dispositivos específicos mediante movimientos programados para realizar diferentes tareas.

Para que un robot pueda realizar la tarea que se le confía, necesita de la interacción y sincronización precisa de los elementos que lo integran. En general los robots están formados por los siguientes componentes:

¹<http://www.wtec.org> 22/11/09,

²<http://www.ifr.org> 22/11/09

- Estructura mecánica. Serie de eslabones unidos por articulaciones que permiten el movimiento relativo entre las partes, formando así una cadena cinemática. Cada uno de los movimientos independientes que puede realizar la cadena cinemática se denomina grado de libertad (GDL).
- Actuadores. Elementos que permiten al robot modificar su medio ambiente. Pueden clasificarse en dos tipos distintos, de acuerdo a su función:
 1. Manipulación: brazos mecánicos y herramientas.
 2. Locomoción: ruedas, patas, alas, hélices, etcétera.
- Sensores. Dispositivos que permiten al robot captar información del medio ambiente y de su estado interno.
- Potencia. Sistema que proporciona la energía necesaria para el funcionamiento de los componentes.
- Control. Combinación de hardware y software que sirven como interfaz entre el sistema de procesamiento del robot y los actuadores y sensores.
- Inteligencia. Programas que permiten al robot procesar la información para la toma de decisiones en función de la tarea confiada.
- Comunicación. Dispositivos que permiten interacción con otros robots o personas.

No todos los robots poseen todos los componentes, pero de manera general la mayoría tiene al menos cuatro de los mencionados anteriormente.

En la literatura se han propuesto diferentes clasificaciones de los robots, de acuerdo a sus características mecánicas, sus aplicaciones o su tipo de control, como se muestra en la Tabla 1.1.

Existe una relación inherente en algunas categorías, p. ej., los robots móviles implementan diferentes tipos de locomoción: ruedas, patas, orugas, etc; por lo que es natural que un robot pertenezca a varias clasificaciones a la vez.

Por locomoción	Por aplicación	Por área de trabajo	Por tipo de control
Patas	Industriales	Fijos	Asistidos
Drones	Educativos	Móviles	Autónomos
Orugas	De servicio	Semifijos	Teleoperados
Ruedas	Experimentales		
Bípedos	Entretenimiento		
Espaciales			
Submarinos			

Tabla 1.1: Clasificación de los robots

1.3 Robots humanoides

Desde hace algunos años se ha tratado de llevar a los robots fuera de las industrias, para utilizarlos en nuevos escenarios tales como casas, oficinas y talleres pequeños. A estos robots que operan fuera de la industria se les conoce como robots de servicio. La IFR define un robot de servicio como “aquel que opera de manera semi-autónoma o completamente autónoma, y realiza servicios útiles para el bienestar de las personas excluyendo las operaciones de fabricación”. Este tipo de robots pueden, a su vez, dividirse en robots de servicio personales y robots de servicio profesionales, dependiendo de su función y uso. Según datos de la UNECE³ (United Nations Economic Commission for Europe, Comisión Económica de las N. U. para Europa), el valor estimado de los robots de servicio (personales y profesionales) vendidos en 2005, fue alrededor de cinco mil millones de dólares. En la Figura 1.3 se muestran a dos robots de servicio comerciales: el robot Wakamaru⁴, un robot construido por Mitsubishi⁵ para asistir en las tareas domésticas en los hogares, y uno de los modelos del robot Roomba, un robot aspiradora construido por la empresa iRobot⁶.

La estructura mecánica de los robots de servicio puede llegar a ser muy diferente de un robot a otro: los robots de servicios pueden ser móviles, semifijos, y fijos; pueden implementar locomoción a base de ruedas u orugas, pueden ser submarinos, bípedos e incluso voladores. Los más populares son

³<http://www.unece.org> 22/11/09,

⁴<http://www.mhi.co.jp/kobe/wakamaru> 22/11/09,

⁵<http://global.mitsubishielectric.com> 22/11/09,

⁶<http://store.irobot.com> 22/11/09

los robots de servicio a ruedas, aunque recientemente los robots humanoides bípedos han tenido un auge importante debido a que presentan ventajas significativas con respecto a los robots a ruedas.

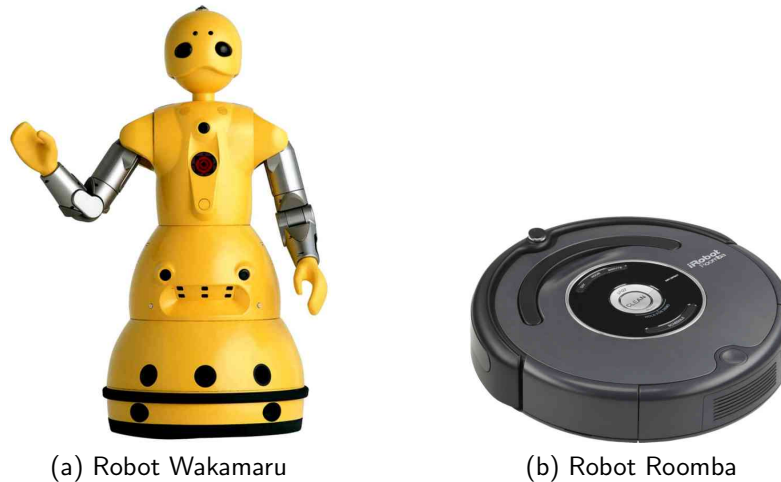


Figura 1.3: Dos robots de servicio en la actualidad

Un robot humanoide es aquel que posee una morfología humana parcial o completa, y manifiesta conductas parecidas a las de un ser humano. Generalmente se relaciona un robot humanoide con un robot bípedo, pero esto no es del todo cierto, pues existen robots humanoides que no poseen piernas ni realizan locomoción bípeda, pero que están provistos por un torso, brazos y cabeza (Figura 1.3.a). En este trabajo nos apegaremos a la idea general de un robot humanoide con morfología humana parcial, y especificaremos como robots humanoides bípedos aquellos que utilizan locomoción sobre dos piernas.

Las ventajas de los robots humanoides bípedos son los principales motivos para su construcción. Entre ellas tenemos:

- Son capaces de moverse en áreas con obstáculos o terrenos que no son completamente planos, lugares que pueden ser de muy difícil acceso para los robots a ruedas.
- Un robot humanoide se adapta de manera sencilla al ambiente de las personas pues la mayoría de estos ambientes son diseñados y construidos para seres humanos.
- Las personas interactúan mejor con robots que presentan morfología y conducta humana [34].

- En el ámbito académico, son excelentes plataformas de investigación para el desarrollo de algoritmos de locomoción, programación genética, reconocimiento de patrones, lógica difusa, redes neuronales, sistemas dinámicos y esquemas de control.
- Debido a su morfología humana, los robots humanoides son atractivos para la industria del entretenimiento [35].
- Existen aplicaciones en las cuales es conveniente que el terreno se dañe lo menos posible. En la agricultura los robots humanoides bípedos causan menos daño en comparación con robots que implementan sistemas de locomoción basados en ruedas u orugas.

1.4 Descripción del proyecto tesis

Alrededor del mundo existen diversos grupos de investigación en el diseño y construcción de robots humanoides bípedos. El interés en estos robots ha motivado la creación de organizaciones dedicadas a fomentar la investigación y el desarrollo en este campo, tales como ROBOCUP⁷ y FIRA⁸, con eventos internacionales como torneos de futbol soccer entre robots.

Hoy en día los robots humanoides son construidos por empresas transnacionales como Honda, Toyota y Sony, las cuales invierten importantes cantidades de dinero para su construcción. Las instituciones académicas también realizan investigación en robótica humanoide, aunque con presupuestos más económicos. El tamaño de un robot bípedo define muchas de sus características, pues la complejidad en su construcción aumenta conforme a su tamaño. Las grandes empresas desarrollan robots de tamaño considerable, de 100 a 170 cm; también es común la construcción de robots humanoides de tamaño medio, de entre 50 y 100 cm, y de robots humanoides pequeños de entre 10 y 50 cm.

Los robots humanoides de tamaño pequeño son una oportunidad asequible para las instituciones académicas. Su costo relativamente bajo los convierte en atractivas plataformas experimentales para el desarrollo de algoritmos de locomoción (exploración, navegación y localización), de control y de interacción hombre-máquina.

⁷<http://www.robocup.org/> 22/11/09,

⁸<http://www.fira.net> 22/11/09

En este proyecto de tesis proponemos el diseño y realización de un sistema robótico humanoide llamado *Kokone* (niños en náhuatl). El objetivo principal es obtener una plataforma robótica educativa de bajo costo, accesible y de código abierto. El software de control de *Kokone* estará disponible para desarrollar e implementar algoritmos de visión, estrategias de locomoción y métodos que mejoren la interacción entre las personas y fomenten el interés hacia los robots humanoides. Los objetivos particulares del proyecto son:

- Diseñar un robot humanoide pequeño (menor a 40 cm) de piezas sencillas con una estructura de fácil ensamblaje.
- Construir el robot con base en el diseño propuesto, utilizando materiales de bajo costo.
- Realizar pruebas mecánicas y electrónicas que validen la estructura del robot.
- Modelar geoméricamente las piernas del robot para facilitar el control de sus movimientos, y proponer una solución factible para resolverlo.
- Validar los modelos geoméricos graficando las posiciones de las piernas.
- Creación de una librería de funciones que abstraiga el control de *Kokone*, proporcionando métodos sencillos para realizar los movimientos y facilitar el desarrollo de estrategias de control más complejas.
- Proponer, validar e implementar un algoritmo de caminado en *Kokone* basado en los conceptos de la caminata estática.

1.5 Organización de la tesis

La organización del documento es la siguiente: En el capítulo 2 se comenta el estado del arte de los robots humanoides más destacados, desde los robots de tamaño natural, hasta los robots humanoides pequeños. En el capítulo 3 se describe el diseño de *Kokone*, el esquema general de la plataforma robótica y los subsistemas mecánico y electrónico que forman el robot. En el capítulo

4 se plantea la cinemática de las piernas del robot, y se describen los métodos propuestos para resolverla. El algoritmo de caminado y el esquema de control del robot se describen en el capítulo 5. En el capítulo 6, las conclusiones y trabajo futuro sobre el proyecto son discutidos. Finalmente, en el apéndice A se describen las funciones de la biblioteca de control y se muestra un ejemplo de una aplicación.

2

Estado del arte en Robots humanoides

A diferencia de otros robots, los robots humanoides poseen características particulares que los vuelven dispositivos con un enorme potencial para el desarrollo de nuestra sociedad. Gracias al avance en la tecnología y las investigaciones, en un futuro la idea de una "máquina compleja" puede cambiar a la de "asistente autónomo". Así, se espera una integración útil de estas máquinas a la sociedad, como muchos otros inventos la han tenido.

2.1 Introducción

Los robots de servicio están cambiando la vida de las personas y, sin lugar a dudas, los robots humanoides ofrecen un enorme potencial de beneficios a la sociedad. Más allá del aspecto tecnológico y científico en estos robots, yace la importancia de su construcción. Sus ventajas impulsan las investigaciones para desarrollar mejores prototipos, y así, incrementar la competitividad en robótica de los países productores. Se cree que la robótica de servicio, en especial los robots humanoides, beneficiará particularmente a los sectores de salud, de seguridad pública y de educación. En el sector

salud, por ejemplo, los robots humanoides son idóneos para asistir a personas de la tercera edad para su cuidado, así como ayudar a personas con diferentes discapacidades. La seguridad pública es un punto primordial en las políticas de los países en general: un robot de servicio puede patrullar ciertas zonas de una ciudad y alertar sobre situaciones sospechosas y/o peligrosas. En educación, la construcción de robots humanoides promoverá e impulsará el interés hacia la robótica, formando especialistas en el área tanto en la academia como en la industria.

La robótica es un área multidisciplinaria, por lo que el desarrollo de robots humanoides se beneficia gracias a los avances en computación, electrónica y mecánica. Es por ello que las investigaciones y resultados que se obtengan en robótica humanoide, pueden aprovecharse en otras áreas tecnológicas, p. ej., los avances en la interacción humano-robot pueden emplearse a problemas de toma de decisiones en procesos industriales, los resultados en visión robótica pueden aplicarse a sistemas de seguridad o a aplicaciones médicas. En nuestro país, desde hace varios años se ha realizado investigación en robótica; sin embargo, aún falta trabajo adicional para lograr una competitividad internacional en robots de servicio. Las proyecciones del mercado mundial son de crecimiento exponencial para la robótica, lo que abre oportunidades y un nicho de crecimiento para México [27].

Actualmente existen muchos tipos de robots humanoides. Como mencionamos en § 1.4, una clasificación podría ser con base en su tamaño: robots de tamaño real de 100 a 170 cm, medianos de 50 a 100 cm, y robots humanoides pequeños, de entre 10 a 50 cm. Así, basados en este aspecto, presentamos a los robots humanoides más representativos.

2.2 Robots humanoides de tamaño real

Desde la década de 1980's se han construido máquinas bípedas de tamaño similar a las personas. Después de años de investigación y desarrollo, los primeros robots humanoides hicieron su aparición en los años 1990's. La construcción de robots humanoides de tamaño real es un gran reto tecnológico. Para el desarrollo de un solo prototipo se necesitan fuertes recursos económicos y sobre todo, un respaldo en tecnología. De tal manera, únicamente grandes empresas e instituciones académicas de prestigio son los principales fabricantes de este tipo de robots.

En esta sección presentamos a los robots humanoides ASIMO [14], HUBO [18], REEM-B [8] y HRP3 Promet Mk-II [24], considerados como los más destacados representantes de esta categoría de robots.

2.2.1 Robot ASIMO

El robot ASIMO (Figura 2.1.a) de la compañía japonesa Honda es uno de los ejemplos más completos en robots humanoides bípedos hoy en día. ASIMO (Advanced Step in Innovative Mobility, Paso Avanzado en Movilidad Innovadora) fue concluido el 20 de noviembre del año 2000. La versión más reciente mide 130 cm y pesa 54 kg. Está provisto de 34 GDL distribuidos de la siguiente manera: 3 en la cabeza, 7 en cada brazo, 2 en cada mano, 1 en la cadera y 6 en cada pierna. ASIMO implementa la tecnología llamada i-WALK, (Intelligent Real-Time Flexible Walking, Caminado Flexible e Inteligente en Tiempo Real), que le permite girar cuando camina sin pausas en su movimiento. Usando la información de sus sensores, ASIMO navega en su ambiente evitando obstáculos y reconociendo su posición. Es capaz de ubicar una fuente de sonido y reconocer algunas palabras; por ejemplo, cuando alguien lo llama, voltea hacia donde escucha su nombre.

ASIMO posee también una gran capacidad de interacción con las personas gracias a su tecnología de reconocimiento visual. Es capaz de interpretar los movimientos de una persona e interactuar con esta. Puede reconocer los rostros de 10 personas diferentes y asociarlas con su nombre. Corre a una velocidad de 6 km/h en línea recta y a 5 km/h de manera circular.

2.2.2 Robot KHR-3: HUBO

La construcción del robot HUBO (Figura 2.1.b) fue llevada a cabo por el KAIST (Korea Advanced Institute of Science and Technology, Instituto Avanzado de Ciencia y Tecnología de Korea). Este robot es la continuación de los trabajos realizados en robótica humanoide en este instituto. HUBO tiene como predecesores al robot KHR-1 y KHR-2, además de su análogo Albert HUBO (en honor a Albert Einstein, un robot humanoide con características similares a HUBO, pero con la peculiaridad de que su rostro es una recreación del rostro del famoso científico). HUBO cuenta con 41 GDL: 6

en la cabeza, 4 en cada brazo, 7 en cada mano, 1 en el cuerpo y 6 en cada pierna. Mide 125 cm y pesa 55 kg. Para realizar el cómputo, este robot cuenta con un sistema PC empotrado con un procesador Intel¹ Pentium III a 933 MHz. El robot utiliza un sistema operativo Microsoft² Windows XP con extensión para tiempo real. En la cabeza se encuentran 2 cámaras CCD para el sistema de visión, cuenta con sensores de fuerza en los tobillos y en las muñecas, además de acelerómetros en las plantas de los pies y sensores inerciales en el torso.

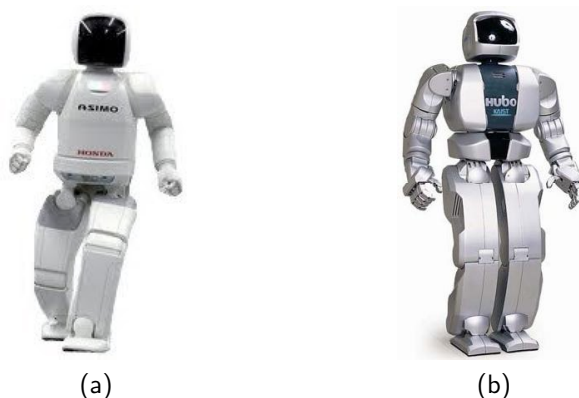


Figura 2.1: Robots ASIMO y HUBO

2.2.3 Robot REEM-B

Considerado uno de los robots humanoides más fuertes, el robot REEM-B (Figura 2.2.a) fue construido por la compañía Pal Technology Robotics³ con sede en España, y tiene como predecesor al robot REEM-A. Puede cargar hasta 12 kg con sus brazos, mide 147 cm de altura y pesa 60 kg. Cuenta con 40 GDL: 6 en cada pierna, 5 en cada brazo, 2 en la cadera, 3 en la cabeza, 2 en una mano tipo pinza y 11 en la otra mano de forma humana.

Puede caminar a una velocidad de 1.5 km/h y tiene una autonomía de energía de 120 minutos. El procesamiento de información se realiza con un sistema de 2 computadoras. La primera, basada

¹<http://www.intel.com> 23/11/09,

²<http://www.microsoft.com> 23/11/09,

³<http://www.pal-robotics.com> 23/11/09,

en un procesador AMD⁴ Geode a 500 MHz, se encarga del control del robot. La segunda está basada en un procesador Intel Core 2 Duo con a 1.66 GHz y realiza las funciones multimedia, como por ejemplo las rutinas de visión artificial. Este robot es capaz de reconocer rostros, subir escaleras y sentarse. También reconoce diferentes objetos y puede generar un mapa del ambiente donde navega.

2.2.4 Robot HRP-3 Promet Mk-II

La empresa japonesa KAWADA⁵ ha realizado diferentes robots humanoides durante los últimos años y el robot HRP-3 Promet Mk-II (Figura 2.2.b) es su versión más reciente. Tiene como antecesoros a los robots HRP-1 y HRP-2, así como un prototipo experimental, el HRP-3. Su estructura mecánica fue elaborada para prevenir la penetración de polvo u otras sustancias que puedan dañarlo. El diseño de sus manos le permite la manipulación de diversos objetos y cuenta con un avanzado software para la interacción en ambientes reales. Construido sobre la base robótica del HRP-2, este robot mide 160.6 cm y pesa 68 kg incluyendo las baterías.

Este robot cuenta con 42 GDL configurados de la siguiente manera: 2 en la cabeza, 14 en ambos brazos, 12 en ambas manos, 2 en la cintura y 12 en ambas piernas. El principal objetivo de su construcción es contar con un robot asistente autónomo en los talleres pequeños y en las casas.

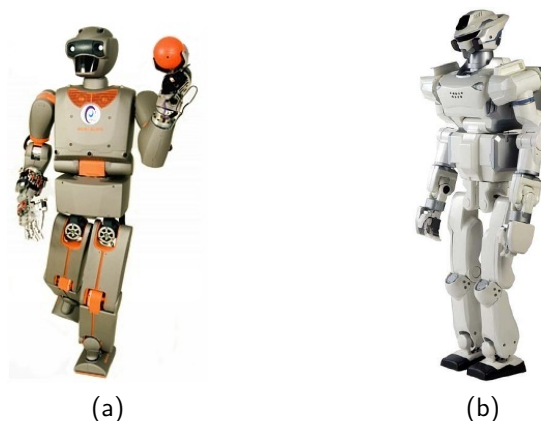


Figura 2.2: Robot REEM-B y HRP-3

⁴<http://www.amd.com> 23/11/09

⁵<http://global.kawada.jp> 23/11/09

Los robots humanoides presentados son algunos de los más representativos que existen hoy en día. Faltaría espacio para citar a todos los robots humanoides bípedos, por lo que sólo mencionaremos a otros robots interesantes de tamaño real: los robots humanoides de la empresa Toyota Motor Corporation⁶, los robots H6 y H7 [9] de la Universidad de Tokyo⁷ y el robot humanoide Wabian-2 [13] de la Universidad de Waseda en Japón⁸.

2.3 Robots humanoides medianos

Los robots de tamaño mediano cuentan con características parecidas a los robots humanoides de tamaño real y a los de tamaño pequeño. Son construidos por instituciones educativas y por empresas, con un mercado de consumo creciente para el entretenimiento.

El robot Qrio [28] desarrollado por la compañía japonesa Sony y el robot HOAP-3 [3] construido por Fujitsu, son ejemplos muy interesantes de este tipo de robots.

2.3.1 Robot SDR-4X: Qrio

El robot SDR-4X, también conocido como Qrio (Figura 2.3.a), es la culminación de la investigación en robots humanoides por parte de SONY. Tiene como predecesor al robot SDR-3X. Las siglas SDR derivan de SONY Dream Robot o en español Robot Soñado por Sony, y sus orígenes se remontan al año de 1997 cuando la compañía empezó a desarrollar sus primeros prototipos. Qrio (como también es conocido) mide 58 cm y tiene un peso aproximado de 7 kg. Camina a una velocidad de 1.188 km/h y cuenta con 38 grados de libertad repartidos de la siguiente manera: 4 en la cabeza, 2 en el cuerpo, 5 en los brazos, 5 en cada mano y 6 en cada pierna. Posee una mejor capacidad de interacción y reconocimiento de su ambiente que sus antecesores. Cuenta con un sistema operativo de tiempo real llamado APERIOS desarrollado por Sony. Así mismo, posee diversos sensores de distancia, aceleración, presión y de contacto.

⁶<http://www.toyota.co.jp> 23/11/09,

⁷<http://www.jsk.t.u-tokyo.ac.jp> 23/11/09,

⁸<http://www.humanoid.waseda.ac.jp/> 23/11/09

2.3.2 Robot HOAP-3

El robot HOAP-3 (Figura 2.3.b) es construido por la compañía japonesa Fujitsu y es sucesor de los robots HOAP-1 y HOAP-2. Las siglas de su nombre provienen de Humanoid Open Architecture Platform, (Plataforma Humanoide de Arquitectura Abierta). Entre sus características podemos mencionar sus 28 GDL distribuidos de la siguiente manera: 3 en la cabeza, 1 en cada mano, 1 en la cintura, 5 en cada brazo y 6 en cada pierna. Sus diversos sensores le permiten interactuar con su ambiente; es capaz de caminar y levantar objetos. Un aspecto interesante es el uso de RTLinux como su sistema operativo de tiempo real.

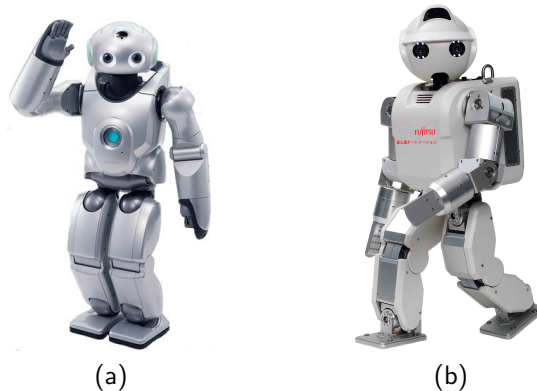


Figura 2.3: Robot Qrio y HOAP-3

2.4 Robots humanoides pequeños

Dado los altos costos para la fabricación de robots humanoides de tamaño real, los robots de tamaño pequeño, menos onerosos en su fabricación, son una solución atractiva para la enseñanza e investigación en robótica humanoide. Actualmente podemos encontrar 2 tipos de robots de tamaño pequeño: los robots comerciales y los construidos por instituciones académicas.

En los últimos años, un creciente número de robots humanoides para el entretenimiento y la educación están disponibles a la venta en el mundo. Esto es el resultado del fuerte crecimiento en la

construcción y el diseño de robots bípedos pequeños y constituye los primeros indicios de un mercado de consumo real en nuestra sociedad. Diversas empresas ofrecen robots de diferentes tamaños y configuraciones, variando principalmente en el número de GDL que poseen y en su capacidad electrónica. La mayoría de los robots comerciales están diseñados para agregarles sensores de diferentes tipos y cuentan con el software necesario para su programación. Entre los más populares podemos encontrar el robot Robonova de HITEC [32] y el robot KHR-2 [31] de KONDO.

2.4.1 Robot Robonova

Quizás el más popular de los robots humanoides pequeños comercialmente disponibles sea Robonova. Construido por la empresa Hitec, mide 40 cm de altura, con un peso aproximado de 2.50 kg y cuenta con 16 grados de libertad: 3 en cada brazo y 5 en cada pierna. La electrónica que lleva consigo permite conectarle diversos dispositivos como giroscopios y sensores de ultrasonidos, además cuenta con el software necesario para su programación y la realización de pruebas. Es una de las primeras opciones para las escuelas en la enseñanza de robótica humanoide, gracias a su versatilidad y flexibilidad. La Figura 2.4.a muestra al robot Robonova.

2.4.2 Robot KHR-2

Este robot es construido por la empresa japonesa KONDO y tiene como predecesor al KHR-1. Su altura es de 35 cm y pesa 1.2 kg. Los 17 GDL que posee están distribuidos de la siguiente manera: 1 en la cabeza, 3 en cada brazo y 5 en cada pierna. Al igual que el Robonova, es muy popular en las escuelas de robótica humanoide, pues permite agregarle diversos sensores conectándolos directamente a su tarjeta electrónica. También, su estructura mecánica permite personalizar el robot agregando diferentes piezas. En la Figura 2.4.b se muestra el robot KHR-2.

En ocasiones, la falta de opciones comerciales de robots humanoides que satisfagan las demandas de las instituciones académicas, así como el desafío tecnológico que representa el desarrollo de un robot de esta naturaleza, motivan a los centros de investigación y universidades del mundo a la construcción de robots de tipo académico. El objetivo es contar con diseños propios de plataformas

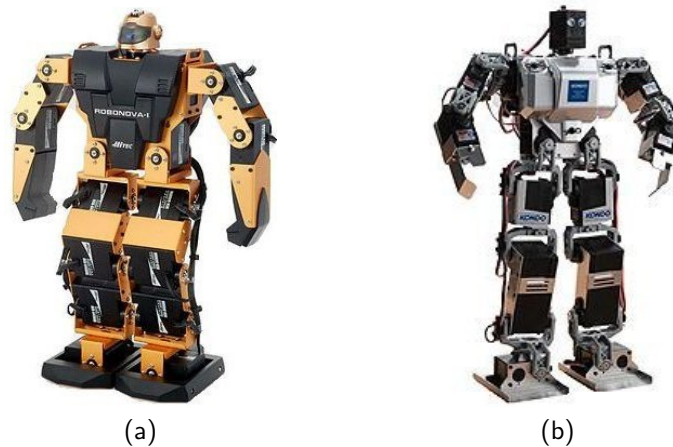


Figura 2.4: Robonova y Kondo KHR-2

robóticas para la implementación de diversas técnicas en locomoción, navegación y visión. Entre los trabajos más representativos se encuentran el robot HR-2 [17], el robot TWNHR-IV [5], el robot Tao-PIE-PIE [21], y el robot ZORC [22].

2.4.3 Robot HR-2

El robot HR-2 fue construido por Almir Heralić en el 2005 en la Universidad Tecnológica de Chalmers en Göteborg, Suecia. Tiene una altura de 30 cm y cuenta con 22 GDL: 2 en la cabeza, 4 en cada brazo y 6 en cada pierna. Para el procesamiento de los algoritmos el robot utiliza una computadora siempre conectada a él por medio de un enlace serial. Esto presenta una ventaja cuando se desarrollan e implementan los algoritmos, pero resta al robot autonomía y movilidad en su ambiente. Para este robot se han desarrollado estrategias de visión e interacción que lo hacen muy completo. Así por ejemplo, es capaz de hablar y su sistema de visión estereoscópico le permite reconocer rostros e identificar si se trata de un hombre o mujer. Otra capacidad es la de imitar: una persona le enseña cómo debe mover determinado objeto, acto seguido el robot confirma que ha entendido y realiza el mismo movimiento. La Figura 2.5.a muestra el robot HR-2.

2.4.4 Robot TWNHR-IV

El robot TWNHR-IV fue construido en la Universidad de Tamkang de la ciudad de Taipei, Taiwan. Mide 46 cm de alto y 25 cm de ancho, con un peso de 3.1 kg. Cuenta con 26 grados GDL: 2 en la cabeza, 2 en el cuerpo, 4 en cada brazo y 7 en cada pierna. Una característica en su diseño es la utilización de 2 motores para la articulación de la rodilla para mejorar la solidez de la pierna. El robot TWNHR-IV es capaz de mantener el equilibrio cuando camina o cuando está detenido en una superficie que se mueva; esto lo logra con un controlador basado en lógica difusa, que recibe las señales de un acelerómetro de 3 ejes. La Figura 2.5.b muestra al robot TWNHR-IV y la configuración de sus 26 GDL.

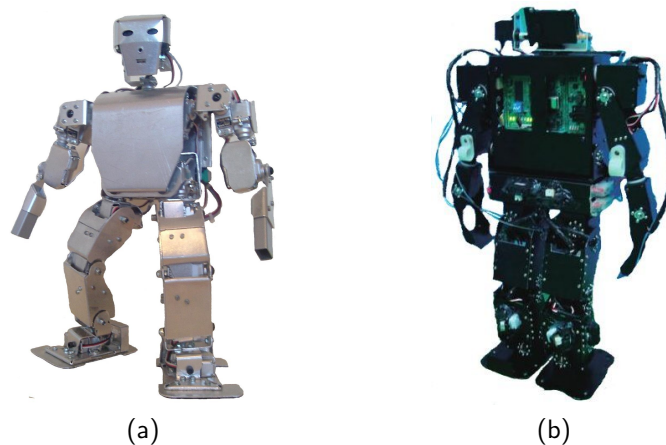


Figura 2.5: HR-2 y TWNHR-IV

2.4.5 Robot Tao-PIE-PIE

El robot Tao-PIE-PIE tiene como objetivo lograr la estabilización y equilibrio en su locomoción bípeda por medio de giroscopios y fue construido en la Universidad de Manitoba en Winnipeg, Canadá. Comparado con los otros robots, cuenta con pocos GDL, distribuidos de la siguiente manera: 3 en cada pierna y 1 en la cabeza, sumando un total de 7. La movilidad del robot es limitada, pero puede presentar un ahorro de energía y de costo. Es la tercera generación de robots construidos en esta

universidad. Al ser un robot autónomo lleva consigo un sistema empotrado Eyebot⁹ y una cámara CMOS para la visión. Tao-PIE-PIE ha competido en ROBOCUP en 2002 y FIRA 2003 obteniendo buenos lugares. La Figura 2.6.a muestra al robot Tao-PIE-PIE.

2.4.6 Robot ZORC

Construido por investigadores de la Universidad de Dortmund en Alemania y por la Universidad Tecnológica de Chalmers en Suecia, el robot ZORC tiene como objetivo servir como prototipo para la implementación de algoritmos genéticos en la solución del problema de la locomoción bípeda. Un robot hermano llamado “Elvina” tiene las mismas características de hardware. ZORC tiene una altura aproximada de 35 cm y un peso de 1.5 kg. Está hecho de material PVC de 0.5 mm de espesor y cuenta con 13 GDL: 2 en cada brazo, 1 en la cadera y 4 en las piernas. La articulaciones son provistas por 2 tipos de actuadores diferentes: los que proporcionan más fuerza están colocados en los tobillos y en las rodillas, los de menor fuerza están colocados en los brazos. Al ser un robot autónomo lleva consigo un sistema empotrado Eyebot. La Figura 2.6 muestra al robot Zorc.

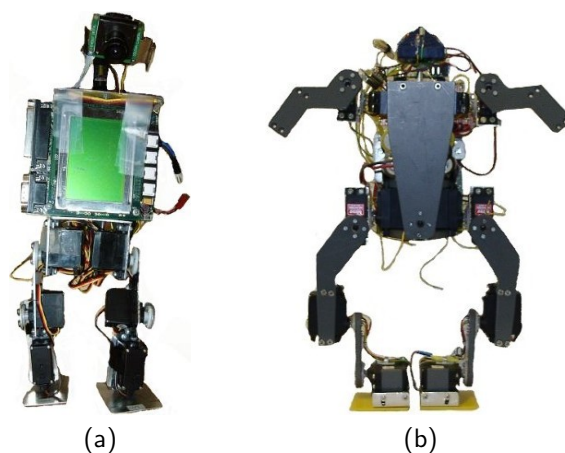


Figura 2.6: Tao-PIE-PIE y ZORC

⁹<http://www.joker-robotics.com/eyebot> 23/11/09

2.5 Robots humanoides en México

En nuestro país se han hecho trabajos muy interesantes en el campo de la robótica humanoide. La Universidad Nacional Autónoma de México (UNAM), con su equipo “Pumas” ha construido los pequeños humanoides “Tafary” y “Miztly”. Estos robots basan su arquitectura en el humanoide Robonova de Hitec, a la que se le agregaron distintos dispositivos, como una cámara para la detección de objetos. Otros robots bípedos en nuestro país son el robot HUNAB-10 que se construyó en el Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) campus Saltillo, el robot “Alphabot”, también construido en el ITESM campus Estado de México, el robot “Roboccio 2” de la Unidad Profesional Interdisciplinaria de Ingeniería y Tecnología Avanzada del IPN (UPIITA), el robot “Patás” de la Unidad Profesional Interdisciplinaria de Ingeniería y Ciencias Sociales y Administrativas del IPN (UPIICSA).

Con la literatura revisada, se recabó la información necesaria para definir puntos importantes del proyecto, de los cuales podemos mencionar el esquema general del sistema, diseño del robot, software de diseño, elección de los materiales para la construcción y programas de control del robot. El siguiente capítulo describe el sistema robótico realizado en este trabajo de tesis, detallando las características de cada subsistema.

3

Diseño de *Kokone*

Diseñar un robot es plasmar en un bosquejo todas sus características. Kokone, como todo sistema mecatrónico, requiere de la interacción de elementos mecánicos, electrónicos y programas de control. El diseño contempla cada aspecto individual de estos subsistemas, con el fin de obtener un sistema estable. Sin duda, las computadoras han facilitado el desarrollo de los robots, no sólo en su control, sino también en su fabricación. Gracias al diseño asistido por computadora, es posible realizar, visualizar y validar diseños en tiempo real ahorrando recursos importantes.

3.1 Introducción

Como se discutió en el capítulo anterior, existen diferentes tipos de robots humanoides y por ende, diversos métodos para su diseño, fabricación y control. Cada elemento comentado en §1.2 es de vital importancia, pues la funcionalidad del sistema completo dependerá de la correcta implementación de cada uno.

El sistema robótico debe ser capaz de realizar correctamente su tarea encomendada, para ello,

todos los subsistemas deben de estar perfectamente sincronizados y trabajar en conjunto para garantizar la correctitud del robot.

3.2 Esquema general del sistema

Dado que el objetivo principal de este trabajo es el desarrollo una plataforma robótica educativa y asequible, se propone un esquema de construcción similar al implementado en el robot HR-2 [17] (presentado en el párrafo § 2.4), donde el procesamiento de la información se realiza en una computadora externa a la estructura mecánica del robot. Este esquema facilita el desarrollo y pruebas de los algoritmos de control, sin embargo, la autonomía motriz del robot se ve afectada por el uso de un cordón umbilical para asegurar la comunicación entre ambos elementos.

La Figura 3.1 muestra el esquema general del sistema. La computadora tipo PC es el “cerebro” de *Kokone*, pues se encarga de realizar todo el procesamiento de los programas de control y envía las instrucciones adecuadas a los motores del robot. *Kokone*, que identificamos como el componente activo y motriz del sistema, lleva en la espalda una tarjeta electrónica encargada de controlar los actuadores y de recibir las instrucciones enviadas por la PC. Un enlace serial (estándar RS-574) se encarga de comunicar al robot con la computadora. El robot puede ser controlado desde cualquier computadora con sistema operativo Linux/Unix compatibles con la API POSIX¹ (para los servicios básicos de E/S). Este aspecto representa una ventaja cuando se realicen pruebas fuera de laboratorio.

Con este esquema se logra una plataforma totalmente abierta para la enseñanza en robótica humanoide. Todas las funciones necesarias para el control de los actuadores y la comunicación con la computadora (desarrolladas en C estándar), se encuentran disponibles para facilitar la programación de estrategias de control más complejas. El hardware del robot es sencillo en su diseño, su estructura está diseñada con piezas de aluminio de fácil ensamble y que no requieren un mantenimiento constante.

¹<http://standards.ieee.org/regauth/posix/> 23/11/09

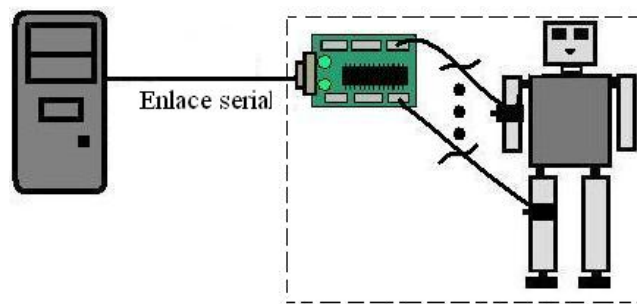


Figura 3.1: Esquema general del sistema

3.3 Motorización

La motorización permite el movimiento de cada uno de los grados de libertad de la estructura mecánica del robot, mediante el uso de actuadores. En el caso particular de *Kokone*, para lograr una movilidad controlada de los GDL se utilizan servomotores (o servos). Estos actuadores son muy populares en modelismo, sistemas de radiocontrol y en robótica. Un servomotor es capaz de ubicarse en cualquier posición dentro de su rango de operación y de mantenerse fijo en dicha posición. Este dispositivo está compuesto por un motor de corriente continua que hace la función de actuador, una caja reductora de engranes que transforma la velocidad del motor en torsión, y un circuito controlador para ubicar la posición del servo. El método de control es a través de una señal de fácil transmisión y recepción, conocida como modulación por ancho de pulso, PWM por sus siglas en inglés (Pulse Width Modulation). La señal consiste de pulsos positivos de 0.9 a 2.1 ms de duración, repetidos 50 veces por segundo (cada 20 ms). En proporción a la anchura del pulso, la posición del servo es definida. Generalmente el rango de operación de los servomotores estándar es de 90° .

Los servomotores usados para la construcción de *Kokone* son modelo BMS-380MAX Blue-Bird² (Figura 3.2). Estos servomotores son del tipo microservos, y poseen características interesantes para los propósitos de este proyecto: su costo es relativamente bajo, poco peso, engranes de metal en su caja reductora y una buena relación torsión-peso. La Tabla 3.1 describe las características más relevantes de estos servomotores.

²<http://www.blue-bird-model.com/> 23/11/09

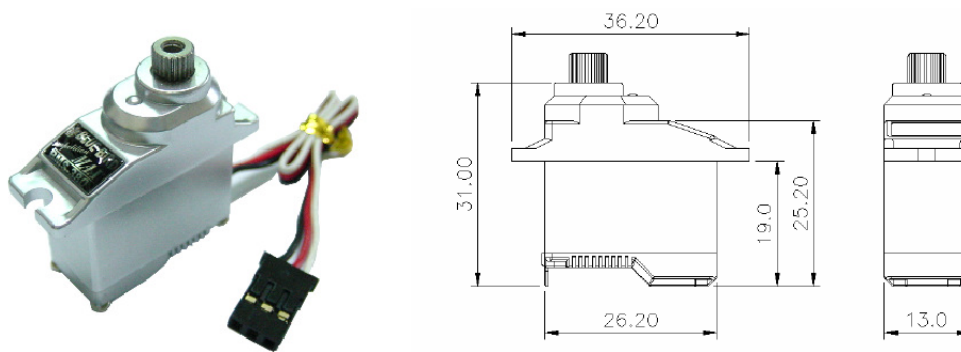


Figura 3.2: Servomotor BMS-380MAX

Característica	BMS-380MAX
Peso	16.4 gr.
Torque a 4.8V	4.2 Kg/cm.
Velocidad a 4.8V	0.16 s/60°
Consumo de corriente	290 mA.
Engranajes de metal	Si, 2 de aluminio
Tipo de motor	Motor de 2 polos

Tabla 3.1: Características del servomotor BMS-380MAX

3.4 Electrónica y potencia

La electrónica de *Kokone* abstrae la comunicación con la computadora y el control de bajo nivel de los servomotores. Existen en el mercado diferentes opciones de tarjetas electrónicas con diversos diseños según el prototipo robótico a realizar. Una tarjeta muy popular, debido a sus características, es la segunda versión del modelo SSC-32 de Lynxmotion³ (Figura 3.3). Esta tarjeta controla los servomotores del robot y permite la comunicación con la computadora (vía el enlace serial).

La tarjeta de control brinda también un fácil control de los actuadores por medio del envío de comandos con un formato sencillo. En particular, esta característica se aprovechó para la realización de los programas de control de los servos y de la comunicación, permitiendo una abstracción del hardware y enfocarse en la programación de alto nivel.

³<http://www.lynxmotion.com/> 23/11/09

Entre sus capacidades tenemos:

- Rango de operación de 0.5 ms a 2.5 ms.
- Controla hasta 32 servomotores simultáneamente.
- Comunicación con la computadora vía un enlace serial.
- 4 entradas analógicas y/o digitales (p.ej. para el uso de sensores).

El procesador central es un microcontrolador ATMEL⁴ modelo ATMEGA168. Este dispositivo se encarga de interpretar los comandos recibidos de la PC, traducirlos a la señal PWM y enviarlos a los motores. Con el fin de evitar ruido eléctrico en las líneas de alimentación, los circuitos de la tarjeta y los servomotores son alimentados por fuentes distintas: la potencia de los servomotores es provista por una fuente conmutada ATX, mientras que la energía de los circuitos lógicos es suministrada por una pila de 9V. Para que los datos sean recibidos e interpretados correctamente por la tarjeta, se configuró el puerto serie con los parámetros mostrados en la Tabla 3.2.

Configuración	Valor
Bits de datos	8
Bits de parada	1
Bits de paridad	0
Lectura	Si
Modo local	Si
Control de flujo	Software
Velocidad	115.2 Kbaud

Tabla 3.2: Configuración del puerto serie

El formato en caracteres ASCII para mover un servomotor o grupo de servomotores, es el que se muestra a continuación:

```
# <ch> P <pw> S <spd> ... # <ch> P <pw> S <spd> T <time> <cr>
```

⁴<http://www.atmel.com> 23/11/09

<ch> = Número de canal en decimal, 0-31.

<pw> = Ancho de pulso en microsegundos, 500 - 2500.

<spd> = Velocidad del movimiento en us por segundo de un canal (opcional).

<time> = Tiempo en ms del movimiento completo, afecta todos los canales, 65535 max.(opcional)

<cr> = Caracter de retorno de carro. (Requerido para iniciar la acción)

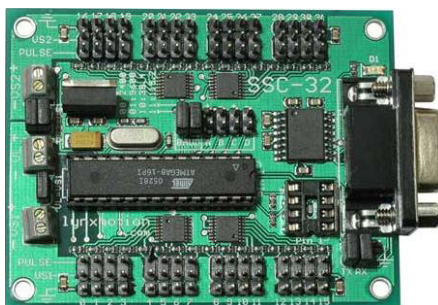


Figura 3.3: Tarjeta de control de servos SSC-32 Lynxmotion

3.5 Estructura mecánica

Sin duda el aspecto mecánico es fundamental para el buen desempeño del robot. Un bajo costo, fácil construcción de las piezas, un sencillo ensamblaje y una altura menor a 40 cm, fueron los objetivos prioritarios en el diseño de *Kokone*. La estructura mecánica de nuestro robot comprende la configuración de los GDL, *i.e.*, de qué manera están distribuidos los distintos GDL en la morfología del robot.

Dada la popularidad del aluminio y su bajo costo, se utilizó este material para la elaboración de las piezas. Fue posible encontrar perfiles de fábrica adecuados para el tamaño de los servomotores, a un costo razonable. Estos perfiles proporcionan rigidez a la estructura del robot, pues no es necesario realizar dobleces que debiliten las piezas, además de facilitar la elaboración de las mismas pues sólo se realizan cortes rectos. Para obtener un ensamblado sencillo, en el diseño de *Kokone* se consideró la reutilización de las piezas, por lo que su estructura está basada en una pieza base de fácil elaboración.

Estas características también son importantes cuando se planea la construcción a largo plazo de más robots con el mismo diseño.

3.5.1 Grados de libertad

Como mencionamos en § 1.2, cada movimiento independiente que puede realizar una cadena cinemática se le conoce como grado de libertad (GDL). En el diseño de cualquier robot, es de suma importancia definir cuántos GDL tendrá y su configuración, pues estos aspectos determinan su movilidad final. Basado en la morfología humana, *Kokone* cuenta con 22 GDL: 6 en cada pierna, 4 en cada brazo y 2 en la cabeza (Figura 3.4).

Las dimensiones físicas del robot, como por ejemplo las distancias que separan las articulaciones entre sí, anchura y altura totales, son determinantes para el diseño de las piezas y el ensamblaje final de la estructura. La distribución de los GDL es una representación de cómo están colocadas las articulaciones, y con base en esta asignación las piezas fueron diseñadas.

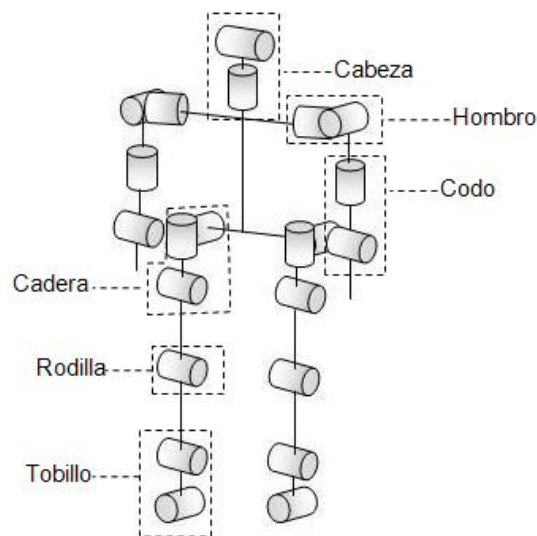


Figura 3.4: GDL del robot

Tomando como base las características de los servomotores (tipo de carcasa y tamaño), se buscó una distribución de los motores que redujera el número de piezas y facilitara el diseño del robot. Los 6 GDL de la pierna son suficientes (como veremos más adelante) para ubicar los pies en

cualquier localización dentro de su rango de operación. La distribución escogida para las piernas de *Kokone* se basó en la pierna humana: 3 GDL en la cadera, 1 en la rodilla y 2 más en el tobillo. Los 3 GDL de la cadera del robot cumplen la función de la articulación tipo rótula de la cadera humana. La colocación del GDL de la rodilla no implica mayor problema debido a que se trata únicamente de un eje de rotación. El diseño del tobillo se realizó de tal manera que la movilidad obtenida sea similar a la articulación tipo rótula guiada del tobillo humano. La configuración de los GDL del brazo se decidió de la siguiente manera: 2 GDL en el hombro y 2 GDL en el codo. Aún cuando el brazo humano tiene más de 4 GDL, de acuerdo a la literatura consultada el uso de 4 GDL en cada brazo es suficiente para los propósitos de los robots humanoides pequeños (mover objetos pequeños, señalar y realizar movimientos coordinados). Los 2 GDL del hombro se aproximan a la rótula del hombro humano. A futuro, se puede implementar un sistema de visión colocando pequeñas cámaras en la cabeza, pues sus 2 GDL le permiten realizar movimientos “pan and tilt”.

3.5.2 Diseño Asistido por Computadora, (Computer Aided Desing, CAD)

El uso de la computadora como asistente en el diseño de productos se ha extendido en muchas áreas. La industria automotriz, la industria aeroespacial y el diseño de barcos son algunos ejemplos. El CAD es el uso de métodos computacionales que facilitan el diseño y modelado de objetos reales o virtuales. Esta técnica permite un ahorro sustancial de recursos, como costos y tiempo. El resultado del CAD es un modelo virtual de un objeto u objetos en particular con las medidas y formas deseadas. Hoy en día existen muchas opciones de software de diseño, la elección de uno en particular dependerá siempre de los recursos del proyecto y de las necesidades del diseñador.

El diseño de la estructura de *Kokone* fue realizado en Solid Edge V19⁵. Este software de diseño maneja interfaces intuitivas y de rápido aprendizaje. Además, cuenta con gran documentación disponible en línea. Una de las características de este software es la reutilización de piezas para formar

⁵<http://www.solidedge.com> 23/11/09

los ensambles (conjunto de piezas acopladas para realizar una función en particular). El diseño es realizado en una ventana 3D que permite visualizar en todo momento la apariencia actual de las piezas y ensambles.

Además de la configuración de los GDL, la morfología humanoide tuvo un papel importante en el diseño de la estructura de *Kokone*. Se aprovecharon la igualdad y relación simétrica de las extremidades del cuerpo humano; así por ejemplo, el diseño de una pierna facilitó el diseño de la otra extremidad. Otro aspecto considerado fue la alineación de las extremidades, característica relacionada con la apariencia: los brazos y las piernas del robot están alineados logrando una apariencia humana. Las piezas y ensambles de la estructura de *Kokone* permiten la movilidad completa de las articulaciones tanto en los brazos como en las piernas. El tamaño final del robot, inferior a los 40 cm, fue un aspecto importante durante todo el proceso de diseño.

Para el diseño de las piezas en particular, se tomaron en cuenta dos aspectos fundamentales: máxima funcionalidad y mínima redundancia. El diseño de *Kokone* está basado en un ensamble base usado en toda la estructura del robot. Esto permitió realizar únicamente los diseños necesarios y reducir el número total de piezas. Las medidas y forma de las piezas están basadas en el servomotor BMS-380MAX. Usando esta técnica, no se presentan partes redundantes sin ninguna función y al mismo tiempo se garantiza el correcto acoplo de los actuadores. Estas dos características también facilitaron la construcción del robot, permitiendo la realización de las piezas rápidamente. Una vez definidos los aspectos de diseño, se realizaron varios esbozos con el fin de cubrir todos los requerimientos mencionados. El diseño final de *Kokone* se muestra en la Figura 3.5. Este diseño es el resultado de diferentes validaciones y pruebas de los requerimientos.

El diseño de *Kokone* fue realizado en 4 etapas: diseño de las piernas, brazos, cabeza y cuerpo del robot. Como mencionamos anteriormente, las medidas y forma de las piezas fueron definidas de acuerdo a las medidas de los servomotores utilizados. Para ello, se realizó el modelo virtual del servomotor y su accesorio llamado cruceta de conexión. (Figura 3.6.a). La cruceta es un accesorio común en los servomotores estándar, su tamaño depende del servo, y su función es ser el medio de conexión entre una pieza y el servomotor.

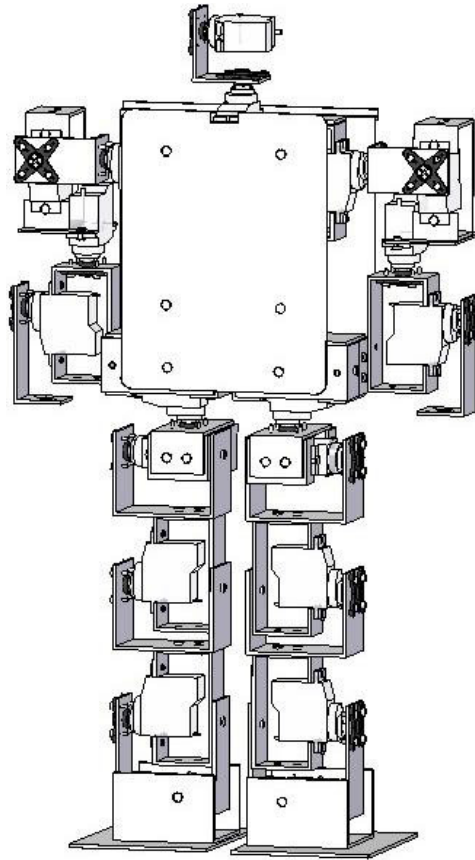


Figura 3.5: Diseño en CAD de *Kokone*

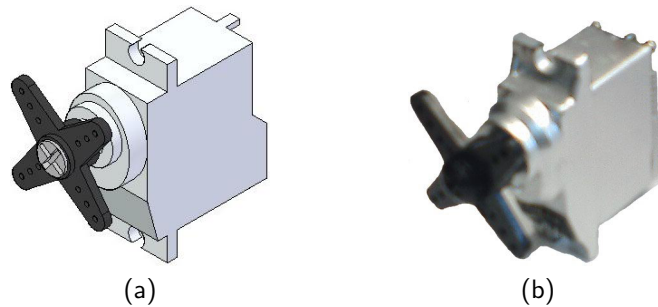


Figura 3.6: Servomotores virtual y real

Todas las extremidades de *Kokone* están basadas en el ensamble E1 (Figura 3.7). Este ensamble está formado por un servomotor y una pieza denominada “sujeta-servo”. Las siguientes secciones detallan cada una de las etapas de diseño, siguiendo un enfoque “top-bottom”.

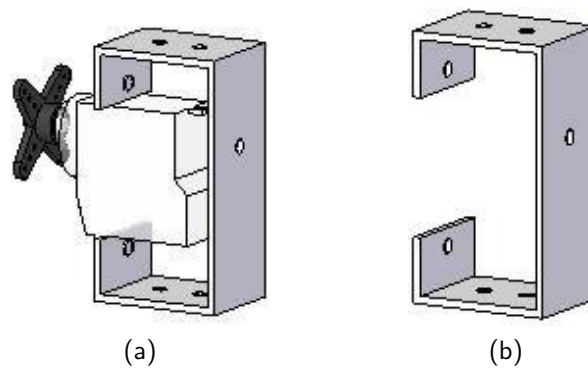


Figura 3.7: Ensamble E1

3.5.2.1. Diseño de las piernas

Para el diseño del robot en general, las piernas juegan un papel importante. Con base en su diseño, se determinaron las proporciones de los brazos y del cuerpo, y así, se logró una apariencia humana proporcionada. Por este motivo, fueron las primeras en diseñarse. Los principales objetivos en su diseño son la completa movilidad de los 6 GDL y usar el menor número de piezas.

Cada pierna del robot está compuesta por 3 ensambles: los ensambles E2, E3 y E4. (Figura 3.8). A su vez, cada ensamble está formado por diferentes piezas y un determinado número de servomotores (el mismo número de GDL que representan). Con el software CAD, se verificaron que los espacios entre las piezas de la pierna fueran los mínimos posibles, y que no existieran colisiones entre éstas. Las proporciones y distancia entre las articulaciones le dan una apariencia semejante a la pierna humana.

El ensamble E2 corresponde a los 3 GDL de la articulación tipo rótula de la cadera del robot (Figura 3.9). Una consideración importante en su diseño fue minimizar las distancias entre las intersecciones de los ejes, con el objeto de simplificar el modelo geométrico de las piernas (como veremos más adelante). El ensamble E2 está formado por 2 sub-ensambles: E2.1 (Figura 3.10.a) y E2.2 (Figura 3.10.b)

El ensamble E3 es bastante sencillo en su diseño (Figura 3.11), ya que corresponde a la articulación de la rodilla, con un sólo GDL. Está basado en el ensamble E1 y una pieza tipo U que permite el

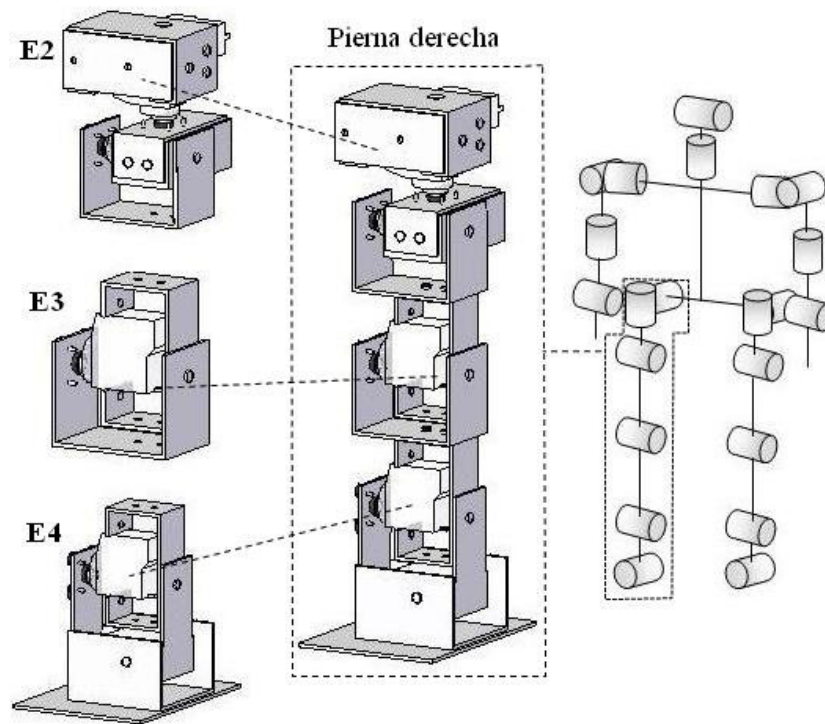


Figura 3.8: Diseño de la pierna

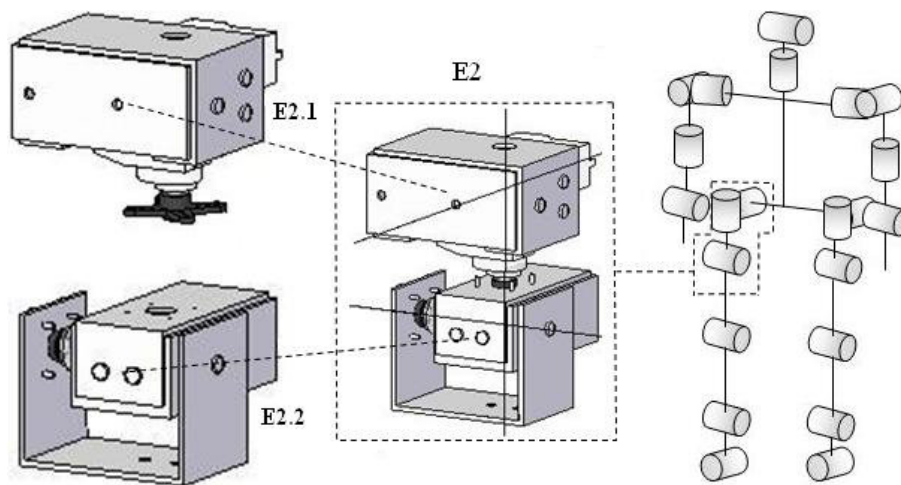


Figura 3.9: Ensamble E2

giro de este ensamble. Para garantizar su completa movilidad, se validó que el giro del servomotor se realice completamente sin obstrucciones, pero manteniendo las distancias mínimas.

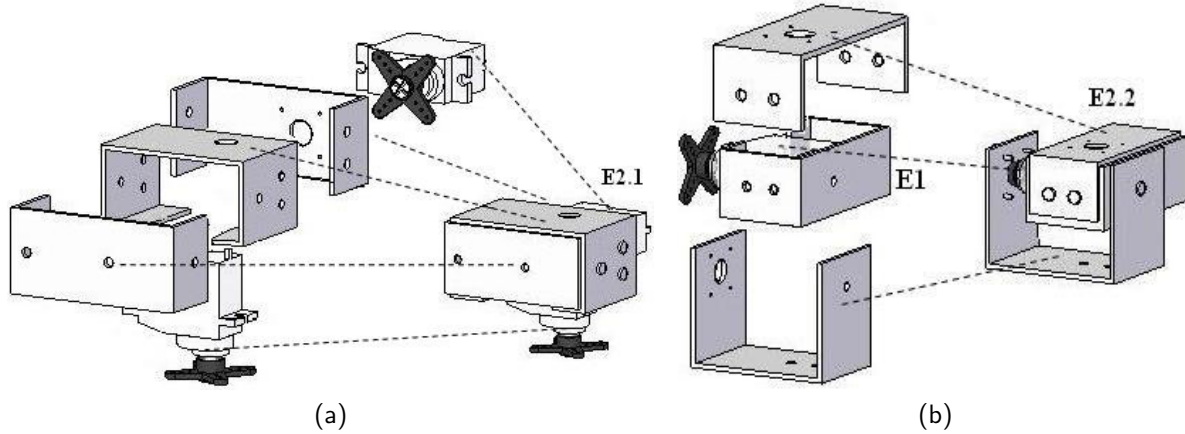


Figura 3.10: sub-ensambles E2.1 y E2.2

Basado en el ensamble E1, el ensamble E4 corresponde a los 2 GDL del tobillo del robot (Figura 3.12). De igual manera que el ensamble E2, el principal aspecto en su diseño fue lograr que los ejes de las articulaciones se encuentren lo más cerca posible.

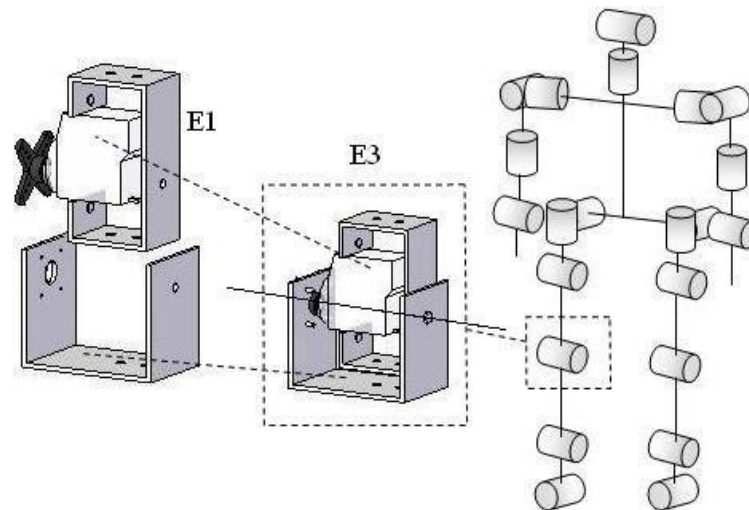


Figura 3.11: Ensamble E3

El diseño la pierna izquierda se facilitó con la pierna derecha concluida. El software Solid Edge permite realizar piezas a partir de otras de diferentes maneras, una de ellas es el “método espejo”. Este método crea una pieza o ensamble idéntico, con la particularidad de que la forma de la pieza

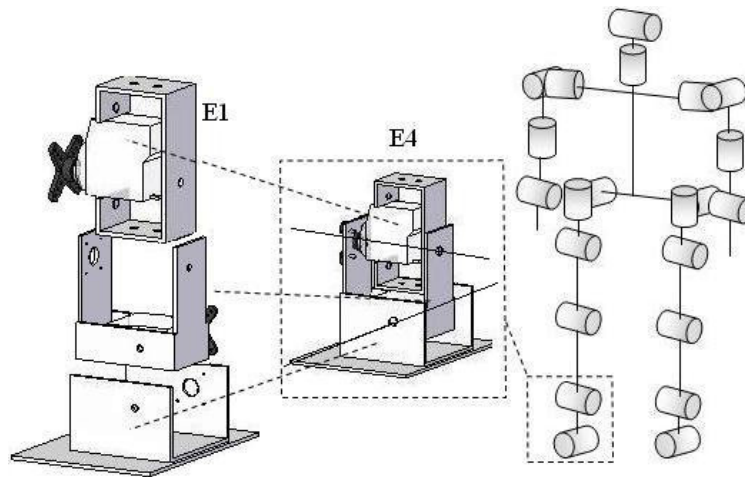


Figura 3.12: Ensamble E4

nueva es el espejo de la pieza original. Para obtener el diseño de la pierna izquierda, aplicamos este método a la pierna derecha completa.

3.5.2.2. Diseño de los brazos

Reutilizando el ensamblaje E1, se facilitó el diseño de los brazos. Siguiendo la distribución de los GDL mostrados en la Figura 3.4, cada brazo de *Kokone* está formado por 2 ensambles: los ensambles E5 y E6 (Figura 3.13). En su diseño están presentes la completa movilidad de los 4 GDL, reducción de las distancias entre los ejes, un fácil ensamblaje del brazo completo y una apariencia semejante al brazo humano.

El ensamblaje E5 (Figura 3.14.a) corresponde a los 2 GDL del hombro de *Kokone*. Aprovechando las crucetas de conexión y su sencillo diseño basado en 2 ensambles E1, permiten ensamblarlo fácilmente sin requerir piezas extras.

Los GDL del codo se implementan en el ensamblaje E6. Su diseño no requiere muchas piezas, pues al igual que los demás ensambles está basado en el ensamblaje E1. En su diseño se considera la integración a futuro de efectores finales para la manipulación de objetos (manos articuladas).

De la misma forma que se realizó el diseño de la pierna izquierda, el diseño del brazo izquierdo fue obtenido aplicando el método 'espejo' al brazo derecho completo.

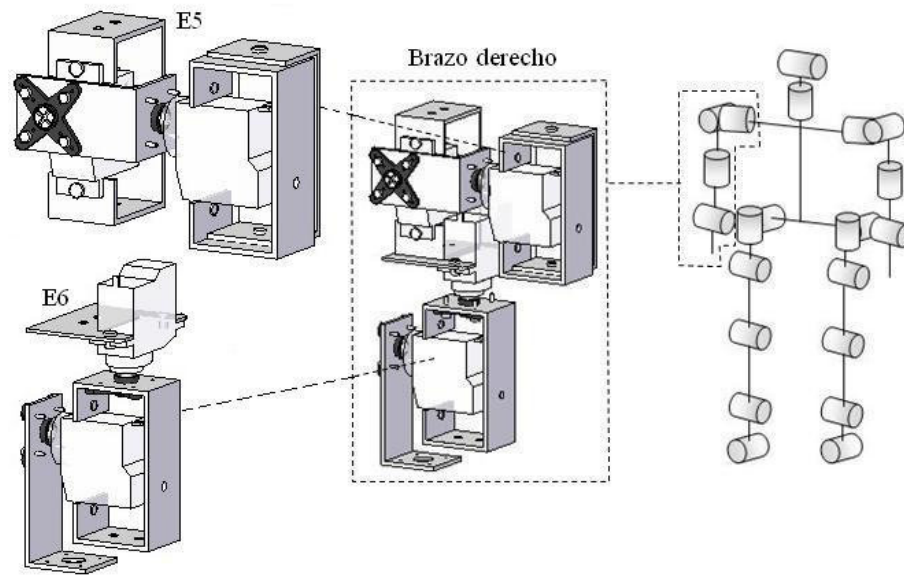


Figura 3.13: Brazo derecho

3.5.2.3. Diseño de la cabeza

El diseño de la cabeza del robot permite a los 2 servomotores realizar los movimientos “pan and tilt” dentro de su rango de operación (Figura 3.15). Su sencillo diseño consiste de sola una pieza, pues se aprovecha las crucetas de conexión para ensamblar los servos. Para fijar la cabeza al cuerpo del robot, se emplean las conexiones de la carcasa de los servos, ahorrando el diseño de una pieza extra. Este mecanismo será utilizado en un trabajo futuro para soportar las cámaras de visión del sistema de visión de *Kokone*.

3.5.2.4. Diseño del cuerpo

El cuerpo de *Kokone* es el punto de unión para las extremidades. Su diseño está basado en las proporciones de las piernas, brazos, cabeza y tarjeta electrónica. Está formado por 2 piezas: la espalda y el pecho del robot, ambos de una sola pieza. Las piernas y espalda del robot forman la cadera completa. En su ensamblaje, se considera la distancia mínima que permite que las piernas giren sin colisionar entre ellas. Los orificios de soporte de los brazos, piernas, cabeza y tarjeta están correctamente alineados para su correcto ensamblaje. El pecho cubre la parte frontal del robot y al

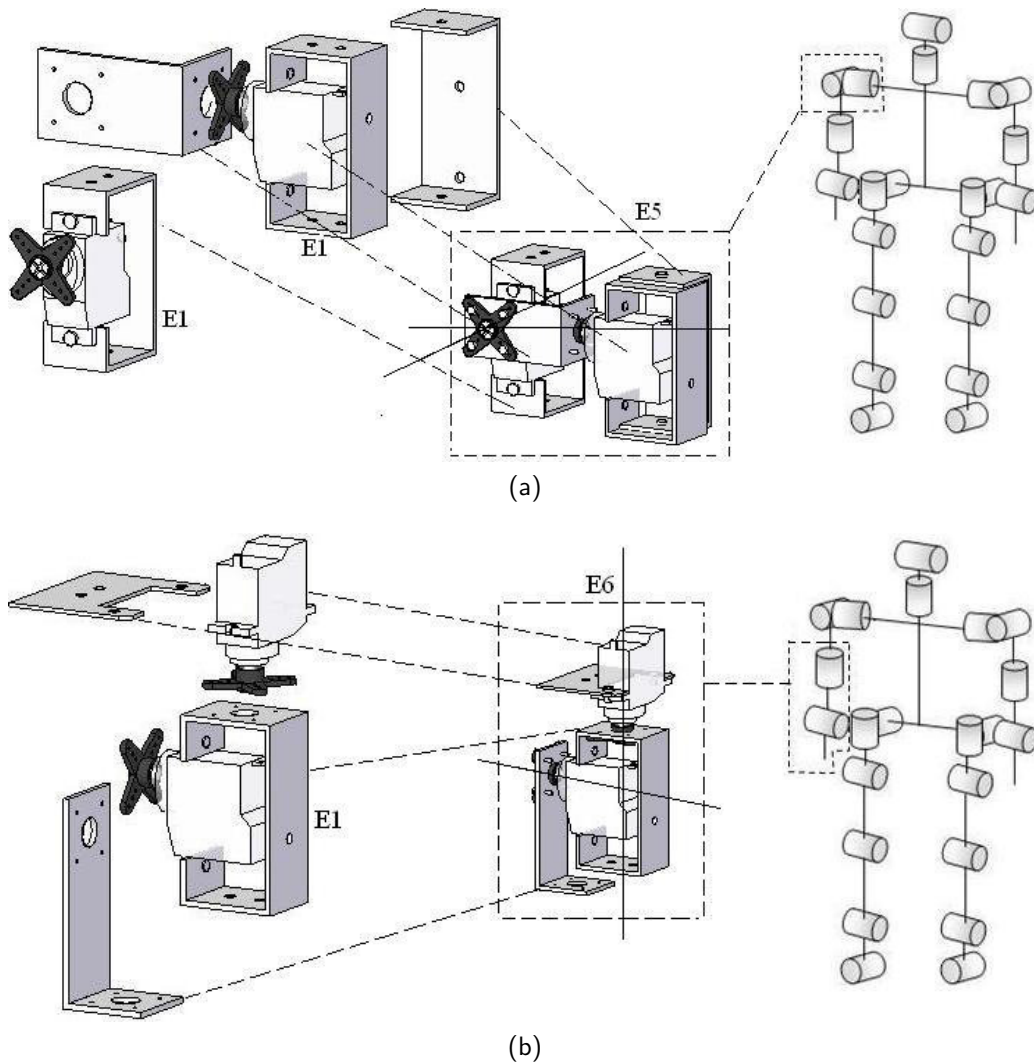


Figura 3.14: Ensamble E5 y E6

igual que la espalda dispone de los orificios de soporte para las extremidades.

3.5.3 Construcción de Kokone

Una vez validado el diseño final, se llevó a cabo la fabricación del robot en colaboración con la empresa “Aluminio Creativo” de Cd. Victoria del Estado de Tamaulipas. La empresa proporcionó los perfiles y herramientas necesarias para su fabricación. Se requirió aproximadamente 1 mes para la construcción completa, desde la elección de los materiales, hasta el ensamble, verificación y ajuste de

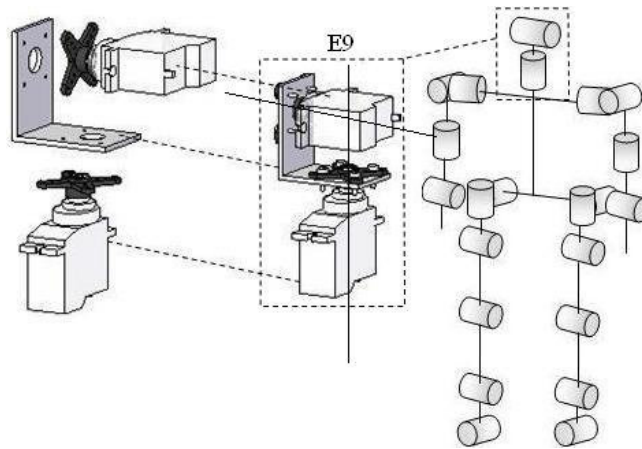


Figura 3.15: Diseño de la cabeza

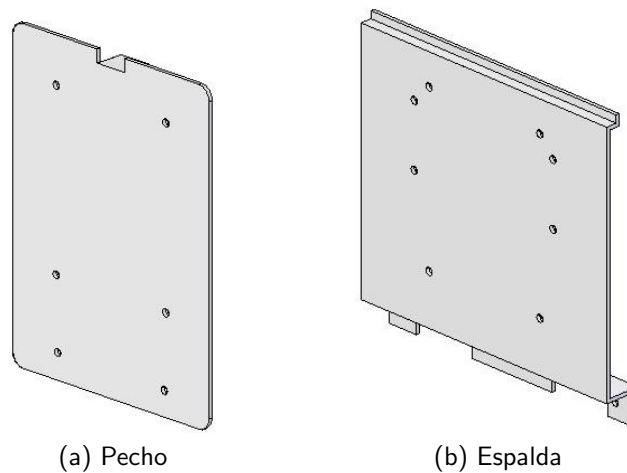
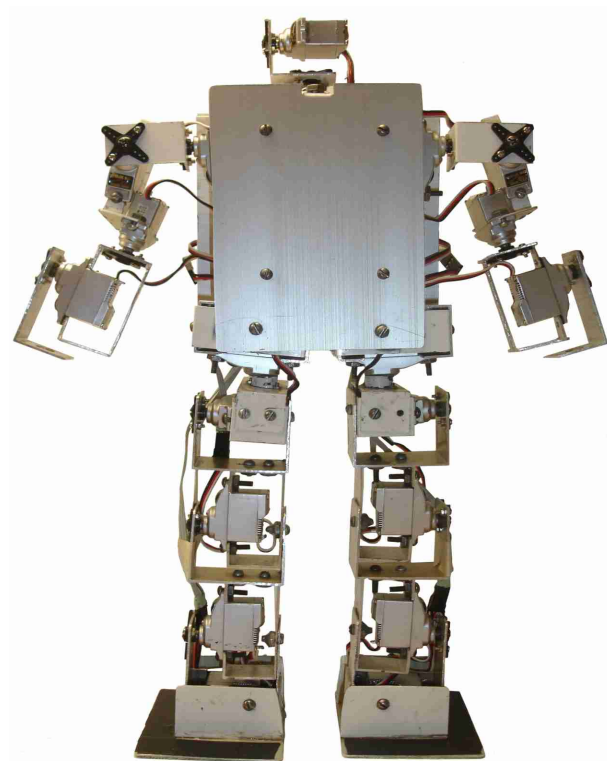
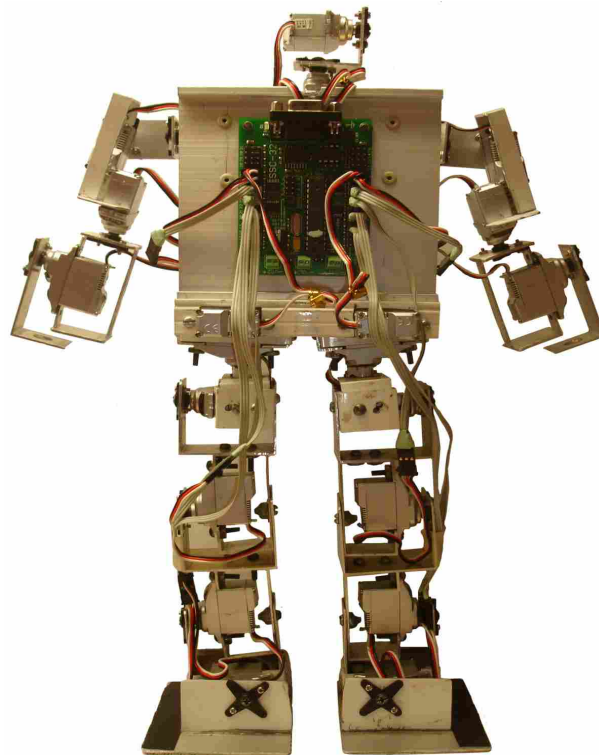


Figura 3.16: Cuerpo

la estructura final. De acuerdo a los movimientos que pueden realizar las extremidades humanas, los servomotores fueron posicionados (a través de la señal PWM) en los ángulos correctos para realizar movimientos coherentes. El tamaño del robot es de 37 cm, con un peso aproximado de 1.2 kg. La versión final se muestra en la Figura 3.17. Dado que uno de los objetivos del robot es realizar un patrón de caminado, es importante modelar geoméricamente las piernas del robot. El modelo geométrico permite desarrollar la biblioteca de funciones para facilitar el control de las piernas. En el siguiente capítulo se detalla el modelo geométrico y los métodos implementados para su solución.



(a) Vista frontal



(b) Vista posterior

Figura 3.17: Realización de *Kokone*

4

Cinemática de *Kokone*

Para realizar la tarea confiada, los robots ejecutan diferentes movimientos interactuando con su entorno, ya sea trasladando piezas en un fábrica o recolectando muestras de rocas en otros planetas. Debido a esto, en robótica es de especial interés el estudio de los movimientos de cada eslabón de la cadena cinemática del robot.

4.1 Introducción

Como comentamos en § 1.2, desde el punto de vista mecánico, los robots son sistemas de cuerpos rígidos conectados por articulaciones. Dado que los robots están diseñados para realizar movimientos, la cinemática es uno de los aspectos fundamentales en el diseño, análisis, control y simulación de robots. La cinemática estudia el movimiento de estos cuerpos sin tener en cuenta las fuerzas o momentos que los generan.

A la posición y orientación de un cuerpo en el espacio¹ se denominan conjuntamente localización. En particular, la cinemática describe la localización, velocidad, aceleración y todas las derivadas de

¹Espacio Euclidiano de tres dimensiones, \mathbb{R}^3

alto orden de la localización de un cuerpo que incluye un mecanismo [1].

Entre las muchas topologías en las que pueden ser conectados los sistemas de eslabones rígidos, existen dos formas en particular en robótica: cadenas cinemáticas abiertas y cadenas cinemáticas cerradas. Una cadena cinemática abierta (Figura 4.1.a) es un sistema de cuerpos rígidos en la cual cada miembro está conectado con otros dos, excepto el primer y último eslabón. Una cadena cinemática cerrada (Figura 4.1.b) es aquella en la cual dos cuerpos están conectados por varias articulaciones, y en la práctica, cada articulación es en sí una cadena cinemática abierta.

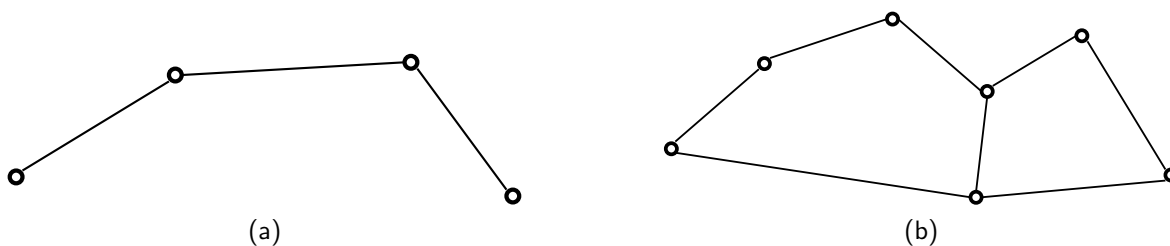


Figura 4.1: Ejemplos de cadena abierta y cerrada

Dado que uno de los objetivos de *Kokone* es ser capaz de realizar un patrones de marcha bípeda, es necesario crear una biblioteca de funciones que brinde a la estrategia de caminado los procedimientos requeridos para controlar los movimientos de las piernas. Así, para incluir estas funciones, se resolvió la cinemática de ambas piernas considerándolas como dos cadenas cinemáticas abiertas, en las cuales los pies son los efectores finales.

Antes de plantear el problema cinemático del robot, presentamos a continuación las herramientas y conceptos matemáticos fundamentales para la localización de objetos en el espacio.

4.2 Representación de la posición y orientación

La cinemática de un cuerpo sólido en el espacio puede ser vista como un estudio de las diferentes formas de representar su localización.

Los desplazamientos de un cuerpo se pueden considerar como traslaciones y rotaciones expresadas por las diferentes representaciones. Ninguna representación es óptima para todos los propósitos,

sin embargo, las ventajas de cada una pueden usarse apropiadamente para facilitar la solución de diferentes problemas.

La forma más común de representar la localización de un sólido en el espacio es mediante un sistema de coordenadas de referencia, o simplemente sistema. Un sistema de referencia i consiste de un origen, denotado como O_i , y una triada de vectores base, mutuamente ortogonales, denotados como $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$, que se consideran fijos con respecto del cuerpo (Figura 4.2.a). La localización de un cuerpo siempre se expresa en relación con otro cuerpo, por lo tanto, se puede expresar la localización de un sistema con respecto a otro sistema. Similarmente, los movimientos de un cuerpo pueden ser expresados como desplazamientos entre dos sistemas de coordenadas, uno de los cuales se considera en movimiento mientras que el otro se considera fijo. Esto indica que el “observador” se encuentra en una posición estacionaria dentro del sistema fijo y que no existe un sistema de referencia absoluto.

4.2.1 Posición y desplazamiento

La posición del origen de un sistema de referencia i con respecto a un sistema de referencia j se denota por el vector de dimensión 3x1:

$${}^j p_i = \begin{bmatrix} {}^j p_i^x \\ {}^j p_i^y \\ {}^j p_i^z \end{bmatrix} \quad (4.1)$$

Las componentes de este vector son las coordenadas cartesianas de O_i en el sistema j , las cuales son las proyecciones del vector ${}^j p_i$ en los correspondientes ejes (Figura 4.2.b).

Una traslación es un desplazamiento en el cual ningún punto en el cuerpo permanece en su posición original, y todas las líneas rectas del cuerpo permanecen paralelas a su orientación inicial. La traslación de un cuerpo en el espacio puede representarse por la combinación de sus posiciones anteriores a la traslación realizada. También, la posición de un cuerpo puede representarse como una traslación que toma el cuerpo desde una posición en la cual el sistema del cuerpo coincide con el sistema estacionario fijo, a una posición en la cual los dos sistemas no coinciden. Así, cual-

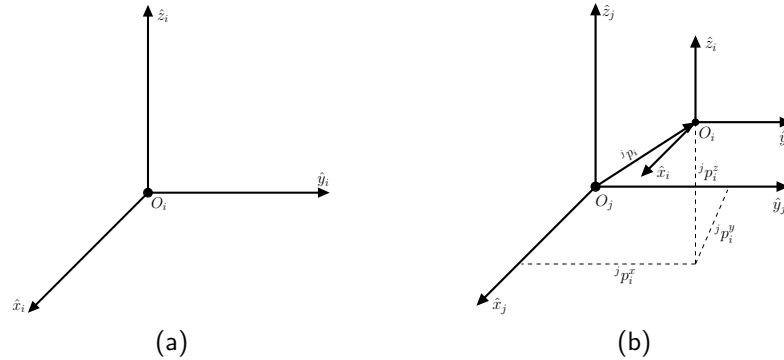


Figura 4.2: Sistema de referencia y coordenadas Cartesianas

quier representación de posición puede ser usada para crear una representación de desplazamiento y viceversa.

4.2.2 Orientación y rotación

Al igual que la traslación, existen muchas maneras de representar la orientación de un cuerpo en el espacio, por lo que en esta sección sólo comentaremos las representaciones más comúnmente usadas en robótica.

Una rotación es un desplazamiento en el que al menos un punto del cuerpo rígido permanece en su posición inicial, y no todas las líneas del cuerpo permanecen paralelas a sus orientaciones iniciales.

Un ejemplo es un cuerpo que orbita circularmente, rotando alrededor de un eje por el centro de su trayectoria circular, cada punto del eje de rotación es un punto en el cuerpo que permanece en su posición. Como en el caso de la posición y la traslación, cualquier representación de orientación puede ser usada para representar una rotación y viceversa.

Matrices de rotación. La rotación de un sistema de coordenadas i relativo al sistema j puede ser denotada expresando los vectores base $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ en términos de los vectores base $(\hat{x}_j, \hat{y}_j, \hat{z}_j)$, esto da como resultado $({}^j\hat{x}_i, {}^j\hat{y}_i, {}^j\hat{z}_i)$, los cuales escritos juntos como una matriz de 3×3 es conocida como la matriz de rotación. Los componentes de esta matriz jR_i son los productos escalares de los

vectores base de los dos sistemas de referencia.

$${}^j R_i = \begin{pmatrix} \hat{x}_i \cdot \hat{x}_j & \hat{y}_i \cdot \hat{x}_j & \hat{z}_i \cdot \hat{x}_j \\ \hat{x}_i \cdot \hat{y}_j & \hat{y}_i \cdot \hat{y}_j & \hat{z}_i \cdot \hat{y}_j \\ \hat{x}_i \cdot \hat{z}_j & \hat{y}_i \cdot \hat{z}_j & \hat{z}_i \cdot \hat{z}_j \end{pmatrix} \quad (4.2)$$

Debido a que los vectores base son vectores unitarios y que el producto escalar de cualesquiera dos vectores unitarios da como resultado el coseno del ángulo entre ellos, a esta matriz también se le conoce como “matriz de cosenos directores”.

En resumen, ${}^j R_i$ es la matriz de rotación que transforma un vector en coordenadas de un sistema i a un vector expresado en coordenadas de un sistema j . También provee una representación de la orientación del sistema i con respecto al sistema j , y por lo tanto, puede ser una representación de rotación del sistema i al sistema j .

Una rotación elemental del sistema i alrededor del eje \hat{z}_j un ángulo θ (Figura 4.3.a) esta definida por la matriz:

$$R(z, \theta) = \begin{pmatrix} \cos(\theta) & -\text{sen}(\theta) & 0 \\ \text{sen}(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.3)$$

El mismo ángulo θ rotado en el eje \hat{y}_j (Figura 4.3.b) se define por la matriz:

$$R(y, \theta) = \begin{pmatrix} \cos(\theta) & 0 & \text{sen}(\theta) \\ 0 & 1 & 0 \\ -\text{sen}(\theta) & 0 & \cos(\theta) \end{pmatrix} \quad (4.4)$$

y alrededor del eje \hat{x}_j (Figura 4.3.c) la matriz es:

$$R(x, \theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\text{sen}(\theta) \\ 0 & \text{sen}(\theta) & \cos(\theta) \end{pmatrix} \quad (4.5)$$

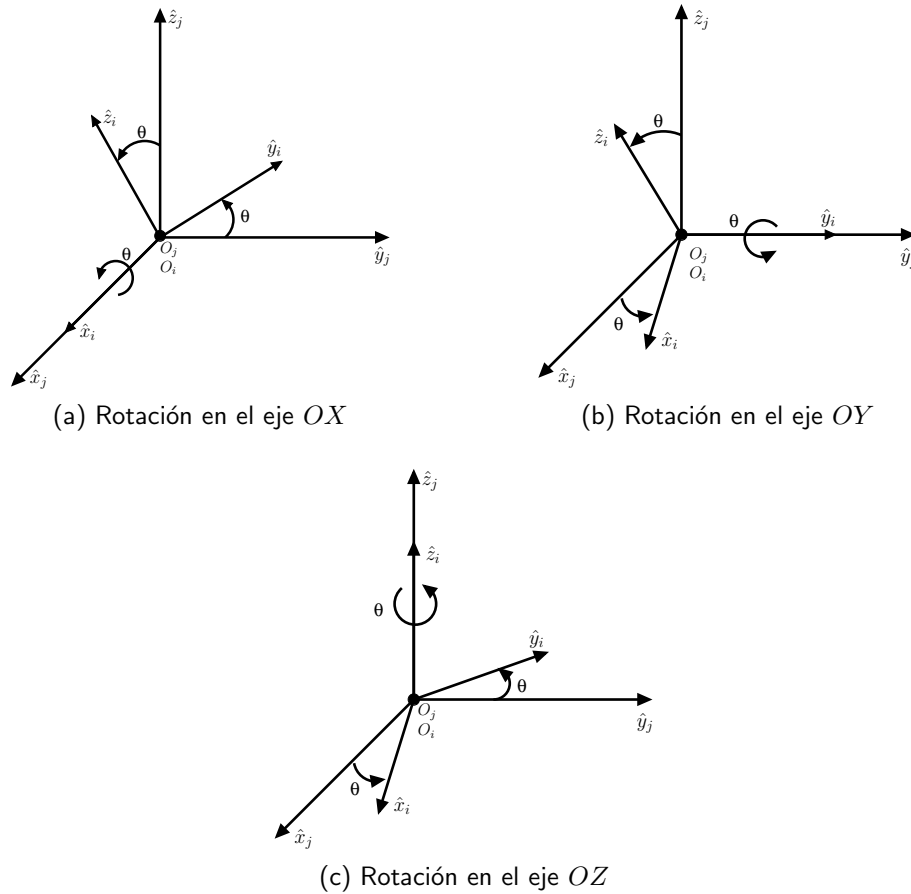


Figura 4.3: Rotaciones en los ejes OX , OY y OZ

Cualquier rotación de un sistema puede representarse como una secuencia ordenada de estas rotaciones básicas. Debido a que los vectores base del sistema i y j son mutuamente ortogonales, las columnas de ${}^j R_i$ son también mutuamente ortogonales.

Una matriz compuesta de vectores ortonormales es conocida como una matriz ortonormal, y tiene la propiedad de que su matriz inversa es igual a su matriz transpuesta. La ortogonalidad de la matriz de rotación se puede apreciar considerando las rotaciones en orden inverso, *p.ej.*, la orientación del sistema j relativa al sistema i , es la matriz de rotación ${}^i R_j$ cuyos renglones son las columnas de ${}^j R_i$. Las matrices de rotación pueden combinarse realizando una simple multiplicación de manera que la

orientación del sistema i con respecto a un sistema k se expresa como:

$${}^k R_i = {}^k R_j {}^j R_i \quad (4.6)$$

La matriz de rotación ${}^j R_i$ está compuestas por nueve elementos, mientras que sólo tres parámetros son requeridos para definir la orientación de un cuerpo en el espacio, por lo tanto, existen seis relaciones auxiliares entre los elementos de la matriz.

Ángulos de Euler. Para una mínima representación, la orientación del sistema i con respecto al sistema j puede ser denotada como un vector de tres ángulos (α, β, γ) . Estos ángulos son conocidos como ángulos de Euler, y cada uno representa una rotación básica sobre uno de los ejes del sistema i . La orientación de cada eje dependerá de la rotación anterior, por lo tanto, el orden de las rotaciones se debe tomar en cuenta. Los símbolos (α, β, γ) generalmente son usados para indicar los ángulos de Euler Z-Y-X. Tomando a los sistemas i y j coincidentes, α es la rotación alrededor del eje \hat{z} del sistema i , β es la rotación alrededor del una vez rotado eje \hat{y} rotado del sistema i , y finalmente, γ es la rotación alrededor del dos veces rotado eje \hat{x} del sistema i . La matriz de rotación equivalente es la siguiente.

Ángulos de Euler Z-Y-X (α, β, γ) :

$$\begin{bmatrix} C_\alpha C_\beta & C_\alpha S_\beta S_\gamma - S_\alpha C_\gamma & C_\alpha S_\beta S_\gamma + S_\alpha C_\gamma \\ S_\alpha C_\beta & S_\alpha S_\beta S_\gamma - C_\alpha C_\gamma & S_\alpha S_\beta S_\gamma + C_\alpha C_\gamma \\ -S_\beta & C_\beta S_\gamma & C_\beta C_\gamma \end{bmatrix} \quad (4.7)$$

Los ángulos de Euler Z-Y-Z y Z-X-Z son otras convenciones usadas de entre doce diferentes orden de rotaciones.

4.3 Transformaciones homogéneas

Las estructuras matemáticas presentadas anteriormente sólo representan la posición o la orientación. Con las transformaciones homogéneas, los vectores de posición y las matrices de rotación son

combinados en una sola estructura matemática compacta.

Cualquier vector ${}^i r$, expresado con respecto a un sistema de referencia i , puede expresarse en las coordenadas de un sistema j si la posición y la orientación del sistema i con respecto al sistema j son conocidos. Usando la notación de los vectores de posición vista en § 4.2.1, la posición del origen del sistema de referencia i con respecto al sistema j se puede denotar como el vector ${}^j p_i = ({}^j p_i^x, {}^j p_i^y, {}^j p_i^z)^T$. Usando la notación de rotación vista en § 4.2.2, la orientación del sistema de referencia i relativo al sistema j se puede denotar por la matriz de rotación ${}^j R_i$, entonces:

$${}^j r = {}^j R_i {}^i r + {}^j p_i \quad (4.8)$$

Esta ecuación puede reescribirse como:

$$\begin{pmatrix} {}^j r \\ 1 \end{pmatrix} = \begin{pmatrix} {}^j R_i & {}^j p_i \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} {}^i r \\ 1 \end{pmatrix}, \quad (4.9)$$

$${}^j r_1^T = {}^j A_i {}^i r_1^T \quad (4.10)$$

donde

$${}^j A_i = \begin{pmatrix} {}^j R_i & {}^j p_i \\ 0^T & 1 \end{pmatrix} \quad (4.11)$$

es la matriz de transformación homogénea de dimensión 4x4, y ${}^j r_1^T$ y ${}^i r_1^T$ son las representaciones homogéneas de los vectores ${}^j r$ y ${}^i r$. La matriz ${}^j A_i$ transforma los vectores en coordenadas del sistema i a coordenadas del sistema j . Su matriz inversa ${}^j A_i^{-1}$ transforma los vectores del sistema j al sistema i

$${}^j A_i^{-1} = {}^i A_j = \begin{pmatrix} {}^j R_i^T & -{}^j R_i^T {}^j p_i \\ 0^T & 1 \end{pmatrix} \quad (4.12)$$

La composición de matrices de transformación homogénea se lleva a cabo mediante la multipli-

cación de matrices, al igual que las matrices de rotación. Por lo tanto, ${}^k A_i = {}^k A_j {}^j A_i$. Dado que la multiplicación de matrices no es conmutativa, el orden es importante.

La matriz de transformación homogénea de una simple rotación alrededor de un eje, en ocasiones es denotada como **Rot** de tal manera que una rotación de un ángulo θ alrededor del eje \hat{z} está dada por la matriz:

$$\mathbf{Rot}(\hat{z}, \theta) = \begin{pmatrix} \cos(\theta) & -\text{sen}(\theta) & 0 & 0 \\ \text{sen}(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.13)$$

De igual manera, la matriz de transformación homogénea de una simple traslación a lo largo de un eje se denota como **Trans**. De modo que una traslación de una distancia d a lo largo del eje \hat{x} es

$$\mathbf{Trans}(\hat{x}, d) = \begin{pmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.14)$$

Las matrices de transformación homogénea son atractivas cuando se desea una notación compacta y/o una programación sencilla. Sin embargo su costo computacional no es muy eficiente, dado que la composición de estas matrices requiere de varias multiplicaciones. Los siguientes ejemplos muestran el uso de las matrices de transformación homogénea.

Ejemplo 1. Dada la traslación del sistema i por la matriz ${}^j A_i$, ¿Cuáles son las coordenadas del vector ${}^j r$, si las coordenadas en el sistema i del vector ${}^i r$ son ${}^i r = (5, 2, -3)$? (Véase la Figura 4.4).

$${}^j A_i = \begin{pmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Aplicando la Ecuación 4.9 tenemos:

$$\begin{pmatrix} j_r \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 5 \\ 2 \\ -3 \\ 1 \end{pmatrix} = \begin{pmatrix} 8 \\ 7 \\ 1 \\ 1 \end{pmatrix}$$

La operación inversa se muestra en el ejemplo 2.

Ejemplo 2. De acuerdo a la matriz anterior jA_i , ¿Cuáles son las coordenadas del vector ${}^i r$ en el sistema i , si las coordenadas del vector en el sistema j son ${}^j r = (8, 7, 1)$? (Véase la Figura 4.4).

Aplicando la Ecuación 4.12 obtenemos la matriz inversa:

$${}^jA_i^{-1} = {}^iA_j = \begin{pmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 1 & -4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Aplicando la Ecuación 4.9 tenemos:

$$\begin{pmatrix} i_r \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & -3 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 1 & -4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 8 \\ 7 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \\ -3 \\ 1 \end{pmatrix}$$

4.4 Representación geométrica

La geometría de un manipulador robótico se define asignando sistemas de referencia a cada uno de los eslabones que integran la cadena cinemática. Cada sistema puede asignarse arbitrariamente en cualquier localización del eslabón. Sin embargo, para lograr eficiencia computacional y llevar una

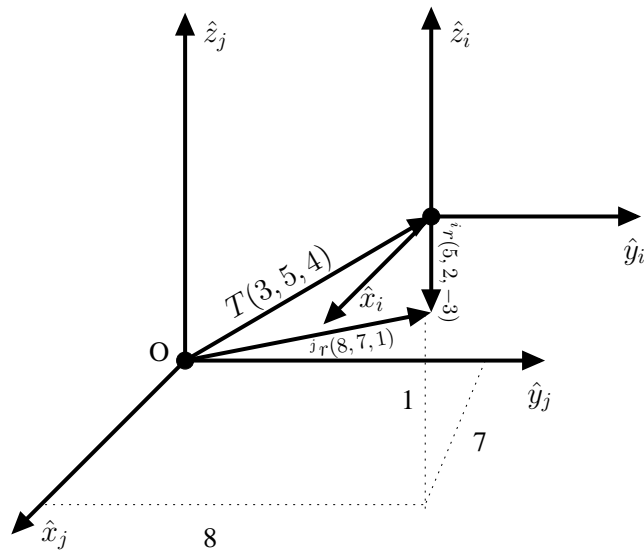


Figura 4.4: Figura del ejemplo 1 y 2

notación consistente, es deseable seguir una convención para fijar los sistemas.

Denavit y Hartenberg [19] propusieron un método sistemático para la asignación de los sistemas de referencia a cada eslabón de una cadena cinemática. Esta convención utiliza cuatro parámetros (en lugar de seis) para localizar un sistema con respecto a otro. Este ahorro de parámetros se debe a la colocación adecuada de los orígenes de los sistemas y sus ejes, de tal manera que el eje \hat{x} de un sistema, intersecte y sea perpendicular al eje \hat{z} del sistema anterior.

En general, un robot de n GDL está formado por $n + 1$ eslabones (considerando la base fija), unidos por n articulaciones. La numeración de los eslabones y articulaciones siguen las siguientes convenciones:

- Los eslabones son numerados empezando con 1 (primer eslabón móvil de la cadena) y terminando con n (último eslabón de la cadena). Se enumera como eslabón 0 a la base fija del robot.
- Cada articulación es numerada comenzando con 1 (primer GDL) y terminando en n .

Con esta numeración, la asignación de los sistemas siguen las siguientes convenciones:

- Localizar el eje de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce desplazamiento.
- Para i de 0 a $n - 1$ situar el eje \hat{z}_i sobre el eje de la articulación $i + 1$.
- Situar el origen del sistema de la base O_0 en cualquier punto del eje \hat{z}_0 . Los ejes \hat{x}_0 e \hat{y}_0 se colocan de modo que formen un sistema dextrógiro con \hat{z}_0 .
- Para i de 1 a $n - 1$, situar el origen del sistema O_i en la intersección del eje \hat{z}_i con la línea normal común a \hat{z}_i y \hat{z}_{i-1} . Si ambos ejes se cortasen se situa O_i en el punto de corte, si fuesen paralelos O_i se situa en la articulación $i + 1$.
- Situar \hat{x}_i en la línea normal común a \hat{z}_{i-1} y \hat{z}_i , y \hat{y}_i se sitúa de modo que forme un sistema dextrógiro con \hat{x}_i y \hat{z}_i .
- Situar el sistema O_n en el extremo del robot, de tal manera que \hat{z}_n coincida con la dirección de \hat{z}_{n-1} y \hat{x}_n sea normal a \hat{z}_{n-1} y \hat{z}_n .

Usando la numeración de los eslabones y la asignación de los sistemas anteriores, los cuatro parámetros que localizan un sistema con respecto a otro se definen como:

- θ_i es el ángulo que hay que girar entorno a \hat{z}_{i-1} para que \hat{x}_{i-1} y \hat{x}_i queden paralelos.
- d_i es la distancia medida a lo largo de \hat{z}_{i-1} que habría que desplazar O_{i-1} para que \hat{x}_i y \hat{x}_{i-1} queden alineados.
- a_i es la distancia medida a lo largo de \hat{x}_i (que ahora coincide con \hat{x}_{i-1}) que habría que desplazar el nuevo O_{i-1} para que coincida con O_i .
- α_i es el ángulo que habría que girar en torno a \hat{x}_i (que ahora coincide con \hat{x}_{i-1}) para que el nuevo S_{i-1} coincida totalmente con S_i .

A estos parámetros se les conoce como parámetros de Denavit-Hartenberg (DH). Siguiendo esta convención, el sistema i puede ser localizado con respecto al sistema de referencia $i - 1$ haciendo

una rotación un ángulo θ_i alrededor del eje \hat{z}_{i-1} , una traslación con distancia d_i a lo largo del eje \hat{z}_{i-1} , otra traslación con distancia a_i a lo largo del eje \hat{x}_i , y una rotación un ángulo α_i alrededor del eje \hat{x}_i . Estas transformaciones se expresan como:

$$Rot(\hat{z}_{i-1}, \theta_i) \cdot Trans(\hat{z}_{i-1}, d_i) \cdot Trans(\hat{x}_i, a_i) \cdot Rot(\hat{x}_i, \alpha_i)$$

La matriz de transformación homogénea resultante es:

$$\begin{aligned} {}^{i-1}A_i &= \\ &\begin{pmatrix} \cos(\theta_i) & -\text{sen}(\theta_i) & 0 & 0 \\ \text{sen}(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\text{sen}(\alpha_i) & 0 \\ 0 & \text{sen}(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos(\theta_i) & -\cos(\alpha_i)\text{sen}(\theta_i) & \text{sen}(\alpha_i)\text{sen}(\theta_i) & a_i\cos(\theta_i) \\ \text{sen}(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\text{sen}(\alpha_i)\cos(\theta_i) & a_i\text{sen}(\theta_i) \\ 0 & \text{sen}(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (4.15)$$

Esta matriz nos permite conocer la localización de un sistema i con respecto a un sistema $i - 1$, es decir, describe la localización de un eslabón con respecto al eslabón anterior. Para una cadena cinemática de n GDL se tendrán n matrices que describirán la localización de los eslabones.

4.4.1 Representación geométrica de Kokone

Uno de los objetivos principales de *Kokone* es poder realizar patrones de marcha bípeda. Como el número de coordenadas requeridas para localizar un cuerpo en un espacio Euclidiano es seis, (tres para la posición y tres más para la orientación), las piernas de *Kokone* cuentan con seis GDL, con el objeto de poder posicionar los pies en cualquier posición y orientación dentro de su espacio de

i	θ	d	a	α
1	$\theta_1 + 90^\circ$	0	0	90°
2	$\theta_2 + 90^\circ$	-4.3	0	90°
3	$\theta_3 + 90^\circ$	0	-6.1	0°
4	θ_4	0	-6.1	0°
5	θ_5	0	-3.3	90°
6	θ_6	0	-2.0	0°

Tabla 4.1: Parámetros DH de ambas piernas

trabajo. Con esto en mente, es posible considerar cada pierna como una cadena cinemática abierta, donde la cadera representa el eslabón fijo (base) para cada pierna y el pie representa el efector final.

Utilizando las convenciones mencionadas anteriores, los sistemas de referencia de la pierna derecha de *Kokone* son mostrados en la Figura 4.5. La ubicación de los sistemas de la pierna izquierda son asignados de igual manera y, por simetría, los parámetros de ambas piernas tiene los mismos valores (Tabla 4.1). Dado que todas las articulaciones de las piernas son rotativas, todos los ejes \hat{z} fueron ubicados en los ejes de rotación. Los ejes \hat{x} e \hat{y} , fueron asignados siguiendo las convenciones presentadas. De acuerdo con la Figura 4.5, la articulación 1 tiene una orientación horizontal, la articulación 2 es perpendicular a la articulación 1 y la intersecta. La articulación 3 es perpendicular a la articulación 2 y la distancia entre ellas es d_2 . La articulación 3 es paralela a la articulación 2, y la distancia entre éstas es a_3 .

La articulación 4 es paralela a la articulación 3, y la distancia entre ellas es d_4 . La articulación 5 es paralela a la articulación 4 y la distancia entre éstas es d_5 . Finalmente, la articulación 6 es perpendicular a la articulación 5, y la distancia entre ambas es d_6 .

Los parámetros DH permiten calcular las matrices ${}^{i-1}AD_i$ y ${}^{i-1}AI_i$, $i = 1 \dots 6$, que modelan geoméricamente las piernas del robot, *i.e.*, describen la localización de un eslabón i con respecto al eslabón $i - 1$ en función de los valores de las coordenadas articulares θd_i y θi_i , $i = 1 \dots 6$.

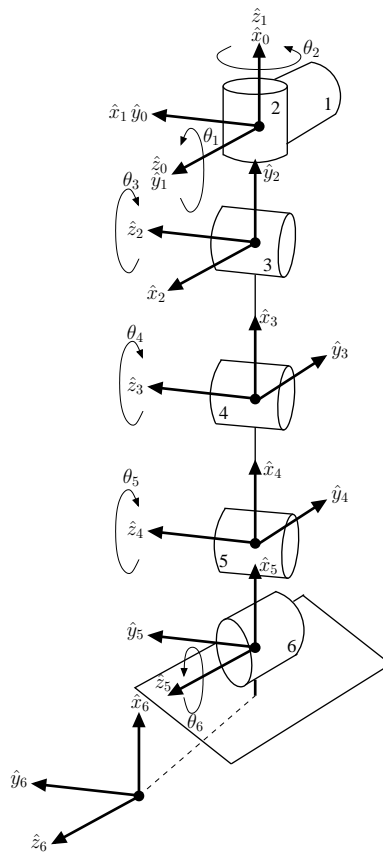


Figura 4.5: Sistemas de referencia asignados a la pierna derecha de *Kokone*

$${}^0AD_1 = \begin{pmatrix} \cos(\theta d_1) & 0 & \text{sen}(\theta d_1) & 0 \\ \text{sen}(\theta d_1) & 0 & -\cos(\theta d_1) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.16)$$

$${}^0AI_1 = \begin{pmatrix} \cos(\theta i_1) & 0 & \text{sen}(\theta i_1) & 0 \\ \text{sen}(\theta i_1) & 0 & -\cos(\theta i_1) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.17)$$

$$\begin{aligned}
{}^1AD_2 = & \begin{pmatrix} \cos(\theta d_2) & 0 & \text{sen}(\theta d_2) & 0 \\ \text{sen}(\theta d_2) & 0 & -\cos(\theta d_2) & 0 \\ 0 & 1 & 0 & -4.3 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^1AI_2 = & \begin{pmatrix} \cos(\theta i_2) & 0 & \text{sen}(\theta i_2) & 0 \\ \text{sen}(\theta i_2) & 0 & -\cos(\theta i_2) & 0 \\ 0 & 1 & 0 & -4.3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
& (4.18) & & (4.19)
\end{aligned}$$

$$\begin{aligned}
{}^2AD_3 = & \begin{pmatrix} \cos(\theta d_3) & -\text{sen}(\theta D_3) & 0 & -6.1 \cos(\theta d_3) \\ \text{sen}(\theta d_3) & \cos(\theta D_3) & 0 & -6.1 \text{sen}(\theta d_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^2AI_3 = & \begin{pmatrix} \cos(\theta i_3) & -\text{sen}(\theta i_3) & 0 & -6.1 \cos(\theta i_3) \\ \text{sen}(\theta i_3) & \cos(\theta i_3) & 0 & -6.1 \text{sen}(\theta i_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
& (4.20) & & (4.21)
\end{aligned}$$

$$\begin{aligned}
{}^3AD_4 = & \begin{pmatrix} \cos(\theta d_4) & -\text{sen}(\theta d_4) & 0 & -6.1 \cos(\theta d_4) \\ \text{sen}(\theta d_4) & \cos(\theta d_4) & 0 & -6.1 \text{sen}(\theta d_4) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} & {}^3AI_4 = & \begin{pmatrix} \cos(\theta i_4) & -\text{sen}(\theta i_4) & 0 & -6.1 \cos(\theta i_4) \\ \text{sen}(\theta i_4) & \cos(\theta i_4) & 0 & -6.1 \text{sen}(\theta i_4) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
& (4.22) & & (4.23)
\end{aligned}$$

$${}^4AD_5 = \begin{pmatrix} \cos(\theta d_5) & 0 & \text{sen}(\theta d_5) & -3.3 \cos(\theta d_5) \\ \text{sen}(\theta d_5) & 0 & -\cos(\theta d_5) & -3.3 \text{sen}(\theta d_5) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.24)$$

$${}^4AI_5 = \begin{pmatrix} \cos(\theta i_5) & 0 & \text{sen}(\theta i_5) & -3.3 \cos(\theta i_5) \\ \text{sen}(\theta i_5) & 0 & -\cos(\theta i_5) & -3.3 \text{sen}(\theta i_5) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.25)$$

$${}^5AD_6 = \begin{pmatrix} \cos(\theta d_6) & -\text{sen}(\theta d_6) & 0 & -2 \cos(\theta d_6) \\ \text{sen}(\theta d_6) & \cos(\theta d_6) & 0 & -2 \text{sen}(\theta d_6) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.26)$$

$${}^5AI_6 = \begin{pmatrix} \cos(\theta i_6) & -\text{sen}(\theta i_6) & 0 & -2 \cos(\theta i_6) \\ \text{sen}(\theta i_6) & \cos(\theta i_6) & 0 & -2 \text{sen}(\theta i_6) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.27)$$

4.5 Cinemática directa

La cinemática directa consiste en encontrar la localización relativa de dos eslabones en particular, de acuerdo a la geometría del robot y los valores sus n articulaciones [1].

El problema cinemático directo es crítico para desarrollar algoritmos de coordinación del manipulador del robot porque los valores de las articulaciones son medidos por los sensores montados en las articulaciones, y es necesario calcular los valores de estas articulaciones con respecto a los sistemas asignados. En la práctica, el problema cinemático directo es resuelto calculando la localización del efector final (sistema n), con respecto a la base del robot (sistema 0). Así, para una cadena cinemática abierta, esta transformación se obtiene mediante la multiplicación de las matrices de transformación homogénea ${}^{i-1}A_i$ que describen la localización de cada eslabón. Así, la matriz

0A_n , representa la localización del extremo final con respecto a su base:

$${}^0A_n = \prod_{i=1}^{i=n} {}^{i-1}A_i \quad (4.28)$$

4.5.1 Cinemática de Kokone

Como mencionamos anteriormente (párrafo § 4.1), las piernas de *Kokone* son modeladas como dos cadenas cinemáticas abiertas, obteniendo los dos conjuntos de matrices ${}^{i-1}AD_i$ y ${}^{i-1}AI_i$, $i = 1 \dots 6$.

Las localizaciones de los pies con respecto a la base de cada una de las piernas, las matrices 0AD_6 y 0AI_6 , quedan descritas de la siguiente manera:

$${}^0AD_6 = \prod_{i=1}^{i=6} {}^{i-1}A_i \quad (4.29)$$

$${}^0AI_6 = \prod_{i=1}^{i=6} {}^{i-1}A_i \quad (4.30)$$

Para completar la descripción de localización de los pies con respecto al cuerpo del robot, se asigna un sistema de referencia a la cadera. Así, las matrices 0AD_6 y 0AI_6 son multiplicadas por dos matrices de traslación A_{td} y A_{ti} , obteniendo las matrices A_{CD} y A_{CI} . Véase la Figura 4.6

$$A_{CD} = A_{td} {}^0AD_6 \quad (4.31)$$

$$A_{CI} = A_{ti} {}^0AI_6 \quad (4.32)$$

$$A_{td} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2.25 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.33) \quad A_{it} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2.25 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.34)$$

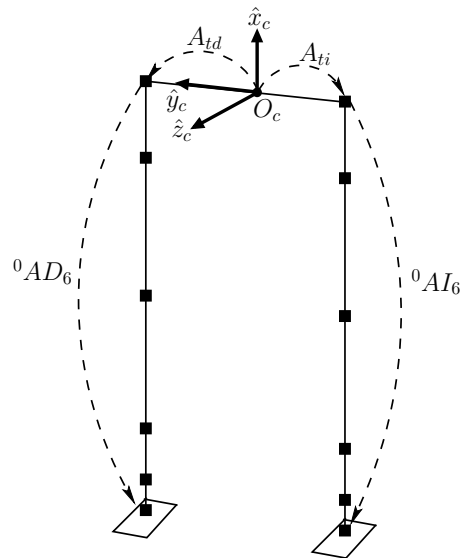


Figura 4.6: Representación de las traslaciones

Los valores de 2.25 en A_{td} y -2.25 en A_{it} son las distancias desde O_c hacia la base de la pierna derecha y hacia la pierna izquierda.

Dado que la cadera del robot es un objeto móvil, es necesario describir la localización de los pies con respecto a un sistema de referencia fijo. Para ello, se elige un punto en el espacio donde se ubica el origen del sistema de referencia global denominado O_g .

De acuerdo al nuevo sistema de referencia, las matrices A_{CD} y A_{CI} son multiplicadas por una matriz A_{mc} que permite pasar de O_g a O_c . Como resultado, se obtienen las matrices A_{DPie} y A_{IPie} ,

que describen la localización de los pies relativo a O_g . Figura 4.7

$$A_{DPie} = A_{mc}A_{CD} \quad (4.35)$$

$$A_{IPie} = A_{mc}A_{CI} \quad (4.36)$$

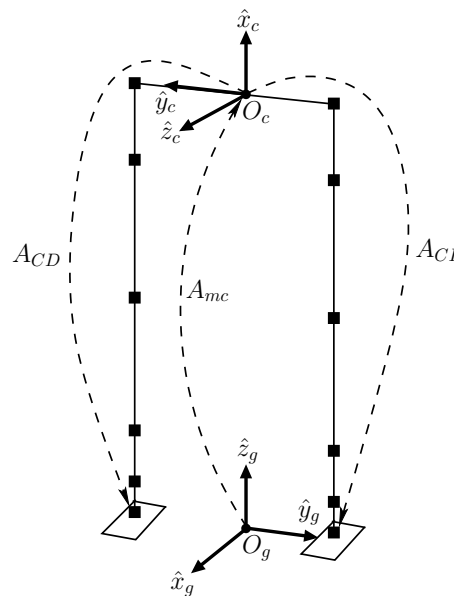


Figura 4.7: Representación de las traslaciones

El ejemplo 3 muestra la forma de usar la expresiones cinemáticas de *Kokone* para calcular las localizaciones de los pies.

Ejemplo 3. Según el desplazamiento de la cadera descrito por $Trans(\hat{z}_c, -1)$, y los valores de θ_d y θ_i mostrados en la Tabla 4.2, ¿Cuál es la localización de los pies con respecto a O_g ?

Para obtener la localización actual de la cadera con respecto a O_g , multiplicamos su localización

i	θd	θi
1	90°	90°
2	90°	90°
3	134°	113°
4	-64°	-46°
5	20°	23°
6	0°	0°

$$Trans(\hat{z}_c, -1) = \begin{pmatrix} 1 & 0 & 0 & {}^9p_c^z = -1 \\ 0 & 1 & 0 & {}^9p_c^y = 0 \\ 0 & 0 & 1 & {}^9p_c^x = 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Tabla 4.2: Parámetros DH del ejemplo 3

por el desplazamiento realizado:

$$A_{mc} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 21.8 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 20.8 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

De acuerdo a la Tabla 4.2, las matrices A_{CD} y A_{CI} son calculadas usando las Ecuaciones 4.31 y 4.32 obteniendo los siguientes valores:

$$A_{CD} = \begin{pmatrix} 1 & 0 & 0 & -19.7201 \\ 0 & 1 & 0 & 2.25 \\ 0 & 0 & 1 & 2.1511 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_{CI} = \begin{pmatrix} 1 & 0 & 0 & -20.8302 \\ 0 & 1 & 0 & -2.25 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Finalmente, las matrices A_{DPie} y A_{IPie} se calculan usando las Ecuaciones 4.35 y 4.36:

$$A_{DPie} = A_{mc}A_{CD} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 21.8 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -19.7201 \\ 0 & 1 & 0 & 2.25 \\ 0 & 0 & 1 & 2.1511 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 2.1511 \\ 0 & -1 & 0 & -2.25 \\ 1 & 0 & 0 & 1.0799 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_{IPie} = A_{mc}A_{CI} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 21.8 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -20.8302 \\ 0 & 1 & 0 & -2.25 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0.0 \\ 0 & -1 & 0 & 2.25 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

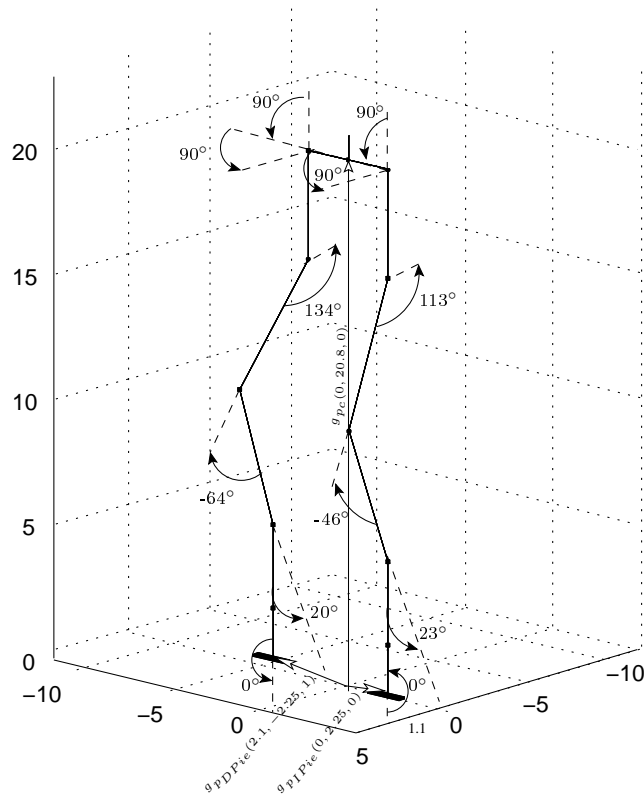


Figura 4.8: Figura del ejemplo 3

4.6 Cinemática inversa

El problema de la cinemática inversa para un manipulador de cadena abierta, consiste en encontrar los valores de sus articulaciones de acuerdo a una localización deseada del efector final y a las

características geométricas de los eslabones. En la práctica los cálculos son hechos con respecto a la base del robot.

Al contrario de la cinemática directa no existe un método sistemático para encontrar la solución de la cinemática inversa. Para robots de pocos GDL (≤ 3), es posible resolver la cinemática inversa por medio de técnicas analíticas como métodos geométricos, desacoplo cinemático o a partir de las mismas matrices de transformación homogénea. Estos métodos son complicados de desarrollar para robots de más GDL. Sin embargo, se han realizado trabajos sobre métodos de programación no lineal y heurísticas para resolver la cinemática inversa de robots de más GDL [25] [37] [26].

Antes de resolver la cinemática inversa del robot, es necesario plantear las expresiones matemáticas que permiten modelar el problema y así elegir la técnica adecuada.

4.6.1 Planteamiento de la cinemática inversa

El problema de la cinemática inversa de *Kokone* consiste en encontrar los valores de θd_i y θi_i , $i = 1 \dots 6$, de acuerdo a la localización de la cadera descrita por A_{mc} , y según la localización de los pies derecho e izquierdo descritos por la matrices A_{DPie} y A_{IPie} .

Con el fin de que los valores de θd_i y θi_i , sean con respecto a O_g , se despeja A_{CD} y A_{CI} de las Ecuaciones 4.35 y 4.36:

$$A_{CD} = A_{mc}^{-1} \cdot A_{DPie} \quad (4.37)$$

$$A_{CI} = A_{mc}^{-1} \cdot A_{IPie} \quad (4.38)$$

Dado que las matrices A_{CD} y A_{CI} están en función de θd y θi , las ecuaciones anteriores pueden reescribirse de la siguiente manera:

$$f(\theta d_i) = A_{mc}^{-1} \cdot A_{DPie} - A_{CD}, \quad i = 1 \dots 6 \quad (4.39)$$

$$g(\theta i_i) = A_{mc}^{-1} \cdot A_{IPie} - A_{CI}, \quad i = 1 \dots 6 \quad (4.40)$$

Así, los valores de θd y θi que hagan que la función sea cero, son los valores correctos para las localizaciones de los pies y de la cadera en particular. Este problema se puede plantear como un problema de optimización cuadrática:

$$\min f(\theta d_i) = \|A_{mc}^{-1} \cdot A_{DPie} - A_{CD}\|, \quad i = 1 \dots 6 \quad (4.41)$$

$$\min g(\theta i_i) = \|A_{mc}^{-1} \cdot A_{IPie} - A_{CI}\|, \quad i = 1 \dots 6 \quad (4.42)$$

Debido a su alta no linealidad, por la cantidad de funciones trascendentales, estas funciones son complicadas de resolver por los métodos tradicionales de optimización. Por lo tanto, una manera de resolver estos problemas es recurrir a técnicas heurísticas. Una de estos métodos son los Algoritmos Genéticos (AG). El uso de AG para resolver el problema de la cinemática inversa data desde la década de 1980's [20], obteniéndose buenos resultados.

Un algoritmo genético es un método iterativo de búsqueda basado en la teoría de la evolución de Charles Darwin: dado un ambiente que sólo puede hospedar un número limitado de individuos, la selección natural favorecerá a aquellos individuos que están mejor adaptados a las condiciones del ambiente [4]. Esta habilidad de supervivencia es guardada en cromosomas, las estructuras genéticas del individuo.

Un AG trabaja sobre un conjunto (población) de soluciones factibles (individuos). Cada individuo es representado por un conjunto de M cromosomas (características de la solución representada) de longitud L . El algoritmo debe encontrar un individuo cuyos valores de los cromosomas mejoren el valor de la función de evaluación (objetivo de optimización). Para conseguir este objetivo, se aplican distintos operadores genéticos (cruza, mutación, selección y elitismo) en los cromosomas de los individuos de la población. El enfoque general de un algoritmo genético se muestra en el Algoritmo 1.

Algoritmo 1 Enfoque general de un Algoritmo Genético**Entrada:** p_m y p_c $t \leftarrow 0$ *Inicializa*(\hat{X}_t) {Inicializa la población aleatoriamente}**repetir** $\hat{F} \leftarrow \text{Aptitud}(\hat{X}_t)$ {Para todos los individuos} $\hat{X}' \leftarrow \text{Seleccion}(\hat{X}_t, \hat{F})$ {Selecciona los mejores individuos de acuerdo a su aptitud} $\hat{X}' \leftarrow \text{Cruza}(\hat{X}', p_c)$ {Para todos los individuos de acuerdo a p_c } $\hat{X}' \leftarrow \text{Mutacion}(\hat{X}', p_m)$ {Para todos los individuos de acuerdo a p_m } $\hat{X}_{t+1} \leftarrow \text{Nueva_poblacion}(\hat{X}'_t)$ $t \leftarrow t + 1$ **hasta que** Condición de parada

El papel de la selección es elegir a los mejores individuos basados en su aptitud. En particular permite que los mejores individuos se conviertan en padres en la próxima generación. El elitismo consiste en preservar al mejor individuo de la población en todas las generaciones. La función de evaluación es la base de la selección. Su papel es de representar los requisitos para adaptarse, i.e., es una función que asigna una medida de calidad a los individuos. La función de evaluación también es llamada función de aptitud, en la terminología de Computación Evolutiva.

La cruce es una operación binaria que mezcla cromosomas (información sobre las características de las soluciones) de dos individuos padres en uno o dos individuos hijos. Es una operación estocástica en dos maneras: la elección de cuándo aplicar la cruce definida por un parámetro de probabilidad p_c , y cuáles cromosomas serán combinados (generalmente se usa una probabilidad de 0.5).

La mutación es una operación unaria. Es una variación en uno o más cromosomas de un individuo, y al igual que la cruce, es una operación estocástica en dos maneras: cuándo aplicarla, definido por p_m y cuál cromosoma será mutado (probabilidad de 0.5).

Existen otros parámetros importantes en un AG, como son la condición de paro del algoritmo y el número de individuos de la población. Su elección depende del problema a resolver y de los resultados de las pruebas. Generalmente la condición de paro es un número de iteraciones definido o una precisión deseada.

Existen diversas maneras de implementar un AG. Generalmente la diferencia en uno y otro es en el tipo de representación de los cromosomas y los operadores genéticos. Para conocer más acerca de

los algoritmos genéticos y en general sobre computación evolutiva puede consultarse el documento [4].

La representación de los cromosomas se puede realizar de varias maneras: vectores de números reales, vectores de números enteros y representación binaria por mencionar algunas. La elección adecuada depende de la naturaleza del problema y la precisión que se requiera. Al igual que la representación, existen muchos tipos de operadores genéticos de selección, mutación y cruza. La elección y forma de implementarse depende de cuál ofrece mejores resultados.

4.6.2 AG para resolver la cinemática inversa de Kokone

Para el caso particular de *Kokone*, se plantea definir un esquema de marcha bípeda describiendo las trayectorias espaciales que deben describir el cuerpo y los pies del robot. Utilizando estas trayectorias como parámetros de entrada, es posible aplicar el AG a cada posición instantánea para encontrar los valores para cada una de las 12 articulaciones de las piernas y determinar así las trayectorias temporales para cada motor.

Para cada posición instantánea del robot, las variables θd_i y θi_i , $i = 1 \dots 6$, son representadas por dos cadenas de seis cromosomas, una para cada pierna, representadas por vectores de números reales con el objeto de obtener soluciones suficientemente precisas.

Tomando en cuenta los movimientos que la pierna humana puede realizar y el rango de operación de los servomotores, los espacios de búsqueda de las variables se han acotado (Tabla 4.3). Estos límites también descartan la posibilidad de encontrar soluciones que satisfagan el problema pero que no sean realizables físicamente por *Kokone*. Las funciones de aptitud están expresadas en las ecuaciones 4.41 y 4.42 de nuestra implementación. Se utiliza como esquema de selección la técnica por ruleta proporcional propuesta por De Jong en 1975 [23] (Algoritmo 2), ya que se trata de uno de los métodos de selección más usados en la actualidad.

Existen muchos operadores genéticos de cruza y mutación, incluso es posible crear un operador con base en los operadores originales y a las necesidades del problema. Se implementó una cruza de dos puntos en el algoritmo. Aleatoriamente se eligen dos números entre cero y seis (el número de cro-

θd y θi	Rango
1	$\frac{\pi}{4} - \frac{3\pi}{4}$
2	$\frac{\pi}{4} - \frac{3\pi}{4}$
3	$\frac{\pi}{4} - \frac{3\pi}{4}$
4	$-\frac{\pi}{2} - 0$
5	$-\frac{\pi}{4} - \frac{\pi}{4}$
6	$-\frac{\pi}{4} - \frac{\pi}{4}$

Tabla 4.3: Rango de las variables

mosomas de la cadena). Estos números son los puntos que indican cuáles conjuntos de cromosomas serán combinados. La probabilidad de cruza es $p_c = 0.7$.

Algoritmo 2 Selección proporcional por ruleta

Entrada: N {Número de individuos}

$suma \leftarrow 0$

para $i = 1$ a N **hacer**

$suma \leftarrow suma + \text{Valoresperado}(\text{individuo}_i)$

fin para

$T \leftarrow suma \div N$

para $i = 1$ a N **hacer**

$suma \leftarrow 0$

$r \leftarrow \text{aleatorio}(0, T)$

$j \leftarrow 1$

mientras $r \leq T$ y $j \leq N - 1$ **hacer**

$suma \leftarrow suma + \text{Valoresperado}(\text{individuo}_j)$

fin mientras

$\text{individuo}_{\text{nuevo}_i} \leftarrow \text{individuo}_j$

fin para

Con el objetivo de lograr soluciones más precisas, se realizó una mutación modificada. La idea principal es reducir el espacio de búsqueda conforme el algoritmo se acerca a la solución. Se le asigna un número Gaussiano al cromosoma i del individuo a modificar, con media μ igual al cromosoma i del mejor individuo. Para definir la desviación estándar σ , se agregó conocimiento del problema a la mutación: si la aptitud del individuo es mayor que 1.0 se considera $\sigma = 0.1$ en otro caso se considera $\text{aptitud} \div 1.5$. Estos valores se obtuvieron realizando diversos experimentos y considerando aquellos valores que mostraron mejor rendimiento. La probabilidad de mutación es $p_m = 0.3$, (Algoritmo 3).

Dado que la resolución teórica de los servos es 0.075° por cada 1 ms, se considera que un error de 0.001 es suficiente para realizar los movimientos del robot. Cuando la aptitud del mejor individuo es menor o igual que 0.001 o se realicen ocho mil generaciones el algoritmo termina.

Algoritmo 3 Mutación Gaussiana

Entrada: $p_m, N, Poblacion_{nueva}, mejor$ $\{N = \text{Número de individuos}\}$

para $i = 1$ a N **hacer**

si $aptitud(mejor) \geq 1.0$ **entonces**

$\sigma \leftarrow 0.10$

si no

$\sigma \leftarrow aptitud(mejor) \div 1.5$

fin si

para $j = 1$ a $Longitud_cromosoma$ **hacer**

si $flip(p_m)$ **entonces**

$aleatorio \leftarrow Gaussiano(cromosoma(mejor, j), \sigma)$ $\{\text{Número Gaussiano}\}$

$asigna(Poblacion_{nueva}(i), j, aleatorio,)$ $\{\text{cromosoma } j \text{ del individuo } i \leftarrow aleatorio\}$

fin si

fin para

fin para

Sin embargo, como cualquier heurística, el algoritmo genético no garantiza lograr la aptitud deseada al finalizar. Una alternativa es reiniciar el algoritmo para aumentar la probabilidad de éxito. Por lo tanto, el algoritmo podría reiniciarse hasta cinco veces en caso de ser necesario. Si aún no se llega a la solución, el algoritmo termina indicando la mejor aptitud calculada. Para determinar este número de reinicios, se probó el algoritmo con 30 posiciones diferentes de los pies y del cuerpo del robot (30 veces cada posición). La Tabla 4.4 muestra los resultados obtenidos por el algoritmo sin reinicialización y la Tabla 4.5 muestra los resultados obtenidos por el algoritmo con reinicialización.

De las dos tablas observamos que la reinicialización ayuda a encontrar mejores resultados. La solución promedio del algoritmo sin reinicio es de 0.020587, la cual es superior a la solución deseada de 0.001. Por otro lado, el algoritmo con reinicio obtuvo una solución promedio de 0.000910, la cual es inferior a 0.001, y por lo tanto una mejor solución. Las mejores soluciones calculadas por ambos algoritmos no difieren mucho: 0.000938 para el algoritmo sin reinicio, y 0.000860 para el algoritmo con reinicios. Sin embargo, una diferencia notoria se puede apreciar en las peores soluciones calculadas por los algoritmos: 0.617623 sin reinicio y 0.001436 con reinicio. Por último, la desviación estándar

del algoritmo con reinicio es de 0.100429, cuyo valor nos indica una dispersión mayor de los resultados con respecto a su valor promedio. En cambio, la desviación estándar del algoritmo con reinicio es de 0.000260, la cual nos indica una dispersión menor con respecto a su media.

Posición	Solución promedio	Solución mejor	Solución peor	Desviación estándar
1	0.006373	0.000571	0.191190	0.010014
2	0.007071	0.000988	0.212141	0.007718
3	0.000915	0.000953	0.027439	0.000085
4	0.000821	0.000969	0.024628	0.000165
5	0.001134	0.000561	0.034030	0.001480
6	0.075629	0.000979	2.268866	0.409443
7	0.108896	0.000986	3.266881	0.591645
8	0.000890	0.000797	0.026714	0.000117
9	0.000896	0.000703	0.026874	0.000100
10	0.000871	0.000974	0.026121	0.000126
11	0.000898	0.000926	0.026933	0.000081
12	0.091865	0.000773	2.755942	0.498623
13	0.000880	0.000590	0.026388	0.000141
14	0.000869	0.000907	0.026068	0.000104
15	0.037814	0.000761	1.134406	0.202302
16	0.000847	0.000979	0.025419	0.000138
17	0.002007	0.000760	0.060204	0.006450
18	0.001133	0.003860	0.033986	0.000785
19	0.104494	0.000945	3.134805	0.386696
20	0.000873	0.000986	0.026202	0.000124
21	0.051280	0.000944	1.538400	0.276225
22	0.000825	0.000699	0.024749	0.000156
23	0.074853	0.000680	2.245602	0.404557
24	0.001030	0.000831	0.030900	0.000568
25	0.002814	0.000962	0.084434	0.010456
26	0.001191	0.000807	0.035736	0.001447
27	0.037752	0.000690	1.132558	0.202152
28	0.000871	0.000793	0.026118	0.000110
29	0.000981	0.000962	0.029435	0.000736
30	0.000850	0.000796	0.025513	0.000133
Promedio	0.020587	0.000938	0.617623	0.100429

Tabla 4.4: Resultados del AG sin reinicio

Posición	Solución promedio	Solución mejor	Solución peor	Desviación estándar
1	0.001926	0.000614	0.018296	0.003710
2	0.001016	0.000960	0.000963	0.000774
3	0.000839	0.000898	0.000990	0.000120
4	0.000861	0.000847	0.000935	0.000096
5	0.000878	0.000979	0.000892	0.000127
6	0.000850	0.000725	0.000863	0.000114
7	0.000884	0.000857	0.000869	0.000093
8	0.000850	0.000919	0.000938	0.000133
9	0.000878	0.000529	0.000985	0.000127
10	0.000872	0.000978	0.000939	0.000125
11	0.000900	0.000746	0.000979	0.000084
12	0.000887	0.000639	0.000805	0.000109
13	0.000876	0.000996	0.000671	0.000118
14	0.000894	0.000667	0.000896	0.000083
15	0.000847	0.000996	0.000991	0.000164
16	0.000875	0.000967	0.000544	0.000130
17	0.000882	0.000958	0.000784	0.000111
18	0.000866	0.000979	0.000986	0.000111
19	0.000867	0.000904	0.000956	0.000110
20	0.000838	0.000973	0.000733	0.000173
21	0.000848	0.000770	0.000959	0.000142
22	0.000864	0.000929	0.000877	0.000133
23	0.000903	0.000857	0.000902	0.000075
24	0.000844	0.000867	0.000609	0.000137
25	0.000872	0.000600	0.000702	0.000112
26	0.000851	0.000868	0.000960	0.000140
27	0.000877	0.000943	0.000572	0.000111
28	0.000904	0.000947	0.000858	0.000107
29	0.000873	0.000970	0.000681	0.000126
30	0.000867	0.000904	0.000935	0.000116
Promedio	0.000910	0.000860	0.001436	0.000260

Tabla 4.5: Resultados del AG con reinicio

El siguiente ejemplo muestra los resultados de aplicar el algoritmo genético según las localizaciones de la cadera y de los pies dados.

Ejemplo 4. Según la localización de la cadera y los pies descritos por la matrices A_{mc} , A_{DPie} y

A_{IPie} , ¿cuáles son los valores de θd_i y θi_i que satisfacen estas localizaciones?

$$A_{mc} = \begin{pmatrix} 0 & 0 & 1 & 1.0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 21.3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_{DPie} = \begin{pmatrix} 0 & 0 & 1 & 7.0 \\ 0 & -1 & 0 & -2.25 \\ 1 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_{IPie} = \begin{pmatrix} 0 & 0 & 1 & -2.3 \\ 0 & -1 & 0 & 2.25 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Aplicando el AG se obtuvieron los resultados de la Tabla 4.6.

i	θd	θi
1	90°	90°
2	90°	90°
3	134°	113°
4	-64°	-46°
5	-20°	23°
6	0°	0°

Tabla 4.6: Tabla del ejemplo 4

Así, para mover los pies a una localización en particular, sólo es necesario definir las localizaciones deseadas de los pies y del cuerpo del robot. Todas las funciones descritas anteriormente se han implementado bajo la forma de una biblioteca de funciones que permiten resolver los problemas de cinemática directa e inversa de *Kokone*. Esta librería se utiliza como piedra angular sobre la cual es posible construir esquemas más complejos de control, como se verá en el siguiente capítulo, en el cual se describen un algoritmo de caminado y el proceso de control del robot propuestos para verificar la realización física de *Kokone*.

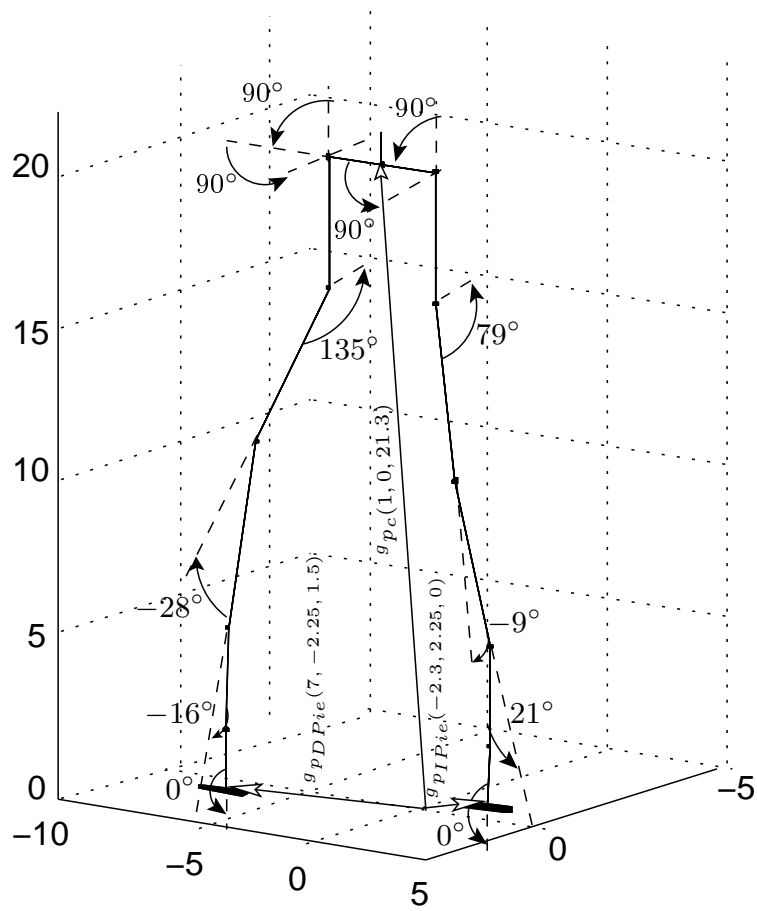


Figura 4.9: Figura del ejemplo 4

5

Primeros pasos de *Kokone*

La capacidad de caminar es uno de los aspectos más interesantes en los robots humanoides. Como mencionamos en § 1.3, la caminata bípeda presenta ventajas importantes comparada con otros métodos de desplazamiento. Por este motivo, la academia y empresas han realizado investigaciones dedicando recursos y tiempo, con el fin de lograr esquemas de control para el caminado de los robots humanoides.

5.1 Introducción

El comportamiento e “inteligencia” de un robot radica en su capacidad de procesar la información, y por ende en sus esquemas de control. Un esquema de control manipula las entradas de un proceso para que las variables del mismo se conformen a valores deseados, denominados valores de referencia. Existen varios procesos en un robot que requieren esquemas de control, *p.ej.*, la interacción con su ambiente, los movimientos de los actuadores, la comunicación y la lectura de los diferentes sensores por mencionar algunos.

Debido a la cantidad considerable de procesos en un robot, se han propuesto esquemas de control

a diferentes niveles, desde el control electrónico (como en el caso de los servomotores), hasta control de la inteligencia artificial (*p.ej.*, interacción humano-robot).

5.2 Control de Kokone

El control del robot se realiza mediante una biblioteca de funciones, que integra la resolución de las cinemáticas directa e inversa de *Kokone* descritas en el capítulo anterior. Escritos en lenguaje C, los métodos brindan un control transparente a las aplicaciones sobre los movimientos de las piernas. Así, en cualquier futura aplicación sólo se tendrá que especificar la localización de los pies para mover las dos extremidades. Esto permite abstraer el control electrónico y los métodos de comunicación con la tarjeta electrónica, formando una estructura jerárquica de control, Figura 5.1. (Las funciones que integran la biblioteca y su invocación son mostradas en el apéndice A).

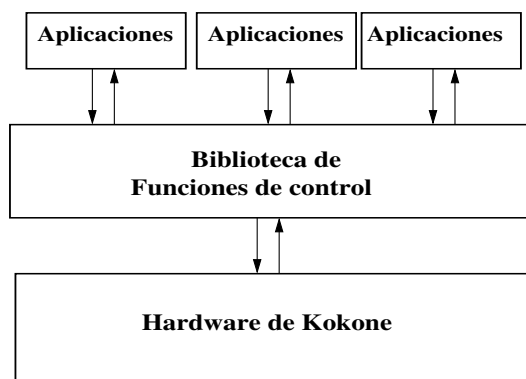


Figura 5.1: Jerarquía de control

5.3 Caminado bípedo

El caminado bípedo en un robot de tamaño real es un proceso que envuelve diferentes subsistemas. Para lograrlo, se requiere de estrategias de control avanzadas que se han desarrollado a través de años de investigación. También, como resultado de estas investigaciones, se han propuesto conceptos para facilitar la comprensión de este proceso.

Hace ya varios años se logró que un robot realizara un patrón de caminado, quedando claro que se trata de un problema complejo de resolver en todas sus características. En la actualidad, se pueden encontrar diversos trabajos alrededor de este tema, con diferentes métodos de solución. Algunos tópicos principales son: criterios de estabilidad [7], generación de la marcha[16][10][11], diseño y aplicaciones de los robots bípedos[12][6] y consumo mínimo de energía [33].

El andar humano puede verse desde diferentes puntos de vista: mecánico, geométrico, energético y anatómico. Desde la perspectiva de diseño mecánico y energético, es importante definir los tipos de actuadores que proveen de movimiento a un robot. En base a este aspecto, la caminata bípeda puede clasificarse como caminata pasiva y caminata activa. De acuerdo al consumo mínimo de energía en la caminata bípeda humana, se han planteado diversos mecanismos que no requieren de actuadores para producir movimiento: el desplazamiento es producido por la fuerza de gravedad que provoca una pequeña pendiente. Este tipo de caminata es conocida como caminata pasiva. Su principal ventaja es el consumo mínimo de energía, sin embargo, su movilidad es limitada, y es imposible subir pendientes con este tipo de caminado. Algunos trabajos en caminata pasiva se pueden encontrar en [15], [36] y [29].

Al contrario de la caminata pasiva, la caminata activa necesita de actuadores que realicen los movimientos del robot. Debido a esto, se requieren de esquemas de control para manipular los actuadores. Una de las ventajas de este tipo de caminado, es que permite mejor y mayor movilidad al robot para realizar diferentes movimientos como subir escaleras, caminar en círculo, equilibrarse en una pierna etc. Todos los robots mostrados en el Capítulo 2 implementan caminata activa.

El proceso de caminata estable de los robots humanoides tiene las bases en la locomoción bípeda humana. Desde este punto de vista, el ciclo de caminado consiste de dos fases: soporte simple y soporte doble. Durante la fase de soporte simple sólo un pie se encuentran en contacto con el piso, mientras que el otro se encuentra levantado y moviéndose de atrás hacia adelante. La fase de soporte doble ocurre cuando ambos pies se encuentran en contacto con la superficie del suelo. En el caminado humano la fase de soporte doble dura aproximadamente el 20 % de un ciclo para dar un paso [30], un ciclo de paso es la mitad del ciclo completo de caminado, la otra mitad es para dar el siguiente paso.

Sin duda el criterio de estabilidad de la caminata bípeda es el tópico más importante en robots bípedos. Según este criterio básicamente existen dos tipos de caminado: caminado con equilibrio estático y caminado con equilibrio dinámico. En el caminado con equilibrio estático se asume que el robot siempre se encuentra estable. Esto significa que si en cualquier momento el movimiento es detenido por cualquier causa, el robot estará en una posición estable indefinidamente. Lograr la estabilidad en el caminado estático requiere de la interacción de los siguientes conceptos: centro de masa (CM), proyección del centro de masa (PCM) y el polígono de soporte (PS). El CM (también conocido como centro de gravedad), es la parte dentro del robot que se considera como un punto en donde se concentra toda la masa del robot, incluyendo la de las extremidades, el torso y la cabeza. El PCM es el punto en la superficie del suelo donde el centro de masa es proyectado. El polígono de soporte es el área formada por la superficie de los pies del robot según en que fase de caminado se encuentre.

En la fase de soporte simple es el área del pie que hace contacto con el piso, en la fase de soporte doble el polígono es la superficie de los dos pies que hacen contacto con el suelo. La Figura 5.2.a muestra los tres conceptos mencionados. La estabilidad en la caminata estática se logra manteniendo siempre el centro de masa dentro del polígono de soporte en ambas fases. En el caminado dinámico, el PCM puede estar fuera del polígono de soporte por periodos limitados de tiempo. Esto permite un andar más rápido y ágil, pero se requiere de un sistema que garantice la estabilidad. No existe un criterio absoluto que indique si el caminado dinámico es estable, sin embargo, si el robot posee articulaciones fuertes en los tobillos, y siempre mantiene por lo menos un pie en el suelo, entonces el concepto de punto de momento cero (Zero Moment Point, ZMP) puede ser usado como criterio de estabilidad.

El ZMP es definido como el punto en el suelo en el cual las fuerzas de reacción en contacto con el pie no producen momento alguno (a excepción del eje vertical), *i.e.*, el pie no se resiste a cambiar su posición debido a las fuerzas de reacción. Mientras el ZMP se encuentre dentro del polígono de soporte, el robot se puede considerar estable y habilitado para caminar, (Figura 5.2.b).

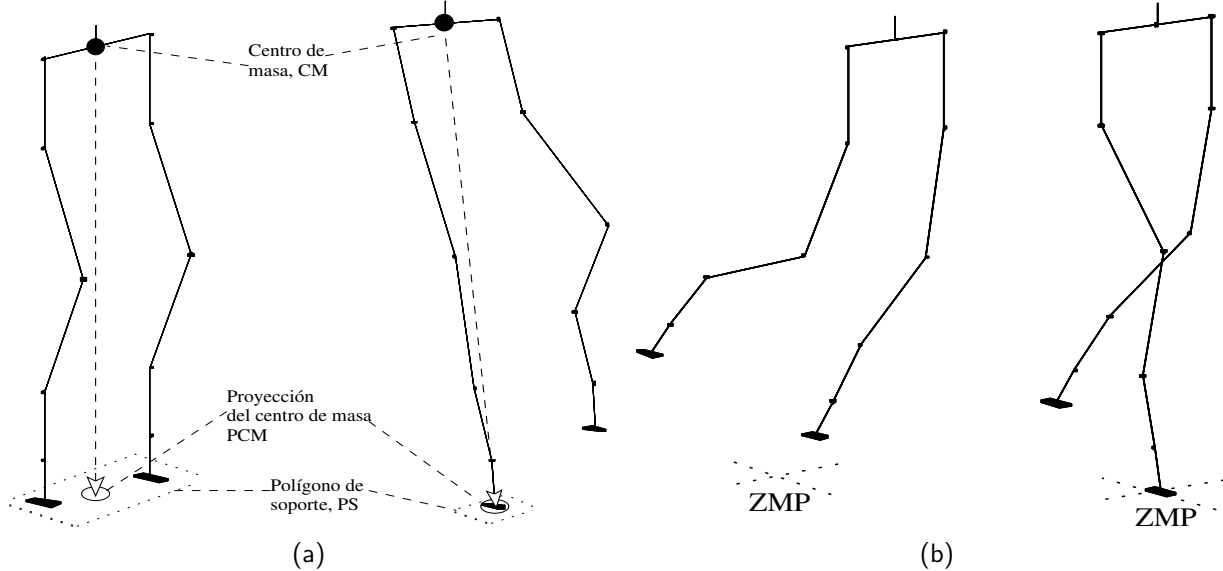


Figura 5.2: CM, PCM, PS, y ZMP

5.4 Aplicación de caminado

Como primera aplicación de *Kokone*, se propone un algoritmo de caminado, basado en los conceptos de la caminata con equilibrio estático. Para lograr que *Kokone* realice un caminado sencillo, la locomoción es realizada repitiendo ocho posiciones estables.

El algoritmo consta de doce posiciones en total: cuatro de inicialización y ocho posiciones del ciclo. La estabilidad en cada etapa está basada en los conceptos de la caminata con equilibrio estático: las localizaciones de la cadera y de los pies están precalculadas para lograr que la proyección del centro de masa se encuentre siempre dentro del polígono de soporte, y así, lograr el equilibrio en el robot.

Las primeras cuatro posiciones se encargan de preparar al robot para iniciar el ciclo de caminado. En la posición uno, se disminuye el centro de gravedad para lograr una mejor estabilización en el robot, las siguientes tres posiciones forman un medio paso con el pie izquierdo, (Figura 5.3). En el ciclo de caminado se repiten las fases de soporte simple y soporte doble: dependiendo en que fase se encuentre, el centro de masase es desplazado hacia el polígono de soporte moviendo la cadera de un lado hacia otro y avanzando. (Figura 5.4). Iterando el ciclo un número de veces deseados se obtiene

el caminado estático.

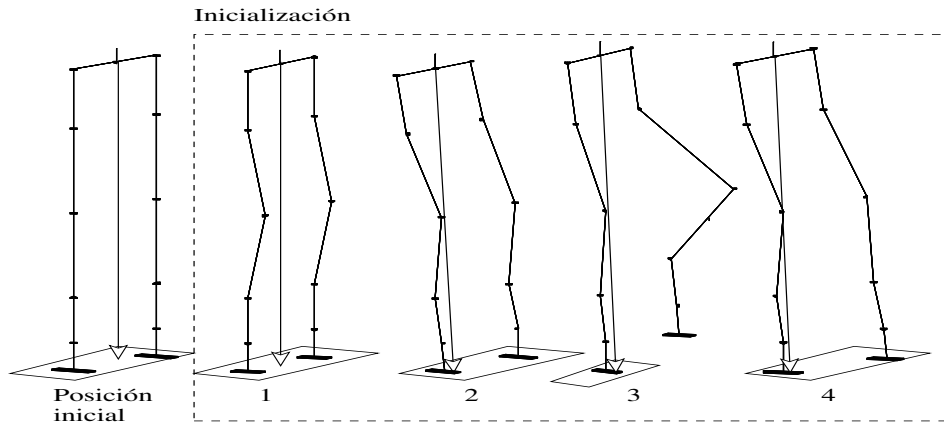


Figura 5.3: Inicialización

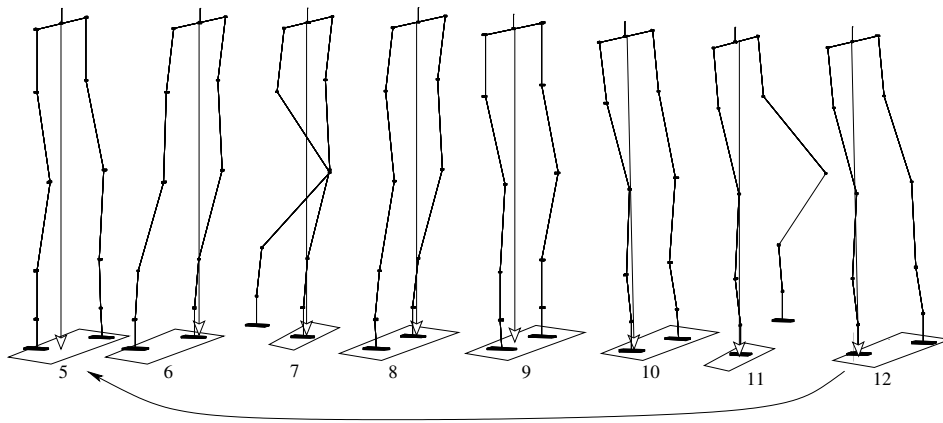


Figura 5.4: Ciclo de caminado

La metodología de control para implementar la aplicación de caminado en *Kokone* se muestran en la Figura 5.5. En cada posición de caminado, la aplicación utiliza las funciones de control de la biblioteca. El algoritmo genético recibe las localizaciones de los pies y cadera, en base el modelo cinemático de las piernas, calcula los valores de θd_i y θi_i $i = 1 \dots 6$ que satisfacen los localizaciones dadas (cinemática inversa). Los ángulos obtenidos y la posición de la cadera son escritos en dos archivos para graficarse y validar los movimientos resultantes. Los ángulos son convertidos al formato propio de la tarjeta de control de los servomotores, para ser enviados a la tarjeta a través del puerto

serie. La tarjeta controladora se encarga de traducir las posiciones calculadas en su respectiva señal PWM correspondiente, y transmitirla a cada servomotor.

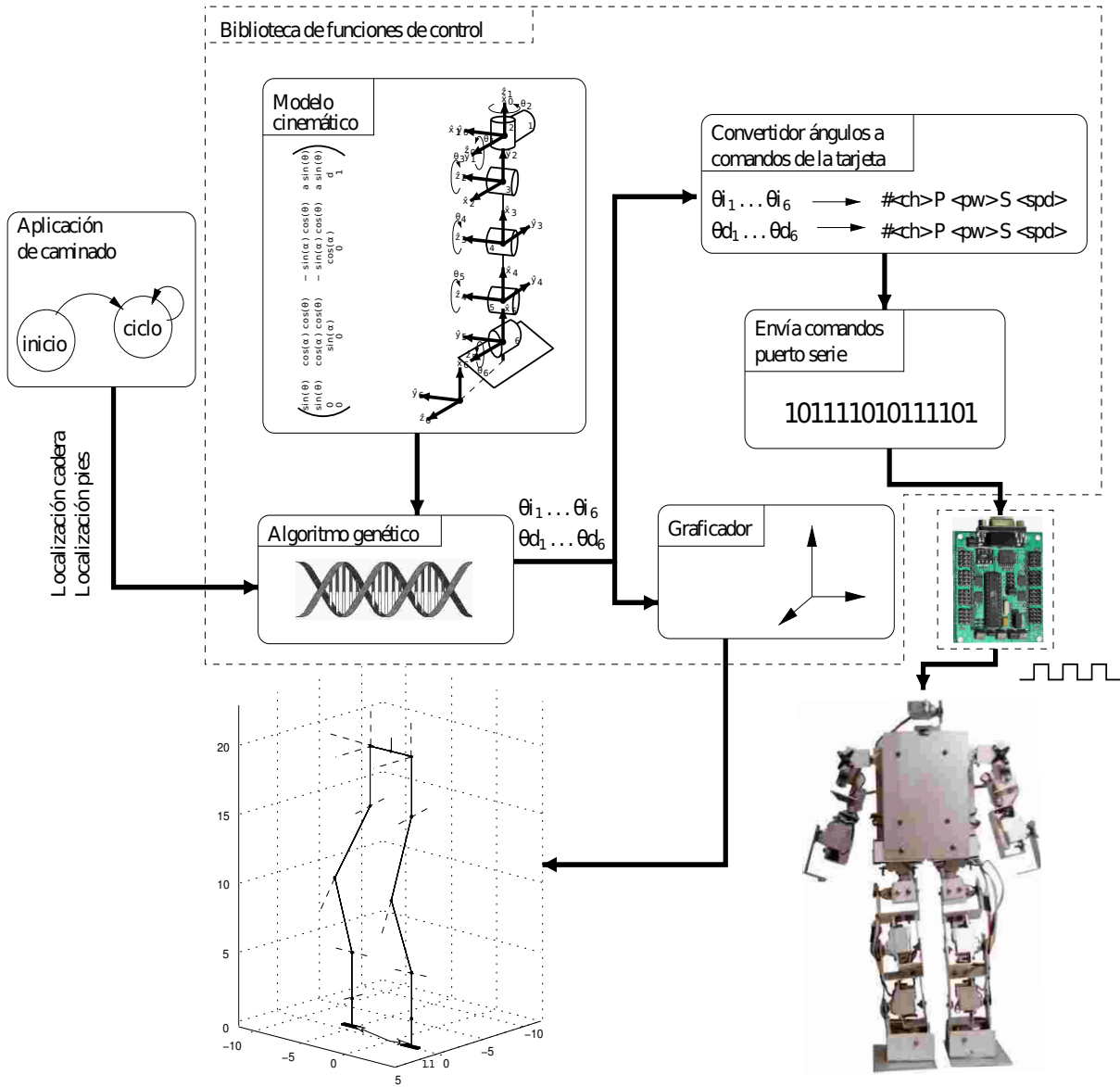


Figura 5.5: Metodología de control

El ciclo de caminado implementado en *Kokone* se muestra en la Figura 5.6, (de la posición cinco a la posición doce).

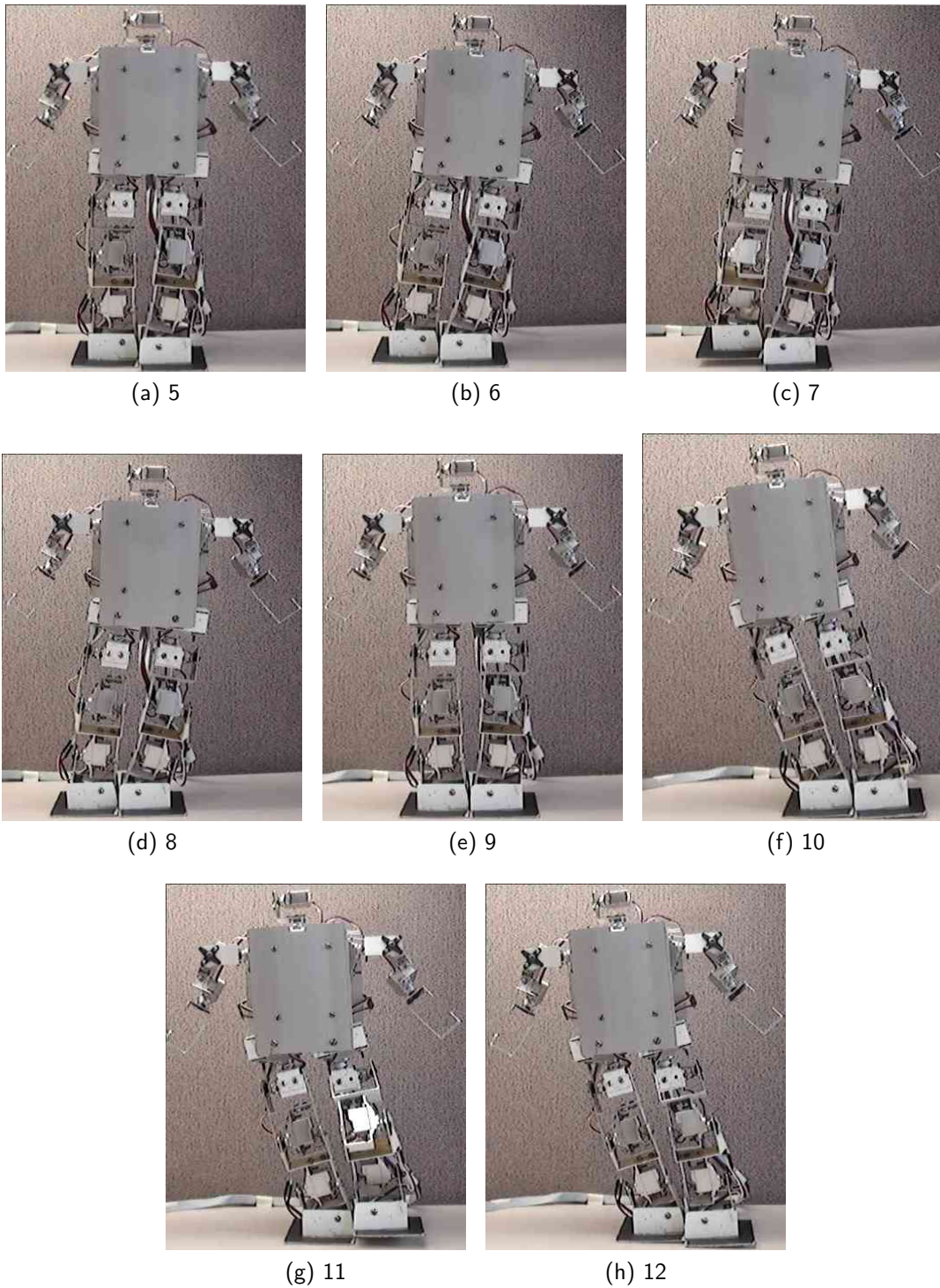


Figura 5.6: Ciclo de caminado implementado en *Kokone*

6

Conclusiones

En un principio la demanda de servicios y productos fue el detonante de la automatización de procesos. Esta situación permitió a los robots explotar su potencial de producción. Hoy en día, los robots están convirtiéndose en una herramienta importantísima para el desarrollo científico y tecnológico de la humanidad. En un futuro no muy distante, será imposible imaginar la vida sin robots, al igual que hoy en día es imposible imaginarla sin computadoras.

Es interesante cómo las primeras civilizaciones construyeron artefactos curiosos con la inquietud de mejorar un procedimiento. Esta misma inquietud llevó a las industrias a implementar robots para aumentar su producción y satisfacer la demanda de la sociedad. Reducir la brecha entre robots y personas es la tendencia actual, para ello, los robots deben ser capaces de adaptarse al ambiente humano y mejorar la calidad de vida de las personas. Sin duda, son los robots humanoides, como asistentes autónomos, los candidatos ideales para esta integración deseada.

6.1 Kokone: plataforma robótica de investigación

Asistir en el desarrollo e investigación de métodos de control que faciliten la enseñanza en robótica humanoide es el objetivo primordial de *Kokone*. La plataforma robótica desarrollada proporciona los mecanismos necesarios de hardware y software para programar aplicaciones más complejas. El esquema general (computadora, robot, enlace serial) aprovecha la flexibilidad y portabilidad del software de la PC, en beneficio de un rápido desarrollo de aplicaciones y facilitar las pruebas de laboratorio.

La configuración de los 22 GDL de *Kokone* permite realizar la mayoría de los movimientos de las extremidades del cuerpo humano. Su diseño mecánico de bajo costo, con materiales asequibles y de fácil ensamble, no requiere constante mantenimiento, además, el diseño de las piezas evita la redundancia para maximizar la funcionalidad de cada una. El diseño también es adecuado como modelo para la fabricación de más robots humanoides. La tarjeta controladora SSC-32 abstrae el control de bajo nivel de los actuadores (señal PWM), y su interfaz serial permite la comunicación con funciones más complejas para el control de los actuadores.

El modelo cinemático de las piernas fue obtenido mediante el método de Denavit-Hartenberg. Este modelo, basado en matrices de transformación homogénea, permite conocer la localización de los pies de acuerdo a los valores de sus articulaciones con respecto a un sistema de referencia fijo. El problema cinemático inverso de *Kokone*, consiste en encontrar los ángulos de las articulaciones según las localizaciones de los pies y de la cadera. Se propuso un algoritmo genético específico como método computacional para resolver la cinemática inversa del robot.

La biblioteca de funciones integra el modelo cinemático, el algoritmo genético, la configuración del puerto serie y los procedimientos necesarios para controlar los movimientos de las extremidades de *Kokone*. Estas funciones están disponibles para cualquier aplicación que implemente estrategias más complejas en el robot.

Como primera aplicación se propuso un algoritmo de caminado basado en los conceptos de la caminata con equilibrio estático. La localización de los pies y cadera son precalculadas con el fin de

mantener siempre el centro de masa dentro del polígono de soporte. Sin embargo, la caminata es un poco lenta, y cada paso lleva varios segundos.

Los resultados de este trabajo, fueron presentados en el 2009 6th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2009) celebrado en la Ciudad de Toluca Estado de México, México, con el artículo "Design of *Kokone*, a small humanoid robot".

6.2 Futuras capacidades y aplicaciones

Kokone está diseñado para extender sus capacidades electrónicas y mecánicas. La cabeza del robot está preparada para realizar movimientos "pan and tilt", por lo que una o dos cámaras pueden ser montadas en ella para proporcionar visión artificial. Los brazos están diseñados para la integración de efectores finales y así manipular objetos pequeños. *Kokone* es capaz de utilizar hasta cuatro sensores externos, como acelerómetros, giroscopios y sensores de presión, que podrán integrarse directamente en la tarjeta electrónica de control SSC-32.

Es posible utilizar nuevas y mejores heurísticas para encontrar las soluciones de la cinemática rápidamente, ésta es una de las ventajas de usar un método de programación no lineal para resolverla. Con el fin de obtener un caminado más ágil y natural, algunas estrategias de caminado pueden ser implementadas, *p.ej.*, realizar un modelo basado en el péndulo invertido para calcular la posición y orientación del cuerpo del robot. Con la ayuda de sensores de presión en los pies, es factible calcular el ZMP y lograr una caminata dinámica. La plataforma es adecuada para experimentar con otras estrategias de control: reconocimiento de objetos, coordinación brazos-ojos y controladores de equilibrio. Para implementar estos métodos, es posible utilizar diversas técnicas computacionales como redes neuronales, lógica difusa y distintas heurísticas.



Biblioteca de control

En este apéndice se muestran las funciones que integran la biblioteca de control. También, se detalla una aplicación que realiza movimientos básicos de las piernas usando estas funciones.

Básicamente, existen dos grupos de funciones: las funciones para resolver la cinemática y las funciones para el control del robot.

A.1 Funciones para resolver la cinemática

```
void multiplica_matrices(double A[][4], double B[][4], double (*C)[4][4]);
```

Realiza la operación $A \times B = C$, donde A, B, y C, son matrices de dimensión 4x4.

```
void matriz_objetivo(double cadera[][4], double pie[][4], double (*obj)[][4]);
```

Calcula la matriz objetivo $A_{mc}^{-1} \cdot A_{DPie}$, de acuerdo a las posiciones de la cadera y del pie, que se proporcionan como parámetros.

```
void genetico(double obj[][4], double (*coordenadas_articulares)[6], int bandera);
```

El algoritmo genético recibe como parámetros la matriz objetivo, un vector donde guarda los valores de los ángulos calculados, y una bandera que indica la pierna a resolver.

```
double funcion_cinematica(double (*q)[6], double obj[][4], int bandera, int bandera2);
```

Esta función es usada por el AG. De acuerdo a las variables solución del individuo (parámetro (*q)[6]), el modelo cinemático y la matriz objetivo, se calculan las aptitudes. El parámetro bandera indica cuál pierna es la que se evalúa y Bandera2 sirve para controlar impresiones de resultados.

A.2 Funciones para la configuración del robot

Esta funciones son utilizadas por un programa denominado "trayectoria", que lee el archivo de ángulos y ejecuta los movimientos resultantes en el robot.

```
int inicializa_robot();
```

Esta función se encarga de configurar el puerto serie de acuerdo a los parámetros mostrados en la Tabla 3.2. Devuelve el descriptor de archivo para el envío de las instrucciones.

```
void carga_posiciones_iniciales(int *posiciones_iniciales_izq, int *posiciones_iniciales_der);
```

Esta función ajusta los motores a su posición inicial.

```
void obten_angulos(char *angulos_pierna_c, int *angulos_pierna_n);
```

Esta función se encarga de leer el archivo de ángulos (soluciones), dividirlos y cargarlos a memoria.

```
void construye_comando(char *comando, int *pier_i, int *pier_d, int *ini_i, int *ini_d, int fase);
```

Esta función traduce los ángulos de ambas piernas a comandos de la tarjeta electrónica.

```
void envia_comando(int descriptor, int longitud, char * comando);
```

Envía el comando final por el puerto serie hacia el robot.


```
char comando[102];
FILE *archivo_angulos;
////////////////////////////////////////////////////////////////////////////////
//Abriendo archivo de angulos
archivo_angulos = fopen("angulos.txt","r");
printf("Abriendo_archivo_de_angulos...\n");

//Abriendo y cargando posiciones iniciales de los motores del archivo
carga_posiciones_iniciales(posiciones_iniciales_izq , posiciones_iniciales_der);
printf("Cargando_posicion_inicial_de_los_motores...\n");

//Inicializando el puerto para enviar los comandos al robot
descriptor_puerto_serial = inicializa_robot();
printf("Leyendo_trayectoria....\n");

//Ciclo para convertir los angulos leidos del archivo a comandos y enviarlos
while(!feof(archivo_angulos))
{
    fgets(angulos_pierna_izq1 ,31 ,archivo_angulos);
    printf("angulos:_%" , angulos_pierna_izq1);
    fgets(angulos_pierna_der1 ,31 ,archivo_angulos);
    printf("angulos:_%" , angulos_pierna_der1);
    obten_angulos(angulos_pierna_izq1 , angulos_pierna_izq2);
    obten_angulos(angulos_pierna_der1 , angulos_pierna_der2);

    construye_comando(comando , angulos_pierna_izq2 , angulos_pierna_der2 , piniciales_i , piniciales_d , fase);
    longitud = strlen(comando);

    printf("comando:_%_%\n" , comando , longitud);
    comando[longitud] = '\r';
    envia_comando(descriptor_puerto_serial , longitud , comando);
    usleep(800000);
}

close(descriptor_puerto_serial);
fclose(archivo_angulos);
return 0;
}
```


Bibliografía

- [1] O. Khatib B. Siciliano. *Handbook of Robotics*. Springer, 2008.
- [2] George Bekey and Junku Yuh. The status of robotics, second part. *IEEE Robotics and Automation Magazine*, 15(1):76–81, March 2008.
- [3] Andrej Kos Damir Omrc̃en, Aleš Ude. Learning primitive actions through object exploration. In *In Proceedings of the 2008 8th IEEE-RAS International Conference on Humanoid Robots*, pages 306–311, December 2008.
- [4] A. Eiben and J. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.
- [5] Ching-Chang Wong et al. Static balancing control of humanoid robot based on accelerometer. In *SICE Annual Conference 2008*, pages 2836 – 2840, August 20-22 2008.
- [6] M. I. C. Dede et. al. Cerberus the humanoid robot: Part i – design. In *In Proceedings of the 2005 Florida Conference on Recents Advances in Robotics*, pages 1–8, May 2005.
- [7] Qiang Huang et. al. Planning walking patterns for a biped robot. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 17(3):280 – 289, 2001.
- [8] Ricardo Tellez et al. Reem-b: an autonomous lightweight human-size humanoid robot. In *In Proceedings of the 8th IEEE-RAS International Conference on Humanoid Robots*, pages 462–464, 2008.
- [9] S. Kagami et. al. Online 3d vision, motion planning and bipedal locomotion control coupling system of humanoid robot:h7. In *In Procceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2557 – 2562, 2002.
- [10] S. Kajita et. al. A realtime pattern generator for biped walking. In *In Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 31–37, May 2002.

- [11] S. Kajita et. al. Biped walking pattern generation by using preview control of zero-moment point. In *In Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pages 1620–1626, September 2003.
- [12] W. Kwon et. al. Biped humanoid robot mahru iii. In *In Proceedings of the 2007th IEEE-RAS International Conference on Humanoid Robots*, pages 583 – 588, 2007.
- [13] Y. Ogura et. al. Development of a new humanoid robot wabian-2. In *In Proceedings of the IEEE International Conference Robotics and Automation*, pages 76 – 81, 2006.
- [14] Y. Sakagami et al. The intelligent asimo: System overview and integration. In *In Proceedings of the IEEE International Conference On Intelligent Robots and Systems*, pages 2478 – 2483, 2002.
- [15] R. Ito H. Minakata and S. Tadakuma. An experimental study of pseudo passive walking robot-discussions about control parameters and expansion to flexible shoe system. In *International Workshop on 2002 7th Advanced Motion Control*, pages 449 – 454, 2002.
- [16] Han Youngjoon Ha Seungsuk and Hahn Hernsoo. Adaptive gait pattern generation of biped robot based on human's gait pattern analysis. *International Journal of Mechanical Systems Science and Engineering*, 1(2):80–85, June 2007.
- [17] Almir Heralić. Design and control of the prototype humanoid robot hr-2. Technical report, Chalmers University of Technology, 2005.
- [18] Jung-Yup Kim Ill-Woo Park and Jun-Ho Oh. Online biped walking pattern generation for humanoid robot khr-3(kaist humanoid robot – 3: Hubo). In *In Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots*, pages 398–403, 2006.
- [19] Denavit J. and Hartenberg R. S. A kinematic notation for lower pair mechanisms based on matrices. *Trans ASME J. Appl. Mech*, 23:215–221, 1955.

- [20] A. R. Khoogar J. K. Parker and D. E. Goldberg. Inverse kinematics of redundant robots using genetic algorithms. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 271 – 276, May 1989.
- [21] Sara McGrath Jacky Baltes and John Anderson. Active balancing using gyroscopes for a small humanoid robot. In *In Proceedings of the 2th International Conference on Autonomous Robots and Agents*, December 13-15 2004.
- [22] Peter Nordin ² Jens Ziegler ¹, Krister Wolff ² and Wolfgang Banzhaf ¹. Constructing a small humanoid walking robot as a platform for the genetic evolution of walking. Technical report, ¹University of Dortmund, Dept. Computer Science, Dortmund, Germany. ²Chalmers University of Technology Dept. of Physics Göteborg Sweden, 2001.
- [23] Kenneth Alan De Jong. *An Analysis of the Behavior of a class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, Ann Arbour. Department of Computer and Communication Sciences, 1975.
- [24] et al. Kenji Kaneko. Humanoid robot hrp-3. In *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept, 22-26 2008.
- [25] E. Delgado L. F. Giraldo and G. Castellanos. Inverse kinematics of a robot arm using genetic algorithms. *Avances en Sistemas e Informática*, 3(1):29–34, June 2006.
- [26] I. L. Salcedo M. S. Dutra and L. M. Prieto. New technique for inverse kinematics problem using simulated annealing. In *International Conference on Enginnering Optimization EngOpt*, 2008.
- [27] E. Morales Manzanares and L. Sucar Succar. (in spanish) los robots del futuro y su importancia para méxico. *Komputer Sapiens*, 1(2):7–12, September 2009.
- [28] Tatsuzo Ishida Masahiro Fujita, Yoshihiro Kuroki and Toshi T. Doi. Autonomous behavior control architecture of entertainment humanoid robot sdr-4x. In *In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 960–967, October 2003.

- [29] Tad McGeer. Passive walking with knees. In *in Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1640 – 1645, 1990.
- [30] T. A. McMahon. *Muscles, Reflexes and Locomotion*. Princeton Univ. Press, 1984.
- [31] Kondo Kagaku's official homepage. <http://www.kondo-robot.com/>. Last visit on August 14, 2009.
- [32] Robonova's official homepage. <http://www.hitecrobotics.com/>. Last visit on August 14, 2009.
- [33] L. Roussel, C. Canudas de Wit, and A. Goswami. Generation of energy optimal complete gait cycles for biped robots. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2036–2041, 1998.
- [34] Mattias Wahde and Jimmy Pettersson. A brief review of bipedal robotics research. In *In Proceedings of the 8th Mechatronics Forum International Conference*, pages 480–488, 2002.
- [35] Krister Wolff and Peter Nordin. Evolution of efficient gait with an autonomous biped robot using visual feedback. In *In Proceedings of the 2nd IEEE-RAS International Conference on Humanoid Robots*, pages 99–106, 2001.
- [36] A. Sano Y. Ikemata and H. Fujimoto. Analysis of stable limit cycle in passive walking. In *SICE 2003 Annual Conference*, volume 1, pages 117 – 122, 2003.
- [37] H. M. Hertunc Z. Bingul and C. Oysu. Applying neural network to inverse kinematics problem for 6r manipulator with offset wrist. In *Adaptive and Neural Computing Algorithms International Conference*, pages 112–115, 2005.