

UNIVERSIDAD AUTÓNOMA DE SINALOA
DOCTORADO EN CIENCIAS DE LA INFORMACIÓN



TELEOPERACIÓN EN LÍNEA DEL ROBOT HUMANOIDE
BIOLOID PREMIUMTM MEDIANTE CAPTURA DE
MOVIMIENTOS

TESIS

como requisito para obtener el grado de
DOCTOR EN CIENCIAS DE LA INFORMACIÓN
presentada por:

DORA AYDEE RODRÍGUEZ VEGA

DIRECTORES

DR. ULISES ZALDÍVAR COLADO
DR. JOSÉ VÍCTOR NUÑEZ NALDA

Culiacán, Sinaloa, México Diciembre del 2016

Resumen

No han sido pocos los investigadores en el área de robótica que buscan desarrollar robots humanoides con la idea de que en algún momento, robots humanoides y humanos puedan trabajar juntos, para ello, el robot debe comportarse de forma similar a los seres humanos; el generar estos esquemas es un tema de interés en la actualidad.

Tareas como aprendizaje por imitación y teleoperación de robots humanoides han sido abordadas por algunos grupos de investigación, utilizando diversos sistemas de captura de movimientos que van desde una cámara detectando marcadores de colores, hasta sistemas ópticos e inerciales.

Se ha logrado imitar la parte alta del cuerpo en línea para robots del tamaño aproximado a un humano con más de 30 grados de libertad; en cuanto a imitación de cuerpo entero implica resolver la estabilidad del robot, dependiendo de la estrategia utilizada para este fin, la imitación puede llevarse a cabo en línea o fuera de línea, sin embargo, la mayoría de los trabajos solo identifica la posición de los pies y calcula una postura estable para realizar la imitación de la zancada.

La selección de la estrategia para resolver la cinemática inversa y la estabilidad es crucial para lograr la teleoperación en línea para posturas y movimientos simples como caminar.

En este trabajo de investigación se propone el uso de resortes virtuales utilizando la metodología de Borst e Indugula para resolver el mapeo cinemático entre el sistema de captura de movimiento y el robot humanoide, así mismo se propone la creación de un ambiente sintético tridimensional que represente el comportamiento dinámico del robot humanoide utilizando el motor de física PhisXTM ya que este puede aportar mayor realismo a la simulación que los simuladores actuales debido a que se ha reportado que es mejor en la detección de colisiones para formas cóncavas. Se utilizan segmentos de captura de movimientos obtenidos de la base de datos de la Carnegie Mellon University para reproducir movimientos en línea en el robot humanoide Bioloid PremiumTM de 18 grados de libertad.

Se observa que la metodología propuesta funciona muy bien para realizar la teleoperación

en línea de cuerpo entero mediante captura de movimientos, sin embargo no de forma estable ya que realiza los movimientos directamente como los realiza el ser humano, si se desea agregar una estrategia de estabilidad, por ejemplo modificar la posición de la pelvis para mantener el ZMP en la zona de estabilidad definida por el polígono de soporte, se modifican las posturas de las piernas por lo que ya no sería imitación de la parte baja del cuerpo; dado que solo se utilizan los dos grados de libertad del hombro para imitación de movimientos en la parte alta, la imitación de movimientos sería muy limitada.

En cuanto al ambiente sintético tridimensional con comportamiento dinámico se presenta un modelo virtual del robot humanoide que proporciona información sobre masa, centro de masa y tensor de inercia de acuerdo a la forma y material de cada una de sus piezas, se compararon estos datos con los obtenidos mediante el software de diseño mecánico SolidWorkTM, cuando las diferencias de éstos datos fueron amplias, se alimentó el modelo propuesto con los datos proporcionados por el software de diseño mecánico; el robot humanoide con comportamiento dinámico puede representar caídas para movimientos inestables, sin embargo, al no considerar la fricción ni el comportamiento real de los motores, el comportamiento del humanoide virtual aún no es igual al real.

Palabras Clave: *Teleoperacion, Modelo Virtual, Comportamiento Dinámico, Robot Humanoide.*

Índice general

1. Introducción	1
1.1. Motivación y antecedentes	1
1.2. Planteamiento del Problema	4
1.2.1. Objetivos	5
1.2.2. Justificación	6
1.2.3. Hipótesis	7
1.2.4. Metodología	7
1.3. Organización de la Tesis	9
2. Trabajos Relacionados	11
2.1. Sistemas de Captura de Movimientos	11
2.2. Mapeo del Movimiento Capturado al Humanoide	18
2.2.1. Cinematica Inversa	18
2.2.2. Redes Neuronales	18
2.2.3. Regresión y aprendizaje	19
2.2.4. Optimización	20
2.2.5. Primitivas de movimiento	22
2.2.6. Modelo Humano Normalizado	23
2.2.7. Resortes Virtuales	23
2.3. Estabilidad	25
2.4. Simulación	28
2.5. Evaluación del Movimiento Realizado por el Humanoide (Cuantitativa y Cualitativa)	32
2.6. Análisis y Perspectivas	33
2.7. Conclusiones	35
3. Equipo Utilizado	36
3.1. Captura de Movimientos	36
3.1.1. Archivo ASF: Definición de Segmentos	38
3.1.2. Archivo AMC: Movimiento del Esqueleto	40
3.1.3. Datos requeridos por el humanoide	41
3.2. Robot Humanoide	42
3.2.1. Descripción	42
3.2.2. Cinemática	49
3.2.3. Articulaciones	50

3.2.4.	Cinemática Inversa	58
3.2.5.	Dinámica	62
3.2.6.	ZMP	69
4.	Resultados	74
4.1.	Modelo Virtual con Comportamiento Cinemático	74
4.2.	Modelo Virtual con Comportamiento Dinámico	77
4.2.1.	Arquitectura del Sistema	78
4.2.2.	Formas	80
4.2.3.	Actor	80
4.2.4.	Estructura Padres e Hijos	81
4.2.5.	Uniones	81
4.2.6.	Parámetros Dinámicos	82
4.2.7.	Resultados	86
4.3.	Teleoperación de Humanoide Virtual con Comportamiento Dinámico median- te captura de movimientos utilizando el modelo de Borst e Indugula	88
4.3.1.	Modelo Borst e Indugula	90
4.3.2.	Resultados	92
4.3.3.	Conclusiones	94
4.4.	Teleoperación en línea de Robot Humanoide mediante el modelo Borst e Indugula	95
4.4.1.	Interfaz del sistema de Teleoperación	96
4.4.2.	Resultados	97
4.5.	Publicaciones	99
4.5.1.	Artículos	99
5.	Análisis de Resultados y Trabajo Futuro	102
5.1.	Análisis de Resultados	102
5.2.	Conclusiones	106
5.3.	Trabajo futuro	107

Índice de figuras

1.1. Arquitectura Modular Propuesta	9
2.1. Sistemas de captura de Movimientos mediante sensores inerciales de la compañía Xsens Technologies TM	12
2.2. Sistema de Captura de Movimientos propuesto por [71].	13
2.3. Sistema de Captura de Movimientos mediante marcadores reflexivos	14
2.4. Procesamiento de Imágenes para Captura de Movimientos [4]	14
2.5. Sistema de Captura de Movimientos Kinect TM [20]	15
2.6. Procesamiento de imágenes para captura de movimiento sin marcas propuesto por Do[14]	16
2.7. Banda de fibra Óptica de la compañía Measurand's Inc. utilizado por Aleotti [27] para capturar movimientos de un brazo.	16
2.8. Plataforma para calcular el ZMP utilizada por Seungsu en [35, 36]	17
2.9. Posiciones de las marcas en el actor y en el androide [12]	20
2.10. Velocidad de una de las uniones, las líneas muestras los límites de cada segmento de movimiento [11]	22
2.11. Modelo Humano Normalizado propuesto por Montecillo [16]	23
2.12. Control de torso de Humanoide mediante resortes traslacionales [44]	24
2.13. Modelo Spring-damper en un roboto virtual SCARA [45]	25
2.14. Control de Torso de Robot Humanoide mediante elementos resorte Amortiguador.a) Elementos Virtuales, b) Configuración detallada de la mano. [46]	26
2.15. Restricciones de contacto con la superficie [5].	27
2.16. Rango de ajuste de la cintura para mantener la estabilidad [75].	28
2.17. Modelo Virtual que no incluye motor de física	29
2.18. Simulación en OpenHRP que no considera autocolisiones [33]	30
2.19. Simulación de objetos cóncavos	30
3.1. Sistemas de captura de Movimientos	37
3.2. Identificación de segmentos del esqueleto[70]	38
3.3. Ejemplo de archivo <i>asf</i>	40
3.4. Ejemplo de archivo <i>amc</i>	40
3.5. Identificación de articulaciones del esqueleto de Captura de Movimiento utilizados para la teleoperación del robot humanoide	42
3.6. Robot Humanoide Bioloid Premium TM en su configuración A.	43
3.7. Unidad de Control CM-530	44

3.8. Posiciones válidas para el motor Dynamixel AX-12 de Robotis TM	49
3.9. Diagrama de flujo para la Unidad de Control CM-530	50
3.10. Articulaciones del Robot	51
3.11. Dimensiones del Robot	51
3.12. Definición de ZMP[103]	71
3.13. Relación entre el CoM y el ZMP	72
4.1. Ejemplos de modelos geométricos utilizados	75
4.2. Modelo Virtual con Comportamiento Dinámico	75
4.3. Diagrama de flujo de la programación del Robot Humanoide para la teleoperación del humanoide virtual mediante manipulación del humanoide real	76
4.4. Comparación entre el humanoide real y el humanoide virtual con comportamiento cinemático	77
4.5. Arquitectura del sistema	79
4.6. Ejemplos de formas	80
4.7. Ejemplos de actores, (A) Torso, (B) Codo, (C) Tobillo1, (D) Tobillo2	81
4.8. Estructura Padres e Hijos	82
4.9. Ejemplos de uniones en PhysX TM : (A) Prismáticas, (B) Esféricas Y (C) Rotacionales	82
4.10. Robot interactuando con esfera	86
4.11. Robot bajo efectos de la fuerza de gravedad	87
4.12. Comportamiento del robot al caer sobre dos esferas	87
4.13. Arquitectura del Sistema	89
4.14. Humanoide Dinámico	94
4.15. Humanoide Cinemático y Dinámico	94
4.16. Comparación entre MOCAP y humanoide dinámico	95
4.17. Interfaz del sistema de Teleoperación	96
4.18. Comparación entre robot virtual y robot real para posición inicial	98
4.19. Comparación de la caída del robot virtual y el real	99
5.1. Posición solicitada al robot virtual que el robot real no puede realizar	103
5.2. El modelo virtual con comportamiento dinámico resuelve las restricciones cinemáticas	104

Índice de tablas

3.1.	Grados de libertad para cada articulación del esqueleto de captura de movimiento.	45
3.2.	Grados de libertad del esqueleto de captura de movimientos utilizados para teleoperar el robot humanoide de 18 grados de libertad	46
3.3.	Especificaciones del motor Dynamixel AX-12 de ROBOTIS TM	46
3.4.	Sección EEPROM de la Tabla de Control del motor Dynamixel AX-12 de ROBOTIS TM	47
3.5.	Sección RAM de la Tabla de Control del motor Dynamixel AX-12 de ROBOTIS TM	48
3.6.	Dimensiones del Robot	52
3.7.	Masa de cada eslabón del Robot Humanoide Bioloid Premium TM	69
3.8.	Centros de Masa para los eslabones del Robot Humanoide Bioloid Premium TM 70	
3.9.	Tensores de Inercia para los eslabones del Robot Humanoide Bioloid Premium TM 73	
4.2.	Comparación de Masas reales y del modelo virtual	83
4.3.	Centros de Masa para las articulaciones del Robot Humanoide Bioloid Premium TM 84	
4.4.	Tensores de Inercia para las articulaciones del Robot Humanoide Bioloid Premium TM	85
4.5.	Comparación entre el movimiento deseado y el obtenido	93

Capítulo 1

Introducción

1.1. Motivación y antecedentes

Investigadores a nivel mundial han trabajado en el desarrollo de robots humanoides con la finalidad de que se asemejen a los humanos y trabajen juntos. Es fundamental para un robot humanoide actuar como un ser humano, para así ser percibido y aceptado de forma natural en un contexto social. Asimismo, proveer a un robot humanoide de la capacidad para realizar tareas de forma similar al ser humano brinda un mayor realismo a la interacción. El generar esquemas para que el robot tenga movimientos similares a los del humano y que interactúe con ellos es, por lo tanto, un tópico de interés actual en el área. La imitación de movimientos es una forma de generar movimiento en robots humanoides de manera estable, en [1] se definen tres niveles de imitación en robótica: Mímica, emulación e imitación de acuerdo a los objetivos y acciones realizadas para lograrlos.

Se habla de mímica cuando se copian las acciones directamente sin analizar o entender el objetivo, si definimos el objetivo como mantener la estabilidad, es posible copiar las acciones (movimientos) sin mantener la estabilidad.

La emulación se enfoca en lograr el objetivo, sin copiar las acciones directamente, si definimos nuevamente el objetivo como la estabilidad, durante la emulación se mantiene la estabilidad aunque las acciones para mantenerla no son copiadas directamente, es decir, se modifican las posturas y/o las trayectorias.

En la imitación se busca copiar las acciones y lograr el objetivo, es decir se buscaría

mantener la estabilidad y que las posturas y/o trayectorias sean copiadas; este nivel es el que se pretende alcanzar en el presente trabajo de investigación.

Cuando un robot realiza movimientos programados a partir de la observación y el aprendizaje de los movimientos realizados por un maestro experimentado (humano), es llamado enfoque de aprendizaje por imitación. Este enfoque es más flexible que el de la pre-programación, en el cual los movimientos programados del robot no consideran el comportamiento humano [2].

En diversos trabajos realizados se considera el aprendizaje por imitación mediante la captura de trayectorias de movimiento del cuerpo humano [3–5]. Kuniyoshi [6] usa reconocimiento visual y análisis de secuencias de movimientos para reproducirlo en robots. En [7] se presenta el aprendizaje de un robot para imitar movimientos que puede reaccionar favorablemente en situaciones desconocidas.

La captura consiste en posicionar marcas en diferentes articulaciones del cuerpo y posteriormente almacenar las trayectorias generadas por estas marcas [3, 5, 8–17]. En cuanto a los sistemas sin marcadores en [18, 19], se analizan siluetas de imágenes para determinar posiciones del cuerpo humano y [4] identifica posición de manos y cabeza por color de la piel, mientras que [20–25] se usa el dispositivo Kinect TM para captura de movimiento.

Para interpretar la posición del sistema de captura de movimientos e introducir estas poses deseadas al robot algunos investigadores resuelven la cinemática inversa mediante la metodología de mínimos cuadrados amortiguados [10, 26], otros autores utilizan estrategias como redes neuronales [12, 27–31], regresión y aprendizaje [32], optimización [14, 33–43] y resortes virtuales [44–48]. Este proceso puede ser realizado fuera de línea, es decir, se realiza la captura de movimientos y al terminar el proceso se realiza el cálculo de los ajustes necesarios y posteriormente, el movimiento se reproduce en los humanoides.

En los últimos años se ha trabajado en la realización de este proceso de captura de trayectorias de movimientos y su imitación por un robot, pero en línea [12, 14, 46], en este nuevo enfoque, el algoritmo para calcular los ajustes trabaja tan pronto llegan las muestras

y entrega los resultados con cierta latencia, la cual es deseable que sea mínima para que no sea percibida por el usuario; estos trabajos consideran solamente el torso y los brazos, por lo que no es necesario tomar en cuenta la estabilidad del robot, lo cual reduce notoriamente el tiempo de cálculo de los ajustes.

En [15, 25, 49, 50], se realiza la imitación de movimiento de cuerpo entero, tomando en cuenta el equilibrio del robot, pero no es en línea, es decir, se hace la captura de movimiento, se hacen los ajustes de estabilidad y en un tiempo posterior se realiza la reproducción de movimientos en el robot, por lo que se prioriza la precisión en los cálculos de ajustes y estabilidad sobre el tiempo requerido para calcularlos.

En [10, 22, 26, 37, 38, 51–53] se mencionan trabajos de imitación de cuerpo entero en línea y tomando en cuenta la estabilidad, sin embargo, no se incluye el caminado, solo se consideran poses, es decir el robot imita los movimientos y puede cambiar de soporte doble a soporte simple manteniendo el equilibrio.

Finalmente, en [16] se realiza imitación de cuerpo entero para posturas en soporte doble y simple de manera estable; cuando se incluye el caminado, la latencia se incrementa cuatro segundos, debido a que es el tiempo requerido por el algoritmo para identificar si se realizó zancada izquierda o derecha y realizar una rutina estable preprogramada para las zancada identificada.

En todos los trabajos mencionados, se habla de imitación de movimientos la cual es solucionada mediante diferentes algoritmos de cinemática inversa y de estabilidad para lo cual se analiza el modelo dinámico del robot; ambas estrategias llevan un tiempo de cálculo considerable lo que aumenta la latencia y en muchos casos impide la teleoperación o imitación de movimientos en línea.

1.2. Planteamiento del Problema

Obtener el modelo cinemático y dinámico para un robot en específico son tareas que tienen cierto nivel de complejidad, realizar la imitación de movimientos mediante cinemática inversa y estabilizar un robot mediante su modelo dinámico implica cierta cantidad de cálculos que puede incrementar la latencia y hasta impedir su realización en línea. Obtener un algoritmo que pueda reducir el tiempo de cálculo para los ajustes requeridos en la imitación de movimientos y que dichos movimientos mantengan el balance en el robot humanoide es deseable.

En el proceso de verificar la imitación de movimientos en el robot, es deseable contar con un ambiente virtual que nos permita comprobar que los datos enviados son correctos y no produzcan, en el peor de los casos, una caída del robot.

En este contexto existen diferentes tipos de simuladores como WEBOTSTM, Gazebo, UBERSIM, UCHILSIM, Microsoft Robotics Developer Studio (MRDS), OpenHRP, entre otros; dichos simuladores (excepto OpenHRP) utilizan una librería que funciona como motor de física, la cual permite detectar colisiones y modelar el comportamiento dinámico del objeto virtual mediante parámetros como forma, masa, centro de masa y tensor de inercia.

Existen varios motores de física entre ellos están Open Dynamics Engine (ODE), Bullet, Vortex, Havok, Newton y PhysX. Boeing [54] y Seugling [55] muestran análisis comparativos indicando que PhysX es el mejor motor de física para formas cóncavas. Esto es importante ya que aporta mayor realismo a la simulación; si se trabaja únicamente con formas convexas la simulación detectaría colisiones cuando visualmente se observa que los objetos aún no colisionan.

De los simuladores mencionados, WEBOTSTM, Gazebo, UBERSIM, UCHILSIM, usan ODE como motor de modelado en física, mientras que MRDS usa PhysXTM, sin embargo, MRSD encapsula la capa del motor de física, lo que hace difícil el acceso a datos del modelo dinámico. Por otra parte en [56], se presenta el desarrollo de HUBOT una plataforma que permite modelar y simular el comportamiento cinemático y dinámico de un

robot, sin embargo, por su reciente creación, aún no esta disponible para su evaluación y por lo tanto no ha sido validado para trabajar con los datos capturados en línea. Por otra parte, OpenHRP desarrolló su propio motor de dinámica, desarrollando las ecuaciones de movimiento y detección de colisiones.

Lo anterior hace evidente la necesidad de una interface que permita monitorear simultáneamente los movimientos humanos capturados, los movimientos seleccionados para que realice el robot en el ambiente virtual, así como los realizados por el robot físicamente.

1.2.1. Objetivos

Objetivo General

Realizar la teleoperación en línea de un robot humanoide Bioloid Premium TM mediante captura de movimientos, incluyendo posturas y caminado frontal y lateral.

Objetivos Secundarios

- Creación de un ambiente sintético tridimensional interactivo que permita representar el comportamiento cinemático y dinámico de robots humanoides basado en el motor de física PhysXTM.
- Utilizar el modelo resorte amortiguador para la imitación de movimientos humanos en el robot humanoide Bioloid Premium.
- Proponer una estrategia de estabilización o balanceo para los movimientos a realizar por el robot humanoide.

1.2.2. Justificación

Las aplicaciones de robots humanoides incluyen entretenimiento [57–59], salud [60, 61], rehabilitación [62] y apoyo para niños con problemas de aprendizaje [63], de socialización [30, 64] así como motivación para mejorar habilidades físicas o verbales [65].

Las características de un robot humanoide pueden resumirse en [66]

1. Deben trabajar en ambientes diseñados para humanos.
2. Deben utilizar herramientas diseñadas para humanos.
3. Deben tener forma similar a los humanos.

La caracterización y control de sistemas humanoides tiene un impacto mas allá de la robótica, puede proporcionar [67]

- Ayuda para entender funciones biológicas del ser humano (biomecánica)
- Herramientas para diseñar máquinas y espacio donde trabajan los humanos (ergonomía)
- Ambientes de simulación para estudiar efectos de alteraciones musculo-esqueléticas.
- Diseñar y estudiar sistemas de rehabilitación.
- Ayuda para realizar animaciones por computadora.

En el futuro, los robots humanoides trabajarán, para y con los humanos, por lo que deben ser capaces de manipular objetos, trasladarse de un punto a otro y responder a estímulos de manera similar a los humanos o mejor.

Es importante remarcar que el uso de robots humanoides se ve impulsado por algunas sociedades en las que el número de jóvenes es cada vez menor con respecto al número de ancianos[68].

Los robots humanoides son un instrumento que ha impulsado la telepresencia, al contar con una entidad que puede hacer cosas por nosotros en lugares remotos, la inteligencia artificial ya que pueden imitar físicamente nuestros movimientos comportamiento y la interacción hombre-máquina, ya que tienen integrados varios sensores que les permiten percibir de una manera compleja [69].

La teleoperación de un robot humanoide permite al ser humano alcanzar ambientes que pueden ser nocivos o bien que están a distancias remotas. El hacer uso de un sistema de captura de movimiento permite que un usuario sin capacitación previa pueda manipular el robot humanoide.

1.2.3. Hipótesis

Es posible realizar teleoperación en línea y estable de un robot humanoide Bioloid PremiumTM mediante captura de movimientos utilizando el modelo Resorte Amortiguador. Así mismo, la creación de un ambiente sintético tridimensional interactivo que incluya un motor de física permitirá representar el comportamiento cinemático y dinámico del robot humanoide Bioloid PremiumTM. Lo que será de utilidad en la verificación de algoritmos de balanceo o estabilidad para imitación y/o teleoperación de movimientos humanos capturados en el robot humanoide Bioloid PremiumTM.

1.2.4. Metodología

A continuación se describe la metodología propuesta para llevar a cabo este proyecto de tesis, definiendo el hardware y software a utilizar y el orden de los procesos a desarrollar. En la Figura 1.1 se presenta la arquitectura modular de la propuesta.

El robot humanoide es el Bioloid PremiumTM con 18 grados de libertad, el sistema de captura de movimientos es óptico con 12 cámaras y 41 marcadores. Toda la programación del sistema se realiza en el lenguaje de programación C++, bajo la plataforma de programación Visual C++ Express edition. Los modelos geométricos del humanoide virtual los provee

la compañía RobotisTM, bajo el formato IGES (Initial Graphics Exchange Specification). Se utiliza el software para intercambio de formatos Deep ExplorationTM, y mediante la herramienta MakeTRI se generan los archivos en su formato final (*tri*). Los archivos en formato *tri* son utilizados por la librería de renderizado gráfico OpenGL para la visualización en la escena del humanoide.

El módulo de Comportamiento Cinemático recibe la posición de los marcadores del sistema de captura de movimientos, las cuales se transmiten al robot virtual, lo que permite que se realice su movimiento con un comportamiento cinemático.

El comportamiento dinámico del robot virtual está basado en el motor de modelado basado en física PhysX de NVIDIATM, el cual resuelve las ecuaciones de movimiento dinámico del robot virtual en el módulo de Comportamiento Dinámico.

El cálculo de las fuerzas necesarias a aplicar a los elementos del robot virtual para que éste alcance las posiciones deseadas es realizado por el módulo de Resorte-Amortiguador, el cual utiliza las posiciones de los modelos dinámicos y cinemáticos del robot virtual para computar dicho cálculo.

Para lograr el equilibrio o estabilidad del robot virtual durante la manipulación, se obtiene un modelo de balance dinámico mediante el módulo de Estabilidad, que utiliza los datos de ZMP y CoM para lograr dicho balance.

Para la captura de movimientos de utilizaron segmentos de video obtenidos de la base de datos de Carnegie Mellon University [70], dicha información representa las poses de los segmentos del esqueleto las cuales son enviadas al módulo de Comportamiento Cinemático para su posterior procesamiento. Los datos ajustados y representados en el robot virtual son enviados al kit de desarrollo (SDK) del robot humanoide (Bioloid PremiumTM), los cuales indican al robot que realice el posicionamiento deseado.

Es necesario aclarar que aunque se usan segmentos de video de captura de movimientos, se utilizan únicamente los datos de un frame a la vez y éstos no reciben algún tipo de preprocesamiento, por esta razón decimos que la teleoperación es en línea.

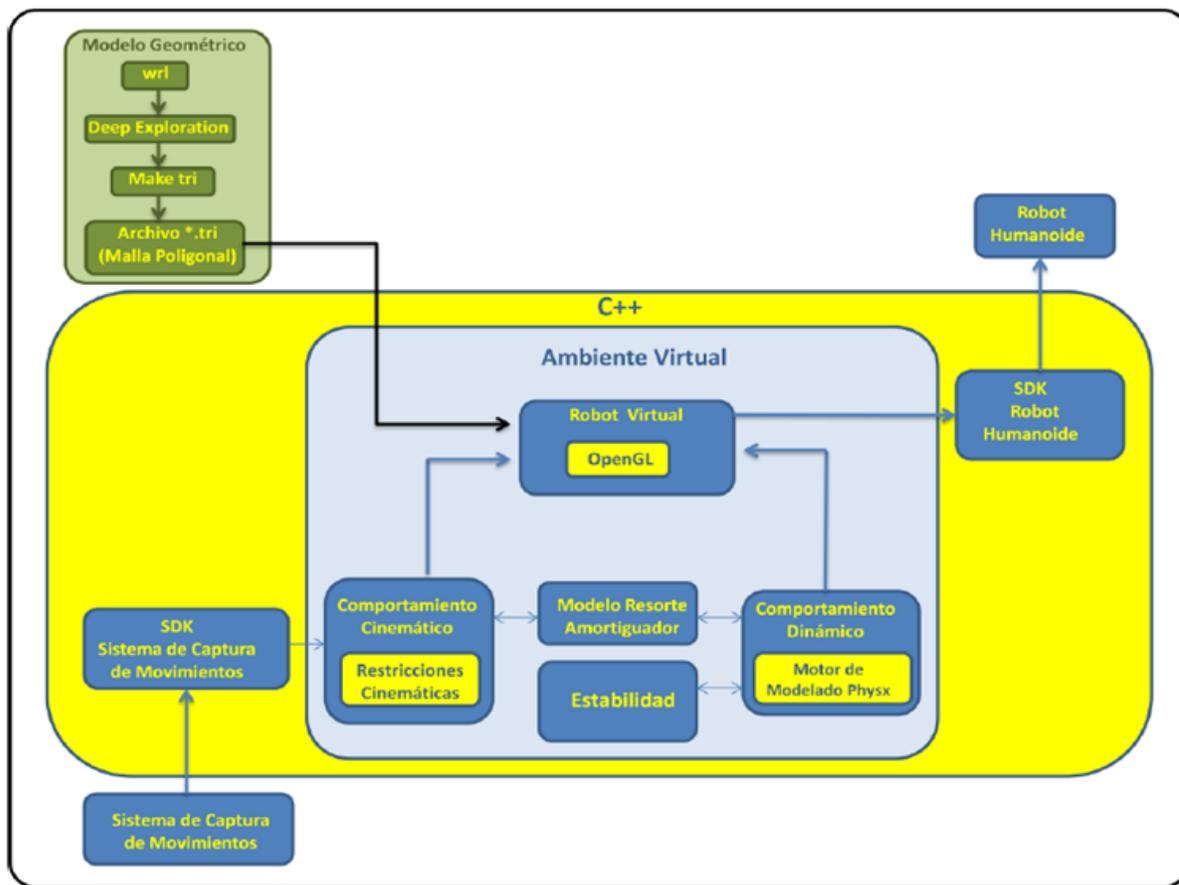


Figura 1.1: Arquitectura Modular Propuesta

1.3. Organización de la Tesis

La tesis está organizada de la siguiente manera: en el Capítulo 2 se presenta el estado del arte sobre teleoperación e imitación de humanoides incluyendo sistemas de captura de movimientos, estabilidad, evaluación del movimiento realizado, así como simuladores utilizados.

En el Capítulo 3 se describe el equipo utilizado para la captura de movimientos describiendo los tipos de archivos así como la interpretación de la información, así mismo, se describe el robot humanoide a utilizar para la teleoperación: microprocesador, motores así como los modelo cinemático y dinámico.

En el Capítulo 4 se presentan los resultados obtenidos durante el desarrollo de esta investigación: primero se muestra el ambiente virtual desarrollado basado en el motor de

física PhysXTM, el cual permite verificar el control de postura así como posibles caídas del robot, después se presenta el modelo resorte - amortiguador propuesto por Borst e Indugula utilizado para la teleoperación, y finalmente se describe el software desarrollado para realizar la teleoperación en línea del robot humanoide mediante captura de movimientos utilizando el modelo Borst e Indugula.

En el Capítulo 5 se muestra el análisis de los resultados mostrados en el Capítulo 4, las conclusiones y posibles trabajos futuros.

Capítulo 2

Trabajos Relacionados

En este capítulo se presenta una revisión de los puntos requeridos para llevar a cabo la teleoperación de humanoides mediante captura de movimiento, posteriormente se muestran los trabajos que han logrado realizar teleoperación de cuerpo entero en línea y de forma estable mediante captura de movimientos para finalmente mencionar los puntos claves en los que todavía hay trabajo por realizar.

Los puntos a analizar fueron:

- a. Sistema de captura de movimientos
- b. Mapeo de los movimientos capturados al humanoide
- c. Estabilidad de los movimientos realizados
- d. Simuladores utilizados
- e. Evaluación del movimiento logrado (cuantitativa y cualitativa)

2.1. Sistemas de Captura de Movimientos

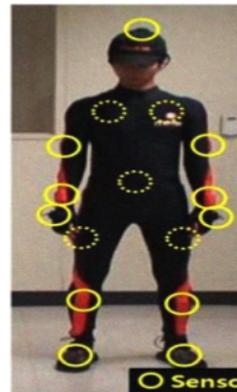
Por tratarse de teleoperación de robots humanoides de cuerpo entero, controlarlo mediante interfaces manuales (tipo joysticks) sería demasiado complejo, por otra parte, si se

agregan interfaces de control por posturas limitan la teleoperación. Algunos investigadores han propuesto el uso de sistemas de captura de movimientos, de tal manera que el operador no requiere entrenamiento específico ni destreza alguna, sólo realiza los movimientos de forma natural.

En cuanto a la captura de movimientos usada en teleoperación están los sistemas inerciales y los ópticos, ambos son invasivos: el operador debe portar en su cuerpo cierto número de sensores o marcadores. Como ejemplo de sistemas inerciales en la Figura 2.1 , se observa el sistema de captura de movimientos inercial MVN de la compañía Xsens TechnologiesTM usado por Sung-Kyun [17] y Koenemann [10, 26] el cual consta de 16 sensores, éstos pueden transmitir su posición a una velocidad de 120 Hz, con la desventaja de un error acumulativo después de la calibración debido al tipo de sensores.



(a) Sensores inerciales usado en [10]



(b) Traje con sensores inerciales usado en [17]

Figura 2.1: Sistemas de captura de Movimientos mediante sensores inerciales de la compañía Xsens TechnologiesTM

En [71] se propone un sistema de captura de movimientos más básico que consiste en 8 sensores para medir variables articulares de las piernas y brazos que se muestra en la Figura 2.2

Otros investigadores usan sistemas de captura de movimientos ópticos; [5, 11–14, 16, 35, 36] usan cámaras infrarrojas para localizar marcadores reflexivos colocados en el cuerpo



Figura 2.2: Sistema de Captura de Movimientos propuesto por [71].

humano, ejemplos de estos sistemas son ViconTM, OptitrackTM, MotionAnalysisTM. La Figura 2.3, muestra estos sistemas en operación; a diferencia de los sistemas inerciales, los sistemas ópticos tiene como desventaja que solo pueden usarse en interiores y en un área limitada por las cámaras, de tal manera que el marcador sea visto por al menos tres cámaras.

Por otra parte Riley y Liu [3, 9] proponen un sistema de visión más sencillo, colocando una cámara en la cabeza del robot y detectan marcadores de colores, el problema de usar solo una cámara y un número reducido de marcadores es que, al realizar ciertos movimientos, la cámara no ve ciertos marcadores resultando en detección de posiciones equivocadas. La oclusión de algunos marcadores en un problema abordado por Han en [72]. Bandera [4] detecta la posición de las manos y la cabeza por el color de la piel, si bien este procedimiento es mucho más rápido que los sistemas con marcadores, el nivel de detalle es mucho menor pues sólo se detectan las manos y la cabeza, resultando en más de una solución posible. La Figura 2.4 muestra el procesamiento de imágenes propuesto.

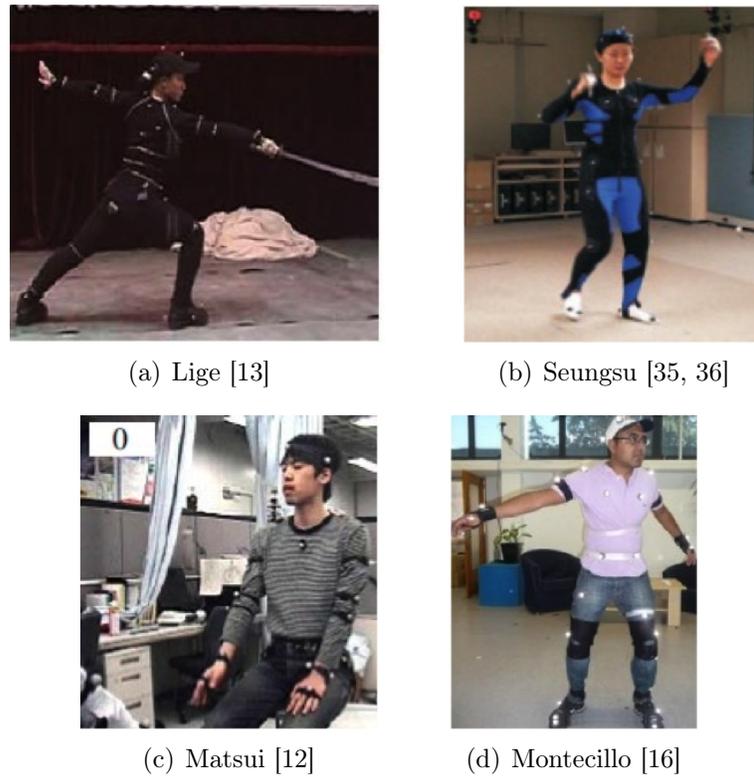


Figura 2.3: Sistema de Captura de Movimientos mediante marcadores reflexivos

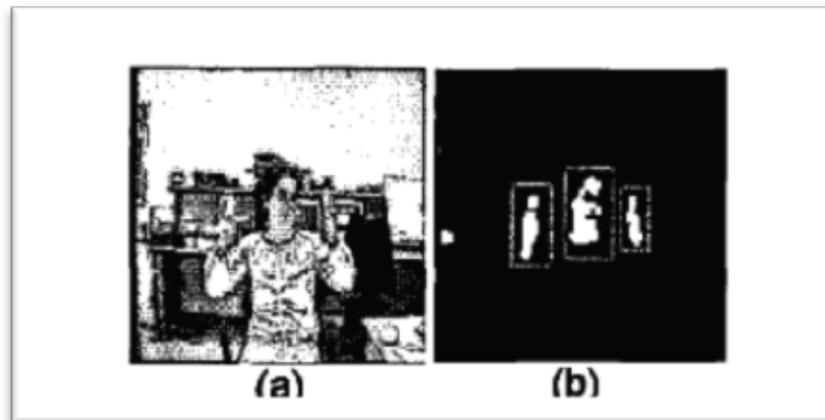


Figura 2.4: Procesamiento de Imágenes para Captura de Movimientos [4]

En [20–25] se utiliza el dispositivo Kinect de Microsoft, el cual cuenta con una cámara RGB y un sensor de profundidad que le permite representar posturas y gestos sin hacer uso de marcadores. Aunque es un equipo de captura de movimiento de bajo costo, los precisión

en movimientos es baja, además ciertas posturas como cruce de pies o manos son detectadas de forma errónea.



Figura 2.5: Sistema de Captura de Movimientos KinectTM [20]

Do [14] propone un sistema combinado con marcas y sin marcas. Para actividades que requieren mucho detalle como manipulación de objetos, se usan marcas; para actividades que requieren mayor velocidad en la teleoperación no se utilizan las marcas. La Figura 2.6 muestra el procesamiento de imágenes para captura de movimientos sin marcas propuesto por Do. Así mismo, hace uso de un motor de mapeo maestro propuesto por Azad en [73] para tomar la información de los sistemas de captura y poder adaptar esta información a robots virtuales y físicos con diferentes configuraciones cinemáticas.

En [27] se utiliza una banda de fibra óptica de la compañía Measurand's Inc. para capturar los movimientos del brazo de un demostrador y reproducirlos en el robot manipulador Pandi-1. La banda mide 1.84 metros y proporciona información de la curvatura en forma de vector en 16 puntos de la banda, para cada punto obtienen vectores de inclinación y rotación a una frecuencia de 110 Hz. La Figura 2.7 muestra el sensor mencionado.

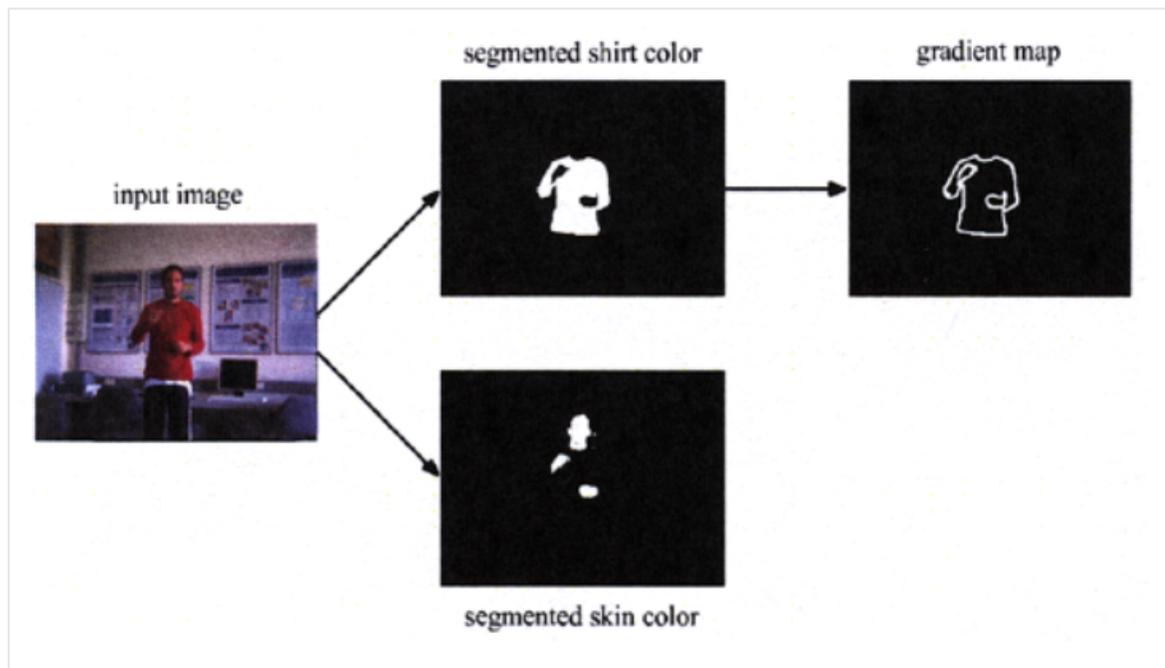


Figura 2.6: Procesamiento de imágenes para captura de movimiento sin marcas propuesto por Do[14]



Figura 2.7: Banda de fibra Óptica de la compañía Measurand's Inc. utilizado por Aleotti [27] para capturar movimientos de un brazo.

Seungsu [35, 36] además de hacer uso del sistema de cámaras y marcadores hacen uso de una plataforma que mide fuerzas y torques contra la superficie lo que hace más rápido y exacto el cálculo de la trayectoria del ZMP del movimiento realizado por el humano, sin embargo el tamaño de la plataforma limita el área de trabajo, por lo que este sólo puede ser utilizado para detectar posturas en soporte doble y simple pero no funciona para caminar,

Figura 2.8.

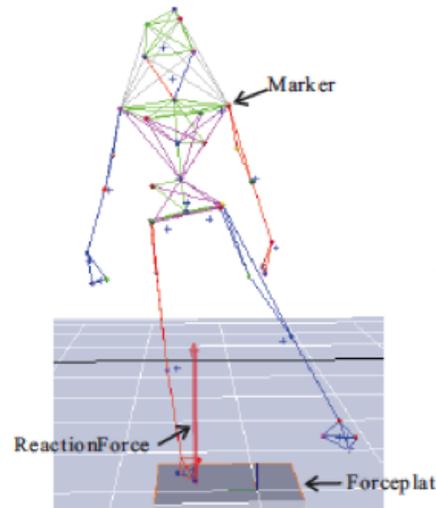


Figura 2.8: Plataforma para calcular el ZMP utilizada por Seungsu en [35, 36]

En conclusión se tienen sistemas de captura de movimientos invasivos y no invasivos, los invasivos requieren que el demostrador use algún tipo de marcador (reflexivos, inerciales, marcas de colores, etc), los no invasivos como el KinectTM no requieren que el demostrador use algún tipo de marcador, sin embargo la información que pueden aportar es muy limitada. Entre los sistemas de captura de movimiento invasivos los más simples utilizan marcas de colores que son identificadas por una cámara, sin embargo se presenta el problema de oclusión de marcas; entre los sistemas más complejos están los sistemas que utilizan sensores inerciales y los ópticos; los primeros pueden utilizarse en zonas abiertas sin limitantes de espacio, su desventaja es que presentan un error acumulativo; los sistemas de captura de movimientos ópticos son más precisos que los inerciales sin embargo, solo pueden utilizarse en un área limitada por las cámaras y con condiciones de iluminación restringidas.

2.2. Mapeo del Movimiento Capturado al Humanoide

Para interpretar la posición del sistema de captura de movimientos e introducir estas posturas deseadas al robot, algunos investigadores resuelven la cinemática inversa y para ello utilizan estrategias como las que se presentan a continuación:

2.2.1. Cinematica Inversa

Koenemann [10, 26] utiliza un sistema de captura de movimientos de sensores inerciales para reproducir movimiento en línea en el robot humanoide NAOTM; para el mapeo del movimiento identifican la posición de los efectores finales (manos y pies) y mediante un algoritmo basado en el método de mínimos cuadrados amortiguados con una matriz de pesos para evitar los límites de las uniones, calcula la cinemática inversa.

Boutin [49] genera locomoción en humanoides basado en captura de movimientos, para ello, se proponen adaptadores autoajustables, es decir, que pueden utilizarse para robots de diferentes masas y tamaños, para producir caminados similares a los del humano de manera estable, se propone un algoritmo en el que transfiere las posiciones angulares del movimiento capturado al modelo cinemático del humanoide y después genera la trayectoria de los pies y del ZMP para mantener el balance y respetar el contacto con la superficie; finalmente se generan las nuevas posiciones angulares mediante cinemática inversa, se realizaron experimentos en simulaciones para los robot HRP y HOAP3, sin embargo, la velocidad del caminado del humanoide resulta menor a la del operador.

2.2.2. Redes Neuronales

En [27] se propone enseñar mediante demostración al robot manipulador Pandi-1, el demostrador usa sensores de fibra óptica y se propone utilizar redes neuronales para aprender a mapear la posición entre el brazo humano y un brazo manipulador, en sus experimentos presentan un clasificador de mínima distancia y una red de retropropagación multicapa,

los valores de entrada son 7 sensores colocados en el brazo de la persona y los valores de salida son tres variables articulares de un robot manipulador, sus resultados indican que para movimientos simples el uso de las redes neuronales propuestas funciona muy bien, pero para movimientos complejos la imitación del movimiento no es satisfactoria por lo que explorarán utilizar la velocidad como variable de entrada.

En [28] se utiliza el sistema de captura de movimientos inercial y el robot NAOTM para realizar la teleoperación, inicialmente se asocia cada articulación con dos sensores del sistema de captura de movimientos, el sistema propone que se realice el aprendizaje de movimientos utilizando una red neuronal de retropropagación para cada articulación, cada red tiene una capa oculta de 20 neuronas, los datos de entrada son las posiciones angulares de cada sensor y las salidas son las posiciones angulares de cada articulación. El error lo presentan como el porcentaje de rango de movimiento del motor, la media es de 5.55%, misma que atribuyen a los movimientos utilizados para entrenar la red.

Matsui [12] genera movimientos naturales en un androide, utilizando un sistema de captura de movimientos ópticos y replicando los movimientos en el androide Repliee Q2s; propone usar una red neuronal con retroalimentación para minimizar la diferencia entre la posición del humanoide y la posición deseada. La red utilizada toma como datos de entrada la posición en x , y y z de 21 marcadores y genera como salida 21 posiciones de marcadores. La Figura 2.9 muestra al actor y al androide con sus respectivas marcas.

El uso de las redes neuronales tiene como desventaja para la teleoperación en línea que requiere entrenamiento previo, y los resultados dependen en gran medida de la realización de este proceso por el demostrador.

2.2.3. Regresión y aprendizaje

Shon [32] presenta el problema de imitación como un problema de regresión, se utiliza el sistema de captura de movimiento óptico de ViconTM y se muestran los resultados con el robot humanoide HOAP2; se propone una solución basada en algoritmos de aprendizaje pro-

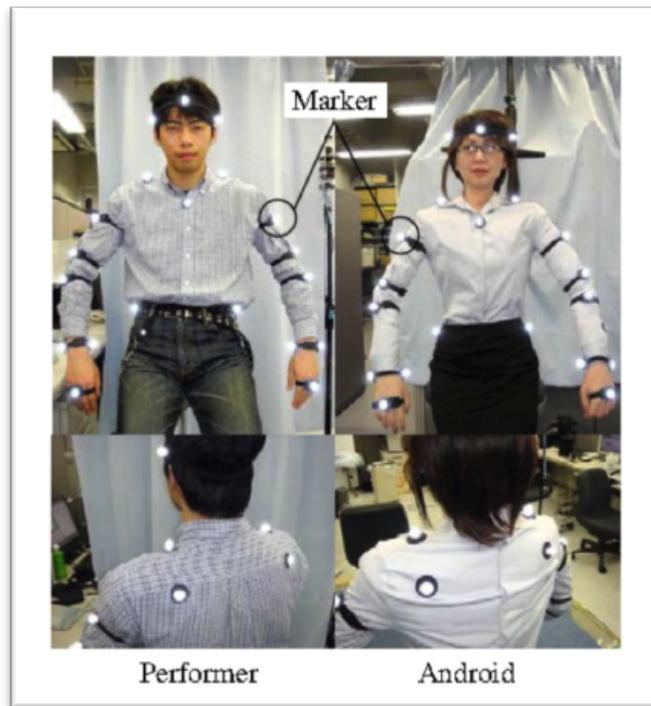


Figura 2.9: Posiciones de las marcas en el actor y en el androide [12]

babilísticos. Después de capturar el movimiento, un resolvidor de cinemática inversa realiza el mapeo de la posición de los marcadores a un modelo de esqueleto humano. Posteriormente se realiza un proceso de entrenamiento basado en algoritmos de aprendizaje probabilísticos con los datos generados por el sistema de captura de movimiento realizados por el humano. El sistema es capaz de interpolar entre posturas de entrenamiento y posturas nuevas, sin embargo, no realiza teleoperación de forma continua y tampoco considera la estabilidad.

2.2.4. Optimización

Suleiman [33] presenta un algoritmo dinámico basado en optimización; usan segmentos de captura de movimiento de la base de datos de la Carnegie Mellon University [70] y realizan los experimentos con el robot HRP en simulación. El algoritmo propuesto permite el cálculo del gradiente con respecto al control analítico de propiedades. Se consideran los límites físicos, es decir, ángulo, velocidad angular y torque del humanoide. Los límites físicos

del humanoide son transformados en restricciones del problema de optimización y la función a ser minimizada es la integral de la diferencia entre los valores angulares del humanoide y los correspondientes en el humanoide virtual.

Ga-Ram [34] propone una combinación de una base de datos y un algoritmo evolutivo para la imitación de movimientos humanos en el brazo del robot humanoide MAHRU. La base de datos se genera offline, se capturan los movimientos y se optimizan para proporcionar torques mínimos, tomando en cuenta diferencias geométricas y dinámicas. Para realizar en movimiento en línea se hace uso de la base de datos generada mediante el algoritmo de búsqueda evolutivo que además actualiza la base de datos, sin embargo no considera autocolisiones ni la estabilidad en los movimientos realizados.

Seungsu [35, 36] realiza imitación de movimientos estables en el humanoide MAHRU, para ello, propone usar una metodología para la parte alta del cuerpo y otra para la parte baja; para la parte alta define un parámetro de escalamiento de los brazos del humano con respecto al brazo del humanoide, y resuelve la cinemática inversa como un problema de optimización, minimizando la diferencia entre las trayectorias del humano y las realizadas por el humanoide, para la parte baja, argumenta que sólo es necesario conocer la posición y orientación del pie para determinar la postura de la pierna.

El trabajo presentado por Yamane en [37, 38] muestra un controlador para imitar el movimiento y mantener la estabilidad, para ello, utiliza los segmentos de captura de movimientos disponibles en la base de datos de la Carnegie Mellon University [70] y realiza los experimentos en el robot SARCOS. El controlador combina el control de balance diseñado para el modelo dinámico y un controlador de seguimiento mediante la ecuación

$$\ddot{q} = \ddot{q}_{ref} + kd(\dot{q}_{ref} - \dot{q}) + kp(q_{ref} - q) \quad (2.1)$$

donde las constantes kd y kp pueden ser diferentes para cada unión.

Posteriormente se utiliza un optimizador para minimizar los errores en centro de presión, torque, aceleraciones en uniones y eslabones de contacto así como el amortiguamiento en

torques y fuerzas de contacto.

2.2.5. Primitivas de movimiento

Nakaoka [11] genera movimientos para un robot humanoide mediante captura de movimientos, para ello define primitivas de movimiento para la parte alta del cuerpo y posturas esenciales para la parte baja del cuerpo; para la parte alta las primitivas se obtienen mediante cinemática inversa y son modificadas para mantener la velocidad de los movimiento mediante la metodología propuesta por Pollard en [74]; para la parte baja propone tres posturas esenciales: soporte doble, soporte simple, y soporte doble bajando la altura de la cintura. La Figura 2.10 muestra los segmentos de movimiento y las posturas donde deben terminar las trayectorias, fue necesario reducir la velocidad de reproducción 2.5 veces. Los resultados fueron verificados mediante el simulador OpenHRP para el robot HRP1S.

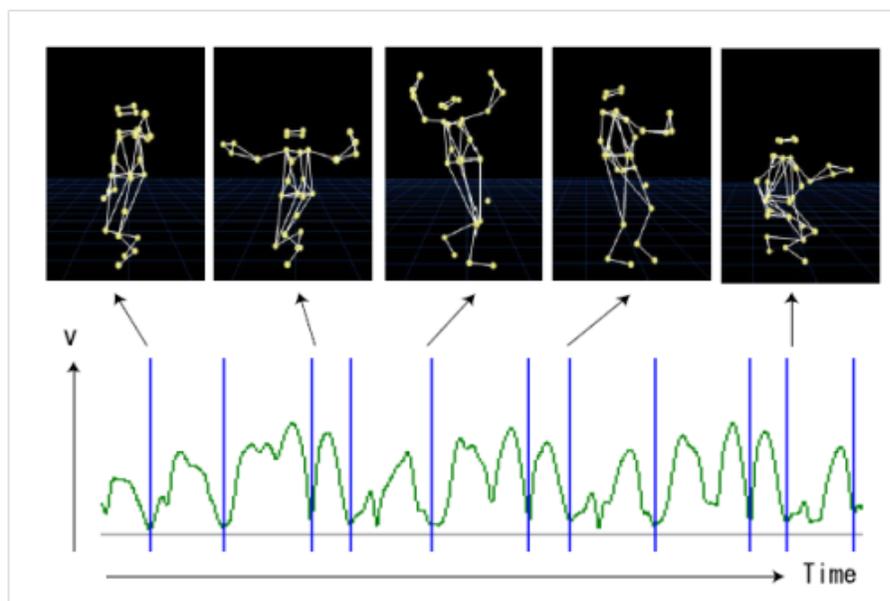


Figura 2.10: Velocidad de una de las uniones, las líneas muestran los límites de cada segmento de movimiento [11]

Lige [13] realiza un procedimiento similar controlando la velocidad de la danza para darle ritmo y evalúan además de la similitud de movimientos en tiempo en que son realizados. Sus resultados, utilizando un sistema de captura de movimiento ópticos y experimentando

en el robot BHR-02 de 32 grados de libertad, muestran que algunas posturas del operador no pueden ser realizadas por el humanoide debido a restricciones mecánicas. El uso de primitivas de movimiento limita la teleoperación a estas primitivas predefinidas.

2.2.6. Modelo Humano Normalizado

Por otra parte Montecillo [16] transfiere movimiento de un humano a un humanoide mediante un sistema de captura de movimientos ópticos y utilizando el robot humanoide HPR2; propone un modelo para representar el movimiento humano, basado en el argumento de que usar la trayectoria angular de cada marca sería muy lento para reproducir los movimientos en línea, por lo que propone un Modelo Humano Normalizado basado en planos para representar posturas que proporcionen información de manos, torso, cabeza, cintura, y pies, como se muestra en la Figura 2.11, facilitando así transferir movimientos a la cadena cinemática del humanoide.

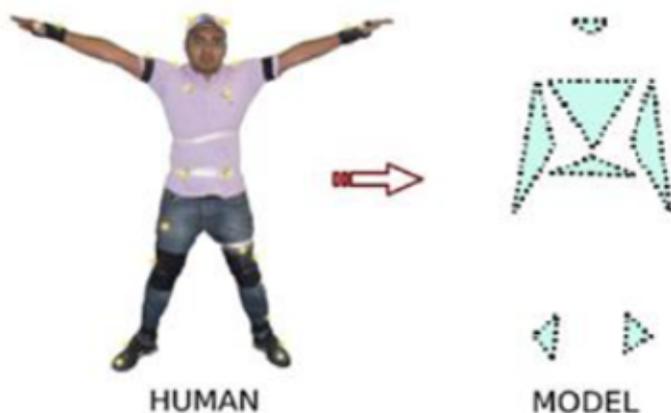


Figura 2.11: Modelo Humano Normalizado propuesto por Montecillo [16]

2.2.7. Resortes Virtuales

Ott, Zaldívar y Sung [44–46], proponen un método que funciona en línea para la manipulación de objetos. Las trayectorias capturadas de la posición deseada se pasan a un

controlador tipo spring-damper (resorte-amortiguador); el controlador genera la fuerza virtual en dirección a la posición deseada, la cual debe ser transformada debido a la dinámica del robot.

Los componentes de tipo spring-damper crean fuerzas cuando el componente virtual interactúa con el robot. Esta estrategia es compacta y no requiere mucha computación. Las fuerzas de los resortes permiten una simplificación en la simulación de la dinámica del robot, de manera que el movimiento real del robot puede ser generado en base a este controlador manejando efectivamente fricciones y otras cuestiones dinámicas. El controlador no requiere manejar la cinemática inversa; en esta estrategia el punto crítico son las oscilaciones producidas en el movimiento realizado por el robot debido a los parámetros del spring-damper. Ott [44] utiliza un sistema de captura de movimiento óptico y reproduce movimientos en la parte alta de un robot humanoide de 38 grados de libertad. Zaldívar [45] presenta un modelo basado en resorte-amortiguador para lograr el comportamiento dinámico y estable de un robot virtual, además muestra una comparación de estabilidad entre dos tipos de resortes: lineal y torsional demostrando que el resorte lineal mantiene un comportamiento dinámico más estable que el torsional, lo anterior mediante la interface háptica Phantom OmniTM. Sung [46] controla la parte alta del robot MAHRU mediante el sistema de captura de movimientos inercial. Las figuras 2.12, 2.13 y 2.14 muestra los modelos propuestos.

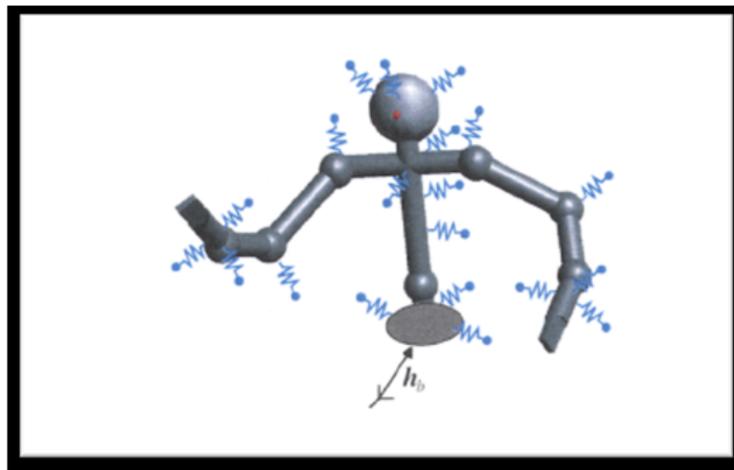


Figura 2.12: Control de torso de Humanoide mediante resortes traslacionales [44]

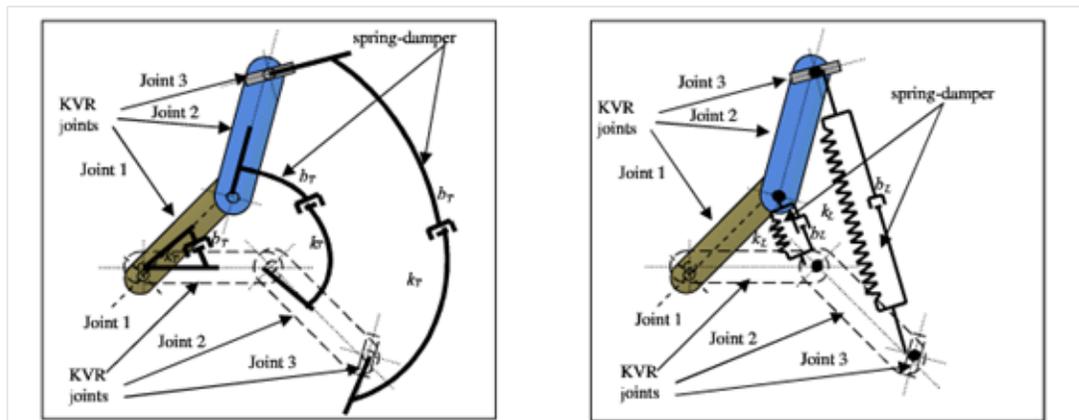


Figura 2.13: Modelo Spring-damper en un roboto virtual SCARA [45]

2.3. Estabilidad

Ya que se cuenta con el mapeo cinemático, el siguiente paso es trabajar en la estabilidad del robot, es decir, que el robot copie los movimientos del demostrador manteniendo el balance.

Xiaojun [5] afirma que las restricciones de contacto con la superficie son cruciales en la locomoción de humanoides, debido a que las extremidades del robot son diferentes a las del humano a imitar, si estas restricciones no se cumplen pueden suceder tres tipos de errores: *trampling*, *slumping* y *glissading*.

Para caso de cambio de soporte doble a simple, el pie de soporte debe estar en la superficie con la posición y orientación deseada; en el caso de *trampling* el pie de soporte está en el aire, como se muestra en la Figura 2.15 inciso a; por el contrario, si el pie de soporte está bajo la superficie tampoco se cumple la restricción, esto se muestra en el inciso b de la Figura 2.15 y se le denomina *slumping*; en ambos casos estas posturas no pueden ser reproducidas por el humanoide de manera estable. Durante la fase de soporte doble los pies deben permanecer fijos sobre la superficie con la posición y orientación deseada, si el controlador utilizado envía posiciones y/o orientaciones variables no se cumple esta restricción esto se observa en el inciso c y se le denomina *glissading*.

Por otra parte Qiang [75] afirma que la posición de la cintura tiene un mayor efecto en

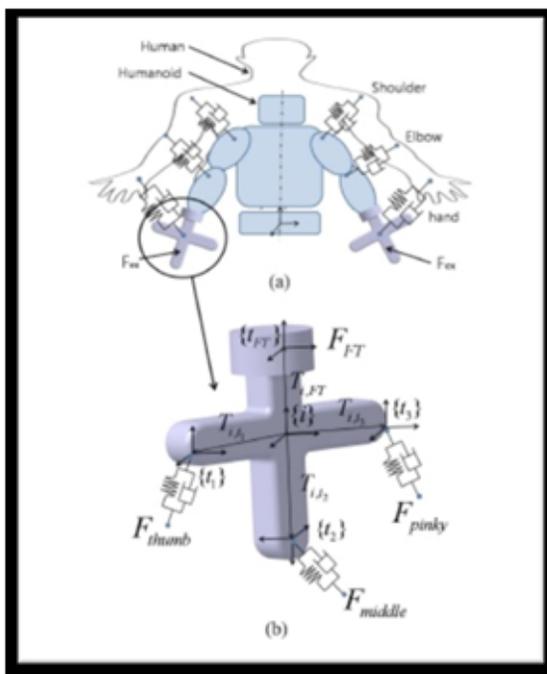


Figura 2.14: Control de Torso de Robot Humanoide mediante elementos resorte Amortiguador. a) Elementos Virtuales, b) Configuración detallada de la mano. [46]

la estabilidad que la posición de los pies y define una región de ajuste de posiciones de la cintura, como se muestra en la Figura 2.16

Controlar la trayectoria del ZMP (punto de momento cero) dentro de la región permitida permite enviar al robot movimientos y/o posturas que lo mantengan estable. La región permitida está definida por el polígono que forman los pies en doble soporte y en la planta del pie en soporte simple. Esta estrategia es utilizada por Nakaoka[11] y Xiaojun[5], en la imitación de danzas japonesas offline y por Suleiman [33] en la imitación de cuerpo entero offline. Seungsu [35, 36] hace uso de los datos de fuerza, torque y centro de masa para modificar la trayectoria de la pelvis para así satisfacer el criterio del ZMP y mantener la estabilidad. Montecillo [16] planea la trayectoria del centro de masa para satisfacer el criterio del ZMP desarrollado por Kajita [76]. Koenemann [10] controla la trayectoria del centro de masa para lograr la estabilidad en posturas en imitación en línea con el humanoide NAOTM.

Por otra parte Nunez [77] presenta una propuesta para controlar la postura de un humanoide basado en fuerzas inerciales, que permite realizar diferentes tipos de locomoción como

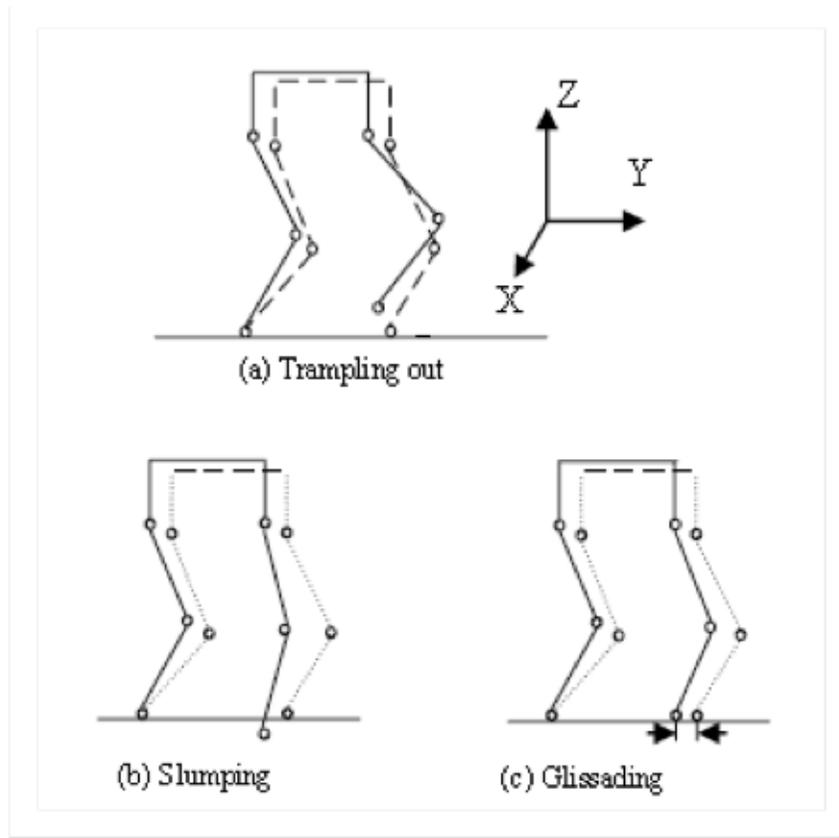


Figura 2.15: Restricciones de contacto con la superficie [5].

caminar y correr de forma estable. La generación de movimiento se basa en la posición de las extremidades que no están en contacto con el suelo y en la generalización de las fuerzas inerciales permitiendo controlar la trayectoria del ZMP alrededor del CoM para que el movimiento sea estable. El controlador de posturas está basado en la ecuación de Lagrange de movimiento que permite relacionar la fuerza inercial y la aceleración angular de cada grado de libertad. Dividir el modelo en extremidades que no están en contacto con el suelo permite encontrar los ángulos de soporte de la extremidad de soporte que compensa el movimiento del resto del cuerpo. La idea básica del controlador es encontrar los ángulos de la pierna de soporte para obtener la fuerza externa deseada y compensar el movimiento deseado de la parte alta del cuerpo. Un trabajo similar se presenta por Kajita [78] pero éste usa velocidad en lugar de aceleración.

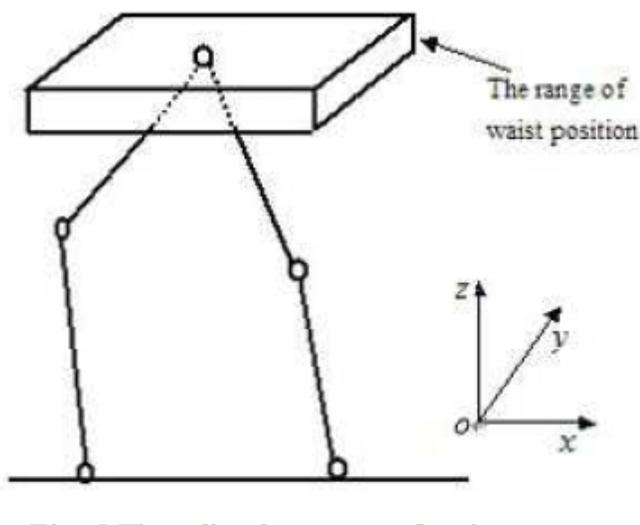


Figura 2.16: Rango de ajuste de la cintura para mantener la estabilidad [75].

2.4. Simulación

Antes de enviar al robot los movimientos que ya pasaron por controladores cinemáticos y dinámicos es necesario verificarlos en un simulador. Se han desarrollado simuladores de humanoides que sirven como modelos de pruebas, mismos que además, podrían ser utilizados para la depuración de rutinas a realizar por estos robots.

Aunque el campo de modelado dinámico y simulación ha avanzado mucho en las últimas décadas en control de movimientos complejos de cuerpo entero de robots humanoides presenta retos adicionales a los simuladores. Las características más importantes de un simulador son: estabilidad, velocidad de respuesta, precisión y exactitud en el reporte de contacto [79]

Un modelo virtual que solo hace uso de una librería de gráficos, puede mostrar el comportamiento del robot, es posible indicar trayectorias y analizarlas, sin embargo, durante colisiones y autocolisiones los objetos se traslapan, es decir, si se presenta un objeto en el ambiente virtual el robot, el eslabón del robot puede sobreponerse visualmente al objeto sin indicar la colisión, en caso de que la colisión sea entre dos eslabones del robot, éstos se sobreponen visualmente, además el robot se presenta flotando en el ambiente virtual, no tiene masa, por lo que no es posible analizar comportamiento debido al peso o a la inercia al implementar las trayectorias. En [80] se presenta un modelo virtual para el robot Bioid

PremiumTM basado únicamente en la librería de gráficos. La Figura 2.17 muestra un modelo virtual que no incluye motor de física, se observa que permite autocolisiones. En la Figura 2.18 se muestran resultados de la imitación de movimiento en el simulador OpenHRP mostrada en [33] que no consideran las autocolisiones.

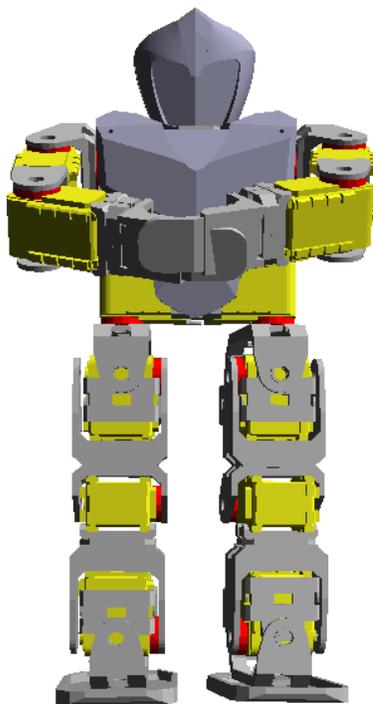


Figura 2.17: Modelo Virtual que no incluye motor de física

Un motor de física es una librería que añade al modelo virtual un comportamiento dinámico, es decir, permite detectar colisiones y modelar el comportamiento dinámico del objeto virtual mediante parámetros como forma y material (densidad). Durante el diseño de controladores en bípedos un motor de física nos permite analizar si la trayectoria producirá una caída en el robot.

Existen varios motores de física entre ellos están ODE, Bullet, Vortex, Havok, Newton y PhysX.

En Boeing [54] y Seugling [55] se muestran análisis comparativos indicando que PhysX es el mejor motor de física para formas cóncavas. Esto es importante ya que aporta mayor

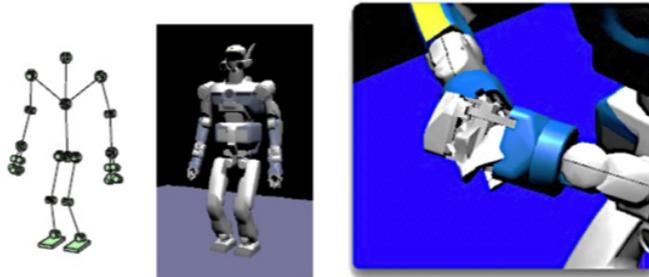


Figura 2.18: Simulación en OpenHRP que no considera autocolisiones [33]

realismo a la simulación, si se trabaja únicamente con formas convexas la simulación detectaría colisiones cuando visualmente se observa que los objetos aún no colisionan. La Figura 2.19 muestra dos objetos con formas cóncavas, para los motores de física que no soportan objetos cóncavos, se indicaría una colisión, sin embargo, PhysxTM no indica la colisión.



Figura 2.19: Simulación de objetos cóncavos

PhysX en su versión 2.3 presenta una herramienta llamada *pmap* utilizada para la detección de colisiones en mallas triangulares cóncavas.

En este contexto existen diferentes tipos de simuladores:

WEBOTSTM es un simulador de robots comercial, proporciona algunos modelos de robots y permite modelar otros, los robots son modelados mediante VRML y los controladores se programan en C++ o Java[81]. En [82] se presenta un ejemplo del uso de este simulador para simular el comportamiento del robot WABIAN.

SimRobot [83], desarrollado por la Universidad Bremen, soporta diferentes tipos de robots y ambientes modelados en XML, con facilidad para programar controladores.

UCHILSIM [84] desarrollado por la Universidad de Chile diseñado para la Liga de cua-

drúpedos de la Robocup. Los robots son modelados en VRML.

UBERSIM [85] desarrollado por Carnegie Melon University modela los robots mediante clases.

Gazebo[86] es un simulador que fue diseñado para analizar estrategias de trabajo colaborativo entre robots.

También hay simuladores para robots específicos como Virtual RE [87] desarrollado por Singapore Polytechnic, basado en UberSim y diseñado para el robot Robo-erectus Junior-BX y iCub simulator [88] para el robot iCub.

Los simuladores mencionados basan su comportamiento dinámico en el motor de física ODE.

Por otra parte ADAMS [89] es una poderosa herramienta comercial para simulación dinámica de múltiples cuerpos rígidos ampliamente usada en industria y algunas universidades, pero genera la cinemática de forma numérica.

OpenHRP desarrolló su propio motor de dinámica basado en restricciones, desarrollando las ecuaciones de movimiento y detección de colisiones, aunque todavía no incluye sensores [90].

Robotics Developer StudioTM [91] basa su comportamiento dinámico en el motor de física PhysX, sin embargo, sólo para un conjunto definido de robots.

Yichao et al. [56] desarrollaron HUBOT una plataforma que permite modelar y simular el comportamiento cinemático y dinámico de un robot, sin embargo, aún no está disponible para su evaluación y por lo tanto no ha sido validado para trabajar con los datos capturados en línea.

Dallali [92] propone ROBOTRAN, un simulador dinámico para el robot COMAN, el cual genera las ecuaciones dinámicas de forma simbólica.

Vonasek [93] propone un simulador con comportamiento dinámico utilizando estructura modular, hace uso de Blender 3D para la visualización.

En conclusión no se ha desarrollado un simulador o modelo virtual con comportamiento

dinámico utilizando el motor de física PhysX, que cómo mencionan Boeing [54] y Seugling [55] presenta un mejor comportamiento para formas cóncavas.

2.5. Evaluación del Movimiento Realizado por el Humanoide (Cuantitativa y Cualitativa)

Ya que se ha realizado la teleoperación con movimientos de forma estable, se verifica la similitud del movimiento realizado con el movimiento deseado. Esta evaluación puede realizarse de forma cuantitativa o cualitativa; la primera se refiere a datos numéricos que permitan evaluar la similitud del movimiento realizado, la cualitativa se enfoca al parecido que observan las personas entre el movimiento realizado por el humano y el realizado por el humanoide.

Para la evaluación cualitativa, Xiaojun [5] propone una función de similitud basada en la trayectoria de los ángulos de unión. Qiang [75] propone una función de similitud que además de la trayectoria de los ángulos de unión incluye la posición de los efectores finales. Lige [13] propone agregar la velocidad a la evaluación de la similitud en movimientos, así como el tiempo en que se realizan los movimientos. Harada [94] propone medir las diferencias angulares de las variables articulares, otorgando cierto peso a los grados de libertad que tengan más influencia en la imitación.

En cuanto a la forma cualitativa el concepto de *embodiment*, se refiere al nivel de presencia que experimentan las personas al interactuar con el robot. Hay trabajos de investigación que evalúan la naturalidad de los movimientos realizados por el humanoide. Sakamoto [95] evalúa el parecido con el ser humano, naturalidad, capacidad de respuesta, contacto visual durante la teleoperación del androide Geminoid HI-1 como medio de telecomunicación. Zuher [96] propone que las personas que observen la imitación del movimiento la califiquen como: mala, pobre, suficiente, buena y excelente.

2.6. Análisis y Perspectivas

La imitación del movimiento ha sido motivo de estudio de algunos investigadores que han realizado trabajos para imitar movimiento offline de la parte baja del cuerpo como los presentados en [49, 50], imitando locomoción o como el trabajo mostrado en [11] realizando imitación del cuerpo entero para reproducir danzas japonesas.

También se proponen estrategias para lograr imitación en línea, para la parte alta del cuerpo como los presentados en [44, 46, 97], donde se propone hacer uso de controladores tipo spring-damper (resorte-amortiguador) ya que es una estrategia compacta y que no requiere mucha computación; por otra parte [34] propone hacer una base de datos para los movimientos y un algoritmo de búsqueda evolutivo.

En cuanto a imitación de movimientos de cuerpo entero en línea los trabajos más destacables son los de Yamane [37, 38], Montecillo [16], Koenemann [10, 26], Gams [51] y Vuga [52]

En [37, 38] se propone un controlador para imitar el movimiento manteniendo la estabilidad. Este enfoque no requiere procesamiento offline, pero solo funciona para posturas en soporte simple o doble, es decir no puede imitar caminado.

En [16] también se presenta imitación en línea de cuerpo entero para posturas y caminado, sin embargo, debe esperar cierta cantidad de muestras para identificar y planear el caminado, por lo que el caminado no es imitado, el tiempo de espera es de 4 segundos en adición a la latencia producida para realizar la imitación de postura.

El trabajo mostrado en [10, 26] realiza imitación en línea con latencia perceptible, se considera la estabilidad para posturas en soporte simple y doble mediante el balanceo del centro de masa. No se incluye el caminado.

En [51] se realiza imitación de cuerpo entero para el robot COMAN utilizando KinectTM para captura de poses, si la proyección del CoM en la superficie está dentro del polígono de soporte realiza la imitación, si se compromete la estabilidad, se prioriza la estabilidad corrigiendo el CoM de la postura a imitar, de la misma manera lo propone Vuga en [52]

utilizando un controlador para la parte alta del cuerpo que se imita directamente cuando el robot está en soporte doble, el controlador de la parte baja imita directamente cuando no está comprometida la estabilidad, cuando el ZMP sale del polígono de soporte se realiza una suave transición entra la posición deseada e inestable a una posición estable.

La teleoperación o imitación de robots en línea de manera estable es un tema de investigación poco abordado, los retos que se han presentado son los siguientes:

A. LATENCIA

Es el tiempo que le lleva al sistema realizar la posición deseada, aunque se ve afectada por el sistema de captura de movimiento, la mayor parte del tiempo de latencia es producida por los controladores cinemático y dinámico, destaca el uso del modelo resorte amortiguador por ser una estrategia compacta para realizar el mapeo cinemático, que utilizada en conjunto con el motor de física PhysX considera las colisiones y autocolisiones.

B. PRECISIÓN

Es que tan parecido es la posición deseada a la real, hay diferentes formas de medirla, Xiaojun [5] y Qiang [75] proponen funciones de similitud basado en posiciones angulares y efectores finales y Harada [94] propone otorgar cierto peso a la variables articulares que tengan más influencia en la imitación. Lige[13] propone agregar la velocidad como variable de similitud.

C. ESTABILIDAD

Las estrategias de estabilidad afectan fuertemente la latencia y la precisión, por lo que debe seleccionarse cuidadosamente. La mayoría de los trabajos basan sus controladores de estabilidad en la trayectoria del ZMP, modificando la posición de la pelvis, Nunez [77] propone un controlador para la postura de un humanoide tomando en cuenta las fuerzas inerciales, lo que permite incrementar la velocidad de los movimientos a estabilizar, sin embargo las ecuación son complejas del orden O^n .

D. SIMULADORES

Por otra parte, el contar con una interfaz gráfica para monitorear la imitación de movi-

mientos humanos en el robot humanoide, que incluya comportamiento dinámico, permitiría verificar diferentes estrategias de control, observando en la interfaz no sólo parámetros matemáticos, sino que además se observa si la estrategia de control mantiene o no al robot en balance, así como las colisiones o autocolisiones en el ambiente virtual. Los simuladores actuales utilizan el motor de física ODE que según Boeing [54] y Seugling [55] aporta menos realismo para formas cóncavas que PhysX.

2.7. Conclusiones

La teleoperación de robots humanoides con fines sociales, médicos o de servicios es una tarea que apenas inicia, se cuenta con trabajos de teleoperación como los realizados por Yamane [37, 38], Montecillo [16], Koenemann [10], Gams [51] y Vuga [52], sin embargo es necesario integrar actividades como caminar, subir escaleras, correr y saltar todo ello de forma natural y estable para que estos robots puedan integrarse a la sociedad e interactuar con los humanos. Así mismo es necesario contar con interfaces gráficas más realistas, es decir, que incluyan comportamiento dinámico, lo que permitiría trabajar en el desarrollo de las estrategias de control aún sin contar con el robot físico.

Capítulo 3

Equipo Utilizado

3.1. Captura de Movimientos

Para la captura de movimientos se utilizó la base de datos del Laboratorio de Captura de Movimientos de la Universidad Carnegie Mellon [70], dicha base de datos cuenta con captura de movimientos para diferentes personas, y en diferentes situaciones capturados con el sistema VICONTM.

El sistema de captura de movimientos utilizado, VICONTM, utiliza 12 cámaras infrarrojas MX-40, cada una es capaz de procesar imágenes de 4 megapíxeles a una velocidad de 120 Hz, las cámaras se localizan en un área rectangular de aproximadamente 3 x 8 metros, todo movimiento humano a capturar debe ser realizado dentro de este espacio. El operador humano usa un traje especial de color negro al cual se le adhieren 41 marcadores reflexivos. La Figura 3.1 muestra la colocación de las cámaras y el uso de los marcadores reflexivos.

Los datos procesados por las cámaras son enviados a una computadora y con el software de sistema de captura de movimientos se obtienen dos tipos de archivos: posición de marcadores y movimiento de esqueleto.

Los archivos de posición de marcadores, indica la posición de cada marcador en el espacio tridimensional y corresponde al usuario interpretar la posición de cada marca así como la relación con el resto de marcadores.

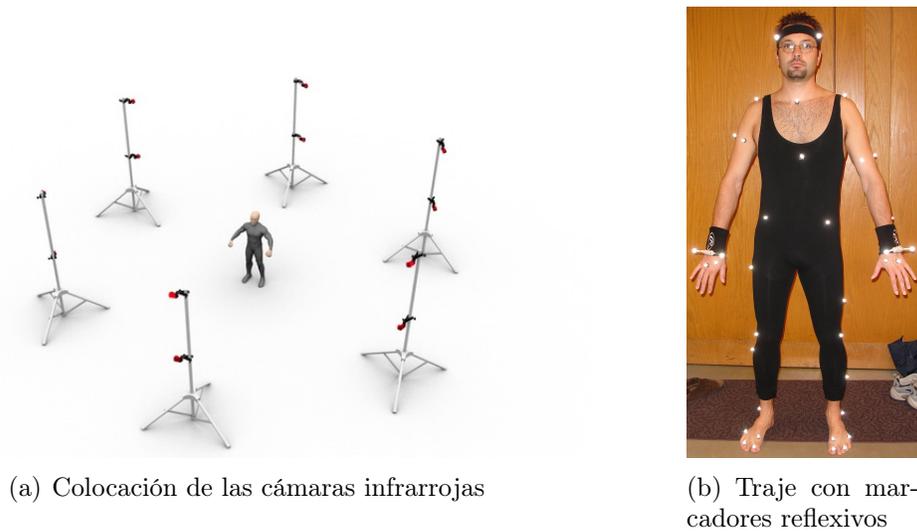


Figura 3.1: Sistemas de captura de Movimientos

Los archivos de movimiento del esqueleto proporcionan información de la formación del esqueleto: segmentos, uniones, medidas, grados de libertad así como las transformaciones matemáticas necesarias para la interpretación de la información. Así mismo proporciona información del movimiento realizado indicando la posición de cada segmento para cada frame.

El sistema $VICON^{TM}$ permite diseñar esqueletos para la captura de movimientos, sin embargo, la base de datos consultada usó los definidos por $VICON^{TM}$ por lo que en este trabajo nos limitamos a usar la información proporcionada por los archivos de movimiento de esqueleto.

La compañía $VICON^{TM}$ seleccionó el formato Aclaim de la compañía del mismo nombre, como el formato de salida para el movimiento del esqueleto proporcionada por su software.

El formato Aclaim usa dos archivos, el primero describe la cinemática del esqueleto en un archivo *asf* (Aclaim Skeleton File); el segundo archivo describe el movimiento del esqueleto en un archivo *amc* (Aclaim Motion Capture data).

Existe también el formato *bvh* (BioVision Hierarchical Data) de la compañía BioVision, muy utilizado en animación, en un solo archivo se indica la estructura del esqueleto y el movimiento.

3.1.1. Archivo ASF: Definición de Segmentos

El archivo *asf* describe como están conectados los segmentos del esqueleto y sus grados de libertad, está representado jerárquicamente con un nodo raíz y varios nodos hijos. La Figura 3.2 muestra los números de indentificación así como las etiquetas asignadas a cada segmento, el nodo de color azul, es el que se identifica como nodo 0 corresponde al nodo raíz, del cual se desprenden 3 cadenas hijas: pierna izquierda, pierna derecha y torso.

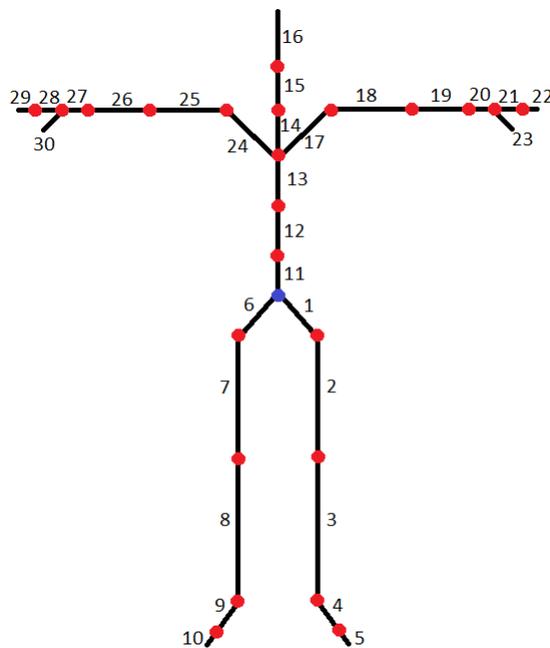


Figura 3.2: Identificación de segmentos del esqueleto[70]

En el archivo cada palabra clave inicia con " : ",

:version Indica la versión del esqueleto

:name Nombre del Esqueleto

:units Unidades a utilizar para masa, longitud y rotación

:documentation Comentarios sobre el archivo

:root Define el nodo raíz, es el nodo principal en la jerarquía definida, no contiene información de longitud ni de la dirección del marco local. Para éste nodo se definen *axis*, *order*, *position* y *orientation*. La primera define el orden de rotación para el nodo raíz, *order* define los canales de movimiento que serán aplicados al nodo raíz, así como el orden en que aparecerán en el archivo *amc*. Las palabras *position* y *orientation* son seguidas de un vector indicando la posición y orientación original del nodo raíz.

:bonedata En esta sección se describen cada uno de los eslabones del esqueleto, para cada eslabón se definen los siguientes parámetros encabezados y finalizados por las palabras reservadas *begin* y *end* respectivamente.

id Número de indentificación único

name Nombre único, es utilizado para definir la jerarquía utilizada en el archivo *amc*

direction Es la dirección del marco local del eslabón padre, indica la dirección del eslabón con respecto al marco del padre.

length Longitud del eslabón.

axis Indica la rotación del marco local con respecto al marco del mundo, los datos de rotación indicados en el archivo *amc* son con respecto a estos marcos locales.

dof Degrees of freedom, indica los grados de libertad posibles de cada articulación.

limits Indica los límites de cada canal de movimiento, para cada canal se indica tanto el límite inferior como el superior

La relación padre-hijo entre los eslabones del esqueleto está definido en la sección *hierarchy* al final del archivo, cada línea indica primero el nodo padre y luego los nodos hijos. En la Figura 3.3 se muestra un ejemplo de archivo *asf*.

```

# AST/ASF file generated using VICON BodyLanguage
# -----
:version 1.10
:name VICON
:units
  mass 1.0
  length 0.45
  angle deg
:documentation
  .ast/.asf automatically generated from VICON data using
  VICON BodyBuilder and BodyLanguage model FoxedUp or BRILLIANT.MOD
:root
  order TX TY TZ RX RY RZ
  axis XYZ
  position 0 0 0
  orientation 0 0 0
:bonedata
begin
  id 1
  name lhipjoint
  direction 0.566809 -0.746272 0.349008
  length 2.40479
  axis 0 0 0 XYZ
end
begin
  id 2
  name lfemur
  direction 0.34202 -0.939693 0
  length 7.1578
  axis 0 0 20 XYZ
  dof rx ry rz
  limits (-150.0 20.0)
          (-70.0 70.0)
          (-60.0 70.0)
end
begin
  id 3
  name ltibia
  direction 0.34202 -0.939693 0
  length 7.49137
  axis 0 0 20 XYZ
  dof rx
  limits (-10.0 170.0)
end
:hierarchy
begin
  root lhipjoint rhipjoint lowerback
  lhipjoint lfemur
  lfemur ltibia
  ltibia lfoot
  lfoot ltoes
  rhipjoint rfemur
  rfemur rtibia
  rtibia rfoot
  rfoot rtoes
  lowerback upperback
  upperback thorax
  thorax lowerneck lclavicle rclavicle
  lowerneck upperneck
  upperneck head
  lclavicle lhumeral
  lhumeral lradius
  lradius lwrist
  lwrist lhand lthumb
  lhand lfingers
  rclavicle rhumerus
  rhumerus rradius
  rradius rwrist
  rwrist rhand rthumb
  rhand rfingers
end

```

(a) Definición de los eslabones del esqueleto

(b) Jerarquía

Figura 3.3: Ejemplo de archivo *asf*

3.1.2. Archivo AMC: Movimiento del Esqueleto

Al capturar el movimiento, cada una de las cámaras preprocesa las imágenes y las envía a la computadora donde se une toda la información proporcionada por las cámaras y se generan los archivos del esqueleto y del movimiento realizado.

El archivo *amc* es un archivo de texto que contiene información sobre el movimiento realizado, por un esqueleto definido en un archivo *asf* específico.

El archivo inicia indicando el archivo *asf* utilizado, así como las unidades para la rotación, posteriormente presenta una lista de los eslabones definidos en el archivo *asf* mostrando en cada renglón un eslabón y su información: posición y orientación para el nodo raíz y orientación en cada eje de rotación indicado para cada eslabón. Es importante notar que las rotaciones son con respecto al marco local y con la jerarquía padres e hijos definida en el archivo *asf*. Para cada frame se presenta de forma consecutiva una lista de segmentos con la información. La Figura 3.4 muestra un ejemplo de archivo *amc*.

Figura 3.4: Ejemplo de archivo *amc*.

```

#!OML:ASF H:\Terrain\Patient Classification 1\PLAYGROUND\JustinFriday\JustinFriday.ASF
:FULLY-SPECIFIED
:DEGREES
1
root 9.37216 17.8693 -17.3198 -2.01677 -7.59696 -3.23164
lowerback 2.30193 -0.395121 1.17299
upperback 0.0030495 -0.462657 2.70388
thorax -1.27453 -0.231833 2.13151
lowerneck -9.32819 -3.76531 -6.70788
upperneck 27.8377 -3.2335 -3.01318
head 10.556 -2.55728 -0.318388
rclavicle 3.64024e-015 -6.75868e-015
rhumerus -29.5133 -11.7797 -80.4307
rradius 21.1829
rwrist -7.55893
rhand -17.4806 -21.0413
rfingers 7.12502
rthumb 8.77158 -50.8391
lclavicle 3.64024e-015 -6.75868e-015
lhumerus 17.2039 -14.515 62.7889
lradius 136.231
lwrist 10.1195
lhand -37.631 -17.4438
lfingers 7.12502
lthumb -10.6834 12.2646
rfemur -0.629535 4.65229 22.5467
rtibia 26.4457
rfoot -15.2124 -9.97437
rtoes 3.93605
lfemur 4.00236 1.20472 -13.8412
ltibia 20.088
lfoot -16.1868 6.57726
ltoes -4.61789
2
root 9.37285 17.8666 -17.3192 -2.06376 -7.58832 -3.1009
lowerback 2.29991 -0.349181 1.09181
upperback 0.0947876 -0.407398 2.60055

```

El archivo *amc* proporciona un total de 62 datos para cada frame, estos datos corresponden a 30 articulaciones y cada articulación tiene de 1 a 3 grados de libertad.

3.1.3. Datos requeridos por el humanoide

Como se muestra en la Figura 3.2, los archivos de movimiento, para el archivo *asf* seleccionado nos proporcionan información de 30 eslabones y un nodo raíz, el mismo archivo *asf* nos indica los grados de libertad para cada eslabón, en total el archivo de movimiento proporciona 62 datos para cada frame.

El robot utilizado cuenta únicamente con 18 grados de libertad, por lo que es necesario definir la información del archivo *amc* que se utilizará. La Figura 3.5 presenta los grados de libertad de cada articulación del esqueleto de captura de movimiento utilizados para la teleoperación del robot humanoide, (marcados de color azul).

La Tabla 3.1 presenta los grados de libertad para cada segmento del esqueleto mientras la Tabla 3.2 indica los grados de libertad seleccionados para utilizar en la teleoperación del

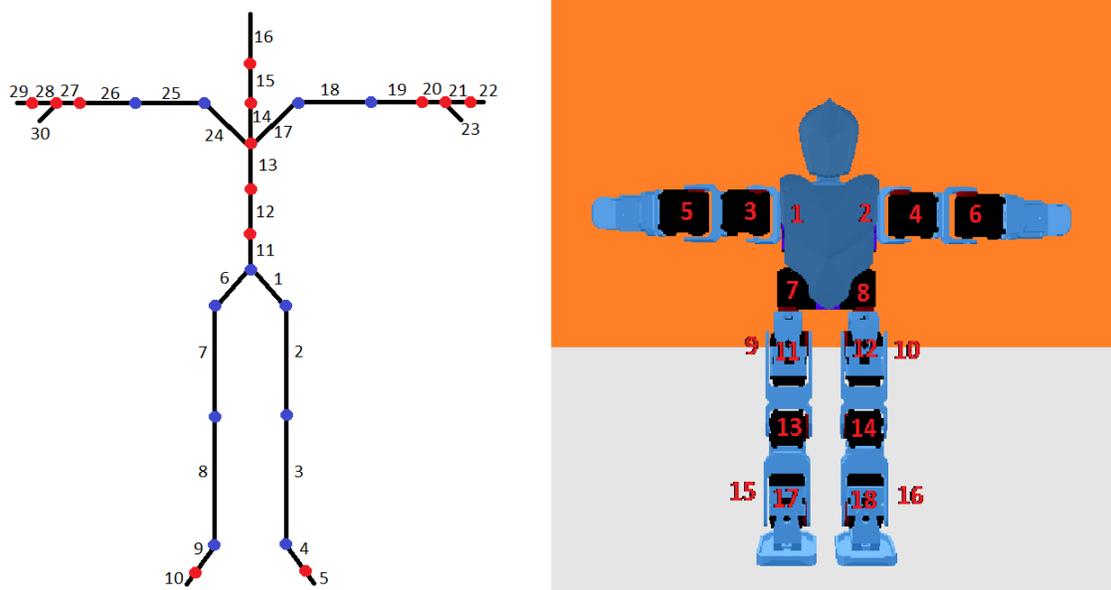


Figura 3.5: Identificación de articulaciones del esqueleto de Captura de Movimiento utilizados para la teleoperación del robot humanoide

robot humanoide. Se realizó una adaptación de 30 a 18 grados de libertad.

3.2. Robot Humanoide

3.2.1. Descripción

El robot a teleoperar es el humanoide Bioloid PremiumTM de la compañía RobotisTM, en su configuración A, contiene una unidad de control denominada CM-530 basada en el microcontrolador ARM Cortex, 18 motores dynamixel AX-12, pesa 1.7 Kg. y mide 39.7 cm. La Figura 3.6, muestra la imagen del robot, los 18 grados de libertad se distribuyen seis en cada pierna y tres en cada brazo [98].

Unidad de Control

La unidad de control puede ser programada mediante el software del fabricante (ROBOTIS) o en C++, el archivo ejecutable es cargado en la unidad de control CM-530 por



Figura 3.6: Robot Humanoide Bioid PremiumTM en su configuración A.

comunicación serial, mediante el autoprogramador precargado en la memoria del microcontrolador del CM-530.

La Figura 3.7 muestra la unidad de control CM-530, se observan los conectores para las cadenas margaritas de los motores, sensores (giroscopio), comunicación inalámbrica (ZIG-BEE), LEDs y botones de control, etc.

La unidad de control es operada por un microcontrolador STM32F103RE de la línea ARM de CORTEX, el microcontrolador de 32 bits cuenta con 512K de memoria Flash, trabaja a 72 MHz, comunicación USB, serial, i2c, CAN, 11 Timers, 3 ADC de 12 bits[99].

Motores

Los motores que utiliza el robot humanoide especificado son Dynamixel AX-12A de la marca RobotisTM, la Tabla 3.3 muestra las características del robot [98].

La unidad de control se conecta a los motores en cadena margarita, cada motor tiene un número de identificación lo que nos permite acceder a su tabla de control para leer y escribir información sobre el estado del motor, dicha tabla tiene una sección de memoria EEPROM y una RAM, a los datos de la primer sección se tiene acceso al momento de configurar el motor, el dato de ID es un ejemplo de éste tipo de información, todos los motores deben

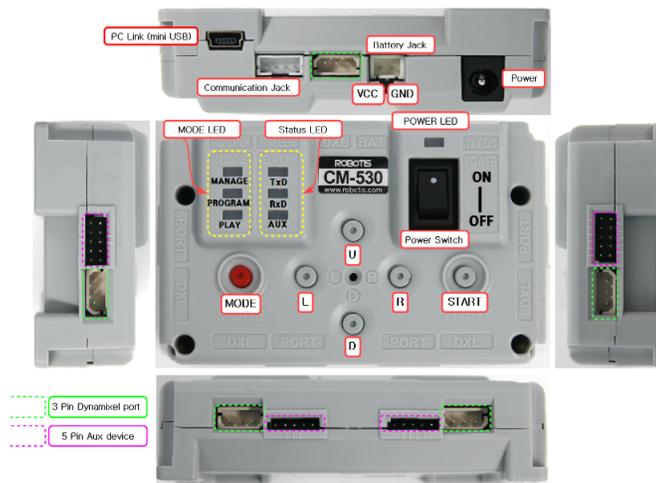


Figura 3.7: Unidad de Control CM-530

configurarse inicialmente para identificarlos al momento de programar las posiciones deseadas. La Tabla 3.4 muestra los datos de la sección EEPROM del motor AX-12 [98].

La sección RAM de la Tabla de Control muestra datos actuales del motor, Posición, velocidad y temperatura actual son ejemplos de datos de lectura, la Tabla 3.5 muestra la sección RAM de la Tabla de Control del motor AX-12 [98].

La posición actual de cada motor es un dato entre 0 y 1023, que representa un rango de 0 a 300 grados, la región de 301 a 359 grados proporciona una lectura invalida. La Figura 3.8 muestra las posiciones para las lecturas 0, 1023 y 512.

Articulación ID	Nombre de la Articulación	Grados de libertad
0	Raíz	TX, TY, TZ, RX, RY, RZ
1	lhipjoint	0
2	lfemur	rx, ry, rz
3	ltibia	rx
4	lfoot	rx, rz
5	ltoes	rx
6	rhipjoint	0
7	rfemur	rx, ry, rz
8	rtibia	rx
9	rfoot	rx, rz
10	rtoes	rx
11	lowerback	rx, ry, rz
12	upperback	rx, ry, rz
13	thorax	rx, ry, rz
14	lowerneck	rx, ry, rz
15	upperneck	rx, ry, rz
16	head	rx, ry, rz
17	lclavicle	ry, rz
18	lhumerus	rx, ry, rz
19	lradius	rx
20	lwrist	ry
21	lhand	rx,rz
22	lfingers	rx
23	lthumb	rx, rz
24	relavicle	ry,rx
25	rhumerus	rx, ry, rz
26	rradius	rx
27	rwrist	ry
28	rhand	rx, rz
29	rfingers	rx
30	rthumb	rx, rz

Tabla 3.1: Grados de libertad para cada articulación del esqueleto de captura de movimiento.

Articulación MOCAP	GDL de MOCAP	Articulación ROBOT Humanoide
Raíz	TX, TY, TZ, RX, RY, RZ	Marco Base
rhumerus	rx	1
lhumerus	rx	2
rhumerus	rz	3
lhumerus	rz	4
rfemur	ry	7
lfemur	ry	8
rfemur	rz	9
lfemur	rz	10
rfemur	rx	11
lfemur	rx	12
rtibia	rx	13
ltibia	rx	14
rfoot	rx	15
lfoot	rx	16
rfoot	rz	17
lfoot	rz	18

Tabla 3.2: Grados de libertad del esqueleto de captura de movimientos utilizados para teleoperar el robot humanoide de 18 grados de libertad

Característica	Dato
Peso	54.6 gramos
Dimensiones	32mm * 50 mm * 40 mm
Resolución	0.29 grados
Relación de reducción	254:1
Torque de Bloqueo	1.5 N.m
Velocidad sin carga	59 rpm
Temperatura de trabajo	-5°C a +70°C
Voltaje	9 a 12 V
Señales de comando	Paquete Digital
Protocolo	Half duplex Serial asíncrono
Velocidad de comunicación	7343bps 1 Mbps
Retroalimentación	Posición, Temperatura, Carga, etc.
Material	Plástico de Ingeniería
Conexión	Conector tipo cadena Margarita

Tabla 3.3: Especificaciones del motor Dynamixel AX-12 de ROBOTISTM.

Dirección (Hex)	Nombre	Descripción	Acceso R/W
0 (0X00)	Model Number(L)	Número de Modelo (Byte parte baja)	R
1 (0X01)	Model Number(H)	Número de Modelo (Byte parte alta)	R
2 (0X02)	Version of Firmware	Versión de Firmware	R
3 (0X03)	ID	ID del Dynamixel	RW
4 (0X04)	Baud Rate	Velocidad de Comunicación	RW
5 (0X05)	Return Delay Time	Tiempo entre envío de instrucción y regreso de status	RW
6 (0X06)	CW Angle Limit(L)	Límite mínimo de la posición del motor (Parte Baja)	RW
7 (0X07)	CW Angle Limit(H)	Límite mínimo de la posición del motor (Parte Alta)	RW
8 (0X08)	CCW Angle Limit(L)	Límite máximo de la posición del motor (Parte Baja)	RW
9 (0X09)	CCW Angle Limit(H)	Límite máximo de la posición del motor (Parte Alta)	RW
11 (0X0B)	the Highest Limit Temperature	Límite interno de temperatura	RW
12 (0X0C)	the Lowest Limit Voltage	Mínimo voltaje requerido	RW
13 (0X0D)	the Highest Limit Voltage	Máximo voltaje soportado	RW
14 (0X0E)	Max Torque(L)	Valores de torque máximo(parte baja)	RW
15 (0X0F)	Max Torque(H)	Valores de torque máximo(parte alta)	RW
16 (0X10)	Status Return Level	Indica cuando regresar un nivel de status	RW
17 (0X11)	Alarm LED	Activa el LED de alarma por mal funcionamiento	RW
18 (0X12)	Alarm Shutdown	Apagado del motor por alarma	RW

Tabla 3.4: Sección EEPROM de la Tabla de Control del motor Dynamixel AX-12 de ROBOTISTM.

Dirección (Hex)	Nombre	Descripción	Acceso R/W
24 (0X18)	Torque Enable	Habilita /Deshabilita Torque	RW
25 (0X19)	LED	Enciende/Apaga LED	RW
26 (0X1A)	CW Compliance Margin	Diferencia entre posición deseada y actual	RW
27 (0X1B)	CCW Compliance Margin	Diferencia entre posición deseada y actual	RW
28 (0X1C)	CW Compliance Slope	Indica el nivel de torque al acercarse a la posición deseada	RW
29 (0X1D)	CCW Compliance Slope	Indica el nivel de torque al acercarse a la posición deseada	RW
30 (0X1E)	Goal Position(L)	Posición deseada (Parte Baja)	RW
31 (0X1F)	Goal Position(H)	Posición deseada (Parte Alta)	RW
32 (0X20)	Moving Speed(L)	Velocidad deseada(Parte Baja)	RW
33 (0X21)	Moving Speed(H)	Velocidad deseada(Parte Alta)	RW
34 (0X22)	Torque Limit(L)	Límite de Torque(Parte Baja)	RW
35 (0X23)	Torque Limit(H)	Límite de Torque(Parte Alta)	RW
36 (0X24)	Present Position(L)	Posición Actual (Parte Baja)	R
37 (0X25)	Present Position(H)	Posición Actual (Parte Alta)	R
38 (0X26)	Present Speed(L)	Velocidad Actual(Parte Baja)	R
39 (0X27)	Present Speed(H)	Velocidad Actual(Parte Alta)	R
40 (0X28)	Present Load(L)	Valor de Carga Actual (Parte Baja)	R
41 (0X29)	Present Load(H)	Valor de Carga Actual (Parte Alta)	R
42 (0X2A)	Present Voltage	Voltaje Actual	R
43 (0X2B)	Present Temperature	Temperatura	Actual R
44 (0X2C)	Registered	Si la instrucción fue registrada	R
46 (0X2E)	Moving	Si hay algún movimiento	R
47 (0X2F)	Lock	Bloque el acceso a Memoria EEPROM	RW
48 (0X30)	Punch(L)	Lowest byte of Punch	RW
49 (0X31)	Punch(H)	Highest byte of Punch	RW

Tabla 3.5: Sección RAM de la Tabla de Control del motor Dynamixel AX-12 de ROBOTISTM.

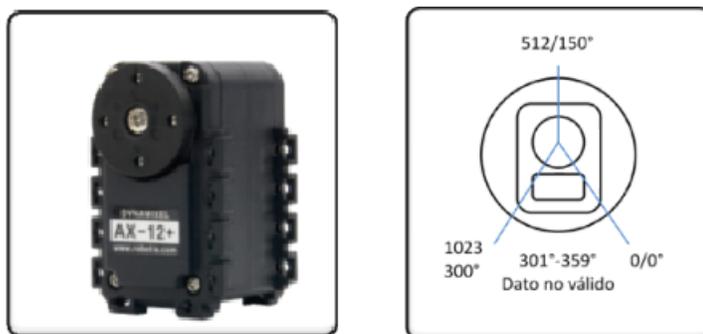


Figura 3.8: Posiciones válidas para el motor Dynamixel AX-12 de RobotisTM

Programación y comunicación Bioloid-PC

El robot Bioloid puede programarse mediante el software RoboplusTM, o en C++, para lo cual es necesario generar el ejecutable (archivo *hex*) y enviarlo al robot mediante el BOOTLOADER (autoprogramador). El software RoboplusTM incluye una terminal de comunicación que nos permite identificar si la unidad de control está conectada, acceder al bootloader, seleccionar el programa y descargarlo, ya que se ha grabado el archivo en la memoria del microcontrolador del CM-530 es posible iniciar el programa, lo cual se puede realizar de dos maneras: teclear el comando *go* en la terminal o presionar el botón *start* de la unidad de control.

Para la teleoperación a realizar, se programa la unidad de control siguiendo el diagrama de flujo mostrado en la Figura 3.9, en la sección de escritura (*w*) donde se envían de la PC a la unidad de control cadenas con las posiciones de los 18 motores para cada frame.

3.2.2. Cinemática

Desde el punto de vista mecánico describiremos la constitución del robot humanoide Bioloid PremiumTM, se muestra las articulaciones, su numeración y las dimensiones, información requerida para obtener las matrices de rotación y homogéneas, mismas que nos proporcionarán información sobre la posición de cierto eslabón con respecto al marco local

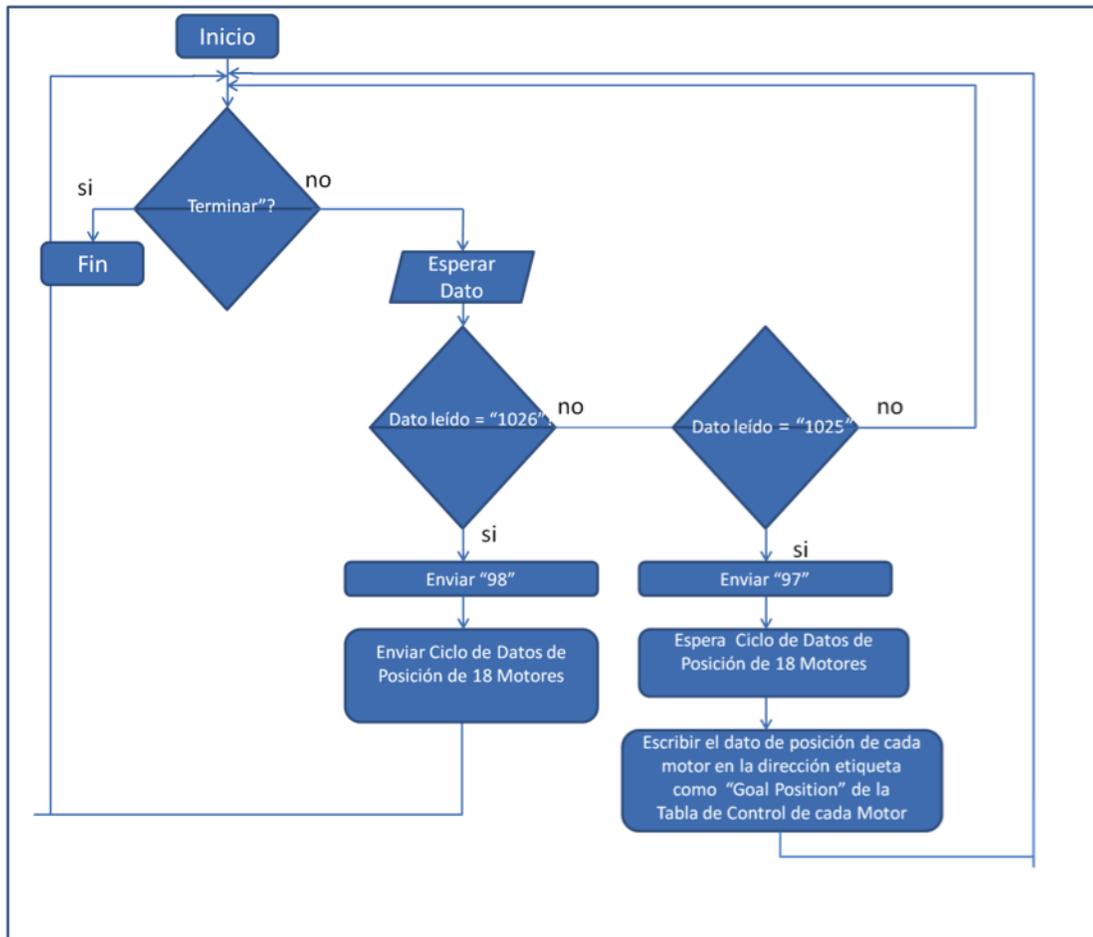


Figura 3.9: Diagrama de flujo para la Unidad de Control CM-530

y/o del mundo para un conjunto dado de variables articulares.

3.2.3. Articulaciones

La Figura 3.10 muestra la numeración de las articulaciones, así como su eje de rotación, se observan cuatro cadenas cinemáticas: brazo derecho, brazo izquierdo, pierna izquierda y pierna derecha, localizando el marco base en la pelvis, se asignan marcos naturales omitiendo la convención de Denavit-Hartenberg.

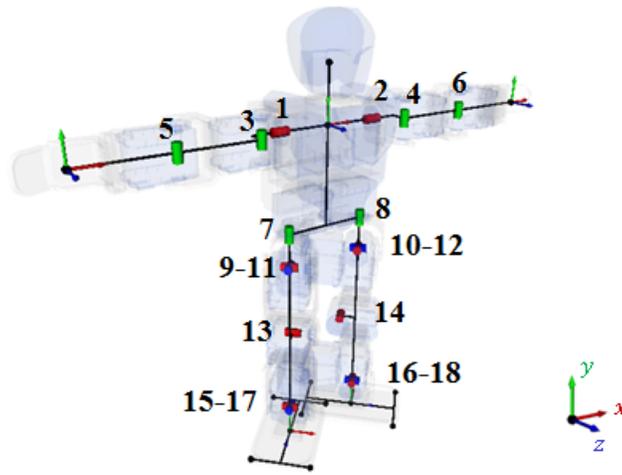


Figura 3.10: Articulaciones del Robot

Dimensiones

La Figura 3.11 muestra la nomenclatura para las distancias entre articulaciones, la Tabla 3.6 presenta el valor de estas distancias.

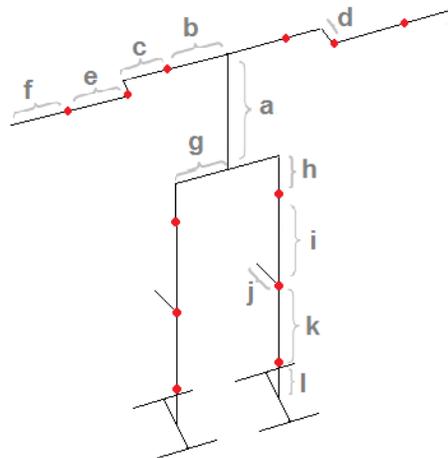


Figura 3.11: Dimensiones del Robot

Tabla 3.6: Dimensiones del Robot

Segmento	Distancia (Valores Absolutos, mm)
a	89.5
b	47
c	25
d	14.5
e	67.5
f	74.5
g	38.5
h	37
i	75
j	14.5
k	75
l	29.5

Localización de la pelvis con respecto al marco del mundo

Antes de describir las matrices homogéneas para cada articulación, es necesario recordar que estamos tratando con un robot móvil por lo que el marco base localizado en la pelvis, tiene una matriz homogénea que indica su posición y orientación con respecto al marco del mundo, las Ecuaciones 3.1 y 3.2 muestran la matriz de Rotación (utilizando ángulos de Euler en convención ZYX) y Homogénea de la Pelvis con respecto al marco del mundo, utilizando las variables ϕ, θ , y ψ para rotación y Xw, Yw y Zw para posición.

$${}^{world}R_{pelvis} = {}^w R_p = R_{z,\psi} R_{y,\phi} R_{x,\theta} \quad (3.1)$$

$${}^w H_p = \begin{bmatrix} c\phi c\psi & c\psi s\theta s\phi - c\theta c\psi & s\theta s\psi + c\theta c\psi s\phi & Xw \\ c\phi s\psi & c\theta c\psi + s\theta s\phi s\psi & c\theta s\phi s\psi - c\psi s\theta & Yw \\ -s\phi & c\phi s\theta & c\theta c\phi & Zw \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

donde la notación $c\theta$ y $s\theta$ denotan $\cos(\theta)$ y $\sin(\theta)$, de la misma manera para las variables ϕ y ψ .

Matrices homogéneas de cada articulación con respecto al marco local

Las matrices homogéneas nos proporcionan información sobre la posición y orientación de la articulación, a continuación se presentan las matrices homogéneas de cada articulación con respecto al marco anterior.

Brazo Derecho

La cadena cinemática del brazo derecho cual está formada por las articulaciones 1, 3 y 5 que rotan en los ejes x , y y y respectivamente. Las ecuaciones 3.3, 3.4, 3.5 y 3.6 muestran las matrices homogéneas de cada articulación.

$${}^P H_1 = \begin{bmatrix} 1 & 0 & 0 & -b \\ 0 & c_1 & -s_1 & a \\ 0 & s_1 & c_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

$${}^1 H_3 = \begin{bmatrix} c_3 & 0 & s_3 & -c \\ 0 & 1 & 0 & 0 \\ -s_3 & 0 & c_3 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$${}^3 H_5 = \begin{bmatrix} c_5 & 0 & s_5 & -e \\ 0 & 1 & 0 & 0 \\ -s_5 & 0 & c_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$${}^5 H_{MD} = \begin{bmatrix} 1 & 0 & 0 & -f \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Brazo Izquierdo

La cadena cinemática del brazo izquierdo cual está formada por las articulaciones 2, 4 y 6 que rotan en los ejes x , y y y respectivamente. Las ecuaciones 3.7, 3.8, 3.9 y 3.10 muestran las matrices homogéneas de cada articulación.

$${}^P H_2 = \begin{bmatrix} 1 & 0 & 0 & b \\ 0 & c_2 & -s_2 & a \\ 0 & s_2 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

$${}^2 H_4 = \begin{bmatrix} c_4 & 0 & s_4 & c \\ 0 & 1 & 0 & 0 \\ -s_4 & 0 & c_4 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

$${}^4 H_6 = \begin{bmatrix} c_6 & 0 & s_6 & e \\ 0 & 1 & 0 & 0 \\ -s_6 & 0 & c_6 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

$${}^6 H_{MI} = \begin{bmatrix} 1 & 0 & 0 & f \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

Pierna Derecha

La cadena cinemática de la pierna derecha, está formada por las articulaciones 7, 9,11,13,15,17 que rotan en los ejes y , z , x , x , z , y x respectivamente. Las ecuaciones 3.11, 3.12, 3.13, 3.14,3.15, 3.16 y 3.17 muestran las matrices homogéneas de cada articulación.

$${}^p H_7 = \begin{bmatrix} c_7 & 0 & s_7 & -g \\ 0 & 1 & 0 & 0 \\ -s_7 & 0 & c_7 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

$${}^7 H_9 = \begin{bmatrix} c_9 & -s_9 & 0 & -h \\ s_9 & c_9 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

$${}^9 H_{11} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{11} & -s_{11} & 0 \\ 0 & s_{11} & c_{11} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

$${}^{11} H_{13} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{13} & -s_{13} & -i \\ 0 & s_{13} & c_{13} & -j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

$${}^{13} H_{15} = \begin{bmatrix} c_{15} & -s_{15} & 0 & 0 \\ s_{15} & c_{15} & 0 & -k \\ 0 & 0 & 1 & j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

$${}^{15} H_{17} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{18} & -s_{18} & 0 \\ 0 & s_{18} & c_{18} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

$${}^{17}H_{PD} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -l \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

Pierna Izquierda

La cadena cinemática de la pierna izquierda, está formada por las articulaciones 8, 10,12,14,16,18 que rotan en los ejes $y, z, x, x, z,$ y x respectivamente. Las ecuaciones 3.18, 3.19, 3.20, 3.21,3.22, 3.23 y 3.24 muestran las matrices homogéneas de cada articulación.

$${}^pH_8 = \begin{bmatrix} c_8 & 0 & s_8 & g \\ 0 & 1 & 0 & 0 \\ -s_8 & 0 & c_8 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

$${}^9H_{10} = \begin{bmatrix} c_{10} & -s_{10} & 0 & -h \\ s_{10} & c_{10} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.19)$$

$${}^{10}H_{12} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{12} & -s_{12} & 0 \\ 0 & s_{12} & c_{12} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.20)$$

$${}^{12}H_{14} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{14} & -s_{14} & -i \\ 0 & s_{14} & c_{14} & -j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.21)$$

$${}^{14}H_{16} = \begin{bmatrix} c_{16} & -s_{16} & 0 & 0 \\ s_{16} & c_{16} & 0 & -k \\ 0 & 0 & 1 & j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

$${}^{16}H_{18} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{18} & -s_{18} & 0 \\ 0 & s_{18} & c_{18} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

$${}^{18}H_{PI} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -l \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.24)$$

Cadenas Cinemáticas

Para obtener posición P y orientación R de cada los efectores finales: mano derecha, mano izquierda, pierna derecha y pierna izquierda, se utilizan las ecuaciones 3.25, 3.26, 3.27, 3.28.

$${}^w H_{MD} = {}^w H_p {}^p H_1 {}^1 H_3 {}^3 H_5 {}^5 H_{MD} \quad (3.25)$$

$${}^w H_{MI} = {}^w H_p {}^p H_2 {}^2 H_4 {}^4 H_6 {}^6 H_{MI} \quad (3.26)$$

$${}^w H_{PD} = {}^w H_p {}^p H_7 {}^7 H_9 {}^9 H_{11} {}^{11} H_{13} {}^{13} H_{15} {}^{15} H_{17} {}^{17} H_{PD} \quad (3.27)$$

$${}^w H_{PI} = {}^w H_p {}^p H_8 {}^8 H_{10} {}^{10} H_{12} {}^{12} H_{14} {}^{14} H_{16} {}^{16} H_{18} {}^{18} H_{PI} \quad (3.28)$$

3.2.4. Cinemática Inversa

En esta sección se describen las ecuaciones para la cinemática inversa del Robot Humanoide Bioloid PremiumTM en su configuración A obtenidas en [100].

Utilizando la nomenclatura para las dimensiones especificadas en la Figura 3.11 y Tabla 3.6, se presentan las siguientes ecuaciones:

Para Brazo Derecho:

$$\theta_1 = \text{atan2}(Y_{RH}, X_{RH}) \quad (3.29)$$

$$\theta_3 = -(\gamma_1 - \gamma_2) \quad (3.30)$$

$$\theta_5 = -\text{atan2}(\sqrt{1 - C_5^2}, C_5) \quad (3.31)$$

$$\gamma_1 = \text{atan2}(s, t) \quad (3.32)$$

$$\gamma_2 = \text{atan2}(f \sin(\theta_5), e + f \cos(\theta_5)) \quad (3.33)$$

$$C_5 = \frac{L_{RA}^2 - e^2 - f^2}{2ef} \quad (3.34)$$

$$s = \sqrt{Y_{RH}^2 + Z_{RH}^2} - d \quad (3.35)$$

$$t = X_{RH} - b - c \quad (3.36)$$

$$L_{RA} = \sqrt{s^2 + t^2} \quad (3.37)$$

donde RH y X_{RH}, Y_{RH} y Z_{RH} se refieren a la posición del efector final (Mano Derecha) y sus coordenadas.

Para brazo Izquierdo se tienen las ecuaciones

$$\theta_2 = \text{atan2}(-Y_{LH}, X_{LH}) \quad (3.38)$$

$$\theta_4 = (\gamma_1 - \gamma_2) \quad (3.39)$$

$$\theta_6 = \text{atan2}(\sqrt{1 - C_6^2}, C_6) \quad (3.40)$$

$$\gamma_1 = \text{atan2}(s, t) \quad (3.41)$$

$$\gamma_2 = \text{atan2}(f \sin(\theta_6), e + f \cos(\theta_6)) \quad (3.42)$$

$$C_6 = \frac{L_{LA}^2 - e^2 - f^2}{2ef} \quad (3.43)$$

$$s = \sqrt{Y_{LH}^2 + Z_{LH}^2} - d \quad (3.44)$$

$$t = X_{LH} - b - c \quad (3.45)$$

$$L_{LA} = \sqrt{s^2 + t^2} \quad (3.46)$$

donde LH y X_{LH}, Y_{LH} y Z_{LH} se refieren a la posición del efector final (Mano Derecha) y sus coordenadas.

Para la parte baja de la pierna derecha se tiene:

$$\theta_{13} = -(\gamma + 2\alpha) \quad (3.47)$$

$$\theta_{15} = -\left(\beta - \frac{\gamma}{2} - \alpha\right) \quad (3.48)$$

$$\theta_{17} = -atan2(X_{RF}, Y_{RF}) \quad (3.49)$$

$$C_\gamma = -\frac{L_{RLeg}^2 - 2L_f^2}{2L_f^2} \quad (3.50)$$

$$L_{RLeg} = \sqrt{x_{RHip}^2 + y_{RHip}^2 + z_{RHip}^2} \quad (3.51)$$

$$L_f = \sqrt{i^2 + j^2} \quad (3.52)$$

$$\alpha = atan2(j, i) \quad (3.53)$$

$$\gamma = \pi - atan2(\sqrt{1 - C_\gamma^2}, C_\gamma) \quad (3.54)$$

$$\beta = atan2(Z_{RHip}, L_{RLeg}^1) \quad (3.55)$$

$$L_{RLeg}^1 = \sqrt{x_{RHip}^2 + y_{RHip}^2} \quad (3.56)$$

donde RF y X_{RF}, Y_{RF} y Z_{RF} se refieren a la posición del efector final (Pie Derecho) y sus coordenadas.

Para la parte baja de la pierna izquierda:

$$\theta_{14} = (\gamma + 2\alpha) \quad (3.57)$$

$$\theta_{16} = (\beta - \frac{\gamma}{2} - \alpha) \quad (3.58)$$

$$\theta_{18} = -atan2(Y_{LF}, X_{LF}) \quad (3.59)$$

$$C_\gamma = \frac{L_{RLeg}^2 - 2L_f^2}{2L_f^2} \quad (3.60)$$

$$L_{RLeg} = \sqrt{x_{LHip}^2 + y_{LHip}^2 + z_{LHip}^2} \quad (3.61)$$

$$L_f = \sqrt{i^2 + j^2} \quad (3.62)$$

$$\alpha = atan2(j, i) \quad (3.63)$$

$$\gamma = \pi - 2(\sqrt{1 - C_\gamma^2}, C_\gamma) \quad (3.64)$$

$$\beta = -atan2(Z_{RHip}, L_{RLeg}^1) \quad (3.65)$$

$$L_{Rleg}^1 = \sqrt{x_{LHip}^2 + y_{LHip}^2} \quad (3.66)$$

donde LF y X_{LF}, Y_{LF} y Z_{LF} se refieren a la posición del efector final (Pie Izquierdo) y sus coordenadas.

3.2.5. Dinámica

El modelo dinámico de un robot humanoide nos permite conocer la relación entre las fuerzas requeridas para alcanzar cierta posición y orientación para un conjunto de valores articulares definidos.

Las ecuaciones de movimiento de Euler-Lagrange describen la evolución de un sistema mecánico sujeto a restricciones holonómicas, dichas ecuaciones se derivan de la segunda ley de Newton.

Las ecuaciones de movimiento son importantes en el diseño de robots, en simulación y animación de movimiento de robots y en el diseño de algoritmos de control.

De acuerdo a la segunda ley de Newton, para un grado de libertad la ecuación de movimiento de una partícula de masa m es:

$$m\ddot{y} = f - mg \quad (3.67)$$

$$f = m\ddot{y} + mg \quad (3.68)$$

el primer elemento de la ecuación 3.68 puede ser reescrito como se muestra en 3.69

$$m\ddot{y} = \frac{d}{dt}(m\dot{y}) = \frac{d}{dt} \frac{\partial}{\partial \dot{y}} \left(\frac{1}{2} m \dot{y}^2 \right) = \frac{d}{dt} \frac{\partial \mathcal{K}}{\partial \dot{y}} \quad (3.69)$$

donde $\mathcal{K} = \frac{1}{2} m \dot{y}^2$ es la energía cinética. De la misma manera es posible expresar la segunda parte de la ecuación 3.68 como se muestra en la Ecuación 3.70

$$mg = \frac{\partial}{\partial y}(mgy) = \frac{\partial \mathcal{P}}{\partial y} \quad (3.70)$$

donde \mathcal{P} es la energía potencial causada por la gravedad.

La diferencia entre la energía cinética y potencial se llama Lagrangiano \mathcal{L} .

$$\mathcal{L} = \mathcal{K} - \mathcal{P} = \frac{1}{2}m\dot{y}^2 - mgy \quad (3.71)$$

Dado que la energía cinética depende de velocidades \dot{y} y la energía potencial depende de posiciones y , las derivadas parciales del Lagrangiano \mathcal{L} con respecto a \dot{y} y a y quedan

$$\frac{\partial \mathcal{L}}{\partial \dot{y}} = \frac{\partial \mathcal{K}}{\partial \dot{y}} \quad (3.72)$$

$$\frac{\partial \mathcal{L}}{\partial y} = -\frac{\partial \mathcal{P}}{\partial y} \quad (3.73)$$

por lo anterior es posible reescribir la Ecuación 3.68 como se muestra en la Ecuación 3.74.

$$f = \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{y}} - \frac{\partial \mathcal{L}}{\partial y} \quad (3.74)$$

Para un robot de n grados de libertad la Ecuación 3.74 puede ser reescritas como:

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{q}_k} - \frac{\partial \mathcal{L}}{\partial q_k} = \tau_k; k = 1 \dots n \quad (3.75)$$

$$\frac{d}{dt} \frac{\partial \mathcal{K}}{\partial \dot{q}_k} + \frac{\partial \mathcal{P}}{\partial q_k} = \tau_k; k = 1 \dots n \quad (3.76)$$

donde τ_k son las fuerzas asociadas con q_k

La energía cinética de un cuerpo rígido es la suma de dos términos: energía cinética

traslacional y rotacional:

$$\mathcal{K} = \frac{1}{2}mv^T v + \frac{1}{2}\omega^T \mathcal{I}\omega \quad (3.77)$$

donde m es la masa, v y ω corresponden a velocidad lineal y angular, respectivamente y \mathcal{I} es el tensor de inercia.

Es necesario expresar el tensor de inercia \mathcal{I} en el marco inercial para realizar el cálculo de $\omega^T \mathcal{I}\omega$, si conocemos I que es el tensor de inercia expresado en el marco local, entonces es posible obtener \mathcal{I} mediante

$$\mathcal{I} = RIR^T \quad (3.78)$$

La matriz I es constante e independiente del movimiento.

Por otra parte las velocidades lineal y angular se obtienen mediante

$$v_i = J_{v_i}(q)\dot{q} \quad (3.79)$$

$$\omega_i = J_{\omega_i}(q)\dot{q} \quad (3.80)$$

Con lo anterior la ecuación 3.77 puede reescribirse como

$$\mathcal{K} = \frac{1}{2}\dot{q}^T \left[\sum_{i=1}^n \{m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega_i}(q)\} \right] \dot{q} \quad (3.81)$$

$$\mathcal{K} = \frac{1}{2}D(q)\dot{q}^T \quad (3.82)$$

donde D corresponde a

$$D = T \left[\sum_{i=1}^n \{m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega_i}(q)\} \right] \quad (3.83)$$

por lo que \mathcal{K} puede expresarse como

$$\mathcal{K} = \frac{1}{2} \dot{q}^T D(q) \dot{q} = \sum_{i,j} d_{i,j}(q) \dot{q}_i \dot{q}_j \quad (3.84)$$

de la misma manera podemos expresar

$$g_k = \frac{\partial \mathcal{P}}{\partial q_k} \quad (3.85)$$

al obtener las derivadas parciales con respecto a \dot{q}_k y q_k y la derivada del tiempo queda una ecuación como se muestra

$$\tau = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) \quad (3.86)$$

Los elementos de la matriz D involucran segundas derivadas de q , los elementos en la matriz C con terminos que incluyen primera derivada de q y los elementos de la matriz g no incluyen derivadas, son solo posiciones de q .

En las siguientes secciones se muestra las ecuaciones así como los elementos necesarios para obtener cada una de las matrices.

Matriz D

La matriz D es una matriz de $n \times n$, para el caso del robot humanoide Bioloid Premium TM analizando n tiene un valor de 24, 6 para la posición y orientación con respecto al mundo ($\phi, \theta, \psi, X_w, Y_w, Z_w$) y 18 variables articulares, se le denomina matriz de Inercia. En la Ecuación 3.87 se muestra como obtenerla.

$$D(q) = \left[\sum_{i=1}^n \{m_i Jv_i(q)^T Jv_i(q) + J\omega_i(q)^T R_i(q) I_i R_i(q)^T J\omega_i(q)\} \right] \quad (3.87)$$

donde m_i es la masa de cada segmento, Jv_i es el jacobiano lineal de cada segmento (al centro de masa), $J\omega_i$ es el jacobiano angular de cada segmento, R_i es la matriz de rotación,

I_i Matriz de Inercia del segmento i evaluada alrededor del centro de masa.

Jacobiano Lineal

Es la matriz de dimensiones $3 \times n$ que relaciona las velocidades y la derivada de la posición, como se muestra en la Ecuación 3.88 , es una de las cantidades más importantes en el análisis y control de movimiento de robots. Para articulaciones del tipo rotacional, la ecuación que define el jacobiano lineal para cada articulación queda como se muestra en la Ecuación 3.89 y el Jacobiano lineal del robot se muestra en la Ecuación 3.90.

$${}^0v_n = Jv\dot{q} \quad (3.88)$$

$$Jv_i = k \times (P_{Efector_Final} - {}^w r_{i-1}) \quad (3.89)$$

donde k es el vector de rotación, ${}^w r_i$ es la posición del centro de masa con respecto al marco global.

$$Jv = [Jv_1 \cdots Jv_n] \quad (3.90)$$

Jacobiano Angular

Matríz de dimensiones $3 \times n$ que relaciona las velocidades angulares con la derivada de la variables articular. Para articulaciones del tipo rotacional el jacobiano de cada articulación queda definido por la ecuación 3.91

$$J_\omega = [z_0 \cdots z_{n-1}] \quad (3.91)$$

donde

$${}^w z_{i-1} = {}^w R_{i-1} k \quad (3.92)$$

con valores de k de acuerdo al vector de rotación para $x = [1, 0, 0]$, $y = [0, 1, 0]$ y $z = [0, 0, 1]$.

Matriz C

Los elementos de la Matriz C es de dimensiones $n \times n$, sus elementos son conocidos como coeficientes de Christoffel, incluye términos que pueden agruparse en dos tipos: términos de coriolis y términos centrífugos, el primero son productos como \dot{q}_i^2 y el segundo del tipo $\dot{q}_i \dot{q}_j$ donde $i \neq j$.

Cada elemento k, j de la matriz $C(q, \dot{q})$ es definido por la Ecuación:

$$c_{kj} = \sum_{i=1}^n c_{ijk}(q) \dot{q}_i \quad (3.93)$$

$$c_{kj} = \sum_{i=1}^n \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_j} + \frac{\partial d_{ki}}{\partial q_j} + \frac{\partial d_{ij}}{\partial q_k} \right\} \dot{q}_i \quad (3.94)$$

donde d se refiere a los elementos de la Matriz D.

Matriz g

Finalmente la Matriz g es de dimensiones $n \times 1$ incluye términos de q pero no de sus derivadas, éstos terminos se obtienen derivando la energía potencial, la cual se obtiene mediante la ecuación

$$\mathcal{P}_i = m_i g^T r_{ci} \quad (3.95)$$

donde g es el vector de la dirección de la gravedad, r_{ci} son las coordenadas del centro de masa del eslabón i . La energía potencial total de un robot de n -segmentos se obtiene mediante

$$\mathcal{P} = \sum_{i=1}^n \mathcal{P}_i = \sum_{i=1}^n m_i g^T r_{ci} \quad (3.96)$$

Cada elemento de la matriz g se obtiene mediante

$$g_k = \frac{\partial \mathcal{P}}{\partial q_k} \quad (3.97)$$

Resultados del Modelo Dinámico

Mediante el software MATLAB, se obtuvieron los jacobianos lineales y angulares para cada segmento del robot, y al conocer los elementos requeridos por la Ecuación 3.87 se calculó la matriz D con dimensiones 24×24 , sin embargo, las ecuaciones resultaron inmanejables de forma simbólica por su tamaño, por lo que se intentó trabajar de forma numérica, sin embargo, éstos resultados no han sido verificados.

La matriz C depende de los elementos de la matriz D , debido a que éstos elementos aún no han sido verificados, no se ha trabajado en la matriz C .

Se obtuvo la matriz g para los 24 grados de libertad.

Aunque se sigue trabajando en la obtención de esta matriz, se descartó utilizarla para estabilizar los movimientos realizados por el robot humanoide.

Parámetros del robot Humanoide Bioloid PremiumTM

Para obtener la masa de cada eslabón se pesaron cada uno de ellos en una báscula con una resolución de 10 gramos. La Tabla 3.7 muestra los datos obtenidos.

Para obtener los centros de masa y tensores de inercia se realizó el ensamble de las piezas de cada eslabón del robot (proporcionadas por el fabricante) en el Software de Diseño Mecánico *SolidWorksTM*, sin embargo, los motores se modelaron como una caja negra de masa homogénea, lo que inicialmente nos hacía concluir que el Centro de Masa sería incorrecto, se consultó Dinnis [101] quien documenta la corrección del centro de masa para este motor, sin embargo las diferencias fueron menores a 5 mm. Los datos obtenidos para

Articulación	Masa (gramos)
0	553
1 = 2	11.38
3 = 4	76.72
5 = 6	77.34
7 = 8	14.9
9 = 10	137
10 = 12	30.61
13 = 14	86.64
15 = 16	137
17 = 18	37.07

Tabla 3.7: Masa de cada eslabón del Robot Humanoide Bioloid PremiumTM

centro de masa con respecto al marco base y tensores de inercia de cada eslabón se muestran en las Tablas 3.8 y 3.9.

3.2.6. ZMP

En 1972 Vukobratovic and Stepanenko[102] definieron el concepto punto de momento cero (Zero Moment Point, ZMP) y desde entonces es un concepto ampliamente utilizado en locomoción de humanoides. La Figura 3.12 muestra como se distribuyen las fuerzas de reacción con la superficie en un pie, misma que puede ser reducida a una fuerza resultante R , localizada en algún punto en los límites de la superficie del pie, a este punto se le conoce como punto de momento cero, ZMP [103].

El ZMP siempre existe en el polígono de soporte[104] el cual se refiere al polígono envolvente que forma la superficie de los dos pies en soporte doble, o la superficie del pie de soporte en soporte simple, como se muestra en la Figura 3.13.

Se le denomina proyección del Centro de Masa (Center of Mass, CoM) en la superficie, al punto donde la línea de gravedad del CoM se intersecta con la superficie, cuando una persona está de pie sobre la superficie en soporte doble el ZMP coincide con la proyección del CoM en la superficie, la persona mantiene el balance si la proyección del CoM en la superficie está dentro del polígono de soporte; por otra parte si el humano se mueve y cambia, por

Eslabón	Centro de masa (mm)
0	00 58 -07
1	-60 89 04
2	60 89 04
3	-97 90 16
4	97 90 16
5	-167 90 15
6	166 90 15
7	-38 -12 -15
8	38 -12 -15
9	-38 -46 -14
10	38 -46 -14
11	-38 -75 00
12	38 -75 00
13	-38 -117 01
14	37 -117 01
15	-38 -167 -14
16	37 -167 -14
17	-43 -207 -11
18	42 -207 -11

Tabla 3.8: Centros de Masa para los eslabones del Robot Humanoide Bioloid PremiumTM

ejemplo, de soporte doble a soporte simple, la proyección con la superficie del CoM puede salir del polígono de soporte, para este caso el ZMP no existe [103].

Vukobratovic y Juricic [105] definen *ZMP* como el punto en la superficie de contacto alrededor del cual la componente horizontal del momento de la fuerza de reacción es cero, Ecuación 3.98.

$$\vec{\tau}_{F.R.}(\vec{r}_{ZMP})_{horizontal} = 0 \quad (3.98)$$

La Ecuación 3.98 significa que el momento de la fuerza de reacción resultante en el ZMP es siempre en el eje vertical.

El ZMP también es definido como un punto en la superficie de contacto en el cual el momento causado por las fuerzas inerciales y gravitacionales no tiene componente en el eje horizontal, Ecuación 3.99

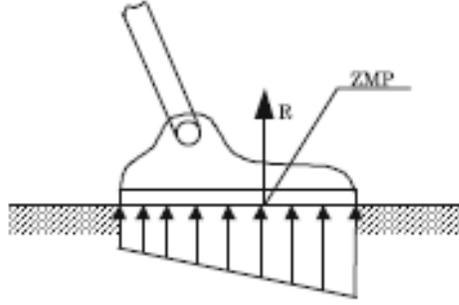


Figura 3.12: Definición de ZMP[103]

$$\vec{\tau}_{inercia+gravedad}(\vec{r}_{ZMP})|_{horizontal=0} \quad (3.99)$$

A partir de la ecuación 3.98, si no hay fuerzas externas, solo la fuerza de gravedad y de la superficie de contacto, la componente horizontal del momento que crea la fuerza de gravedad en el ZMP es igual a la componente horizontal de momento de todo el cuerpo en el ZMP, Ecuación 3.100.

$$\vec{\tau}(\vec{r}_{ZMP})|_{horizontal} = [\vec{r}_{CoM} - \vec{r}_{ZMP} \times M \vec{g}]_{horizontal} \quad (3.100)$$

donde \vec{r}_{CoM} es el Centro de Masa y M es la masa total. Utilizando información de cada eslabón la Ecuación 3.100 puede ser reescrita como se muestra en la Ecuación 3.101

$$\sum_{i=1}^N \left[\vec{r}_i - \vec{r}_{ZMP} \times m_i \vec{a}_i + \frac{d \overleftrightarrow{I}_i \vec{\omega}_i}{dt} \right]_{horizontal} = [\vec{r}_i - \vec{r}_{ZMP} \times M \vec{g}]_{horizontal} \quad (3.101)$$

donde \vec{r}_i, m_i y corresponden a el CoM y masa de cada eslabón, \vec{a}_i y \overleftrightarrow{I}_i indican aceleración lineal y tensores de inercia del CoM de cada segmento y finalmente $\vec{\omega}_i$ representa la velocidad angular de cada segmento. La Ecuación 3.101 es un sistema de dos ecuaciones con dos variables desconocidad: zmp_x y zmp_z , la cual al resolverse da lugar a las Ecuaciones

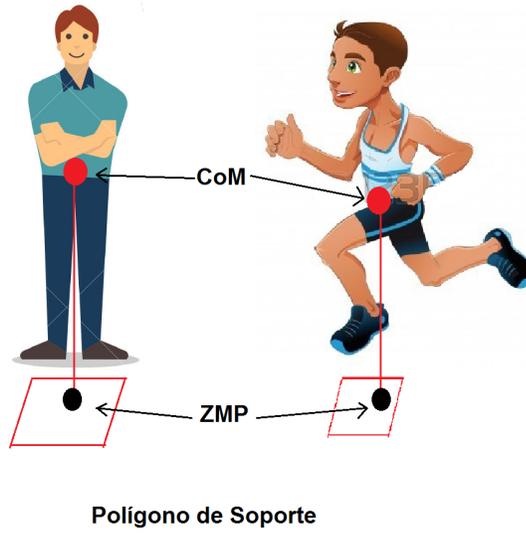


Figura 3.13: Relación entre el CoM y el ZMP

3.102 y 3.103 .

$$zmp_x = - \frac{\sum_{i=1}^N \{ \vec{r}_i \times m_i (\vec{a}_i - \vec{g}) + \left[\frac{d\vec{I}_i}{dt} \vec{\omega}_i \right] \}_z}{M (C\ddot{o}M_y + g)} \quad (3.102)$$

$$zmp_z = - \frac{\sum_{i=1}^N \{ \vec{r}_i \times m_i (\vec{a}_i - \vec{g}) + \left[\frac{d\vec{I}_i}{dt} \vec{\omega}_i \right] \}_x}{M (C\ddot{o}M_y + g)} \quad (3.103)$$

Estas ecuaciones se utilizan para proveer de información de este concepto a la interfaz propuesta, el cálculo es mostrado en línea.

Articulación	Tensor de Inercia (g/mm^2)	Articulación	Tensor de Inercia(g/mm^2)
0	992393 -2073 -1276 -2073 560882 76222 -1276 76222 1132970	.	.
1	5057 -0 -448 0 2105 0 -448 0 4935	2	5057 0 448 0 2105 0 448 0 4935
3	7800 367 0 367 19126 0 0 0 23979	4	7800 -367 0 -367 19126 0 0 0 23979
5	5522 552 3273 522 27123 35 3273 35 28867	6	5522 -552 -3273 -522 27123 35 -3273 35 28867
7	13799 0 0 0 13185 110 0 110 2746	8	13799 0 0 0 13185 110 0 110 2746
9	52973 -114 1010 -110 54243 1444 1010 1444 11067	10	52973 114 -1010 110 54243 1444 -1010 1444 11067
11	16624 0 0 0 13579 1342 0 1342 24938	12	16624 0 0 0 13579 1342 0 1342 24938
13	37213 702 101 702 13545 124 101 124 43536	14	37214 -702 -101 -702 13545 124 -101 124 43537
15	52973 114 1010 114 54243 -1444 1010 -1444 11067	16	52973 -114 -1010 -114 54243 -1444 -1010 -1444 11067
17	33435 779 -617 779 39224 -804 -617 -804 11521	18	33435 -779 617 -779 39224 -804 617 -804 11521

Tabla 3.9: Tensores de Inercia para los eslabones del Robot Humanoide Bioloid PremiumTM

Capítulo 4

Resultados

En esta sección se mencionan los resultados obtenidos durante el desarrollo de la presente investigación, los resultados se presentan de acuerdo a la metodología presentada en la Figura 1.1. Primero se presenta el modelo virtual con comportamiento cinemático, después el modelo virtual con comportamiento dinámico, enseguida un primer experimento de imitación de movimientos mediante captura de movimientos replicados en un humanoide virtual mediante el modelo resorte amortiguador y finalmente la teleoperación de un humanoide mediante captura de movimientos mediante captura de movimientos utilizando en modelo resorte amortiguador.

4.1. Modelo Virtual con Comportamiento Cinemático

Para el desarrollo del modelo virtual se utiliza lenguaje de programación C++ bajo la plataforma Visual C++ Express Edition, los modelos geométricos del humanoide virtual los provee el fabricante de la compañía Robotis TM, bajo el formato IGES (Initial Graphics Exchange Specification). Se utiliza el software para intercambio de formatos Deep ExplorationTM, y mediante la herramienta MakeTRI se generan los archivos en su formato final (*tri*). Los archivos en formato *tri* son utilizados por la librería de renderizado gráfico OpenGL para la visualización en la escena del humanoide. En la Figura 4.6 se muestran

ejemplos de los modelos geométricos utilizados.



Figura 4.1: Ejemplos de modelos geométricos utilizados

Para acomodar cada una de las piezas del robot en el modelo virtual es necesario realizar traslaciones (cambio de posición) y rotación (giros) de las piezas de manera específica, para ello se usaron las matrices homogéneas con los parámetros indicados por el fabricante. La Figura 4.2 muestra el modelo virtual del humanoide resultante.

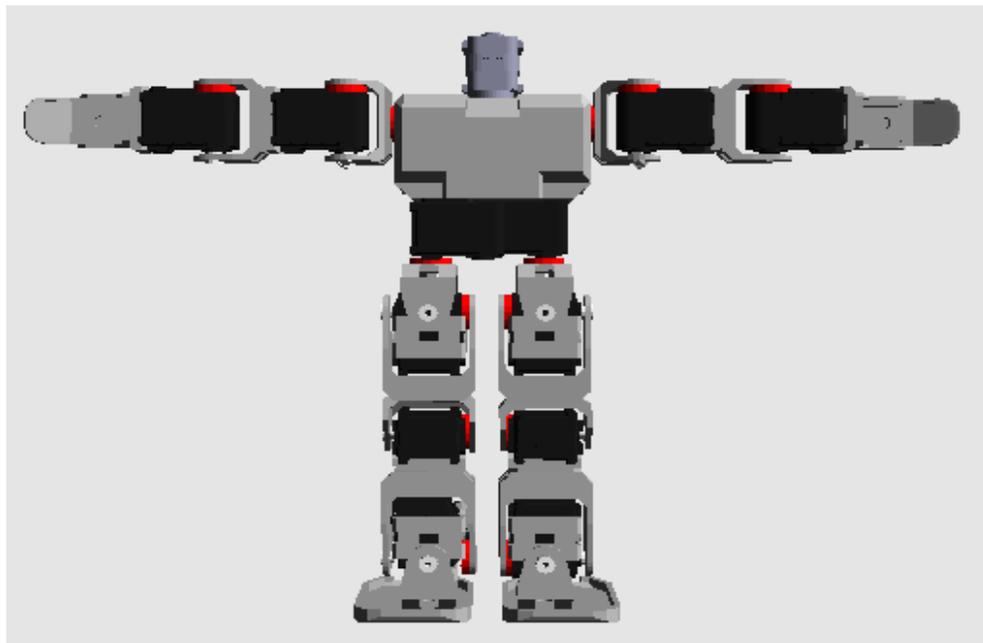


Figura 4.2: Modelo Virtual con Comportamiento Dinámico

Como un primer experimento de comunicación con entre el modelo virtual y el robot humanoide Bioloid PremiumTM en su configuración A con una unidad de control CM-510, se realizó la teleoperación del humanoide virtual mediante la manipulación del humanoide real, para ello se desarrolló un programa para el robot humanoide Bioloid PremiumTM siguiendo el diagrama de flujo de la Figura 4.4

Para que el modelo virtual imite las posiciones de los motores del humanoide real, se

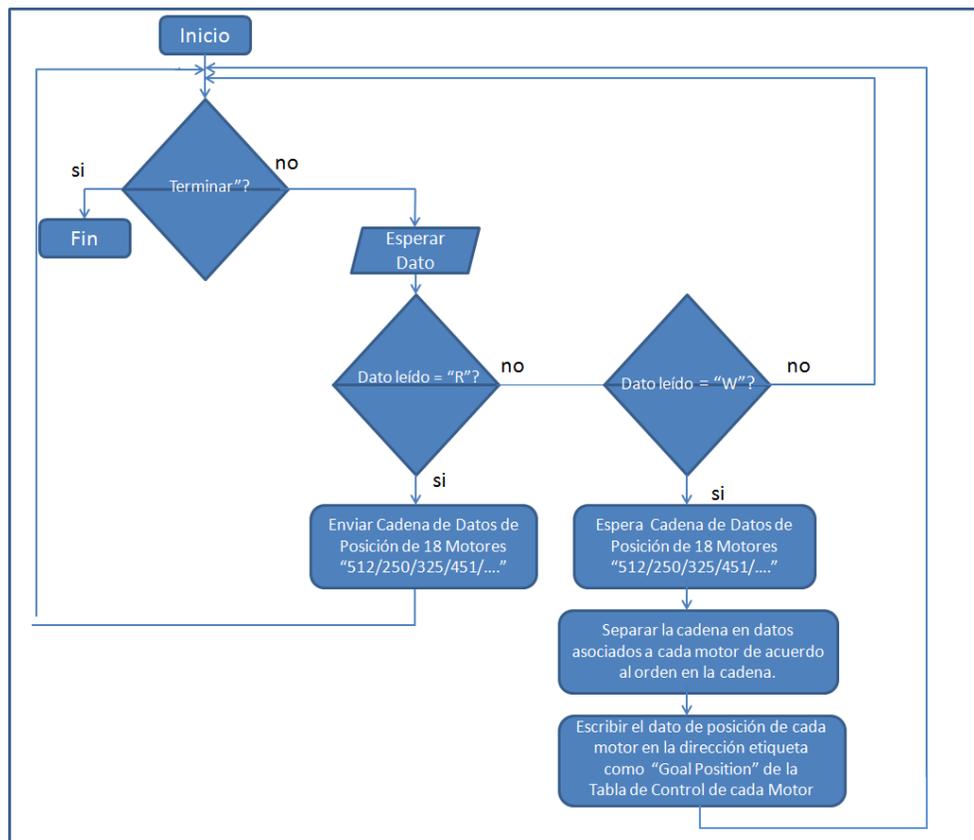


Figura 4.3: Diagrama de flujo de la programación del Robot Humanoide para la teleoperación del humanoide virtual mediante manipulación del humanoide real

identifican las piezas que tendrán valor angular variable, el cual responde a los datos leídos de la posición del motor, previamente convertidos a grados. La comunicación entre la PC y el Robot Humanoide es mediante puerto serial a una velocidad de 57600 baudios. La unidad de control CM-510 espera un la indicación de lectura o escritura mediante las letras *R* o *W*, al recibir la indicación de lectura, regresa una cadena de caracteres indicando la posición de cada uno de los motores separada por una diagonal: 512/1024/650/128/355. , la PC recibe la cadena de caracteres y las separa en un arreglo de datos flotantes, estos datos son transformados a grados mediante la relación mostrada en la ecuación:

$$PosicionAngular = \frac{PosiciodelMotor \times 300}{1024} \quad (4.1)$$

Debido a las traslaciones y rotaciones iniciales de cada una de las piezas para formar el humanoide, ya se tiene un desplazamiento inicial de la posición angular, por lo que es necesario sumarlo a la posición angular leída, la Tabla 1, muestra el desplazamiento inicial para cada uno de los grados de libertad.

Se analizaron diferentes posiciones para verificar que las posiciones del modelo virtual sean las mismas del robot humanoide, la Figura 4.4 muestra las comparaciones de posición del humanoide real y el virtual. Se verificó el tiempo de respuesta entre la lectura del dato y la visualización obteniéndose una medición de 60 milisegundos. En [106] está disponible un video de la teleoperación.



Figura 4.4: Comparación entre el humanoide real y el humanoide virtual con comportamiento cinemático

4.2. Modelo Virtual con Comportamiento Dinámico

Un modelo virtual que solo hace uso de una librería de gráficos, puede mostrar el comportamiento únicamente cinemático del robot, es decir, es posible indicar trayectorias y analizarlas visualmente, sin embargo, durante colisiones y autocolisiones los objetos se interpenetran, es decir, si se presenta un objeto en el ambiente virtual del robot, el eslabón del robot puede superponerse visualmente al objeto virtual sin indicar la colisión, en caso de que la colisión sea entre dos eslabones del robot, éstos se superponen visualmente; por otra parte, el robot se presenta flotando en el ambiente virtual, no tiene masa, por lo que

no es posible analizar comportamiento debido al peso o a la inercia al implementar las trayectorias. En [107] se presenta un modelo virtual para el robot Bioloid PremiumTM basado únicamente en la librería de gráficos. La Figura 2.17 muestra un modelo virtual que no incluye motor de física y se observa que permite autocolisiones.

Un motor de física es una librería que añade al modelo virtual un comportamiento dinámico, es decir, permite detectar colisiones y modelar el comportamiento dinámico del objeto virtual mediante parámetros como forma y material (densidad). Durante el diseño de controladores en bípedos un motor de física nos permite analizar si la trayectoria producirá una caída en el robot.

Existen varios motores de física entre ellos están ODE, Bullet, Vortex, Havok, Newton y PhysX.

Boeing [54] y Seugling [55] muestran análisis comparativos indicando que PhysX es el mejor motor de física para formas cóncavas. Esto es importante ya que aporta mayor realismo a la simulación, si se trabaja únicamente con formas convexas la simulación detectaría colisiones cuando visualmente se observa que los objetos aún no colisionan.

PhysX en su versión 2.3 presenta una herramienta llamada *pmap* utilizada para la detección de colisiones en mallas triangulares cóncavas.

4.2.1. Arquitectura del Sistema

Para el desarrollo del software, se siguió la arquitectura presentada en la Figura 4.5. Se inicia convirtiendo los modelos proporcionados por el fabricante, RobotisTM, en formato IGES (Initial Graphics Exchange Specification), a archivos en formato TRI mediante el software de intercambio de formatos Deep ExplorationTM y la herramienta MakeTRI [108]. La conversión de partes mecánicas 3D a malla poligonal genera un archivo de texto con extensión TRI, este archivo contiene la descripción de los vértices y caras de la malla poligonal de la parte mecánica. Los archivos en formato TRI, son utilizados por la librería del motor de física PhysXTM para crear las formas (Shapes), una o varias formas se utilizan para crear

un actor. Los actores se unen mediante uniones rotacionales de un solo grado de libertad. El movimiento se logra aplicando torques a cada actor.

Es posible tener acceso a datos de cada actor: masa, posición global, centro de masa y tensor de inercia, que pueden ser utilizados para la programación del controlador.

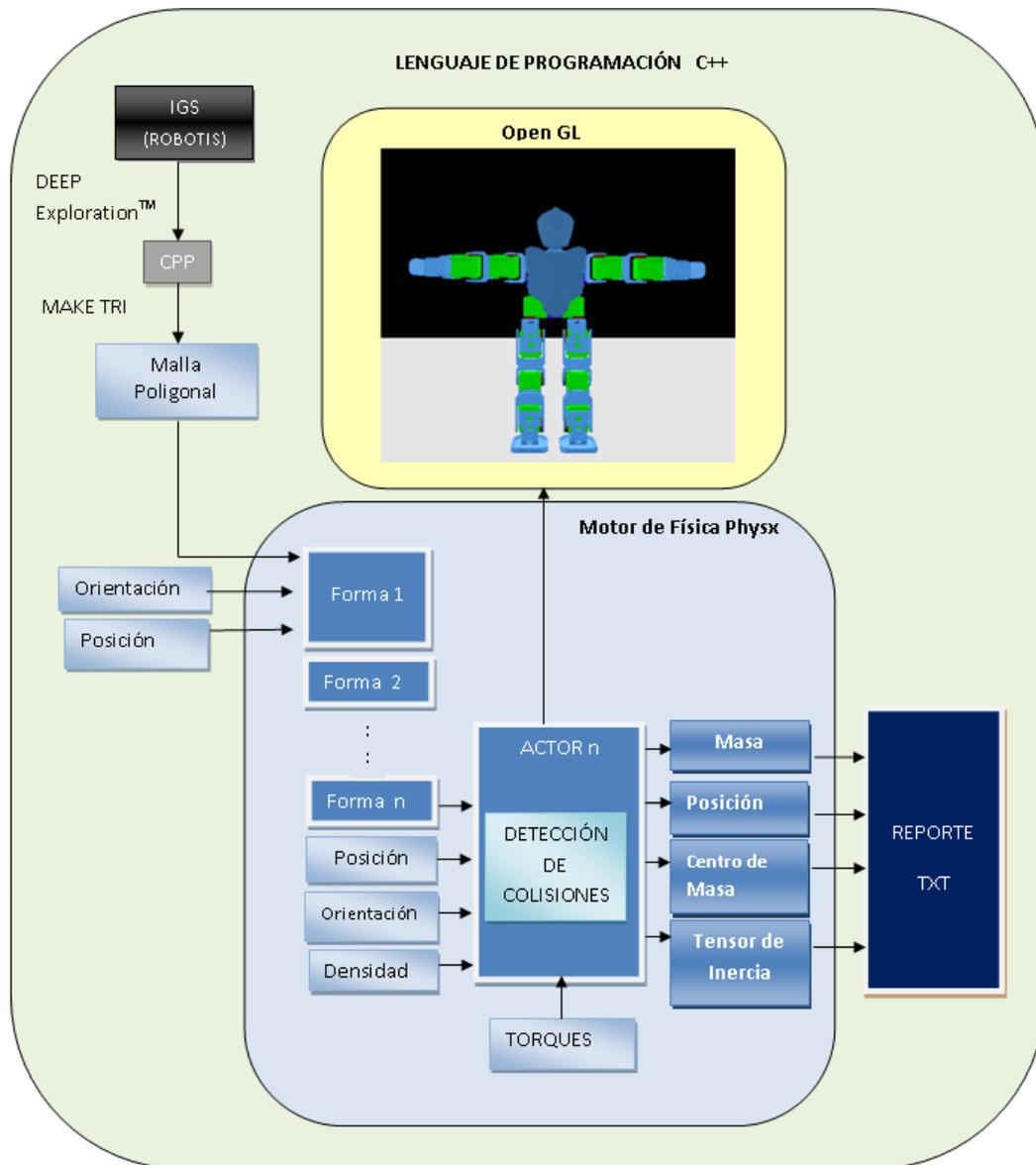


Figura 4.5: Arquitectura del sistema

4.2.2. Formas

Las formas toman la información de la malla triangular creando una estructura de datos que permite la detección de colisiones.

Para cada parte requerida para ensamblar el robot, el fabricante proporciona el modelo geométrico de cada parte en formato IGES (Initial Graphics Exchange Specification) el cual mediante DEEP Exploration es convertida a formato CPP, posteriormente mediante la herramienta MakeTRI, la información de la malla triangular en el archivo CPP es reorganizada y colocada en un archivo TRI.

Se crearon las formas de cada parte requerida para realizar el ensamble del robot excepto tornillos y cables. La Figura 4.6 muestra ejemplos de formas utilizadas.

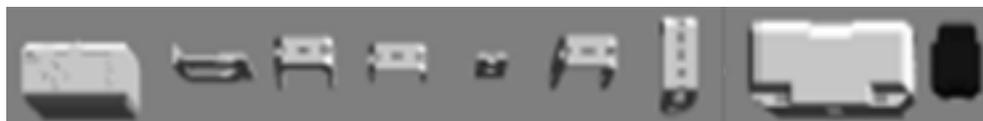


Figura 4.6: Ejemplos de formas

4.2.3. Actor

Un actor es un ente individual en el ambiente virtual con marco de referencia propio y que responde a detección de colisiones con otros actores.

Para PhysXTM un actor es un objeto de simulación, al cual podemos indicarle su posición, orientación, densidad, puede encapsular una o más formas y para moverlo es necesario aplicarle una fuerza o torque. La detección de colisiones asegura que las formas de un actor no se interpenetren con las formas de otro actor.

Se definió un actor para cada eslabón del robot humanoide Bioloid PremiumTM, es decir, para cada conjunto de elementos actuados por un motor. Los actores son: Torso, Hombro 1, Hombro 2, Codo, Ingle, Cadera 1, Cadera 2, Rodilla, Tobillo 1 y Tobillo 2. La Figura 4.7 muestra ejemplos de actores.

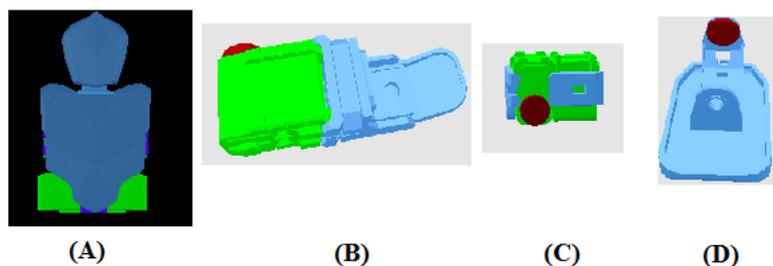


Figura 4.7: Ejemplos de actores, (A) Torso, (B) Codo, (C) Tobillo1, (D) Tobillo2

Para acomodar cada una de las formas en cada actor se indica posición y orientación mediante la matriz homogénea, de la misma manera para acomodar cada actor.

4.2.4. Estructura Padres e Hijos

Las matrices homogéneas indican posición y orientación en la forma para constituir un actor y en el actor para formar el robot Humanoide, ésta información es con respecto a los marcos locales, por lo que es necesario identificar la secuencia tanto en las formas como en los actores, para ello se define una estructura de padres e hijos, a cada forma y actor se le indica el padre y con ello el marco referencial para la posición y orientación indicados. La Figura 4.8 muestra la estructura padres e hijos para los actores.

4.2.5. Uniones

Las uniones son una forma de conectar dos actores. Sin ellas, cualquier fuerza (incluida la gravedad) puede separar los actores.

El motor de física puede respresentar diferentes tipos de uniones entre ellas están las prismáticas, esféricas y rotacionales. La primera permite movimiento de traslación entre dos actores a lo largo de un eje definido. Las uniones esféricas permiten movimiento en los tres ejes. Las uniones rotaciones solo permiten movimiento de rotación en un eje definido. La Figura 4.9 muestra ejemplos de cada una de ellas.

Para el simulador propuesto se seleccionaron uniones rotacionales que solo permiten

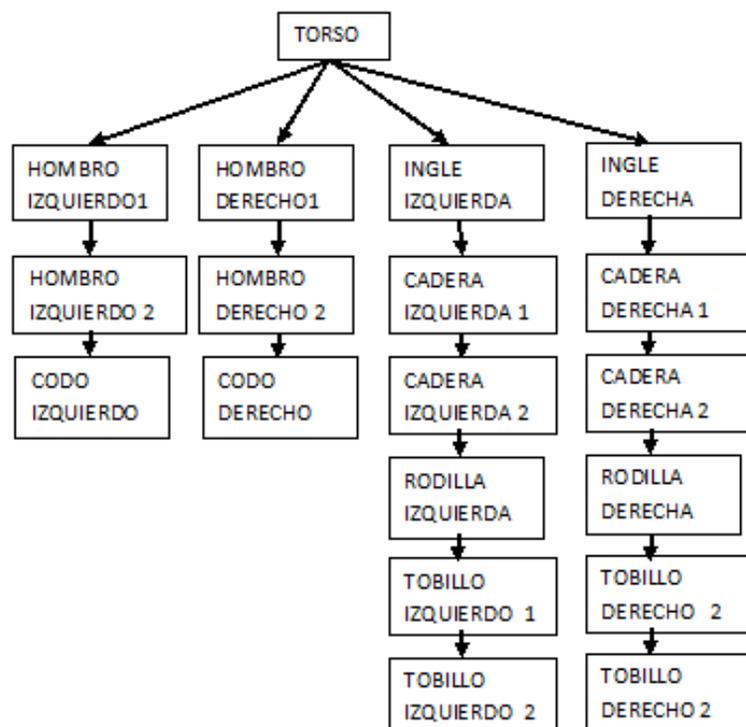
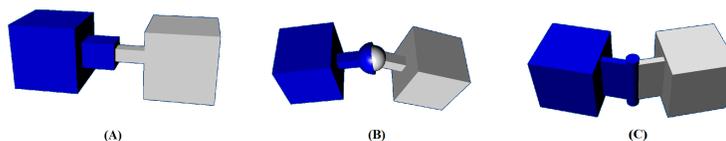


Figura 4.8: Estructura Padres e Hijos

Figura 4.9: Ejemplos de uniones en PhysXTM: (A) Prismáticas, (B) Esféricas Y (C) Rotacionales

movimiento en un eje, indicando los dos actores a unir, el eje de rotación, y el punto de unión. Se colocaron 18 uniones entre cada rotor y estator, el eje fue seleccionado de acuerdo a la orientación del motor.

4.2.6. Parámetros Dinámicos

Al crear los actores (después de indicar formas y densidad) PhysXTM calcula los parámetros dinámicos que caracterizan un cuerpo rígido: masa, centro de masa y tensor de inercia. Para eventualmente lograr movimientos estables en el humanoide, estos parámetros o sus

equivalentes globales, masa total, centro de masa del robot, momento alrededor del centro de masa del robot, resultan de gran utilidad

La Tabla 4.2 muestra la comparación entre las masas reales y las calculadas por el modelo virtual. Para los centros de masa y tensores de inercia se compararon con los obtenidos mediante en ensamble realizado en *SolidWorksTM* y se muestran en las Tablas 4.3 y 4.4

Tabla 4.2: Comparación de Masas reales y del modelo virtual

Segmento	Masa Real(g)	Masa del modelo virtual (g)	Diferencia (g)
0	553	866.79	313.79
1 = 2	11.38	12.85	0.47
3 = 4	76.72	74.52	2.2
5 = 6	77.34	83.72	6.38
7 = 8	14.9	16.37	1.47
9 = 10	137	132.20	4.8
11 = 12	30.61	30.53	0.08
13 = 14	86.64	86.80	0.16
15 = 16	137	132.20	4.8
17 = 18	37.07	42.37	5.3

Articulación	Centro de masa Diseño Mecanico(mm)	Centro de masa Modelo Virtual(mm)	Diferencia
0	00 58 -07	02, 82, -04	02,24,03
1	-60 89 04	-58, 93, 05	02,04,01
2	60 89 04	58, 93, 05	02,04,01
3	-97 90 16	-88, 91, 13	11,01,03
4	97 90 16	89, 90, 13	08,00,03
5	-167 90 15	-159, 91, 12	07,01,03
6	166 90 15	159, 90, 12	07,00,03
7	-38 -12 -15	-39, -12, -22	01,00,07
8	38 -12 -15	38, -12, -22	00,00,07
9	-38 -46 -14	-38, -40, -19	00,06,05
10	38 -46 -14	38, -40, -18	00,06,04
11	-38 -75 00	-37, -73, -05	01,02,05
12	38 -75 00	37, -73, -04	01,02,04
13	-38 -117 01	-39, -117, -06	01,00,05
14	37 -117 01	39, -117, -05	02,00,04
15	-38 -167 -14	-40, -167, -25	02,00,09
16	37 -167 -14	40, -167, -24	03,00,10
17	-43 -207 -11	-39, -207, -12	04,00,01
18	42 -207 -11	40, -207, -11	02,00,00

Tabla 4.3: Centros de Masa para las articulaciones del Robot Humanoide Bioloid PremiumTM

Articulación	Tensor de Inercia SW (g/mm^2)	Tensor de Inercia MV (g/mm^2)	Diferencia
0	992393 -2073 -1276 -2073 560882 76222 -1276 76222 1132970	1533449 -24870 424 -24870 603307 -49744 424 -49744 1627050	-541056 -22797 852 -22797 -5472425 26478 852 26478 -494080
1 = 2	5057 0 448 0 2105 0 448 0 4935	5043 356 554 356 2258 282 554 282 5063	14 356 106 356 153 282 106 282 128
3=4	7800 367 0 367 19126 0 0 0 23979	15445 2266 30 2266 35572 2 30 2 41957	7645 1899 30 1899 16446 2 30 2 17978
5 =6	5522 552 3273 522 27123 35 3273 35 28867	13097 82 3881 82 51641 9 3881 9 54603	7575 470 608 470 24518 26 608 26 25736
7=8	13799 0 0 0 13185 110 0 110 2746	15645 1 0 1 14383 989 0 989 3269	1846 1 0 1 1198 879 0 879 523
9=10	13799 0 0 0 13185 110 0 110 2746	74065 190 230 190 64269 618 230 618 37864	60266 190 230 190 51084 508 230 230 508 35118
11 =12	16624 0 0 0 13579 1342 0 1342 24938	19213 985 434 985 14496 2376 434 2376 27580	2589 985 434 985 917 1034 434 1034 20374
13=14	37213 702 101 702 13545 124 101 124 43536	61967 1512 200 1512 27368 2052 200 2052 63910	24754 810 99 810 13823 1928 99 1928 20375
15=16	52973 114 1010 114 54243 1444 1010 1444 11067	75927 247 303 247 65297 247 303 247 36938	22954 133 707 133 11054 1197 707 1197 25871
17=18	33435 779 617 779 39224 804 617 804 11521	35701 8 10 8 41921 1435 10 1435 12745	2266 771 607 771 2697 631 607 631 1224

Tabla 4.4: Tensores de Inercia para las articulaciones del Robot Humanoide Biolooid PremiumTM

4.2.7. Resultados

Se diseñó un modelo virtual con comportamiento dinámico para el robot Bioloid PremiumTM, el cual permite observar el comportamiento realista del robot ante un conjunto de trayectorias para sus 18 grados de libertad. Se genera además un reporte con los parámetros dinámicos de los eslabones del robot que serán utilizados en el diseño del controlador.

No es necesario definir las restricciones cinemáticas, ya que el sistema detecta automáticamente los límites de cada grado de libertad, al detectar las autocolisiones.

Se realizaron pruebas para verificar que no se interpenetraran los actores del robot con actores externos, el torso permanece fijo, por lo que el robot da la apariencia de estar *colgado*.

La Figura 4.10 muestra el comportamiento del robot al colocar una esfera cerca del brazo izquierdo, derecho y entre las piernas del robot.

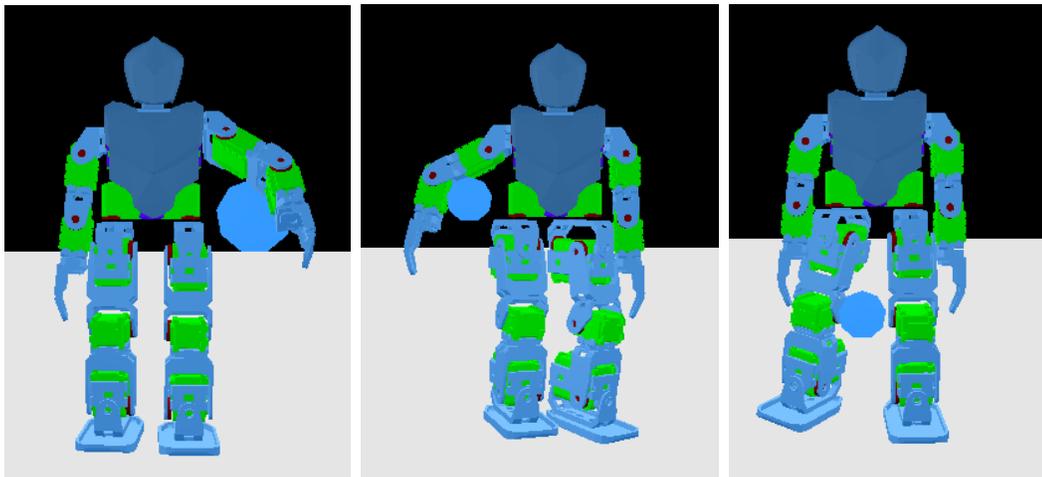


Figura 4.10: Robot interactuando con esfera

La Figura 4.11 muestra cómo cae el robot al soltar el torso que permaneció fijo en los experimentos anteriores,

Se colocaron dos esferas frente al robot para observar su comportamiento al caer cerca de ellas, la Figura 4.12 muestra los resultados.

Se analizaron las diferencias en los parámetros de masa, centros de masa y tensor de inercia. En el caso de masa, aunque para los segmentos del 1 al 18 las diferencias fueron menores

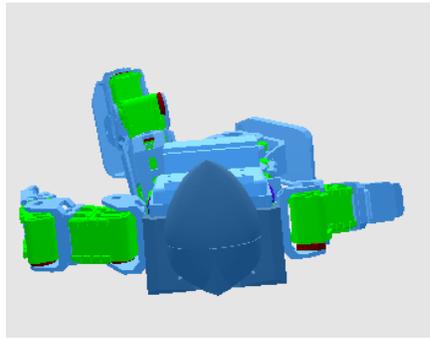


Figura 4.11: Robot bajo efectos de la fuerza de gravedad

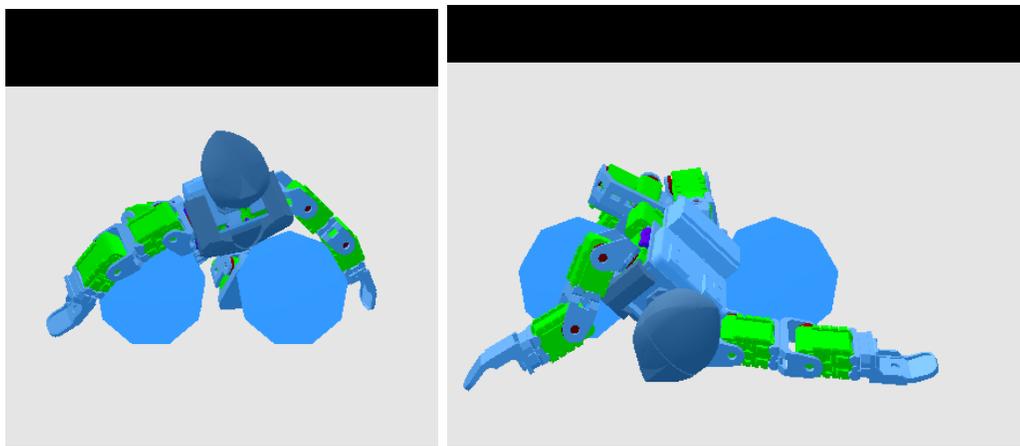


Figura 4.12: Comportamiento del robot al caer sobre dos esferas

a 7 gramos, para el torso la diferencia fue de 313.79 gramos, ésto puede atribuirse a que algunas de las piezas proporcionadas por el fabricante no tienen el nivel de detalle requerido; para centros de masa la diferencia mayor fue de 11 mm en un eje de uno de los eslabones. En cuanto a los tensores de inercia se despliega la diferencia entre las matrices proporcionadas por el ensamble realizado en SolidWorksTM y la obtenidas en el modelo virtual, aunque no se realizó una interpretación de estas diferencias se observa que el comportamiento del modelo virtual es más realista cuando se utilizan las matrices proporcionadas por SolidWorksTM.

Cabe mencionar que no se modeló la fricción de los elementos, ni de contacto con la superficie, por lo que este modelo virtual puede ser mejorado al definir correctamente estos parámetros.

4.3. Teleoperación de Humanoide Virtual con Comportamiento Dinámico mediante captura de movimientos utilizando el modelo de Borst e Indugula

Los modelos virtuales pueden controlarse de forma cinemática o dinámica; los modelos con comportamiento cinemático pueden posicionarse en cualquier parte de la escena sin importar si hay otro objeto con comportamiento cinemático en la misma posición, es decir, pueden presentarse traslapes entre objetos. Un modelo virtual con comportamiento dinámico respeta las leyes de física: un cuerpo permanecerá en reposo o en movimiento a menos que alguna fuerza actúe sobre él; la aceleración de un objeto es directamente proporcional a la fuerza que actúa sobre él, e inversamente proporcional a su masa, la dirección de la aceleración es la misma de la fuerza aplicada; siempre que un objeto ejerce una fuerza sobre un segundo objeto, este ejerce una fuerza de igual magnitud y dirección pero en sentido opuesto sobre el primero. En consecuencia, al posicionar un objeto de este tipo en algún lugar de la escena este caerá hasta chocar con la superficie con velocidad y aceleración que dependerán de su peso y de su forma, así mismo detecta colisiones con objetos presentes en la escena virtual.

El posicionamiento de un modelo virtual con comportamiento dinámico se realiza mediante fuerzas y torques, por lo que para lograr la teleoperación de dicho humanoide sería necesario calcular los torques para las 18 articulaciones para que el robot logre la postura solicitada para cada frame; para ello sería necesario resolver el modelo dinámico del robot.

En esta sección se propone utilizar el modelo Borst e Indugula[109][110] para calcular los torques requeridos en cada articulación para llegar a la postura solicitada. Dicho modelo toma los datos de las variables articulares del sistema de captura de movimiento para mover un modelo virtual con comportamiento cinemático, éste, se une mediante un resorte - amortiguador a un modelo con comportamiento dinámico.

El modelo Borst e Indugula [109, 110] ha sido aplicado para el agarre de objetos en entor-

nos virtuales en [111] y [97], por otra parte en [44] y [46] se utilizan resortes-amortiguadores para imitación de movimientos en parta alta de robots humanoides.

La Figura 4.13 muestra la arquitectura del sistema propuesta, en la escena virtual se crean dos robots humanoides con las formas, actores y uniones descritos en la sección anterior, uno de ellos tendrá comportamiento cinemático, es decir, se controlará por posiciones, el otro tendrá comportamiento dinámico. Los torques requeridos para que el robot con comportamiento dinámico logre las posturas del humanoide con comportamiento cinemático será calculadas mediante el modelo Borst e Indugula.

En la Tabla 3.2 se observan la relación de los datos proporcionados por el sistema de captura de movimientos que serán utilizados en las articulaciones del robot para realizar la teleoperación, estos datos son utilizados para mover el humanoide con comportamiento cinemático.

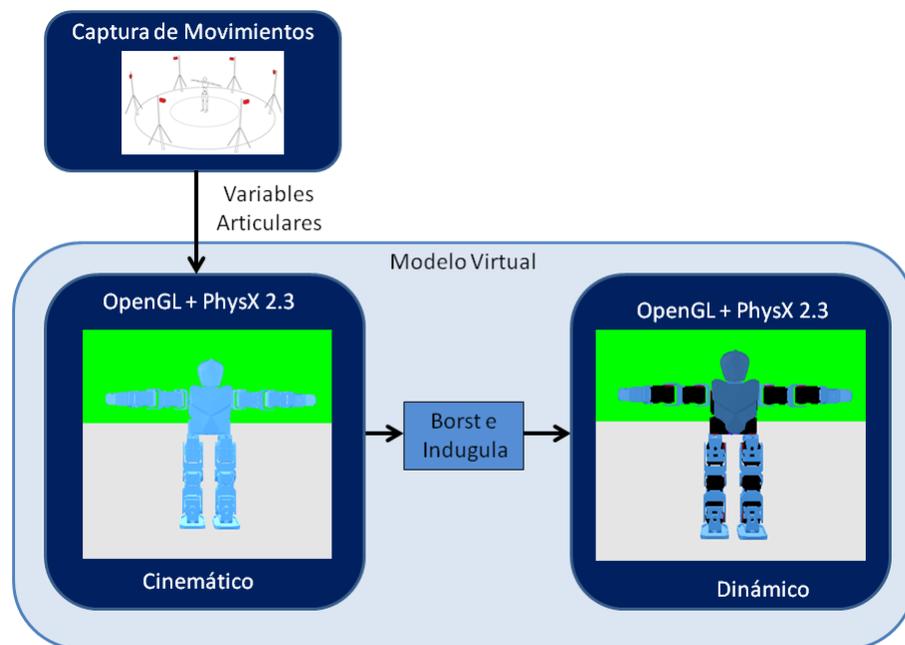


Figura 4.13: Arquitectura del Sistema

4.3.1. Modelo Borst e Indugula

Calcula las fuerzas y torques necesarios para mover el humanoide dinámico a la posición del humanoide cinemático, para calcular las fuerza requeridas se hace uso de resortes lineales y para los torques resortes torsionales.

Resorte Lineal

El resorte lineal permite conocer la fuerza necesaria para mover al actor a la posición deseada. La Ecuación 4.2 muestra que la Fuerza, \vec{F} , depende de la diferencia entre la posición del actor cinemático \vec{P}_c y el actor dinámico \vec{P}_d , así como la diferencia entre las velocidades dinámica \vec{v}_d y cinemática \vec{v}_c , cada una de ellas modificada por las constantes de resorte K_L y de amortiguamiento b_L .

$$\vec{F} = K_L(\vec{P}_c - \vec{P}_d) - b_L(\vec{v}_d - \vec{v}_c) \quad (4.2)$$

Para calcular las \vec{v}_c y \vec{v}_d se utilizan las Ecuaciones 4.3 y 4.4, donde P_c^t y P_d^t representan las posiciones del actor cinemático y dinámico en el frame actual, de la misma manera P_c^{t-1} y P_d^{t-1} simbolizan las posiciones cinemática y dinámica para el frame anterior; Δt es el tiempo transcurrido entre un frame y el otro.

$$\vec{v}_c = \frac{P_c^t - P_c^{t-1}}{\Delta t} \quad (4.3)$$

$$\vec{v}_d = \frac{P_d^t - P_d^{t-1}}{\Delta t} \quad (4.4)$$

Resorte Torsional

Permite conocer el torque requerido para mover al actor a la posición angular deseada. La magnitud del torque se calcula mediante la Ecuación 4.5 donde θ_c y θ_d representan las posiciones angulares del actor cinemático y dinámico, respectivamente, así mismo, ω_d

y ω_c simbolizan las velocidades angulares dinámica y cinemática; ambas diferencias son modificadas por las constantes de resorte k_T y amortiguamiento b_T .

$$\tau = k_T(\theta_c - \theta_d) - b_T(\omega_d - \omega_c) \quad (4.5)$$

La dirección del torque para cada actor es el eje de la unión, excepto para el actor base (Torso) ya que éste puede rotar en tres ejes, por lo que la Ecuación 4.5 incrementa su complejidad, para éste actor se calcula el torque de resorte y de amortiguamiento por separado, ya que es muy probable que los dos tengan direcciones diferentes.

Para el torque de resorte, es necesario conocer la diferencia entre las posiciones angular cinemática y dinámica, para ello se hace uso de cuaterniones, recordando que si q_c representa el cuaternión para Torso Cinemático y q_d para el Torso Dinámico, los dos con respecto al mismo marco, $q_c^*q_d$ representa el desplazamiento angular entre q_c y q_d con respecto al marco de q_c , q_c^* es el cuaternión conjugado de q_c . El eje y ángulo extraído de $q_c^*q_d$, θ_q y se modifica por la constante de resorte $K_{T_{torso}}$ como se muestra en la ecuación 4.6. El torque resultante es con respecto al marco del torso dinámico.

$$\tau = k_{T_{torso}}(\theta_q) \quad (4.6)$$

Para el torque de amortiguamiento es necesario calcular primero las diferencias en posición angular para el torso cinemático y dinámico entre los frames actual y anterior mediante las Ecuaciones 4.7 y 4.8.

$$qc = qc^t(qc^{t-1})^* \quad (4.7)$$

$$qd = qd^t(qd^{t-1})^* \quad (4.8)$$

donde qc^t y qd^t son los cuaterniones que indican la orientación del torso en el frame

actual y qc^{t-1} y qd^{t-1} son los cuaterniones del frame anterior, para calcular la diferencia de velocidades angulares se utiliza la ecuaciones 4.10 y 4.9.

$$\vec{\omega}_d = \frac{\vec{\theta}_{qd}}{\Delta t} \quad (4.9)$$

$$\vec{\omega}_c = \frac{\vec{\theta}_{qc}}{\Delta t} \quad (4.10)$$

donde $\vec{\theta}_{qc}$ son los ángulos extraído de qc y $\vec{\theta}_{qd}$ son los ángulos extraído de qd en los ejes x, y, z .

Finalmente se obtiene el torque de amortiguamiento mediante 4.11

$$\tau = -b_{T_{Torsio}}(\vec{\omega}_d - \vec{\omega}_c) \quad (4.11)$$

4.3.2. Resultados

Se utilizaron resortes lineales en el marco base (Pelvis) y en los 4 efectores finales (Manos y pies), para los 18 grados de libertad del robot se utilizaron 18 resortes torsionales.

Se utilizó un archivo *amc* correspondiente a caminado con un total de 517 frames, se realizó una comparación entre la posición angular de cada segmento en el humanoide cinemático y el dinámico, la Tabla 4.5 muestra la diferencia absoluta promedio entre la variable cinemática y dinámica, calculada para los 517 frames.

La Figura 4.14 muestra el humanoide dinámico, mientras que en la Figura 4.15 se presentan los humanoides cinemático y dinámico (traslapados), mostrando diferencias mínimas con los datos mostrados en la Tabla 4.5.

Ahora bien, para evaluar la teleoperación es necesario analizar la información que se deseaba teleoperar y la que se obtiene, es decir, comparar el archivo de captura de movimientos con los movimientos logrados por el humanoide virtual con comportamiento dinámico. La Figura 4.16 muestra comparaciones entre el archivo de captura de movimiento y

Tabla 4.5: Comparación entre el movimiento deseado y el obtenido

Variable	Diferencia Promedio
X	0.87 mm
Y	5.14 mm
Z	2.11 mm
Roll	1.59 grados
Pitch	2.26 grados
Yaw	0.92 grados
Motor 1	1.21 grados
Motor 2	1.59 grados
Motor 3	0.68 grados
Motor 4	1.21 grados
Motor 7	0.79 grados
Motor 8	0.73 grados
Motor 9	0.36 grados
Motor 10	0.44 grados
Motor 11	1.46 grados
Motor 12	1.50 grados
Motor 13	1.51 grados
Motor 14	1.38 grados
Motor 15	1.26 grados
Motor 16	1.08 grados
Motor 17	0.07 grados
Motor 18	0.11 grados

el humanoide virtual dinámico. Se observa que, debido a las limitaciones de la estructura cinemática del robot humanoide, no es posible plasmar información sobre eslabones como el Torso del esqueleto del sistema de captura de movimientos, el cual está dividido en espalda alta, espalda baja y torax; para los brazos, solo se utilizó información de los hombros, ya que la articulación del codo del esqueleto del archivo amc solo proporciona información en el eje x mientras que el humanoide virtual tiene el eslabón del codo con eje de rotación en el eje y .

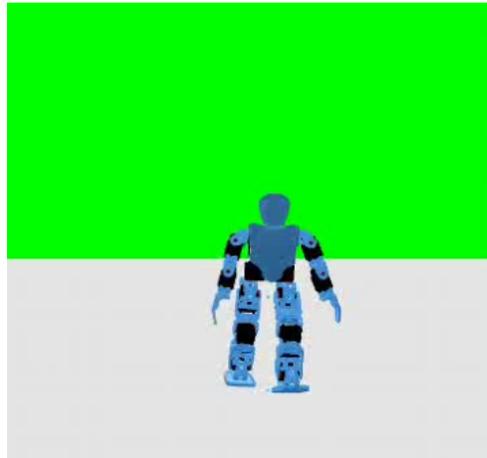


Figura 4.14: Humanoide Dinámico

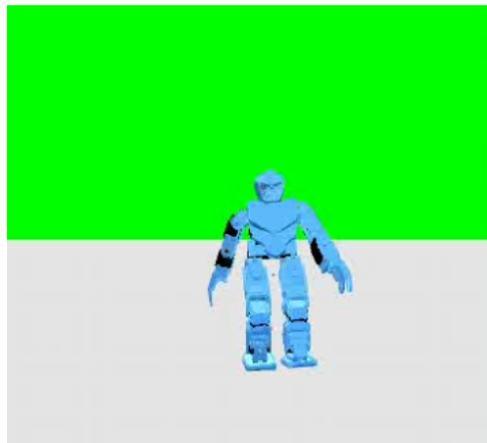


Figura 4.15: Humanoide Cinemático y Dinámico

4.3.3. Conclusiones

Los resultados mostrados en la Tabla 4.5 muestran que es posible teleoperar un humanoide virtual, evitando autocolisiones y/o colisiones con objetos en el ambiente virtual, mediante captura de movimientos, utilizando el modelo de Borst e Indugula con diferencias en posición de cinco mm y en posiciones angulares menores a dos grados; sin embargo, los resortes lineales en los efectores finales (manos y pies) no permiten verificar estrategias de control en locomoción, ya que el robot no se caerá al fallar el control.

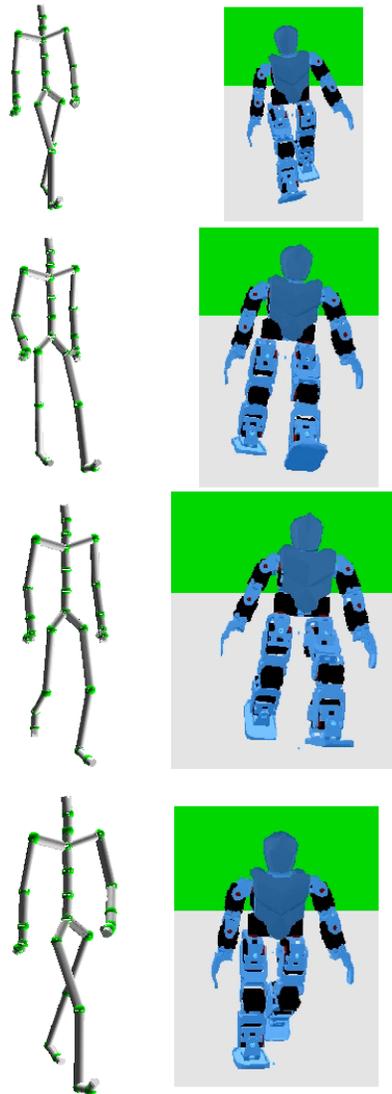


Figura 4.16: Comparación entre MOCAP y humanoide dinámico

4.4. Teleoperación en línea de Robot Humanoide mediante el modelo Borst e Indugula

En esta sección se presenta el sistema desarrollado para realizar la teleoperación en línea del robot Humanoide Bioloid Premium TM en su configuración A mediante captura de

movimientos y utilizando el modelo resorte amortiguador.

4.4.1. Interfaz del sistema de Teleoperación

La Figura 4.17 muestra la arquitectura del sistema propuesto, el sistema fue desarrollado bajo la plataforma Visual C++.

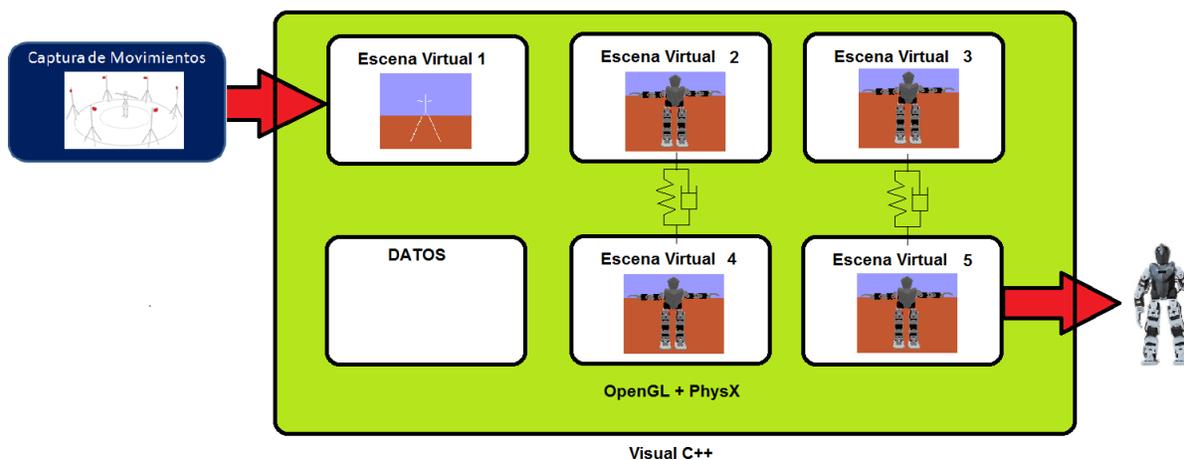


Figura 4.17: Interfaz del sistema de Teleoperación

Para la captura de movimientos, al no contar con el equipo físico, se utilizaron los segmentos de captura disponibles en la base de datos de la Carnegie Mellon University [70], se interpretaron los archivos *asf* y *amc* de captura de movimientos; en la escena virtual uno muestra la interpretación de dicha información. La escena virtual dos muestra un humanoide con comportamiento cinemático creado como se comentó en la sección 4.1, la Tabla 3.2 indica la selección de la información del archivo *amc* que se utilizará para lograr las posturas del humanoide en cada frame. Las articulaciones 5 y 6 se mantienen fijas debido a que los archivos de movimiento utilizados se requiere movimiento en el eje x y éstas articulaciones tienen su grado de libertad en el eje y . La escena virtual cuatro muestra un humanoide con comportamiento dinámico creado como se comentó en la sección 4.2, la masa de cada segmento fue un parámetro de diseño utilizando las masas reales indicadas en la Tabla 3.7. Los centros de masa y tensores de inercia se indicaron como parámetros de diseño de acuerdo a

los datos obtenidos mediante el ensamble realizado en el software *SolidWorksTM*. Las posturas del humanoide en cada frame se alcanzaron mediante el modelo resorte amortiguador de Borst e Indugula [109] [110], se utilizaron 18 resortes torsionales y 4 resortes lineales en los 4 efectores finales (manos y pies) se seleccionaron las constantes de los resortes lineales de tal manera que no impidan la caída del robot cuando el ZMP salga de la zona de estabilidad. Se muestra el ZMP calculado mediante las Ecuaciones 3.102 y 3.103 así como la zona formada por los polígonos de los pies en soporte doble y del pie de soporte en caso de soporte simple. La escena virtual tres presenta un humanoide con comportamiento cinemático que identifica si el ZMP de la escena virtual cuatro (humanoide dinámico), sale de la zona definida por el polígono de soporte y realiza la corrección necesaria, el control utilizado es básico, es proporcional de acuerdo a la posición del ZMP y se modifica la orientación de la cadera en el sentido opuesto. La escena virtual cinco presenta un humanoide con comportamiento dinámico que imita mediante el modelo borst e indugula el comportamiento del humanoide con comportamiento cinemático estabilizado. Las posiciones angulares de este humanoide son las que serán enviadas al robot humanoide real.

Debido a la cantidad de escenas virtuales mostradas, la frecuencia de muestreo se reduce a 10 frames por segundo. Siguiendo el diagrama de flujo de la Figura 3.9 se realiza la comunicación con el Robot Humanoide enviando la posición de los 18 motores.

En la sección de datos se muestra la información para seleccionar mediante el teclado el video de captura de movimiento a utilizar, una vez seleccionado, muestra los frames del video, el frame actual, así como el control de la cámara para la visualización.

4.4.2. Resultados

El primer experimento fue verificar la posición inicial, se selecciona el primer frame de un segmento de movimiento definido y se verifica que todas las escenas virtuales logren la posición solicitada, así como el robot humanoide, como se observa en la Figura 4.18.

El segundo experimento fue observar la caída del robot, se desactivan los resortes en el

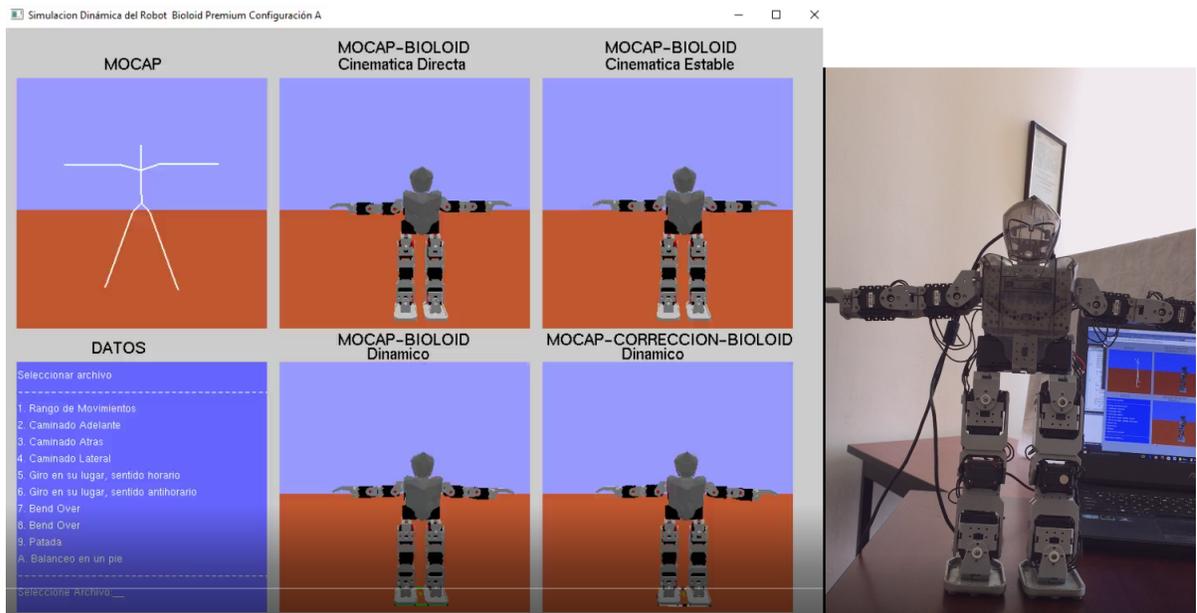


Figura 4.18: Comparación entre robot virtual y robot real para posición inicial

humanoide dinámico y se verifica la caída del humanoide virtual y del humanoide real, las Figura 4.19 muestra comportamientos distintos en el modelo virtual y en el real.

El tercer experimento fue verificar si el robot virtual presenta una caída cuando el ZMP sale de la zona de estabilidad formada por los polígonos de soporte. Se observa que al iniciar la teleoperación, el robot virtual se mantiene estable por más tiempo que el robot real y aunque el ZMP sale de la zona de estabilidad el robot cae muy lentamente, comportamiento que puede ser ocasionado por el uso de resortes en los efectores finales. También se observa que al no controlar la velocidad de los movimientos realizados por el robot humanoide, se presentan cambios bruscos de posición y al reducir la velocidad el movimiento realizado es muy lento.

El cuarto experimento fue realizar la teleoperación de un segmento de captura de movimientos de caminado, sin embargo, el robot se desbalanceó al cambiar de soporte doble a soporte simple para intentar dar el primer paso.

El quinto experimento fue similar al anterior pero para un segmento de captura de movimientos de caminado lateral, el resultado fue el mismo.

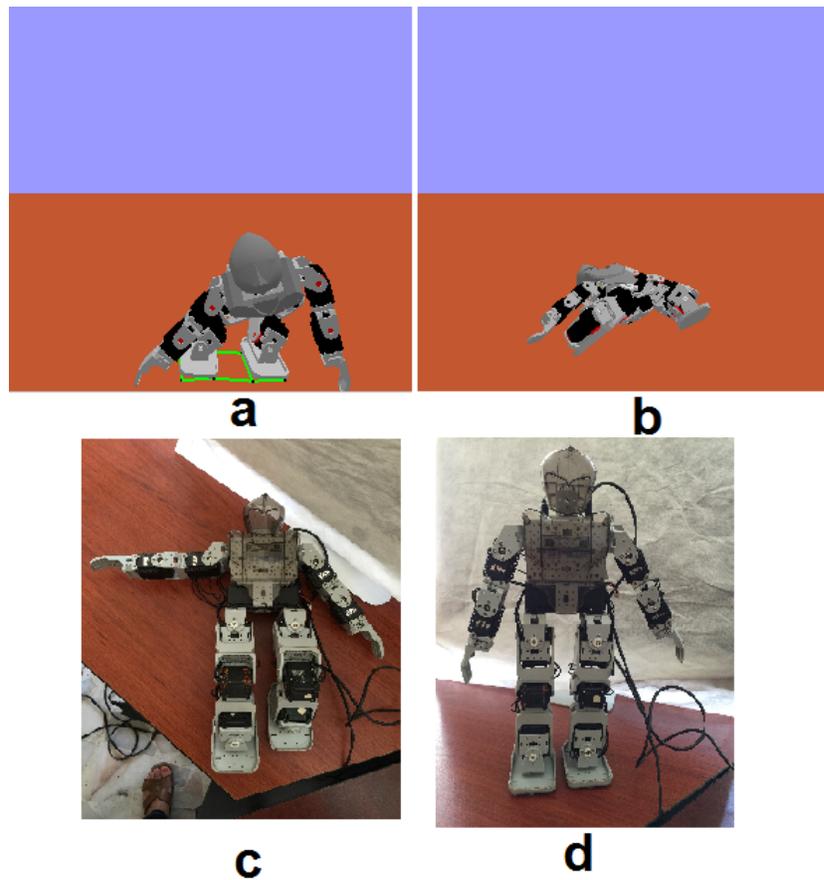


Figura 4.19: Comparación de la caída del robot virtual y el real

4.5. Publicaciones

4.5.1. Artículos

2015 MODELO VIRTUAL CON COMPORTAMIENTO DINAMICO DEL ROBOT HUMANOIDE BIOLOID PREMIUM. ISSN:20077114, Rodríguez Vega Dora, Zaldívar Colador Ulises, Nuñez Nalda José., AMRob Journal, Robotics: Theory and Applications, Vol.3,

2013 INTEGRACION DE UN MOTOR DE MODELADO BASADO EN FÍSICA PARA EVITAR EL CÁLCULO DE CINEMATICA INVERSA EN LA OPERACIÓN GESTUAL DE UN MANIPULADOR DE 7 GDL. ISSN:20077114, Ulises Zaldívar Colado,

Roberto Bernal Guadiana, Dora Aydee Rodriguez Vega, Xiomara Zaldívar Colado, Claudia Marmolejo Rivas, Saúl Guerrero Cárdenas, Carolina Martínez Tirado, Jose Víctor Nuñez Nalda., Amrob Journal, Robotics:Theory and Applications,, Vol.2, Pag.93-98,

2012 EXPLICIT ANALYTIC SOLUTION FOR INVERSE KINEMATICS OF BIOLOID HUMANOID ROBOT. ISSN: 9781467346504, Nunez J Victor, Briseno Abraham, Rodriguez Dora, Ibarra J Manuel, Rodriguez Victor, SBRLARS IEEE 2012, Vol. , Pag.33-38,

2012 MODELADO Y ANIMACIÓN MEDIANTE CAPTURA DE TRAYECTORIAS CON EL USO DE UN BODY MOTION CAPTURE, Rodriguez Vega D, Briseño Cerón A, Saucedo Ugalde I, Nuñez Nalda J,, Academia Journals 2012,

Memorias en Congresos

2015 TELEOPERACION DE HUMANOIDE VIRTUAL MEDIANTE CAPTURA DE MOVIMIENTOS, USANDO EL MODELO RESORTE AMORTIGUADOR, Congreso Mexicano de Robótica 2015.

2014 MODELO VIRTUAL CON COMPORTAMIENTO DINAMICO DEL ROBOT HUMANOIDE BIOLOID PREMIUM, Congreso Mexicano de Robótica 2014.

2013 IMITACIÓN DE MOVIMIENTOS ENTRE ROBOTS HUMANOIDES MEDIANTE TÉCNICAS DE TELEOPERACIÓN ROBOT MAESTRO ESCLAVO, Zaldívar Colado X, Zaldivar Colado U, Hernández Payán J, Bernal Guadiana R, Marmolejo Rivas C, Rodríguez Vega D, Reyes Viera R,Rodriguez López H, Congreso Mexicano de Robotica 2013

2012 TELEOPERACIÓN EN LÍNEA DE UN HUMANOIDE VIRTUAL MEDIANTE MANIPULACIÓN DE HUMANOIDE REAL, Rodriguez Vega D, Zaldivar Colado U, Nuñez Nalda J, Briseño Cerón A,, Congreso Mexicano de Robotica 2012.

2011 INTERFACE PARA INTERACCIÓN CON ROBOTS HUMANOIDES, Víctor Núñez, Josué Sapiens, Dora Rodríguez, Abraham Briceño, Víctor Rodríguez. Congreso Mexicano de Robótica 2011.

Capítulo 5

Análisis de Resultados y Trabajo Futuro

5.1. Análisis de Resultados

Este trabajo de investigación se inició con la idea de realizar teleoperación de un humanoide mediante captura de movimientos, en línea y de forma estable, inicialmente se contaba con el sistema de captura de movimiento e incluso se logró realizar algunos experimentos de captura de movimiento transmisión de la información de posición y orientación de los eslabones en tiempo real a través de una red local, sin embargo por cuestiones institucionales ya no se tiene acceso a dicho equipo por lo que se utilizaron segmentos de captura de movimientos disponibles en la base de datos de la Carnegie Mellon University [70], éstos datos no son preprocesados, es decir, se toma un dato del archivo que indica posición y orientación de cada eslabón del esqueleto de captura de movimiento predefinido y se procesa, es por eso, que se dice que la teleoperación es en línea aunque no es posible obtener los datos de latencia.

Posteriormente se diseñó un ambiente virtual que nos permitiera observar la captura de movimiento y la teleoperación de un humanoide virtual, inicialmente éste humanoide solo utilizaba librería de gráficos, y su comportamiento era manipulado mediante posiciones, las cuales se tomaron de la interpretación y selección de información que proporciona el archivo de captura de movimientos, éste humanoide virtual tuvo como desventaja que no respetaba

las restricciones cinemáticas, es decir, era posible solicitarle una posición factible para el ser humano, pero imposible para el robot por su estructura, como se muestra en la Figura 5.1

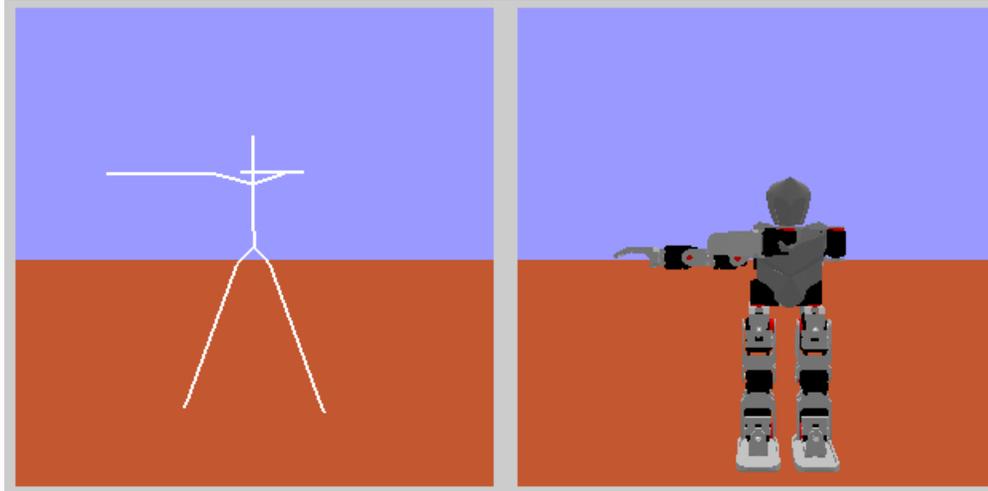


Figura 5.1: Posición solicitada al robot virtual que el robot real no puede realizar

Se incluyó en el ambiente propuesto un modelo virtual con comportamiento dinámico, es decir, mediante la librería de un motor de física ($PhysX^{TM}$) el modelo virtual tiene propiedades como masa, volumen, tensor de inercia y se mueve de acuerdo a las leyes de la física respetando colisiones con otros objetos en el ambiente o con otros eslabones del mismo robot. Con este modelo se resolvieron las restricciones cinemáticas como se muestra en la Fig 5.2, sin embargo surge otro reto, para teleoperar este robot es necesario realizar por fuerzas y torques (ya no por posiciones) por lo que es necesario realizar el cálculo de estas fuerzas. la primera opción es calcular el modelo dinámico del robot el cual conlleva cierto nivel de complejidad, la otra opción es utilizar un modelo resorte-amortiguador que calcule las fuerzas necesarias para que el modelo virtual llegue a cierta posición, en [109, 110] se presenta el modelo de Borst e Indugula para realizar un trabajo similar en una mano virtual.

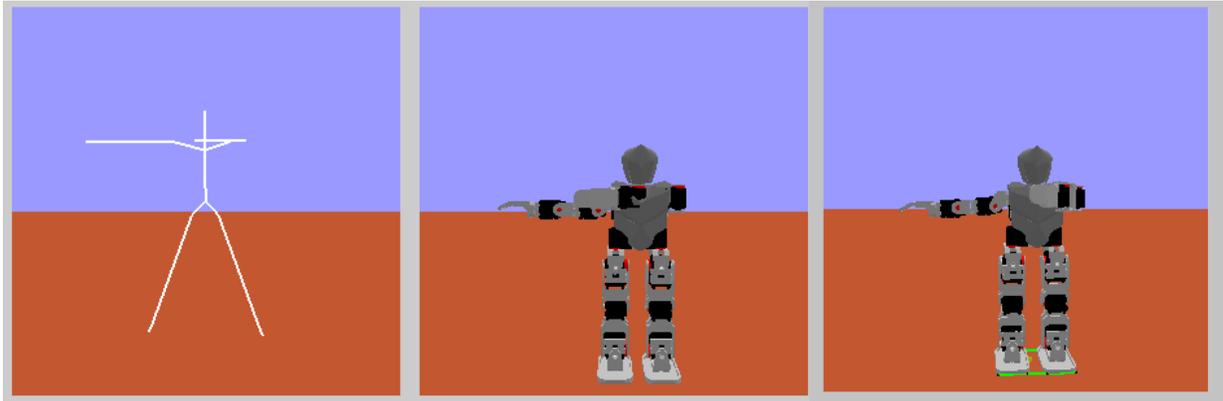


Figura 5.2: El modelo virtual con comportamiento dinámico resuelve las restricciones cinemáticas

El ambiente virtual con un humanoide virtual con comportamiento dinámico controlado por el modelo de resorte - amortiguador de Borst e Indugula, sin embargo los resortes lineales colocados en el torso y en las 4 extremidades, que permiten que la teleoperación del humanoide virtual sea muy precisa (errores menores a 3 grados) son los mismos que no nos permiten observar si el movimiento solicitado provocará que el robot real se desbalancee y se caiga, por lo que se eliminaron los resortes lineales del torso del robot y se redujeron las constantes de resorte y amortiguador de los resortes lineales de las extremidades para que sea posible observar una caída del robot en caso de desbalance.

En este punto, se analizó que algunas posturas que aparecían estables en el modelo virtual no resultaban estables en el robot real. Después de un exhaustivo análisis se observa que no se modeló la fricción de contacto con la superficie, lo que permite al robot virtual deslizar la planta del pie en la superficie manteniendo la estabilidad, situación que en el robot real terminó en una inminente caída.

Todo el esfuerzo inicial se enfocó inicialmente en que el robot real se moviera como el robot virtual con comportamiento dinámico, pero después de observar el problema de la fricción se analizó el comportamiento de la caída del robot desde la posición inicial, al desactivar los motores, el comportamiento del humanoide real fue muy diferente al humanoide

virtual, esto es debido a que no se modeló el comportamiento mecánico del motor, es decir, en el modelo virtual al desactivar las uniones de los motores el humanoide virtual cae considerando únicamente el peso y volumen de sus eslabones (y de los elementos que lo componen) pero no toma en cuenta la fricción producida por el comportamiento del motor. La Figura 4.19 a muestra lo que pasa al modelo virtual al desactivar los motores con un parámetro de fricción fijo (10000), al eliminar la fricción además de desactivar los motores las extremidades inferiores se deslizan en la superficie como se muestra en el inciso b de la Figura. Por otra parte en el robot real, partiendo de la posición inicial, al desactivar los motores presenta dos comportamientos diferentes en el primero de ellos, mostrado en el inciso c, el robot cae al suelo y en el segundo, inciso d, el robot sólo deja caer los brazos pero permanece de pie.

En este punto ya se cuenta con la interpretación de captura de movimientos, teleoperación de humanoide cinemático por posiciones, dinámico por torques virtuales, sin embargo, el robot utilizado, Robot Bioid PremiumTM utiliza motores dinamixel controlados por posiciones y no se tiene acceso a controlarlos por torque, por lo que la teleoperación se realiza utilizando las posiciones del modelo virtual dinámico.

Se observa que el movimiento a teleoperar, estable en el ser humano, resulta inestable tanto en el modelo virtual como en el robot real, aunque la caída se produce en instantes diferentes, este es debido a que, como ya se mencionó antes el modelo virtual no considera las fricciones, y tampoco se modeló el comportamiento del motor.

Se realizó un experimento para tratar de estabilizar el movimiento deseado ajustando la orientación del torso, para el mismo segmento de movimiento aunque en el modelo virtual el comportamiento es estable, el robot real se cae al iniciar el movimiento.

Por último se realizó el experimento de enviar los datos del humanoide virtual con comportamiento cinemático, como era de esperar el robot se cae al inicio del movimiento.

5.2. Conclusiones

Con lo anterior se concluye que realizar la teleoperación de un humanoide virtual mediante captura de movimientos utilizando el modelo de Borst e Indugula no permite identificar si el movimiento teleoperado es inestable para el robot ya que debido a los resortes el robot no caería. Si se reducen las constantes de resorte y amortiguamiento, entonces si es posible observar las caídas del robot sin embargo se hace necesaria una estrategia para mantener la estabilidad; si se redefine la posición de la cintura y/o piernas para ajustar la posición del ZMP, entonces no se imita el movimiento en las piernas y dado que el movimiento de los brazos se limitaría a solo a dos grados de libertad, la imitación realizada sería muy limitada.

Lo anterior confirma lo realizado por Montecillo [16] con el modelo humano normalizado propuesto mostrado en la Figura 2.11 el cual se enfoca en la parte alta del cuerpo y para la parte baja solo observa la posición y orientación de los pies y con esa información imita en soporte doble; para el caminado la posición del pie le permite identificar si es zancada derecha o izquierda y realiza un movimiento preprogramado y estable para la zancada identificada.

En el Capítulo 2 se mencionaron diversas estrategias para controlar la parte alta de diversos humanoides, entre ellas la estrategia de resortes virtuales, sin embargo dichos humanoides tenían configuraciones cinemáticas mayores a las del robot Bioloid Premium TM, algunos autores realizaron teleoperación en robots de pocos grados de libertad (igual o menos de 21) sólo tomando en cuenta los efectores finales imitando sólo posturas en soporte doble y simple.

En cuanto al modelo virtual propuesto se concluye que aunque el modelo propuesto permite observar caídas del robot, para que estas sean más realistas en trayectoria y tiempo es necesario agregar la fricción y el comportamiento real del motor Dynamixel AX-12.

5.3. Trabajo futuro

Se propone la creación de un ambiente virtual con comportamiento dinámico para la representación de movimientos obtenidos mediante captura de movimientos en el humanoide virtual que corresponde al robot humanoide Bioloid Premium TM que permite verificar diferentes estrategias de control para estabilizar el robot. Por lo pronto el ambiente carece de parámetros de fricción y caracterización del motor AX-12 que se consideran como área de oportunidad.

La selección de las constantes de los resortes torsionales y lineales fue seleccionada a prueba y error, por lo que un área de oportunidad es el diseño de un algoritmo de autoajuste.

El motor AX-12 se mueve por posiciones, el manejo de los torques y su control no son accesibles al usuario, por lo que no es posible verificar que los torques virtuales obtenidos corresponden a los requeridos realmente. Como trabajo futuro se propone explorar como verificar estos torques virtuales en un motor real.

Bibliografía

- [1] L. Montesano J. Santos-Victor M. Lopes, F. Melo. Abstraction levels for robotic imitation: overview and computational approaches,. *From Motor Learning to Interaction Learning in Robots, Springer*, 2010.
- [2] I. L. Hsien, L. Yu-Cheng and C. Chi-Li. Evaluation of human-robot arm movement imitation. *Control Conference (ASCC) 2011 8th Asian*, 2011.
- [3] M. Riley, A. Ude K.,K. Wade and C. G. Atkeson. Enabling real-time full-body imitation: a natural way of transferring human movement to humanoids. *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, 2003.
- [4] J. P. Bandera, L. Molina-Tanco, R. Marfil and F. Sandoval. A model-based humanoid perception system for real-time human motion imitation. *Robotics, Automation and Mechatronics, 2004 IEEE Conference on*,, 2004.
- [5] Z. Xiaojun, H. Qiang, D. Peng, W. Dongming, and L. Kejie. Humanoid kinematics mapping and similarity evaluation based on human motion capture. *Information Acquisition, 2004. Proceedings. International Conference on*, 2004.
- [6] H. Inoue Y. Kuniyoshi, M. Inaba. Learning by watching: extracting reusable task knowledge from visual observation of human performance,. *IEEE Trans. Robot. Autom.*, 1994.

- [7] A. Billard S. Calinon, F. Guenter. On learning, representing, and generalizing a task in a humanoid robot,. *IEEE Trans. Syst. Man Cybern.*, 2007.
- [8] W. Sheng A. Thobbi. Imitation learning of hand gestures and its evaluation for humanoid robots. *IEEE Int. Conf. on Information and Automation, ICIA, Harbin, China*, 2010.
- [9] Hsin-Yu Liu ; Wen-June Wang ; Rong-Jyue Wang ; Cheng-Wei Tung ; Pei-Jui Wang ; I-Ping Chang. Image recognition and force measurement application in the humanoid robot imitation. *IEEE Transactions on Instrumentation and Measurement (Volume: 61, Issue: 1)*, 2012.
- [10] J. Koenemann and M. Bennewitz. Whole-body imitation of human motions with a nao humanoid. *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, 2012.
- [11] Nakazawa A. Yokoi-K. Hirukawa H. Ikeuchi K. Nakaoka, S. Generating whole body motions for a biped humanoid robot from captured human dances. *IEEE International Conference on Robotics and Automation*, 2003.
- [12] D. Matsui, T. Minato, K. F. MacDorman and H. Ishiguro. Generating natural motion in an android by mapping human motion. *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005.
- [13] Qiang H. Shusheng L. You S. Zhijie W. Jafri A. R. (Lige, Z. Humanoid motion design considering rhythm based on human motion capture. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.
- [14] M. Do, P. Azad, T. Asfour, and R. Dillmann. Imitation of human motion on a humanoid robot using non-linear optimization. *Humanoids 2008. 8th IEEE-RAS International Conference on*, 2008.

- [15] K. Seungsu, K. ChangHwan, and Y. Bum-Jae. Whole-body motion imitation using human modeling. *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, 2009.
- [16] F. J. Montecillo. *Human Motion Transfer on Humanoid Robot*. PhD thesis, Institut National Polytechnique de Toulouse, 2010.
- [17] Seokmin H. Doik K. Sung-Kyun, K. A walking motion imitation framework of a humanoid robot by human walking recognition from imu motion data. *IEEE-RAS International Conference on Humanoid Robots*, 2009.
- [18] J. Ohya K. Takahashi, T. Sakaguchi. Remarks on a real-time 3d human body posture estimation method using trinocular images. *Proceedings of the 15th Int. Conf. on Pattern Recognition, Barcelona, Spain*, 2000.
- [19] J.R. Wu D. Lee C.F. Juang, C.M. Chang. Computer vision-based human body segmentation and posture estimation. *IEEE Trans. Syst. Man Cybern. Part A*, 2009.
- [20] Motomura S. Kato S. Wakabayashi, A. Communicative humanoid robot control system reflecting human body movement. *International Symposium on Micro-NanoMechatronics and Human Science (MHS)*, 2011.
- [21] C. C. Tung R. Y. Tara H. Fabroyir Z. L. Bao, S. L. Chen and W. C. Teng. Limb vector paralleling: A general approach to translate postures from human to humanoid robots. *Advanced Robotics and Intelligent Systems (ARIS), 2014 International Conference on, Taipei*, 2014.
- [22] W. Indrajit and A. Muis. Development of whole body motion imitation in humanoid robot. *QiR (Quality in Research), International Conference on, Yogyakarta*, 2013.
- [23] J.H. Lee V.V. Nguyen. Full body imitation of human motions with kinect and heterogeneous kinematic structure of humanoid robot. *IEEE SICE International Symposium on System Integration, Kyushu University, Fukuoka, Japan*, 2012.

- [24] W. Adiprawita A.R. Ibrahim. Analytical upper body human motion transfer to naohumanoid robot,. *Int. J. Electr. Eng. Inf.* 4 (4)., 2012.
- [25] Ze-Nian Li Chun Chen Jie Lei, Mingli Song. Whole-body humanoid robot imitation with pose similarity evaluation. *Signal Processing, Elsevier*, 2014.
- [26] F. Burget J. Koenemann and M. Bennewitz. Real-time imitation of human whole-body motions by humanoids. *2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong*, 2014.
- [27] T. Duckett J. Aleotti, A. Skoglund. Position teaching of a robot arm by demonstration with a wearable input device,. *International Conference on Intelligent Manipulation and Grasping*, 2004.
- [28] E. Ratanasena C. Stanton, A. Bogdanovych. Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning. *Proceedings of the Australasian Conference on Robotics and Automation*, 2012.
- [29] W. Hu X. Tong S. Maybank Y. Zhang X. Zhang, C. Li. Human pose estimation and tracking via parsing a tree structure based human model. *IEEE Trans. Syst.Man Cybern. Syst.*, 2014.
- [30] J. Nadel A. Billard, B. Robins and K. Dautenhahn. Building robota, a mini-humanoid robot for the rehabilitation of children with autism. *Assistive Technol., vol. 19*, 2007.
- [31] N. Pollard A. Sofonova and J.K. Hodgins. Optimizing human motion fot the control of a humanoid robot. *Proc. Applied Mathematics and Applications of Mathematics*, 2003.
- [32] Grochow K. Rao R. P. N. Shon, A. P. Robotic imitation from human motion capture using gaussian processes. *5th IEEE-RAS International Conference on the Humanoid Robots*, 2005.

- [33] Yoshida E. Kanehiro F. Laumond J. P. Monin A. Suleiman, W. On human motion imitation by humanoid robot. *IEEE International Conference on Robotics and Automation*, 2008.
- [34] ChangHwan K. Ga-Ram, P. Constructing of optimal database structure by imitation learning based on evolutionary algorithm. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.
- [35] ChangHwan K. Bum-Jae Y. Seungsu, K. Whole-body motion imitation using human modeling. paper presented at the robotics and biomimetics. *IEEE International Conference on Robotics and Biomimetics*, 2008.
- [36] ChangHwan K. Bumjae Y.- Sangrok O. Seungsu, K. Stable whole-body motion generation for humanoid robots to imitate human motions. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009.*, ●.
- [37] K. Yamane, S. O. Anderson, and J. K. Hodgins. Controlling humanoid robots with human motion data: Experimental validation. *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, 2008.
- [38] K. Yamane and J. Hodgins. Simultaneous tracking and balancing of humanoid robots for imitating human motion capture data. *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 2009.
- [39] C. Atkeson A. Ude and M. Riley. Programming full-body movements for humanoid robots by observation. *Robotics and Autonomous Systems*, 2004.
- [40] K. Grochow R. Chalodhorn, D. B. Grimes and R. P. N. Rao. Learning to walk through imitation. *Int. Conf. on Artificial Intelligence (IJCAI)*,, 2007.
- [41] K. Munirathinam, C. Chevallereau, and S. Sakka. Offline imitation of a human motion by a humanoid robot under balance constraint. *New Trends in Medical and Service Robots: Challenges and Solutions, Springer International Publishing*, 2014.

- [42] G. Z. Yang K. Teachasrisaksakul, Z. q. Zhang and B. Lo. Imitation of dynamic walking with bsn for humanoid robot. *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, 2015.
- [43] T. Asfour R. Dillmann C. Simonidis S. Gaertner, M. Do and W. Seemann. Generation of human-like motion for humanoid robots based on marker-based motion capture data. *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, 2010.
- [44] Dongheui L. Nakamura Y. Ott, C. Motion capture based human motion recognition and imitation by direct marker control. *8th IEEE-RAS International Conference on Humanoid Robots, 2008. Humanoids 2008.*, 2008.
- [45] Murillo Campos D. Zaldivar Colado X. Osuna Peraza E. Nuñez Nalda V. Zaldivar Colado, U. Interaction technique for virtual robot stabilization with dynamic behavior. *Fifth International Congress on Electronics and Biomedical Engineering, Computer and Informatics (CONCIBE SCIENCE 2009), Guadalajara, México.*, 2009.
- [46] S. Sung Yul and K. ChangHwan. On-line human motion transition and control for humanoid upper body manipulation. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010.
- [47] D. Ballard J Cooper. Realtime, physics-based marker following. *Motion in Games: 5th International Conference, MIG 2012, Rennes, France, November 15-17, 2012. Proceedings, Springer Berlin Heidelberg*, 2012.
- [48] Jin Tak Kim, Jongwon Lee, Hyogon Kim, Jaehong Seo, Sang-uk Chon, Byungyun Park, Sangdeok Park, and Jungsan Cho. Method for improving the position precision of a hydraulic robot arm: dual virtual spring–damper controller. *Intelligent Service Robotics*, 2016.

- [49] L. Boutin, A. Eon, S. Zegloul, and P. Lacouture. An auto-adaptable algorithm to generate human-like locomotion for different humanoid robots based on motion capture data. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010.
- [50] Y. Woosung, C. Nak Young, K. ChangHwan, and Y. Bum Jae. Locomotion imitation of humanoid using goal-directed self-adjusting adaptor. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006.
- [51] van den Kieboom J. Dzeladini F. Ude A. Gams, A. and A.J. Ijspeert. Real-time full body motion imitation on the coman humanoid robot. *Robotica*,, 2015.
- [52] A. Gams T. Petric N. Sugimoto A. Ude R. Vuga, M. Ogrinc and J. Morimoto. Motion capture and reinforcement learning of dynamically stable humanoid movement primitives,. *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013.
- [53] Chih-Lyang Hwang, Bo-Lin Chen, Hsing-Hao Huang, and Huei-Ting Syu. Hybrid learning model and mmsvm classification for on-line visual imitation of a human with 3-d motions. *Robot. Auton. Syst*, 2015.
- [54] A. Boeing and T. Braunl. Evaluation of real-time physics simulation systems. *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia GRAPHITE '07*, 2007.
- [55] Axel Seugling and Martin Rölin. Evaluation of Physics engines and implementation of a physics module in a 3d-authoring tool. *Umeå University*, 2006.
- [56] S. Yichao, X. Rong, and C. Jian. Hubot—a modeling and simulation platform based on physx for multi-link robot. *Digital Media and Digital Content Management (DMDCM), 2011 Workshop on*, 2011.
- [57] T. Ishida M. Fujita, Y. Kuroki and T. T. Doi.

- [58] Y. Sakagami. The intelligent asimo: System overview and integration. *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*, 2002.
- [59] A. Cichocki N. Yunjun, K. Bonkon and C. Seungjin. Gom-face: Gkp, eog, and emg-based multimodal interface with application to humanoid robot control. *IEEE Trans. Biomed. Eng.*, vol. 61, no. 2, 2014.
- [60] B. Robins J. Wainer, K. Dautenhahn and F. Amirabdollahian. Collaborating with kaspar: Using an autonomous humanoid robot to foster cooperative dyadic play among children with autism,. *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, 2010.
- [61] A. R. Swanson J. A. Crittendon Z. E. Warren E. T. Bekele, U. Lahiri and N. Sarkar. A step towards developing adaptive robot-mediated intervention architecture (aria) for children with autism. *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 2, 2013.
- [62] A.D. Sánchez J. E. Delgado L. E. Cheu W. A. Arévalo J. M. I. Zannatha, A. J. M. Tamayo. Development of a system based on 3d vision, interactive virtual environments, ergonomic signals and a humanoid for stroke rehabilitation,. *Comput. Methods Prog. Biomed.*, vol. 112, 2013.
- [63] M. Villano J. J. Diehl, L. M. Schmitt and C. R. Crowell. The clinical use of robots for individuals with autism spectrum disorders: A critical review,. *Res. Autism Spectrum Disorders*, vol. 6, 2012.
- [64] T. Boekhorst B. Robins, K. Dautenhahn and A. Billard. Robotic assistants in therapy and education of children with autism: Can a small humanoid robot help encourage social interaction skills? *Univers. Access Inf. Soc.*, vol. 4,, 2005.
- [65] J. Roscoe J. Hedgecock D. Stewart M. Galvez Trigo P. Standen, D. Brown and E. Elgajiji. Engaging students with profound and multiple disabilities using humanoid robots. *Universal Access in Human-Computer Interaction: Universal Access to Infor-*

- mation and Knowledge (Lecture Notes in Computer Science)*, vol. 8514, C. Stephanidis and M. Antona, Eds. New York, NY, USA: Springer, 2014.
- [66] S. Kajita, H. Hirukawa, K. Harada, K. Yokoi. Introduction to humanoid robotis. Springer, 2014.
- [67] Luis Sentis. *Synthesis and control of whole-body behaviors in humanode systems*. PhD thesis, Department of Electrical Engineering, Standford University, 2007.
- [68] Y. Ogura, H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, H. Lim and A. Takanishi. Development of a humanoid robot capable of leaning on a walk-assist machine. *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, 2006.
- [69] M. L. Swinson and D. J. Bruemmer. Expanding frontiers of humanoid robotics. *Intelligent Systems and their Applications, IEEE*, 2000.
- [70] Carnegie Mellon University Motion Capture Database. <http://mocap.cs.cmu.edu/>.
- [71] R. Arroyo L. M. Bergasa R. Barea A. Cela, J. J. Yebes and E. Lopez. Complete low-cost implementation of a teleoperated control system for a humanoid robot. *Sensors*, 2013.
- [72] J. Lee R. Destenay H. Han, J. Kwon and B. J. You. Real-time optimization for the high-fidelity of human motion imitation. *Ubiquitous Robots and Ambient Intelligence (URAI), 2014 11th International Conference on, Kuala Lumpur*, 2014.
- [73] Asfour T. Dillmann R. Azad, P. Toward an unified representation for imitation of human motion on humanoids. *IEEE International Conference on Robotics and Automation*, 2007.
- [74] Marcia J. Riley Nancy S. Pollard, Jessica K.Hodgins and Christopher G. Atkeson.

- Adapting human motion for the control of a humanoid robot. *Proceedings of International Conference on Robotics and Automation*, 2002.
- [75] Zhaoqin P. Weimin Z. Lige Z. Kejie L. Qiang, H. Design of humanoid complicated dynamic motion based on human motion capture. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.*, 2005.
- [76] Kanehiro F. Kaneko K. Fujiwara K. Harada K. Yokoi K. Hirukawa H. Kajita, S. Biped walking pattern generation by using preview control of zero-moment point. *IEEE International Conference on Robotics and Automation*, 2003.
- [77] V. Nunez, N. Nadjar-Gauthier, K. Yokoi, P. Blazevic and O. Stasse. Whole body posture controller based on inertial forces. *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, 2006.
- [78] Kanehiro F.-Kaneko K. Fujiwara K. Harada K. Yokoi K. Hirukawa H. Kajita, S. Resolved momentum control: humanoid motion planning based on the linear and angular momentum. *Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.
- [79] V. Padois S. Ivaldi, J. Peters and F. Nori. Tools for simulating humanoid robot dynamics: A survey based on user feedback. *IEEE-RAS International Conference on Humanoid Robots*, 2014.
- [80] Nuñez-Nalda Jose V. Briseño-Cerón Abraham Rodriguez-Vega Dora A., Zaldivar-Colado Ulises. Teleoperación en línea de un humanoide virtual mediante la manipulación de un humanoide real. *Congreso Mexicano de Robótica 2012, Puebla, México*, 2012.
- [81] Página web WEBOTS. <http://www.cyberbotics.com/webots>.
- [82] Aiman Musa M. Omer, Yu Ogura, Hideki Kondo, Hun-ok Lim, and Atsuo Takanishi. Dynamic-based simulation for humanoid robot walking using walking support system.

- Informatics in Control, Automation and Robotics: Selected Papers from the International Conference on Informatics in Control, Automation and Robotics, Springer Berlin Heidelberg, 2009.*
- [83] Spiess K. Laue, T. and T Rofer. Simrobot a general physical robot simulator and its application. *RoboCupLecture Notes in Computer ScienceRoboCup 2005: Robot Soccer World Cup IX*, 2005.
- [84] JuanCristóbal Zagal and Javier Ruiz-del Solar. Uchilsim: A dynamically and visually realistic simulator for the robocup four legged league. *Lecture Notes in Computer Science*.
- [85] Jared Go, Brett Browning, and Manuela Veloso. Accurate and flexible simulation for dynamic, vision-centric robots. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 3, AAMAS '04*, 2004.
- [86] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.
- [87] C.AA Calderon, R.E. Mohan, and Changjiu Zhou. Virtual-re: A humanoid robotic soccer simulator cyberworlds. *International Conference on*, 2008.
- [88] A. Fitzpatrick-P. Metta G. Natale L. Tikhonoff, V. Cangelosi and F. Nori. An open-source simulator for cognitive robotics research the prototype of the icub humanoid robot simulator. *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, 2008.
- [89] Adams Multibody Dynamics Simulation Software.
<http://www.mscsoftware.com/products/cae-tools/adams.aspx>. 2012.

- [90] Kanehiro F. Kajita S. Hirukawa H. Nakaoka S., Hattori S. Constraint based dynamics simulator for humanoid robots with shock absorbing mechanisms. *IEEE/RSJ International Conference on Intelligent Robot and Sysmtes, Proceedings of*, 2007.
- [91] Página WEB Robotics Developer Studio. <http://msdn.microsoft.com/en-us/library/bb648760.aspx>.
- [92] H. Dallali et al. Development of a dynamic simulator for a compliant humanoid robot based on a symbolic multibody approach. *Mechatronics (ICM), 2013 IEEE International Conference on, Vicenza,, 2013*.
- [93] Vojtěch Vonásek, Daniel Fišer, Karel Košnar, and Libor Přeučil. A light-weight robot simulator for modular robotics. *"Modelling and Simulation for Autonomous Systems: First International Workshop, MESAS 2014, Rome, Italy, May 5-6, 2014, Revised Selected Papers, Springer International Publishing, 2014*.
- [94] T. Mori T. Sato T. Harada, S. Taoka. Quantitative evaluation method for pose and motion similarity based on human perception,. *Fourth IEEE/RAS International Conference on Humanoid Robots, vol. 1,, 2004*.
- [95] Kanda T. Ono T. Ishiguro H. Hagita N. Sakamoto, D. Android as a telecommunication medium with a human-like presence. *2nd ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2007*.
- [96] R. Romero F. Zuher. Recognition of human motions for imitation and control of a humanoid robot. *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012*.
- [97] Xiomara Zaldivar-Colado Claudia Marmolejo-Rivas-Roberto Bernal-Guadiana Ulises Zaldivar-Colado, Diego Murillo-Campos. Realistic manipulation of virtual robot in dynamic virtual environment. *Proceedings of the ASME 2010 World Conference on Innovative Virtual RealityWINVR2010, 2010*.

- [98] Pagina WEB ROBOTIS. <http://support.robotis.com/en/>.
- [99] ST Microelectronics. Hoja de datos del microcontrolador stm32f103re.
- [100] Dora. A. Rodriguez-J.Manuel Ibarra Victor M. Rodriguez J. Victor nunez, Abraham Briseno. Explicit analytic solution for inverse kinematics of bioloid humanoid robot. *Brazilian Robotics Symposium and Latin American Robotics Symposium*, 2012.
- [101] Pedro Daniel Dinis Teodoro. Humanoid robot, development of a simulation enviroment of an entertainment humanoid robot. *Master Thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico*, 2007.
- [102] M. Vukobratovic ; J. Stepanenko. On the stability of anthropomorphic systems. *Mathematical Biosciences (Volume: 15)*, 1972.
- [103] S. Kajita, H. Hirukawa, K. Harada, K. Yokoi. Introduction to humanoid robotis. *Springer*, 2014.
- [104] M. Vukobratovic ; B. Borovac. Zero moment point - thirty five years of its life. *International Journal of Humanoid Robotics, Vol. 1, Num. 1*, 2004.
- [105] M. Vukobratovic ; D. Juriric. Contribution to the sintesis of biped gait. *IEEE Transaction on Biomedical Engineering*, 1969.
- [106] Nuñez-Nalda Jose V. Briseño-Cerón Abraham Rodriguez-Vega Dora A., Zaldivar-Colado Ulises. Teleoperación en línea de un humanoide virtual mediante la manipulación de un humanoide real. <https://youtu.be/aV5G6An3Xo4>, 2012.
- [107] Nuñez-Nalda Jose V. Rodriguez-Vega Dora A., Zaldivar-Colado Ulises. Modelo virtual con comportamiento dinámico del robot humanoide bioloid premium TM. *Congreso Mexicano de Robótica 2014, Mazatlán, México*, 2014.

- [108] U. Zaldívar Colado. Planification d'assemblage en environnement virtuel. *Ph.D. thesis, Laboratoire d'Ingenierie des Systemes de Versailles, Université de Versailles Saint Quentin-en-Yvelines, France, 2009.*
- [109] C. W. Borst and A. P. Indugula. Realistic virtual grasping. *IEEE Virtual Reality Conference (VR'05), 2005.*
- [110] C. W. Borst and A. P. Indugula. A spring model for whole-hand virtual grasping. *Presence: Teleoperators and Virtual Environments, 2006.*
- [111] Claudia Marmolejo-Rivas Carlos A. León Sánchez Saúl J. Guerrero-Cárdenas Diego Murillo-Campos Carolina G. Martínez-Tirado Ulises Zaldivar-Colado, Xiomara Zaldivar-Colado. Manipulación dinámica de objetos utilizando un sistema de captura de movimientos de los dedos de la mano aplicando el modelo de resorte-amortiguador. *Memorias del XIII Congreso Mexicano de Robótica, ●.*