



UNIVERSIDAD VERACRUZANA

---

---

FACULTAD DE INGENIERÍA

REGIÓN VERACRUZ

POSGRADO

PROYECTO DE INTERVENCIÓN PROFESIONAL

METODOLOGÍA INTEGRAL PARA EL DISEÑO DE UN SISTEMA  
ROBOTICO DE LOCOMOCION BIPEDA

QUE PARA OBTENER EL GRADO DE:

**MAESTRIA EN INGENIERÍA APLICADA**

PRESENTA:

**ING. JACOBO TORRES FIGUEROA**

DIRECTOR DEL PROYECTO:

**DR. ALEJANDRO VÁZQUEZ SANTACRUZ**

BOCA DEL RÍO, VERACRUZ.

MARZO 2018.

# Contenido

<b>Agradecimientos</b>	<b>5</b>
<b>Resumen</b>	<b>6</b>
<b>Abstract</b>	<b>7</b>
<b>1. Introducción</b>	<b>13</b>
1.1. Objetivo . . . . .	14
1.2. Justificación . . . . .	15
1.3. Hipótesis . . . . .	16
1.4. Factibilidad del proyecto . . . . .	16
<b>2. Marco teórico</b>	<b>17</b>
2.1. Locomoción bípeda en la robótica . . . . .	17
2.1.1. Tipos de robots bípedos . . . . .	19
2.2. Fisiología articular del miembro inferior . . . . .	20
2.2.1. La cadera . . . . .	21
2.2.1.1. Movimientos de flexión . . . . .	22
2.2.1.2. Movimientos de extensión . . . . .	23
2.2.1.3. Movimientos de abducción . . . . .	24
2.2.1.4. Movimientos de aducción . . . . .	24
2.2.1.5. Movimientos de rotación longitudinal . . . . .	25
2.2.1.6. Movimiento de circundación . . . . .	25
2.3. Biomecánica de la marcha humana . . . . .	26
2.4. Ciclo de marcha . . . . .	26
2.5. Conceptos de robótica . . . . .	28
2.5.1. Notación de Denavit-Hartenberg . . . . .	28
2.5.2. Modelo dinámico . . . . .	31

2.5.2.1. Algoritmo iterativo de Newton-Euler para modelo dinámico inverso . . . . .	35
2.6. Modelo V . . . . .	36
2.6.1. Etapas del ciclo de vida del producto definidas por el modelo V . . . . .	36
2.6.1.1. Requisitos . . . . .	36
2.6.1.2. Diseño de sistema . . . . .	37
2.6.1.3. Diseño de dominio específico . . . . .	37
2.6.1.4. Integración del sistema . . . . .	37
2.6.1.5. Verificaron y validación . . . . .	38
2.6.1.6. Modelado y análisis del modelo . . . . .	38
2.6.1.7. Producto . . . . .	38
2.7. Algoritmos genéticos . . . . .	38
2.7.1. Inicialización . . . . .	39
2.7.2. Selección de padres . . . . .	40
2.7.3. Cruza . . . . .	40
2.7.3.1. Cruza de un sólo punto . . . . .	41
2.7.3.2. Cruza de dos puntos o multipunto . . . . .	41
2.7.3.3. Cruza uniforme . . . . .	42
2.7.3.4. Mutación . . . . .	42
2.7.3.5. Selección del más apto . . . . .	43
2.8. Ingeniería de sistemas basado en modelos . . . . .	43
<b>3. Metodología de diseño mecatrónico</b>	<b>47</b>
3.1. Introducción . . . . .	47
<b>4. Diseño del robot bípedo</b>	<b>51</b>
4.1. Fase I - Análisis de los requerimientos . . . . .	51
4.1.1. Requerimientos del usuario . . . . .	53
4.1.2. Análisis operacional . . . . .	55
4.1.2.1. Casos de uso . . . . .	56
4.1.2.2. Contexto . . . . .	57
4.1.2.3. Escenario global . . . . .	58
4.1.2.4. Modos del sistema . . . . .	58
4.2. FASE II - Diseño del sistema . . . . .	59
4.2.1. Requerimientos del sistema . . . . .	59
4.2.2. Interfaces externas . . . . .	61
4.2.3. Escenarios del sistema . . . . .	62

4.2.4. Operaciones . . . . .	64
4.3. Fase III - Diseño de la arquitectura . . . . .	66
4.3.1. Arquitectura lógica . . . . .	66
4.3.2. Arquitectura física . . . . .	72
<b>5. Análisis de ingeniería</b>	<b>74</b>
5.1. Modelos matemáticos . . . . .	74
5.1.1. Modelo cinemático directo . . . . .	74
5.1.2. Modelo dinámico inverso . . . . .	77
5.1.3. Optimización por medio de algoritmos genéticos del sistema . . . . .	83
5.1.3.1. Función objetivo . . . . .	83
5.1.3.2. Vector de diseño . . . . .	83
5.1.3.3. Restricciones . . . . .	86
5.1.3.4. Resultados . . . . .	86
5.2. Modularización, diseño de detalle y validación a nivel de parte . . . . .	90
5.2.1. Diseño del sub-sistema mecánico . . . . .	90
5.2.2. Diseño de elementos de maquinas . . . . .	90
5.2.2.1. Diseño de ejes . . . . .	90
5.2.2.2. Selección rodamientos . . . . .	95
5.2.2.3. Selección de sistemas de transmisión por bandas sincronas	97
5.2.3. Optimización topológica local de los eslabones . . . . .	103
5.2.3.1. Optimización del eslabón L2 . . . . .	104
5.2.3.2. Optimizacion del eslabón L2 . . . . .	110
<b>6. Resultados</b>	<b>115</b>
<b>7. Conclusiones</b>	<b>116</b>
<b>Referencias</b>	<b>118</b>
Anexo 1. Programas para simulacion de modelos matmáticos . . . . .	120
1.2 Algoritmo de optimización por algoritmos genéticos escrito en Matlab . .	133
Apendice 1. Registro de proyecto . . . . .	134
Apéndice 2. Apoyos recibidos durante la maestría . . . . .	134
Apéndice 3. Publicaciones durante la maestría . . . . .	136
3.1 Diseño de contrlador digital para servomecanismo de seguimiento solar para aplicaciones en celdas fotovoltaicas . . . . .	136
3.2 Evidencia de participación en el congreso . . . . .	142

3.3 estudio hidrodinamico de brazo manipulador de dos grados de libertad para robot submarino . . . . .	143
3.4 Evidencia de participación en el congreso . . . . .	148
Apéndice 4. CVU Actualizado . . . . .	149

# Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología por todo el apoyo económico recibido durante este tiempo. Agradezco a mi asesor, el Dr José Alejandro Vásquez Santa Cruz por su ayuda en todo este tiempo, al Dr. Rogelio Portillo por la transmisión del conocimiento.

A mi familia, a Fátima, a mis amigos Gaby, Alan y Chris, pero por sobre todo a HaShem.

# Resumen

La presente tesis aborda el diseño de un sistema robótico de locomoción bípeda, utilizando una metodología basada en el modelo V, aplicando técnicas de la ingeniería de sistemas basado en modelos, utilizando para ello el lenguaje SysML en el entorno de trabajo que proporciona el software Cameo System Modeler. Se utiliza una estrategia de optimización basado en algoritmos genéticos para la obtención de los requerimientos a nivel de parte. Así también se estudia la optimización del sistema mecánico del robot mediante la optimización topológica. También se estudia la selección de diferentes elementos de maquina como diseño de ejes y selección de sistemas de transmisión de potencia y rodamientos.

# Abstract

This thesis deals with the design of a robotic bipedal locomotion system, using a methodology based on the V model, applying techniques of model based system engineering (MBSE), using the SysML language in the Cameo System Modeler software. Use an optimization strategy based on genetic algorithms to obtain the requirements at the part level. The optimization of the robot's mechanical system through topological optimization is also studied. The selection of different machine elements such as shaft design and selection of power transmission systems and bearings was also selected.

# Índice de figuras

2.1.	Robot bípedo planar pasivo, creado por McGreer en la década de los 70's.	17
2.2.	Robot humanoide ASIMO (Ruiz, 2014).	19
2.3.	Diferentes planos de corte en el cuerpo humano (Kapandji, 2012).	20
2.4.	Ejes sobre los que se describen el movimiento articular de la cadera (Kapandji, 2012).	21
2.5.	Flexión activa de la cadera (Kapandji, 2012).	22
2.6.	Flexión pasiva de la cadera (Kapandji, 2012).	23
2.7.	Extensión pasiva de la cadera (Kapandji, 2012).	23
2.8.	Amplitudes máximas en la abducción de la cadera (Kapandji, 2012).	24
2.9.	Diferentes tipos de apreciación de la aducción de cadera (Kapandji, 2012).	25
2.10.	Movimientos de rotación de la cadera (Kapandji, 2012).	25
2.11.	Trayectoria de circundación de la articulación coxofemoral (Kapandji, 2012).	26
2.12.	Parámetros de vinculo y articulares que describen la relación entre los eslabones de un brazo manipulador.	28
2.13.	Tramas adjuntas al vinculo $i$ y al vinculo $i - 1$ , según la convención basada en los parámetros propuestos por Denavit-Hartenberg.	29
2.14.	Eslabón sometido a aceleración lineal, generado por una fuerza $F$ aplicado en su centro de masa.	31
2.15.	Momentos ejercido en un eslabón cuando este se mueve con una velocidad y aceleración anulares.	32
2.16.	Velocidades anulares $\omega$ y lineales $v$ , entre dos eslabones adyacentes.	33
2.17.	Estructura del modelo V sugerido por el lineamiento VDI 2206 (Sell, 2005).	37
2.18.	Gráfica de la función de Ackley, la cual es una función compleja en el ámbito de la optimización por la gran cantidad de mínimos locales (Cuevas, 2016).	39

2.19. Ejemplo de mutación de un solo punto en dos individuos de 16 bits (Cuevas, 2016). . . . .	41
2.20. Ejemplo de crusa multipunto en individuos de codificación binaria (Cuevas, 2016). . . . .	42
2.21. Ejemplo de crusa uniforme en dos especímenes de 16 bits de longitud (Cuevas, 2016). . . . .	42
2.22. Relación ente UML y SysML (Object Management Group, 2018). . . .	44
2.23. Tipos de diagramas de SysML (Object Management Group, 2018). . . .	45
3.1. Descripción de las fases del modelo-V en la metodología propuesta. . .	47
4.1. Estructura del modelo creado en Cameo System Modeler. . . . .	52
4.2. Diagrama de paquetes como forma de visualización de la estructura del modelo. . . . .	52
4.3. Requerimientos del usuario escritos en una hoja de calculo. . . . .	53
4.4. Requerimientos importados en el modelo. . . . .	54
4.5. Representación de los requerimientos del usuario en un diagrama de requerimientos. . . . .	55
4.6. Diagrama de casos de uso que muestra la misión principal del sistema. .	56
4.7. Diagrama de casos de uso que describe el uso nominal del sistema, en interacción con los actores. . . . .	56
4.8. Contexto operacional del sistema RobotBipedo. . . . .	57
4.10. Escenario global para el sistema RobotBipedo y su contexto. . . . .	58
4.9. Contexto operacional desde el punto de vista del sistema RobotBipedo.	58
4.11. Maquina de estados, que modelan los estados básicos del sistema. . .	59
4.12. Diagrama de derivación de los requerimientos del sistema. . . . .	61
4.13. Diagramas de interfaces externas del sistema RobotBipedo. . . . .	61
4.14. Diagrama de secuencia que describe las interacciones de los diferentes actores con el sistema RobotBipedo, en el escenario EjecutarMarcha. .	63
4.15. Operaciones que el robot bípedo debe realizar. . . . .	64
4.16. Matriz de derivación de requerimientos, donde se especifica la relacion de derivacion entre los requerimientos. . . . .	65
4.17. Matriz de satisfacción de requerimientos, que justifica las operaciones del RobotBipedo respecto de los requerimientos del sistema. . . . .	66
4.18. Actividad principal del sistema RobotBipedo. . . . .	67
4.19. Matriz de satisfacción de requerimientos para verificar las actividades propuestas. . . . .	68

<b>ÍNDICE DE FIGURAS</b>	<b>10</b>
4.20. Estructura lógica interna del RobotBipedo. . . . .	68
4.21. Composición interna de la «Unidad de Control» . . . . .	69
4.22. Composición interna del «SubSistemaSensores». . . . .	70
4.23. Composición interna del «SubSistemaPotencia». . . . .	70
4.24. Composición interna del «SubSistemaMecanico». . . . .	71
4.25. Composición interna del «SubSistemaActuadores». . . . .	71
4.26. Arquitectura física del robot bipedo. . . . .	73
5.1. Vista alámbrica de la estructura del robot cuando todas las variables articulares son cero. . . . .	77
5.2. Fase de apoyo simple utilizada en la simulación del modelo dinámico inverso. . . . .	80
5.3. Torques articulares requeridos para la trayectoria deseada en cada eslabón del robot. . . . .	81
5.4. Componentes de las fuerzas netas aplicadas en el centro de masas de cada eslabón del robot. . . . .	82
5.5. Componentes de las velocidades anulares que experimenta cada eslabón del robot. . . . .	82
5.6. Modelo antropomórfico del cuerpo humano (Ruiz, 2014). . . . .	84
5.7. Convergencia de la solución del algoritmo genético. . . . .	88
5.8. Torques articulares obtenidos con los valores óptimos del vector de diseño. . . . .	89
5.9. Fuerzas iniciales aplicadas en el centro de masas de los eslabones. . . . .	89
5.10. Diagrama de cuerpo libre del eje. . . . .	91
5.11. Diagrama de fuerza cortante y momento flector para eje tipo 1. . . . .	92
5.12. Dimensiones del diseño final para el eje para cargas radiales. . . . .	95
5.13. Rodamiento seleccionado por medio de la aplicación del fabricante SKF (SKF, 2017). . . . .	96
5.14. Especificaciones del rodamiento W61700, del fabricante SKF (SKF, 2017)	97
5.15. Gráfica de selección del paso de la banda del tipo PowerGrip G3 del fabricante Gates Corporación (Gates, 2017). . . . .	99
5.16. Ventana principal del software Design Flex Pro del fabricante Gates Corporation, corriendo en el sistema operativo Windows (Gates, 2017). . . . .	100
5.17. a) Tipo de banda seleccionada en el software. b) . . . . .	100
5.18. Parámetros de configuración del motor de CD introducidos en el software Design Flex Pro. . . . .	101
5.19. Parámetros introducidos en el software respecto de la polea conducida y distancia entre ejes. . . . .	101

5.20. Parámetros dimensionales para ambas polea. . . . .	102
5.21. Soluciones factibles para el sistema de transmisión por bandas sincronas tipo PowerGrip GT3. . . . .	102
5.22. Reporte del diseño del sistema de transmisión seleccionado generado por el software Design Flex Pro. . . . .	103
5.23. Fotografía de las poleas seleccionadas para polea motriz (2MR-21S-06) y conducida (2MR-25S-06). . . . .	103
5.24. a) Cargas, sujeteciones y zonas de preservación introducidas al modelo para la optimizacion. b) Malla generada por el software. . . . .	105
5.25. Etapas de diseño y optimizacion del eslabón para el segmento $L_2$ . a) Diseño inicial sin optimizar, b) Forma generada por la optimizacion topologica. . . . .	106
5.26. Re diseño del eslabón basado en la forma optima. . . . .	107
5.27. Esfuerzos de Vom Mises en el eslabón 2, sometido a las fuerzas calculadas por el modelo dinámico inverso. . . . .	108
5.28. Curvas esfuerzo-deformación determinadas experimentalmente para especímenes impresos en 3D de material ABS (Lee, 2013). . . . .	109
5.29. a) Desplazamiento máximo y mínimo calculados para el eslabón 2. b) Factor de seguridad calculado. . . . .	109
5.30. a) Regiones preservadas en la forma inicial de la pieza para el eslabón 3. b) Mallado generado por el software NASTRAN. . . . .	111
5.31. a) Pieza inicial diseñada. b) Forma optimizada. . . . .	112
5.32. Forma final optimizada para el eslabón 3, generada por la optimización topológica. . . . .	113
5.33. Esfuerzos de Von Mises en el eslabón 3. . . . .	113
5.34. a) Desplazamientos máximos y mínimos en el eslabón 3. b) Factor de seguridad calculado por el software. . . . .	114
7.1. . . . .	134
7.2. . . . .	135

# Índice de tablas

3.1. Relación de los diagramas de SysML utilizados en la FASE I-Análisis de los requerimientos. . . . .	48
3.3. Diagramas de SysML utilizados en la FASE III - Diseño de la arquitectura.	49
3.2. Diagramas de SysML utilizados en la FASE II - Diseño del sistema. . . . .	49
4.1. Requerimientos del usuario importado. . . . .	54
4.2. Requerimientos del sistema derivados de los requerimientos del usuario.	60
5.1. Parámetro de masas de eslabones utilizados en la simulación del modelo dinámico inverso . . . . .	77
5.2. Dimensiones de los eslabones para la simulación. . . . .	78
5.3. Parámetros utilizados para la generación de trayectorias articulares en la simulación del modelo dinámico inverso. . . . .	80
5.4. Parámetros de diseño para el eje en el punto A. . . . .	92
5.5. Parámetros de diseño para el eje en el punto A. . . . .	93
5.6. Parámetros de diseño para el eje en el punto A. . . . .	93
5.7. Parámetros de diseño para el eje en el punto A. . . . .	94
6.1. Tabla de resultados de las masas obtenidas mediante el diseño basado en optimización topológica y las masa objetivo por medio de algoritmos genéticos. . . . .	115

# Capítulo 1

## Introducción

El estudio de los sistemas de locomoción en robótica es amplio, enfocándose en diversos sistemas para desplazamiento, los más simples están basados en ruedas en cuanto al control y modelado matemático, sin embargo, existen otros sistemas de locomoción que presentan ventajas respecto de estos a pesar de su incremento en complejidad, como son los sistemas de locomoción basados en patas, donde pueden existir diversas configuraciones más o menos complejas, variando en el número extremidades (Barrientos et al., 1997).

Si bien los sistemas de locomoción basados en patas son particularmente más complejos en su diseño, modelado y construcción, presentan varias ventajas, por ejemplo la capacidad de desplazamiento en ambientes con desniveles considerables, que difícilmente podrían ser transitados por robots con ruedas u orugas. Una de las principales ventajas consiste en la eficiencia energética en la realización de los movimientos necesarios en el balanceo de las extremidades. Los robots con patas tienen la particularidad de que son más adaptables a entornos reales de tránsito humano, es decir, que pueden desplazarse con más naturalidad en un entorno diseñado para desplazamientos basados en sistemas de locomoción de patas, puesto que los humanos cuentan con dicho sistema. Por tanto desde el punto de vista práctico, resulta conveniente el estudio de dichos sistemas robóticos y de forma particular en los robots bípedos, es decir, robots cuyo sistema de locomoción está basado en dos patas (Armada, 2004).

Sin embargo, los robots bípedos presentan, debido a su inherente inestabilidad un reto considerable en su estudio, modelado y aplicación. La complejidad de la marcha y la estabilidad durante la misma, se ha podido abordar desde dos perspectivas válidas, denominadas: marcha estática y dinámica.

La marcha estática conlleva el uso de pies grandes y movimientos lentos durante la marcha, de modo que todo aporte dinámico en el movimiento de las masas sea disminuido, convergiendo en robots lentos, puesto que se busca principalmente el desplazamiento del centro de masas del sistema dentro de la proyección de los pies del robot. Este tipo de marcha fue particularmente popular entre los primeros desarrollos de robots bípedos, puesto que se pueden desarrollar incluso por medios basados en su totalidad en mecanismos, sin necesidad de sistemas de retroalimentación para el ajuste (Borina, 2015).

La marcha dinámica consiste en mantener al robot en una constante caída controlada, imitando de la mejor forma posible el andar humano, abarcando el reto de la alta inestabilidad ocasionada y la dinámica del sistema. La bases teóricas para el estudio de estos robots, se remontan principalmente a los matemáticos rusos como P. L. Chebyshev (1824-1894), I.I. Artobolevsky (1905-1977) y A.E. Kобрinsky (1915-1992), entre otros . Este tipo de sistemas requieren de sofisticados sistemas de control y es por ello que su investigación y desarrollo se remonta a los años 60's, con el advenimiento de las computadoras electrónicas de propósito general (Borina, 2015).

Otro de los retos en el diseño de robots bípedos en la actualidad, es el diseño eficiente y la selección de todos los elementos que lo conforman de forma óptima, siendo dicha tarea de una complejidad considerable, puesto que se requieren realizar elecciones con un objetivo multi-criterio y de igual forma conlleva la interacción fluida y estructurada de diversas ramas de la ingeniería para evitar errores a la hora de la unificación de los diversos subsistemas (Armada, 2004). De modo que para llevar a cabo dicho problema en la actualidad se están buscando metodologías que permitan su desarrollo de forma estructurada, fluida y óptima (Sell, 2005).

En la presente tesis se desarrolla una metodología para el diseño del robot bípedo, además de abordar el problema de optimización multi-objetivo, enfocado particularmente en las piezas mecánicas, en orden de reducir las aportaciones dinámicas de las mismas.

## 1.1. Objetivo

Objetivo general:

Diseñar e implementar mediante una metodología de diseño mecatrónico propuesta un sistema de locomoción bípeda.

Objetivos particulares:

1. Desarrollar una metodología simplificada basada en el modelo-V en el marco de la ingeniería de sistemas basado en modelos (MBSE).
2. Obtener de los modelos matemáticos que describen la cinemática y dinámica del robot.
3. Construir un sistema de locomoción bípeda optimizado de forma global.
4. Obtener un robot bípedo antropomórficamente consistente, de bajo costo y fácil fabricación.
6. validación de las piezas mecánicas diseñadas mediante el método de elementos finitos.

## 1.2. Justificación

El diseño y construcción de robot bípedo con fines experimentales que se aborda en esta tesis, proporciona las bases de conocimiento, datos y experiencia en el manejo y control de sistemas que repliquen la marcha humana, que pueden ser aplicables en trabajos futuros en el ámbito de la robótica asistencial, particularmente en el diseño y desarrollo de exoesqueletos para rehabilitación activa o pasiva.

Por otro lado, a pesar de que existen en el mercado, sistemas robóticos similares, principalmente robots humanoides con características avanzadas en navegación y percepción del ambiente (Borina, 2015), presentan la desventaja de que son comúnmente de arquitectura cerrada , por lo que de cierta forma están limitados en sus fines experimentales, particularmente para el caso de pruebas de algoritmos de control en la marcha, que pueden variar dependiendo de características específicas del ambiente, como por ejemplo la marcha en ambientes similares a un entorno urbano, donde usualmente se presentan pendientes, escaleras, desniveles, etc. De esta forma el sistema robótico bípedo construido, funge como base experimental de arquitectura abierta, para tales fines.

La utilización de una metodología específicamente diseñada para el desarrollo de sistemas mecatrónicos en el ámbito de la ingeniería de sistemas basado en modelos, reduce los tiempos de diseño y el presupuesto del proyecto. Desarrollar sistemas mecatrónicos bajo la dirección de una metodología de diseño permite trabajar de forma ordenada y sistematizada, lo que reduce errores en el ciclo de diseño. Las metodologías de diseño basadas en el paradigma de la MBSE, se ven actualmente como el futuro de

la ingeniería de sistemas, lo que contribuye al desarrollo tecnológico regional.

### **1.3. Hipótesis**

Es posible gestionar el diseño de sistemas mecatrónicos mediante el enfoque de la ingeniería de sistemas basados en modelos en conjunto con la metodología del modelo-V, para mejorar la integración y solución óptima de los problemas de dominio cruzado involucrados en todo el ciclo de vida del sistema.

### **1.4. Factibilidad del proyecto**

En este proyecto se desarrollan las diferentes fases de la metodología principalmente mediante el uso de softwares de modelado matemático y CAD tridimensional, además de que son empleados en la simulación del mismo.

Por otra parte, debido a que la finalidad del proyecto de tesis es poner las bases de conocimiento y experiencia en el manejo de robots bípedos que repliquen la marcha humana en diversos entornos urbanos, con el fin de futuros desarrollos en el ámbito de la robótica asistencial, puede afirmarse que dichos conocimientos son aplicables para el desarrollo de productos comerciales.

# Capítulo 2

## Marco teórico

### 2.1. Locomoción bípeda en la robótica

El estudio reciente de la locomoción bípeda se ha centrado en dos enfoques: locomoción bípeda activa y pasiva. En la locomoción bípeda pasiva, la única fuente generadora de movimiento es la gravedad, por lo que se busca la realización de mecanismo que logren tal cometido sin que existan fuentes de movimiento, como actuadores, buscando particularmente la eficiencia energética durante la marcha. La locomoción activa busca estudiar sistemas donde la fuente generadora de movimiento sea además de la gravedad, dispositivos de actuación como motores eléctricos y dispositivos hidráulicos.

En el ámbito de la locomoción pasiva, destacan los estudios realizados por McGeer en la década de los 80's, quien construyó un mecanismo planar de cuatro barras. El mecanismo de McGeer incluía en las rodillas un sistema de bloqueo para evitar el colapso



Figura 2.1: Robot bípedo planar pasivo, creado por McGeer en la década de los 70's.

de la pierna y pies circulares para dar un contacto rodante sobre el suelo (Borina, 2015) (ver figura 2.1). En la década de los 90's Goswami, Espiau, y Keramane, modificaron el mecanismo de McGeer adicionándole un actuador entre las piernas utilizando un controlador para regular la energía total del bípedo, el sistema consistía en un mecanismo planar de dos eslabones, el cual podía desarrollar marchas estables y era capaz, debido a su sistema de control, de expandir el conjunto de condiciones iniciales que hacían a que convergiera en una marcha estable. Smith y Berkemeier, desarrollaron un sistema de locomoción basado en una llanta con patas, la cual representaba una primera aproximación hacia la marcha tridimensional, el sistema al ser colocado en un plano inclinado podía converger a la marcha y de forma estable indefinidamente. A finales de los 90's Collins crea una versión tridimensional del mecanismo de McGreer, dotándolo de mas articulaciones, pues curvados y brazos, podía andar a una pendiente de 3.1 grados (Borina, 2015).

El desarrollo de robots bípedos activos, surge en la década de los 70's principalmente en Japón, donde el primer intento de locomoción bípeda se debió a las investigaciones realizadas por Kato y Tsuiki en la Universidad de Waseda en 1972, con la creación del robot WL-5 con capacidad de marcha tridimensional y de 11 GDL. Más tarde en la década de los 80's el mismo grupo de investigación desarrolló WL-10RD, un robot bípedo tridimensional de 12 GDL capaz de desplazarse a 0.1 m/s. En la década de los 90's Kajita y Tani desarrollaron el robot bípedo planar de 6 GDL nombrado Meltran II, el cual era capaz de caminar sobre pequeños obstáculos a una velocidad de 0.2 m/s. A finales de los 90's Pratt diseño el robot bípedo nombrado Spring Flamingo, en el LegLab del MIT, que consistía en un robot planar de 7 eslabones, era capaz de caminar sobre obstáculos a una velocidad de 1.2 m/s (Borina, 2015).

En el año 2000 la universidad Waseda en Japón continúa el desarrollo de los primeros robots bípedos, con el robot tridimensional WABIAN, de 52 GDL, 1.84 m de altura y es capaz de desplazarse a 0.21 m/s. Por otra parte uno de los robots más emblemáticos respecto de locomoción bípeda comenzó su desarrollo en la década de los 80's, y no fue sino hasta el año 2000 cuando la compañía Honda presentó al robot ASIMO (standing for Advanced Step in Innovation MObility), el cual es un robot bípedo autónomo tridimensional de 57 GDL, con la capacidad de correr caminar a una velocidad máxima de 9 km/h, además de subir y bajar escaleras, correr y saltar sobre una o ambas piernas, su desarrollo continúa en la actualidad (ver figura 2.2 ) (Borina, 2015).

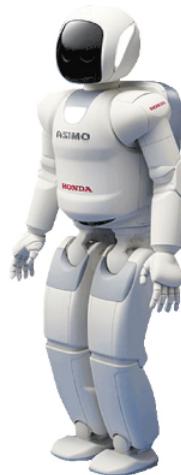


Figura 2.2: Robot humanoide ASIMO (Ruiz, 2014).

En estados unidos su agencia de defensa DARPA (Defense Advanced Research Projects Agency), actualmente desarrolla el robot bípedo tridimensional Atlas, de 28 GDL basado en articulaciones hidráulicas, es capaz de caminar sobre terreno altamente demandante, es decir donde los obstáculos pertenecen no a entornos de laboratorio sino a ambientes reales. En la actualidad se han venido desarrollando multitud de robots humanoides con la capacidad de locomoción bípeda, principalmente para fines de investigación y desarrollo, entre los cuales destaca el robot humanoide desarrollado por la empresa francesa Aldebaran Robotics nombrado NAO, el cual es autónomo y programable, cuenta con hasta 25 GDL. También cuenta con sistemas avanzados de sensorizado que le permiten interactuar de forma fluida con el entorno, así como la capacidad de síntesis de la voz.

### 2.1.1. Tipos de robots bípedos

Un robot bípedo puede describirse como una cadena cinemática abierta, con dos subcadenas denominadas piernas. Por otra parte según el grado de movilidad, existen diferentes enfoques en la construcción de robots bípedos, a continuación se presenta una clasificación siguiendo dichos criterios (Craig, 2006).

**Bidimensionales:** En esta clase particular de robots bípedos, los movimientos de sus eslabones se desarrollan únicamente en un solo plano, denominado sagital.

**Tridimensionales:** Se toman en cuenta las rotaciones durante la marcha bípeda, que originan movimientos fuera del plano sagital.

**Minimalistas:** Cuando el robot bípedo presentan menos de 6 grados de libertad

(GDL) en cada pierna.

**Antropomórficos:** Se busca imitar de forma más precisa la cinemática de las piernas humanas, y por tanto este tipo de robots bípedos presentan 6 ó más GDL por cada pierna.

**Pasivos:** No se utilizan controladores para dirigir los movimientos de las articulaciones.

**Activos:** En este tipo de sistemas se toma en cuenta el control articular de posición del robot.

## 2.2. Fisiología articular del miembro inferior

Los movimientos articulares en el miembro inferior que se originan durante la marcha humana son complejos, sin embargo, pueden ser sintetizados en tres diferentes planos que normalmente son denominados: sagital, frontal y transversal (Kapandji, 2012). El plano frontal divide al cuerpo en parte izquierda y derecha, el plano frontal es perpendicular al plano sagital y divide al cuerpo en parte trasera y delantera, el plano transversal realiza un corte en el cuerpo de forma perpendicular al plano sagital y frontal (ver figura 2.3).

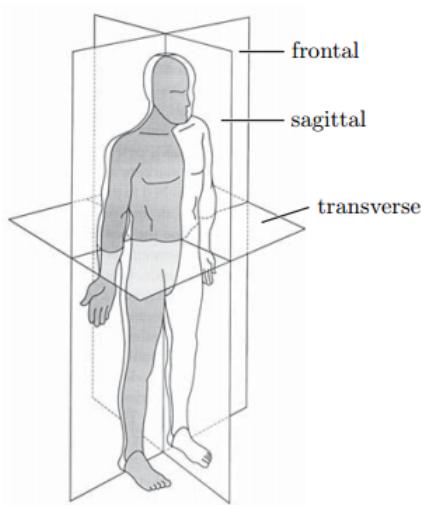


Figura 2.3: Diferentes planos de corte en el cuerpo humano (Kapandji, 2012).

Las articulaciones que componen los movimientos del miembro inferior son:

1. Cadera.
2. Rodilla.

3. Tobillo, y
4. Bóveda plantar.

si bien, fisiológicamente estos movimientos articulares constituyen la movilidad completa del miembro inferior, sin embargo, aquí sólo se describen los movimientos articulares correspondientes a la cadera, y a los de la rodilla y el tobillo de forma simplificada, puesto que en la práctica el robot bípedo diseñado sólo cuenta con seis grados de libertad por cada miembro inferior, despreciando cualquier movimiento articular presente en la bóveda plantar (Kapandji, 2012).

### 2.2.1. La cadera

La articulación de la cadera recibe el nombre de coxofemoral, y posee tres grados de libertad. La función de esta articulación es la de orientar el miembro inferior en todas direcciones. Los tres movimientos propios de la cadera se describen en tres ejes, dos de ellos paralelos a los planos frontal y sagital, XOX' y YOY' respectivamente y un eje perpendicular al plano transversal, nombrado OZ (véase figura 2.4) (Kapandji, 2012).

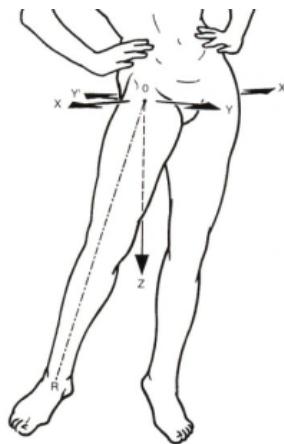


Figura 2.4: Ejes sobre los que se describen el movimiento articular de la cadera (Kapandji, 2012).

En el eje transversal XOX' contenido por el plano frontal, se ejecutan los movimientos de flexo-extensión. De forma similar, en el eje anteroposterior YOY' situado en el plano sagital, se efectúan los movimientos de abducción-aducción, y en el eje vertical OZ (el cual se alinea con el eje longitudinal OR del miembro inferior cuando la cadera está en su posición de referencia) se efectúan los movimientos de rotación externa y rotación interna (Kapandji, 2012).

### 2.2.1.1. Movimientos de flexión

El movimiento de flexión de la cadera se produce cuando la cara anterior del muslo y el tronco se aproximan, de forma que el miembro inferior sobre sale del plano frontal. El movimiento de flexión se clasifica en flexión pasiva y activa, la flexión pasiva ocurre cuando el miembro es ayudado en el movimiento por fuerzas externas; por el contrario, en la flexión pasiva las fuerzas articulares dependen solo del sistema musculo esquelético. En la flexión activa (no es tan amplia como la flexión pasiva), si la rodilla está extendida la flexión no supera los 90, pero si la rodilla está flexionada entonces la flexión de la cadera alcanza los 120 (ver la figura 2.5) (Kapandji, 2012).

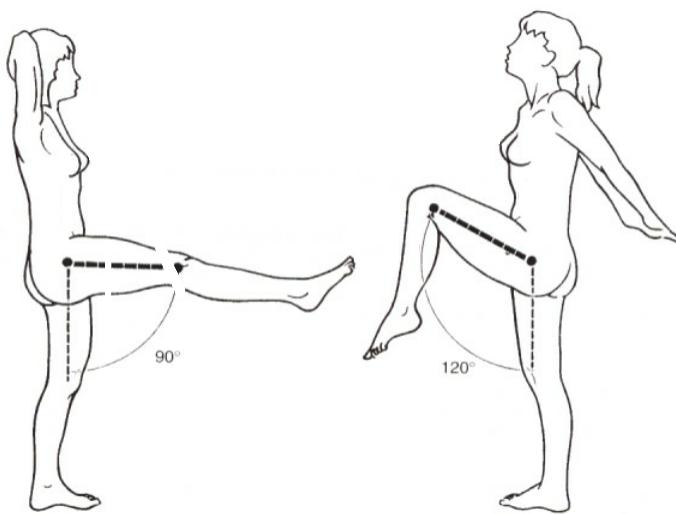


Figura 2.5: Flexión activa de la cadera (Kapandji, 2012).

Por otra parte en la flexión pasiva la amplitud es superior a los 120 siempre, pero también este rango de apertura es modificado por la articulación de la rodilla, ya que cuando esta está flexionada la apertura de flexión de la cadera alcanza los 140, quedando incluso el muslo en contacto con el tronco (Kapandji, 2012) (ver figura 2.6).

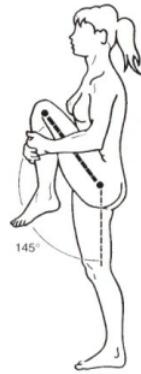


Figura 2.6: Flexión pasiva de la cadera (Kapandji, 2012).

#### 2.2.1.2. Movimientos de extensión

El movimiento de extensión lleva al miembro inferior por detrás del plano frontal, de forma similar a la flexión, la extensión también se clasifica en activa y pasiva. En la extensión activa, cuando la rodilla está extendida la amplitud alcanza los 20, mientras que cuando está flexionada la amplitud de la extensión de la cadera disminuye a 10, como se observa en la figura. Por otra parte en la extensión pasiva se alcanzan los 30 cuando se lleva el miembro inferior hacia atrás (ver figura 2.7)(Kapandji, 2012).

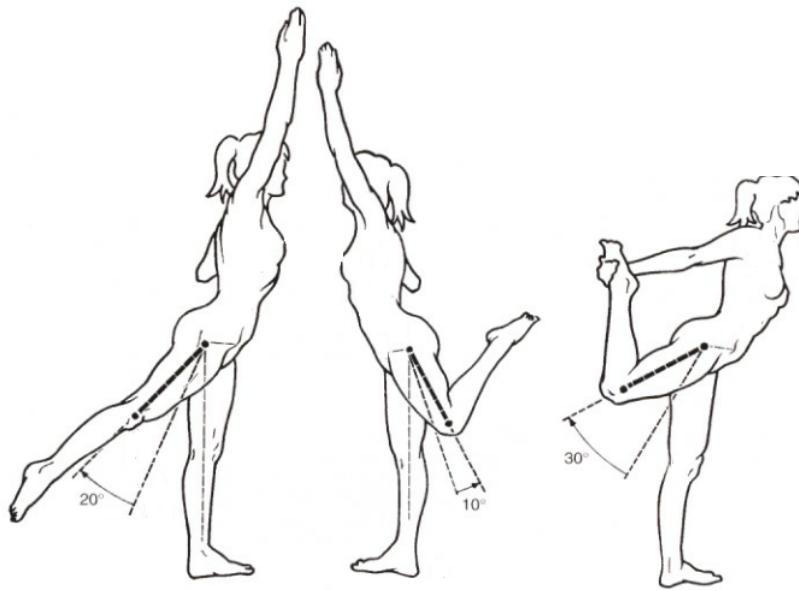


Figura 2.7: Extensión pasiva de la cadera (Kapandji, 2012).

### 2.2.1.3. Movimientos de abducción

El movimiento de abducción conduce al miembro inferior hacia afuera del plano sagital, que es el plano de simetría del cuerpo. La abducción de una sola cadera no es posible para ángulos mayores de 15°, ya que al superarlos este movimiento implica que la otra cadera realice una abducción idéntica, midiendo por lo tanto un ángulo de 30° entre extremidades. La máxima amplitud de abducción ocurre a los 90°, es decir 45° por cada cadera medidos desde el eje de simetría de la pelvis, como se aprecia en la figura 2.8 (Kapandji, 2012).

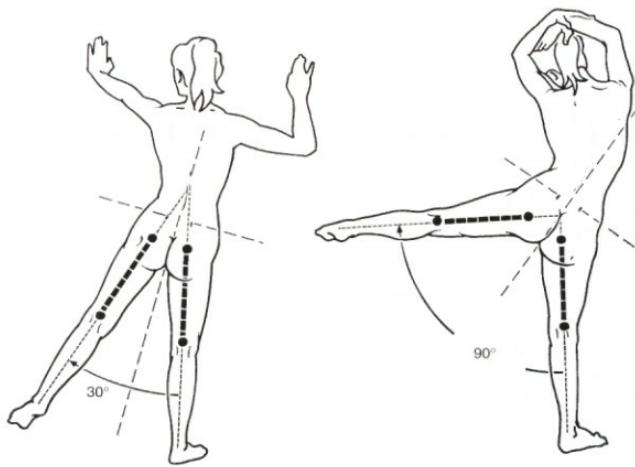


Figura 2.8: Amplitudes máximas en la abducción de la cadera (Kapandji, 2012).

### 2.2.1.4. Movimientos de aducción

La aducción lleva al miembro inferior hacia adentro del plano sagital del cuerpo, aproximándose al eje de simetría del mismo. Sin embargo no existe movimiento de aducción absoluto, sino es medido de forma relativa, por ejemplo, partiendo de una posición de abducción el miembro inferior se aproxima hacia dentro, como se observa en la figura. También es posible medir un movimiento de aducción combinado con extensión de cadera (ver figura), del mismo modo es posible apreciar una aducción combinada con flexión de cadera (ver figura 2.9) (Kapandji, 2012).

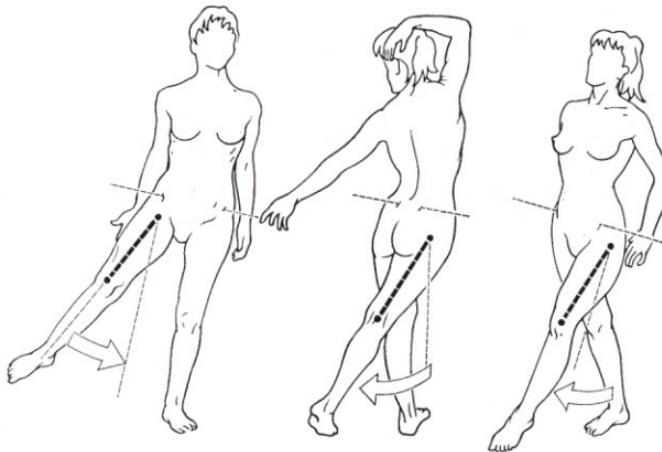


Figura 2.9: Diferentes tipos de apreciación de la aducción de cadera (Kapandji, 2012).

La amplitud máxima de la aducción en todos estos movimientos combinados es de 30.

#### 2.2.1.5. Movimientos de rotación longitudinal

La rotación de la cadera se realiza al rededor del eje mecánico del miembro inferior OR, el cual es coincidente cuando la pierna está en la posición de referencia. El movimiento de rotación externa ocurre cuando el pie es orientado hacia afuera del cuerpo, mientras que la rotación interna lleva al pie hacia dentro, las amplitudes de estos movimientos son de 30 y 60 respectivamente, como se aprecia en la figura 2.10 (Kapandji, 2012).

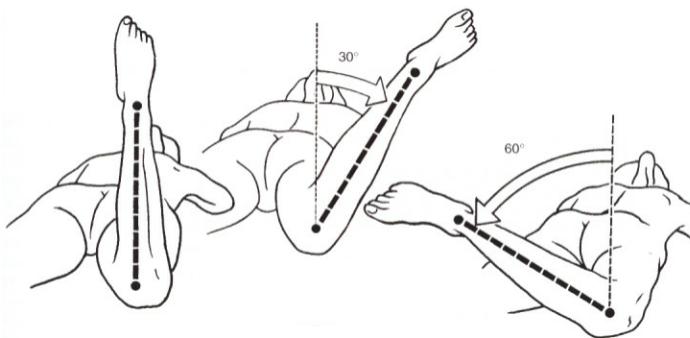


Figura 2.10: Movimientos de rotación de la cadera (Kapandji, 2012).

#### 2.2.1.6. Movimiento de circundación

La combinación de los tres grados de libertad de movimientos en la articulación coxofemoral, conducen al movimiento denominado circundación, es decir que el movi-

miento de la cadera puede modelarse como un cono cuyo vértice es el eje de rotación. El cono no es completamente regular, puesto que los límites angulares para cada movimiento en los grados de libertad no es simétrico. Por tanto la trayectoria descrita por el miembro inferior es más bien una curva sinuosa que pasa por los tres planos de corte del cuerpo, en diferentes sectores. El plano sagital se realizan los movimientos de flexión/extensión, en el plano frontal los movimientos de abducción/aducción y en el plano transversal los movimientos de rotación interna/externa. Un ejemplo del cono sinuoso descrito por la articulación coxofemoral se aprecia en la figura 2.11, donde además se aprecia que la curva sinuosa pasa por los sectores III, II, I IV, V y VIII, de los VIII sectores posibles (Kapandji, 2012).

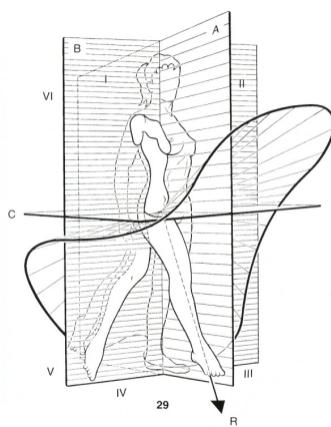


Figura 2.11: Trayectoria de circundación de la articulación coxofemoral (Kapandji, 2012).

## 2.3. Biomecánica de la marcha humana

### 2.4. Ciclo de marcha

La locomoción bípeda es una actividad cíclica que consta de dos fases para cada extremidad, apoyo y balanceo. La marcha es relativamente simétrica por lo que se refiere a los movimientos angulares de las principales articulaciones, patrones de activación muscular y soporte de carga de las extremidades inferiores los cuales dan como resultado, una traslación efectiva del centro de masa del cuerpo en dirección de la locomoción. Por lo tanto, un ciclo de marcha completo está definido como la presencia de una fase de apoyo y una fase de balanceo u oscilación sucesiva que realiza una extremidad (Murray et al. 1964).

La fase de apoyo abarca un 60 % de la zancada y consta de dos períodos de doble

apoyo de la extremidad (inicial y final): cuando el pie contralateral está en contacto con el suelo y un periodo intermedio de apoyo sencillo de la extremidad que sucede cuando la extremidad contralateral se encuentra en la fase de balanceo u oscilación. A su vez, el apoyo se puede dividir en seis partes y periodos.

El contacto inicial, se define como el instante en el que el pie toca el piso. La respuesta ante la carga es un intervalo durante el cual la planta del pie toca al piso y el peso del cuerpo es aceptado en la extremidad de apoyo. El periodo de respuesta ante la carga coincide con el final del doble apoyo inicial de la extremidad en alrededor de 10 a 12 % de la zancada. El apoyo intermedio es el periodo en el cual la tibia gira sobre el pie estacionario en la dirección de la locomoción. El inicio del apoyo intermedio coincide con el apoyo sencillo de la extremidad y dura aproximadamente de 10 a 30 % de la zancada. El apoyo final es el periodo durante el cual el peso del cuerpo es transferido desde las regiones posterior a intermedia del pie al antepié. Esto abarca de 30 a 50 % de la zancada y coincide con el inicio del apoyo doble final de la extremidad. De manera simultánea con el apoyo doble final de la extremidad, sucede el prebalanceo y éste dura casi 50 a 60 % de la zancada. En este instante, el peso se transfiere a la extremidad contralateral como preparación para la fase del balanceo. El final del prebalanceo corresponde cuando la punta del pie se despega del piso, es decir, el momento en el cual el pie rompe el contacto con el suelo, y esto da inicio a la fase de balanceo. La fase de balanceo ocupa 40 % del ciclo de la marcha y se divide en tres periodos; Balanceo inicial, intermedio y final.

Por otra parte, el tiempo de la zancada se refiere al lapso que transcurre al dar una zancada sencilla. La longitud de la zancada se refiere a la distancia cubierta por dicha zancada en la dirección de locomoción. El paso se define como el hecho en que sucede un evento en el pie opuesto. La longitud del paso se refiere a la distancia que cubre un paso en la dirección de locomoción. El ancho del paso se refiere a la distancia, perpendicular a la dirección de locomoción, que cubre dicho paso según lo medido a partir de los puntos de contacto con los talones. Dos pasos consecutivos hacen una zancada. Si bien las variables del paso pueden ser distintas en la derecha y la izquierda en una persona, las variables de la zancada se conservan constantes independientemente de si la zancada está delimitada por los contactos iniciales derecho o izquierdo, debido a que la zancada es la suma de los pasos derecho e izquierdo. La cadencia es una medida de la frecuencia del paso que se define como la cantidad de pasos por unidad de tiempo y, por lo regular, se expresa en pasos por minuto. La velocidad se define como la distancia cubierta, en la dirección de locomoción, por unidad de tiempo y, en general, se expresa en metros por segundo.

## 2.5. Conceptos de robótica

### 2.5.1. Notación de Denavit-Hartenberg

Para la obtención de las ecuaciones cinemáticas es preciso definir antes algunos parámetros que describen tanto a los vínculos como a las articulaciones, en ese sentido un vínculo es visto como una cuerda rígida que define la relación entre dos ejes de articulación adyacentes de un robot manipulador, a estos parámetros de vínculo se le conocen como: longitud de vínculo  $a$  y torsión de vínculo  $\alpha$ , así mismo los otros dos parámetros que tiene que ver con las articulaciones son el desplazamiento de vínculo  $d$  y el angulo de articulación  $\theta$ , todos estos parámetros están representados en la figura 2.12 (Craig, 2006).

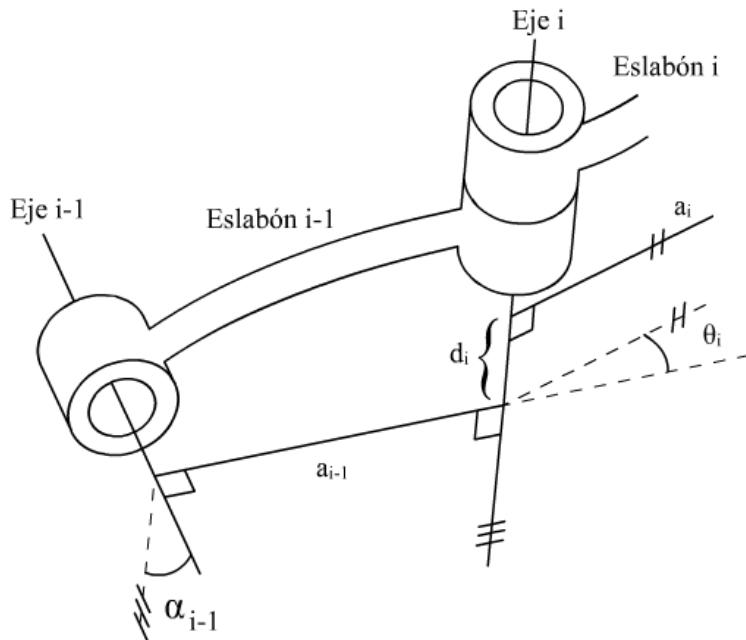


Figura 2.12: Parámetros de vínculo y articulares que describen la relación entre los eslabones de un brazo manipulador.

De la figura 2.12 se observa la interconexión del vínculo  $i - 1$  y el vínculo  $i$ , la distancia  $\alpha_{i-1}$  es la perpendicular común entre los ejes  $i - 1$  e  $i$ , el ángulo  $\alpha_{i-1}$  esta formado por la proyección del eje  $i$  sobre el plano formado por el eje  $i - 1$  y cuya normal es la perpendicular común,  $d_i$  es la distancia desde la intersección de la perpendicular común  $a_{i-1}$  con el eje  $i$  medido a lo largo del mismo, hasta la intersección con la perpendicular  $a_i$ ,  $\theta_i$  es el angulo formado por  $\alpha_{i-1}$  y  $a_i$  en la intersección con el eje  $i$ .

Para describir la posición relativa de cada vínculo respecto de los otros, es necesario adjuntar un sistema de referencia a cada uno de ellos denominada trama, para tal propósito se hace uso del algoritmo propuesto también por Denavit-Hartenberg, el cual utiliza los parámetros de vínculo y articulares antes descritos para indicar como se deben colocar y orientar dichas trama, la convención no es única pero todas en general convergen en la misma idea central, una de ellas se describe a continuación.

El eje  $\hat{Z}$  de la trama  $\{i\}$ , llamado  $\hat{Z}_i$ , coincide con el eje de articulación  $i$ . El origen de la trama  $\{i\}$  se ubica en donde la perpendicular  $a_i$  intercepta al eje de articulación  $i$ . El eje  $\hat{X}_i$  apunta hacia  $a_i$  en la dirección de la articulación  $i$  hacia la articulación  $i + 1$ . En el caso de que  $a_i = 0$ ,  $\hat{X}_i$  es normal al plano de  $\hat{Z}_i$  y de  $\hat{Z}_{i+1}$ . La torsión de vínculo  $\alpha_i$  es definida midiéndola en sentido de la mano derecha sobre  $\hat{X}_i$ , y por último  $\hat{Y}_i$  se forma mediante la regla de la mano derecha para completar la  $i$ -ésima trama, un esquema de las tramas adjuntas es mostrada en la figura 2.13 (Craig, 2006).

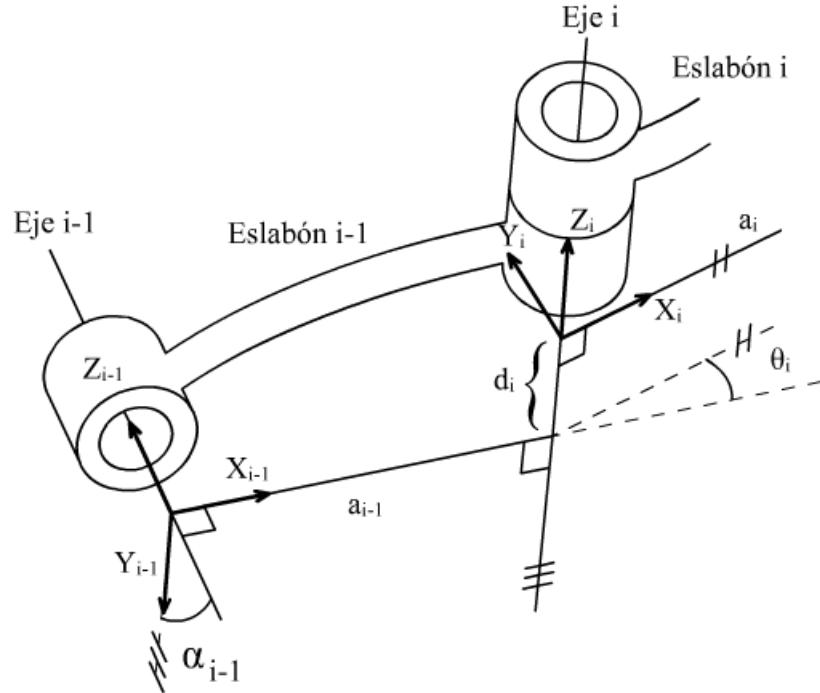


Figura 2.13: Tramas adjuntas al vínculo  $i$  y al vínculo  $i - 1$ , según la convención basada en los parámetros propuestos por Denavit-Hartenberg.

De modo que los cuatro parámetros vínculo-articulares pueden describirse en términos de las tramas asociadas mediante la convención, de la siguiente forma:

- $a_i$  = la distancia de  $\hat{Z}_i$  a  $\hat{Z}_{i+1}$  medida sobre  $\hat{X}_i$ ;

- $\alpha_i$ = el ángulo de  $\hat{Z}_i$  a  $\hat{Z}_{i+1}$  medido sobre  $\hat{X}_i$ ;
- $d_i$ = la distancia de  $\hat{X}_{i-1}$  a  $\hat{X}_i$  medida sobre  $\hat{Z}_i$ ; y
- $\theta_i$ = el ángulo de  $\hat{X}_{i-1}$  a  $\hat{X}_i$  medido sobre  $\hat{Z}_i$ .

De lo cual puede resumirse una serie de pasos iterativos para ubicar las tramas a cualquier mecanismo de robot manipulador:

1. Identificar los ejes de articulación para cada vinculo del mecanismo y trazar lineas a lo largo de dichos ejes. Seleccionar dos ejes adyacentes y etiquetarlos como  $i$  e  $i + 1$ .
2. Ubicar el parámetro de longitud de vinculo  $a$  que es la perpendicular común o el punto de intersección entre las dos líneas y asociar en dicho punto el origen de la trama asociada al vinculo.
3. Asignar el eje  $\hat{Z}_i$  para que apunte sobre el  $i$ -ésimo eje de articulación.
4. Asignar el eje  $\hat{X}_i$  para que apunte sobre la perpendicular común o, si los ejes se interceptan, asignar  $\hat{X}_i$  para que sea normal al plano que contiene los dos ejes.
5. Ej eje  $\hat{Y}_i$  que completa el sistema de coordenadas se determina mediante el uso de la regla de la mano derecha.
6. Asignar la trama  $\{0\}$  para que concuerde con la trama  $\{1\}$  cuando la primera variable de articulación sea cero. Para la última trama  $\{N\}$  seleccionar la ubicación del origen y la dirección de  $\hat{X}_N$  de forma arbitraria, pero de manera que haga que la mayor parte de los parámetros de los vínculos sean cero.

Con el uso de los cuatro parámetros de Denavit-Hartenberg y una vez asociadas las tramas de forma correspondiente, es posible determinar la matriz de transformación homogénea que describe la posición y orientación del efecto final con la base del robot correspondiendo al modelo cinemático directo del manipulador, la cual es descrita mediante la ecuación 2.1 y se obtiene realizando las operaciones de translación y rotación que están determinadas por los cuatro parámetros vinculo-articulares entre el vinculo  $i$  e  $i - 1$  (Craig, 2006).

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.1)$$

donde  $c\theta_i = \cos\theta_i$  y de igual forma debe entenderse que  $s\theta_i = \sin\theta_i$ , por simplicidad y claridad.

### 2.5.2. Modelo dinámico

Conocer las fuerzas y momentos presentes en el robot mientras permanece en movimiento bajo el efecto de fuerzas, velocidades y aceleraciones, es crucial en muchos aspectos para el diseño del robot, como primera instancia mediante el modelo dinámico es posible conocer los torques o fuerzas articulares requeridas para que el robot se mueva de una forma determinada. Existen diversas formas de abordar el calculo del modelo dinámico inverso, los más comunes son el método de las ecuaciones de movimiento de Euler-Lagrange, que hace uso del lagrangiano del sistema considerando un balance de las energías potencial y cinética, y el método iterativo de Newton-Euler. Con ambas metodologías se puede llegar al modelo dinámico inverso, el cual consiste en: conociendo las trayectorias articulares deseadas del robot, en posición  $\theta$ , velocidad  $\dot{\theta}$  y aceleración  $\ddot{\theta}$ , así como todos los parámetros iniciales (masas e inercias de los eslabones) y dimensionales (longitudes de eslabones y distancias a sus respectivos de masas) del robot, obtener los torques articulares que se requiere para que el robot se mueva siguiendo dichas trayectorias articulares deseadas (Craig, 2006).

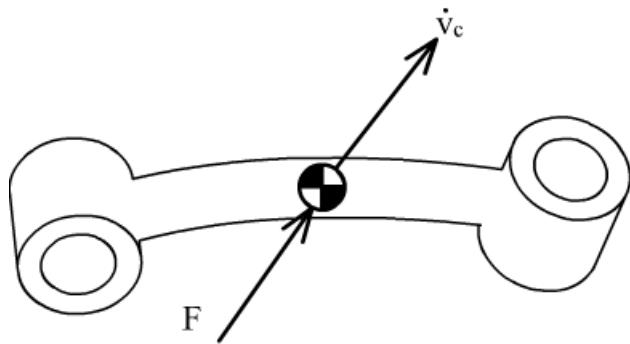


Figura 2.14: Eslabón sometido a aceleración lineal, generado por una fuerza  $F$  aplicado en su centro de masa.

El algoritmo iterativo de Newton-Euler, parte del balance de fuerzas  $F$  y momentos  $N$  que actúan en un eslabón, esto considerando que el movimiento descrito por dicho eslabón puede descomponerse en un movimiento lineal y otro de rotación. Como se observa en la figura 2.14, para que un eslabón de masa  $m$  acelere de forma lineal con una aceleración  $\dot{v}_c$ , se requiere de una fuerza  $F$  aplicada en su centro de masa, la cual está determinada según la segunda ley de Newton como:

$$F = m\dot{v}_c \quad (2.2)$$

Por otra parte la ecuación de Euler que es equivalente a la de Newton, toma en cuenta el momento  $N$  que experimenta el eslabón cuando está sometido a una velocidad angular  $\omega$  y a una aceleración angular  $\dot{\omega}$ , medido en el centro de masa del mismo (ver figura 2.15), resultando en:

$$N = I_c\dot{\omega} + \omega \times I_c\omega \quad (2.3)$$

Donde  $I_c$  es el tensor de inercia descrito respecto del sistema coordenado solidario al centro de masa del eslabón.

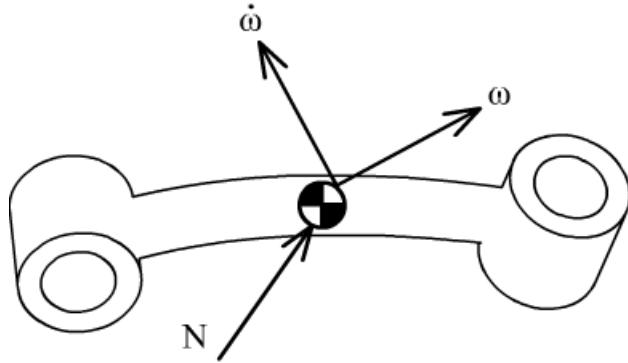


Figura 2.15: Momentos ejercido en un eslabón cuando este se mueve con una velocidad y aceleración anulares.

A partir de las ecuaciones (2) y (3) se observa que para conocer la fuerza y el momento experimentado por un eslabón en movimiento, se requiere conocer su aceleraciones lineal  $\dot{v}_c$  y angular  $\dot{\omega}$ , así como la velocidad angular  $\omega$  que experimenta. Sin embargo un robot manipulador posee varios eslabones vinculados entre si mediante articulaciones, con lo cual la fuerza y momento experimentado por cada eslabón es el resultado de un balance de fuerzas y momentos provocados por la interacción con los eslabones adyacentes. Por tanto el algoritmo de Newton-Euler consiste en determinar las fuerzas y momentos que experimenta cada eslabón del robot de forma iterativa, denominadas iteraciones salientes para calcular las aceleraciones y velocidades de cada eslabón, considerando como se afectan entre sí comenzando desde el primer eslabón hasta el eslabón  $n$ , e iteraciones entrantes para calcular las fuerzas y momentos aplicados a cada eslabón comenzando con el efecto final hasta la base del robot (Craig, 2006).

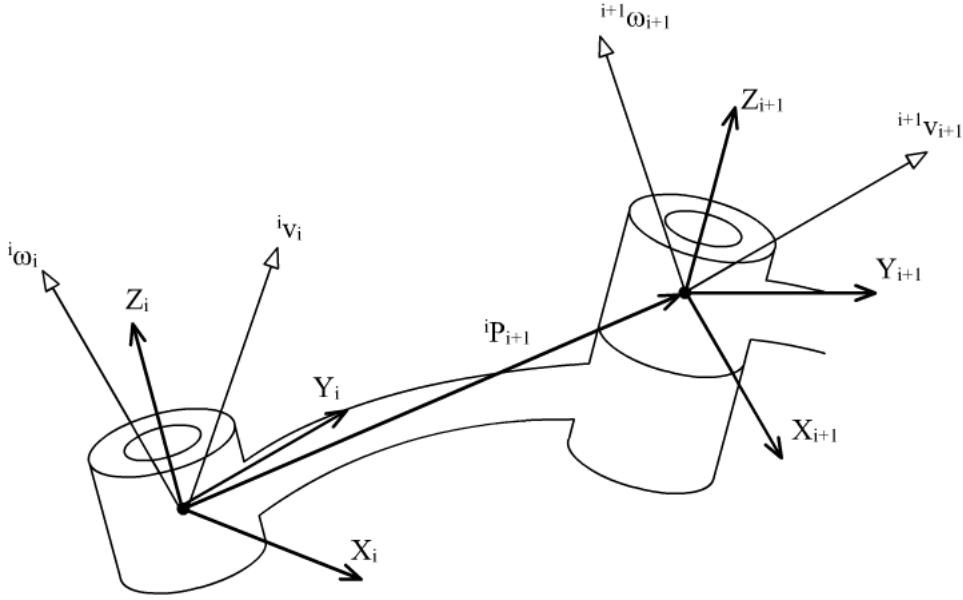


Figura 2.16: Velocidades anulares  $\omega$  y lineales  $v$ , entre dos eslabones adyacentes.

**Iteraciones salientes** Dado que los eslabones del robot están vinculados unos con otros, existe una propagación de velocidades angulares y aceleraciones lineales y angulares, como se observa en la figura 2.16, dicha propagación comienza con el primer eslabón del robot, hasta el efecto final. De la figura 2.16 se observa que la velocidad angular propagada por el eslabón  $i$ -ésimo, al eslabón  $i+1$ , está determinada por (4) (Craig, 2006) (Schilling, 1990):

$${}^{i+1}\omega_{i+1} = {}_i^{i+1} R^i \omega_i + \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (2.4)$$

donde la velocidad angular propagada  ${}^{i+1}\omega_{i+1}$ , es la suma de la velocidad angular del eslabón anterior proyectada en el sistema coordinado  $i+1$  mediante  ${}_i^{i+1}R^i \omega_i$ , más el componente de velocidad angular  $\dot{\theta}_{i+1}$  del eslabón  $i+1$ , considerando su componente vectorial en el vector unitario  ${}^{i+1}\hat{Z}_{i+1}$  como  $\dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1}$ .

De forma similar la aceleración angular propagada al eslabón  $i+1$  puede obtenerse al derivar la ecuación (), resultando en:

$${}^{i+1}\ddot{\omega}_{i+1} = {}_i^{i+1} R^i \ddot{\omega}_i + {}_i^{i+1} R^i \omega_i \times \dot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} + \ddot{\theta}_{i+1} {}^{i+1}\hat{Z}_{i+1} \quad (2.5)$$

La velocidad lineal propagada está determinada por:

$${}^{i+1}v_{i+1} = {}_i^{i+1} R ({}^i v_i + {}^i \omega_i \times {}^i P_{i+1}) \quad (2.6)$$

donde  ${}^i P_{i+1}$  es el vector de posición que ubica el origen del sistema de coordenadas

del eslabón  $i + 1$  en términos del sistema de coordenadas del eslabón  $i$  (Craig, 2006).

De () se deduce que la velocidad lineal propagada al eslabón  $i + 1$ :  ${}^{i+1}v_{i+1}$ , es la proyección al sistema coordenado  $i + 1$ , de la suma de la velocidad lineal del eslabón  $i$ :  ${}^i v_i$  más la componente de velocidad lineal provocada por la rotación del eslabón  $i$ :  ${}^i \omega_i \times {}^i P_{i+1}$ , mediante la matriz de rotación que va del eslabón  $i$  al  $i + 1$ :  ${}^i R$ .

Al derivar (6) se obtiene la aceleración lineal propagada  ${}^{i+1}\dot{v}_{i+1}$  como:

$${}^{i+1}\dot{v}_{i+1} = {}^i R [{}^i \dot{\omega}_i \times {}^i P_{i+1} + {}^i \omega_i \times ({}^i \omega_i \times {}^i P_{i+1}) + {}^i \dot{v}_i] \quad (2.7)$$

de () se aprecia que el método también requiere conocer la aceleración lineal en el centro de masa  ${}^i \dot{v}_{C_i}$  descrito en un sistema de coordenadas solidario al centro de masa  $\{C\}$ :

$${}^i \dot{v}_{C_i} = {}^i \dot{\omega}_i \times {}^i P_{C_i} + {}^i \omega_i \times ({}^i \omega_i \times {}^i P_{C_i}) + {}^i \dot{v}_i \quad (2.8)$$

donde  ${}^i P_{C_i}$  es el vector de posición que ubica al origen del sistema de coordenadas solidario al centro de masa  $\{C\}$ , respecto del origen del sistema de coordenadas del eslabón  $i$ -esimo.

**Iteraciones entrantes** Conociendo las aceleraciones anulares y lineales es posible aplicar las ecuaciones de Newton-Euler para conocer las fuerzas  $F_i$  y momentos  $N_i$  resultantes que afectan a cada eslabón. De la figura 2.10 se observan las fuerzas y momentos que son ejercidos entre dos eslabones adyacentes (Craig, 2006). La fuerza resultante aplicada en el centro de masas en el eslabón  $i$ -esimo, se obtiene al realizar la sumaria de todas las fuerzas que afectan a mismo:

$${}^i F_i = {}^i f_i - {}^i_{i+1} R {}^{i+1} f_{i+1} \quad (2.9)$$

de forma similar el momento resultante aplicado en el centro de masas del eslabón  $i$ -esimo está dato por la sumatorio de todos los momentos que afectan al eslabón como:

$${}^i N_i = {}^i n_i - {}^i_{i+1} R {}^{i+1} n_{i+1} - {}^i P_{C_i} \times {}^i F_i - {}^i P_{i+1} \times {}^i_{i+1} R {}^{i+1} f_{i+1} \quad (2.10)$$

Las expresiones de propagación hacia atrás de las fuerzas y momentos desde el eslabón  $n$  hasta el eslabón 1, se obtiene al despejar de () y ()  ${}^i f_i$  y  ${}^i n_i$  respectivamente:

$${}^i f_i = {}^i F_i + {}^i_{i+1} R {}^{i+1} f_{i+1} \quad (2.11)$$

$${}^i n_i = {}^i N_i + {}_{i+1}^i R^{i+1} n_{i+1} + {}^i P_{C_i} \times {}^i F_i + {}^i P_{i+1} \times {}_{i+1}^i R^{i+1} f_{i+1} \quad (2.12)$$

donde  ${}^i f_i$  y  ${}^i n_i$  son la fuerza y el momento aplicados en la articulación del eslabón i-esimo, de tal forma que es posible calcular el torque  ${}^i \tau_i$  que experimenta el eslabón i-esimo como:

$${}^i \tau_i = {}^i n_i^{Ti} \hat{Z}_i \quad (2.13)$$

es decir, el torque  ${}^i \tau_i$  es la componente en el eje de rotación  $Z$  (eje de rotación asignado por convención de Denavit-Hartenberg) paralelo al vector unitario  ${}^i \hat{Z}_i$  del sistema de coordenadas solidario al eslabón i-esimo (Craig, 2006).

#### 2.5.2.1. Algoritmo iterativo de Newton-Euler para modelo dinámico inverso

A continuación se presenta el algoritmo iterativo basado en las ecuaciones de Newton-Euler para la obtención del modelo dinámico inverso para robots con articulaciones rotacionales. El cual consta de dos etapas de iteraciones: las iteraciones hacia delante que van desde el primer eslabón móvil del robot hasta el eslabón n, donde se calculan las aceleraciones angulares y lineales propagadas. En a segunda etapa de iteraciones hacia atrás, se obtienen los momentos de torsión asociados a cada eslabón del robot, desde el eslabón n hasta el primer eslabón móvil del robot (Craig, 2006).

##### Iteraciones hacia adelante: Para i=0 hasta n-1

$${}^{i+1} \omega_{i+1} = {}_i^{i+1} R^i \omega_i + \dot{\theta}_{i+1} {}^{i+1} \hat{Z}_{i+1} \quad (2.14)$$

$${}^{i+1} \dot{\omega}_{i+1} = {}_i^{i+1} R^i \dot{\omega}_i + {}_i^{i+1} R^i \omega_i \times \dot{\theta}_{i+1} {}^{i+1} \hat{Z}_{i+1} + \ddot{\theta}_{i+1} {}^{i+1} \hat{Z}_{i+1} \quad (2.15)$$

$${}^{i+1} \dot{v}_{i+1} = {}_i^{i+1} R [{}^i \dot{\omega}_i \times {}^i P_{i+1} + {}^i \omega_i \times ({}^i \omega_i \times {}^i P_{i+1}) + {}^i \dot{v}_i] \quad (2.16)$$

$${}^i \dot{v}_{C_i} = {}^i \dot{\omega}_i \times {}^i P_{C_i} + {}^i \omega_i \times ({}^i \omega_i \times {}^i P_{C_i}) + {}^i \dot{v}_i \quad (2.17)$$

$${}^{i+1} F_{i+1} = m_{i+1} {}^{i+1} \dot{v}_{C_{i+1}} \quad (2.18)$$

$${}^{i+1}N_{i+1} = {}^{C_{i+1}}I_{i+1}{}^{i+1}\dot{\omega}_{i+1} + {}^{i+1}\omega_{i+1} \times {}^{C_{i+1}}I_{i+1}{}^{i+1}\omega_{i+1} \quad (2.19)$$

**Iteraciones hacia atrás: Para i = n, hasta 1**

$${}^i f_i = {}^i F_i + {}^i_{i+1} R^{i+1} f_{i+1} \quad (2.20)$$

$${}^i n_i = {}^i N_i + {}^i_{i+1} R^{i+1} n_{i+1} + {}^i P_{C_i} \times {}^i F_i + {}^i P_{i+1} \times {}^i_{i+1} R^{i+1} f_{i+1} \quad (2.21)$$

$${}^i \tau_i = {}^i n_i^{Ti} \hat{Z}_i \quad (2.22)$$

## 2.6. Modelo V

El modelo V surge como herramienta en el desarrollo de sistemas de software en 1997, creado por la Administración Federal Alemana. Sin embargo debido a sus características fue paulatinamente adoptado en el diseño mecatrónico ya que el modelo esta basado en la descomposición del sistema en sub-sistemas de forma descendente, partiendo del sistema total en su complejidad hasta reducirlo en sistemas de dominio específico, para posteriormente realizar validaciones ascendentes reintegrando el sistema ten su totalidad. De forma que en todo momento se hace presente la necesidad de la validación de los requisitos de los sistemas mediante el estado actual del modelo.

Por tal motivo y realizando algunas adaptaciones y modificaciones la Asociación Alemana de Ingenieros (VDI por sus siglas en Alemán), sugirió el uso de la metodología del modelo V para sistemas mechatrónicos mediante el documento «VDI lineamiento 2206: Metodología de diseño para sistemas mechatrónicos», misma que es regida por la estructura mostrada en la figura 2.17. Las fases del modelo se describen a continuación (Gausemeler, 2002).

### 2.6.1. Etapas del ciclo de vida del producto definidas por el modelo V

#### 2.6.1.1. Requisitos

En esta fase se definen las tareas a realizar por el sistema mediante requisitos, que se convierten en restricciones de diseño para las fases posteriores. De forma que el sistema queda completamente definido.

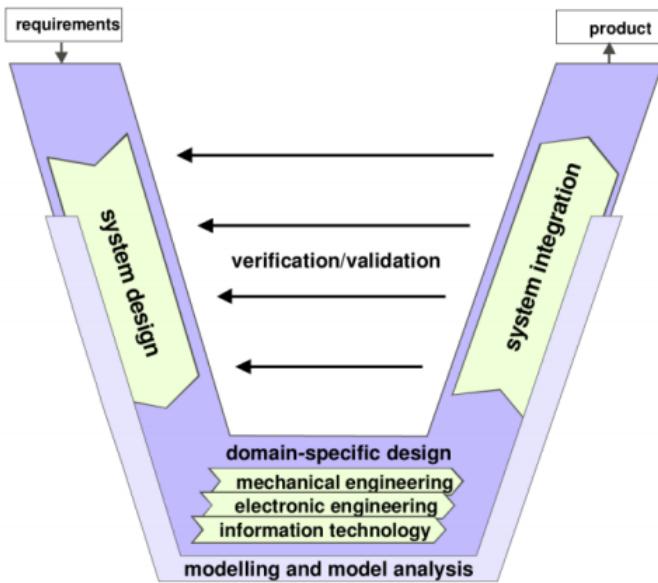


Figura 2.17: Estructura del modelo V sugerido por el lineamiento VDI 2206 (Sell, 2005).

#### 2.6.1.2. Diseño de sistema

Tiene como objetivo la modularización del sistema total en sub-sistemas de dominio específico, mediante la conceptualización física y lógica de las funciones del sistema total. De forma que las soluciones a cada función o estructura lógica se convierte en un sub-sistema, normalmente satisfecho por un principio físico, que pertenece a un dominio específico de ingeniería.

#### 2.6.1.3. Diseño de dominio específico

En esta fase se desarrolla la solución dada a cada función conceptual de la fase previa, de forma detallada en el dominio específico mediante modelos (físicos, analíticos y numéricos), dando prioridad a las funciones críticas.

#### 2.6.1.4. Integración del sistema

Una vez establecidas las soluciones a nivel sub-sistemas se precede en esta fase a la reintegración del sistema de forma ascendente, para analizar las dependencias e interacciones multi-dominio.

### 2.6.1.5. Verificaron y validación

En el proceso de reintegración del sistema total a partir de los diferentes sub-sistemas tiene que estar regido por una verificación y validación continua para garantizar que las características del sistemas son las deseadas o esperadas.

### 2.6.1.6. Modelado y análisis del modelo

Los modelos están presentes en todas las fases del modelo V, permitiendo realizar las validaciones y reestructuración de los elementos del sistema normalmente mediante herramientas de simulación numérica.

### 2.6.1.7. Producto

Es el resultado final de todo el proceso en el modelo V, sin embargo esto no sugiere necesariamente que sea un producto terminado, pero si en una etapa más madura, haciendo posible la iteración en el mismo modelo V (Gausemeler, 2002).

## 2.7. Algoritmos genéticos

Los algoritmos genéticos forman parte de los algoritmos evolutivos, surgieron en el ámbito de los sistemas adaptativos artificiales SAS en la década de los setentas, aplicando en sus metaherurísticas conceptos neodarwinianos , es decir la selección natural propia de la teoría de Darwin y la crusa y mutación tomados de los trabajos en genética de Mendel. El primer trabajo en mencionar los operadores específicos para permitir la «evolución» en los algoritmos genéticos fue el desarrollado por Holland, sin embargo, su trabajo fue posible gracias a investigaciones previas en dicho campo (Cuevas, 2016).

Los parámetros propuestos por Holland para permitir la evolución son: selección, crusa y mutación. Algunas de las ventajas de los algoritmos genéticos es que pueden trabajar con poco información acerca de la función objetivo, y al ser de carácter poblacional es menos propenso a caer en mínimos locales en la función objetivo, donde los métodos clásicos de gradiente tienen problemas, como por ejemplo en la función mostrada en la figura 2.18 se observan una gran cantidad de mínimos locales.

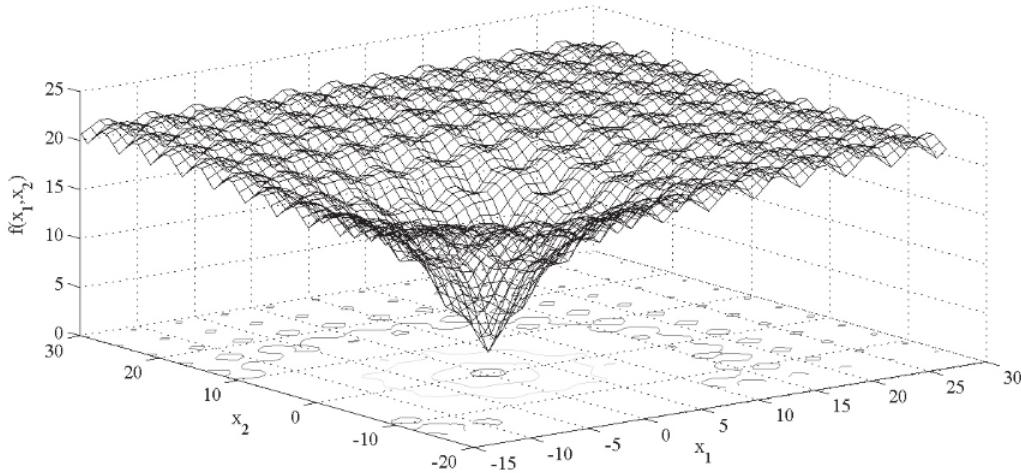


Figura 2.18: Gráfica de la función de Ackley, la cual es una función compleja en el ámbito de la optimización por la gran cantidad de mínimos locales (Cuevas, 2016).

El procedimiento general dentro de los algoritmos genéticos es el de un programa que genera una población, del cual selecciona un par de padres que se consideran mejor adaptados utilizando el criterio de la evaluación de la función objetivo, para posteriormente realizar una crusa de sus genes y generar una nueva población de hijos los cuales mediante técnicas de probabilidad son mutados, e iterando sobre esta misma idea se llega a la población optima, donde se refleja que no existe diferencias significativas entre individuos ya que todos se aproximan a la solución optima de la función objetivo (Cuevas, 2016).

### 2.7.1. Inicialización

Es el primer procedimiento realizado por el AG, donde se genera la población a partir de números binarios aun que también es posible realizarlo con números decimales. La población debe de estar dentro de un rango específico de búsqueda, estableciendo para ello los límites inferior y superior (Cuevas, 2016). Normalmente la población está descrita en términos de vectores binario, los cuales son generados mediante:

$$b_{i,j} 2 * \text{rand}() - 1 \quad (2.23)$$

$$i = 1, \dots, N_p, \quad , j = 1, \dots, N_b * d$$

donde  $N_p$  es el tamaño de la población,  $N_b$  es el numero de bits con que se representa a cada elemento, y  $d$  el numero de dimensiones del problema.. Por procedimiento se realiza

un conversión a números decimales después de la inicialización binaria mediante:

$$x_{i,n} = l_i + \frac{\sum_{k=0}^{N_b-1} (2^k * b_{i,k+1})}{\sum_{k=0}^{N_b-1} (2^k)} (u_i - l_i) \quad (2.24)$$

Dependiendo de la cantidad de números binario utilizados en la representación de la población, mayor será la resolución de los números decimales creados (Cuevas, 2016).

### 2.7.2. Selección de padres

Una etapa importante en el AG es la adecuada selección de los padres en una población para su posterior cruce, existen diferentes métodos de selección, a continuación se enlistan los principales o mas utilizados.

- Selección proporcional.
- Selección por rangos.
- Selección por torneo.
- Selección de estado estable.

La forma de selección mediante el método de selección proporcional es la evaluación de la función objetivo de las soluciones candidatas, es decir lo individuos. Dentro de este método se destacan las siguientes formas de selección. La información detallada de los procedimientos de selección puede encontrarse en (Cuevas, 2016).

1. Método de la ruleta.
2. Sobrante estocástico.
3. Universal estocástica.
4. Muestreo determinista.

### 2.7.3. Cruza

De forma análoga al intercambio de información biológica entre los seres vivos, los AG permiten emular este comportamiento de intercambio, no considerando genes de los individuos, sino compartiendo información procedente de los vectores que los constituyen. Existen diferentes aproximaciones a la cruce entre individuos en los AG.

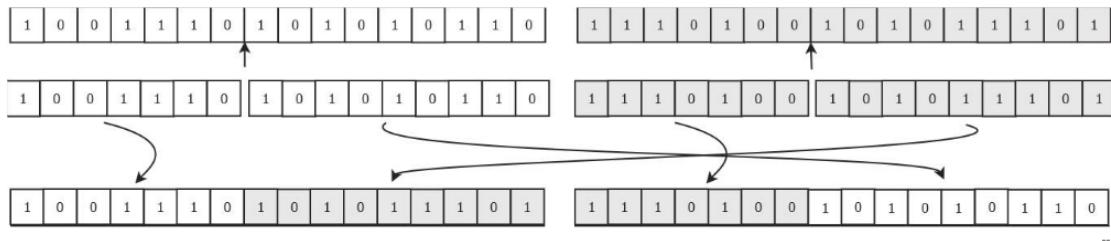


Figura 2.19: Ejemplo de mutación de un solo punto en dos individuos de 16 bits (Cuevas, 2016).

A continuación se describen los procedimientos de cruce cuando los individuos son representados por cadenas de números binarios o decimales (Cuevas, 2016).

#### 2.7.3.1. Cruza de un sólo punto

Consiste en la división de las cadenas de un par de individuos en una posición determinada, para posteriormente combinar cada parte seccionada de ambos para crear un par nuevo de individuos. Como se observa en la figura 2.19, dos individuos de 16 bits de longitud de cadena son seccionados en el bit 7, de forma que cada individuo queda seccionado en dos cadenas de 7 bits y 9 bits, las mismas que son combinadas: los 7 bits de un individuo mas los 9 bits del otro, de forma que se crean dos especímenes nuevos.

#### 2.7.3.2. Cruza de dos puntos o multipunto

La crusa de punto simple es comúnmente empleada debido en gran parte por su simplicidad, sin embargo, resulta inconveniente cuando los individuos están construidos por cadenas muy largas, aumentando la probabilidad de no explorar todas las posibles cruzas y por tanto dejar de un lado posibles soluciones. Este inconveniente es compensado al considerar la crusa por dos puntos o multipunto, que en esencia es el mismo procedimiento que para un solo punto solo que en este caso la cadena de los individuos es segmentada en n puntos por lo general para  $n < 3$ . En ejemplo de este tipo de crusa se aprecia en la figura 2.20 (Cuevas, 2016).

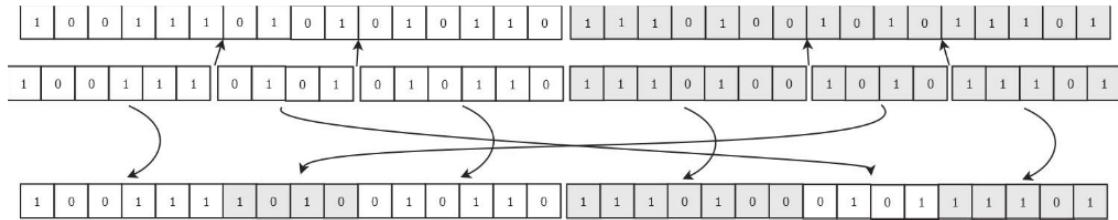


Figura 2.20: Ejemplo de crusa multipunto en individuos de codificación binaria (Cuevas, 2016).

#### 2.7.3.3. Cruza uniforme

En este tipo de crusa no se especifica de forma determinada el lugar de la segmentación en las cadenas de los especímenes, sino que se crea una combinación de los datos de un par de ellos mediante la selecciona de información individual utilizando para ello una selección por probabilidad generalmente no mayor a 0.5. Un ejemplo de crusa distribuida en dos especímenes de 16 bist se aprecia en la figura 2.21, donde se observa que debido a las características de probabilidad es posible crear dos nuevos especialmente diferentes sin necesidad de establecer el lugar específico de corte en ambas cadenas (Cuevas, 2016).

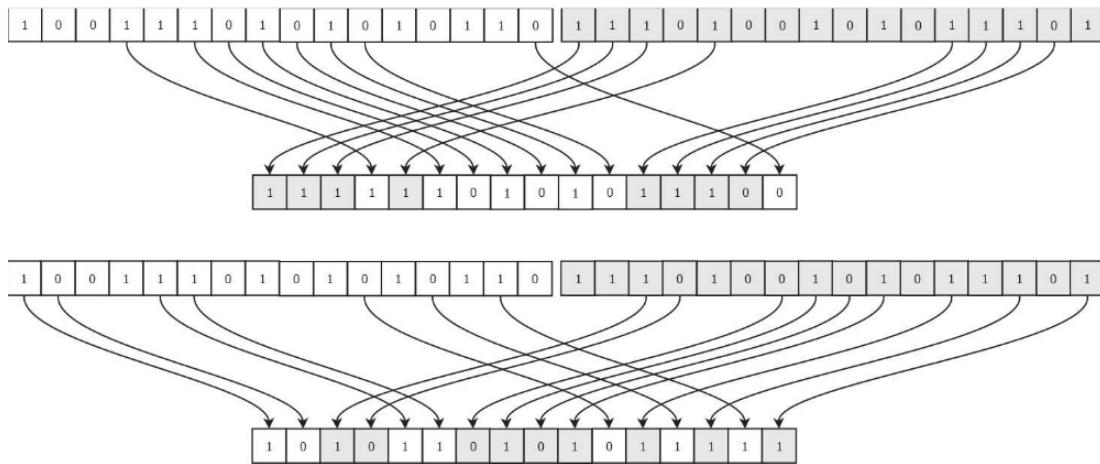


Figura 2.21: Ejemplo de crusa uniforme en dos especímenes de 16 bits de longitud (Cuevas, 2016).

#### 2.7.3.4. Mutación

La mutación para individuos codificados en cadenas binarias o decimales, se logra intercambiando la información de determinada posición en su cadena seleccionada de

forma aleatoria:  $x = \{x_1, \dots, x_m, \dots, x_i\}$  para  $i = 1, \dots, n$ , por otro numero generado de forma aleatoria mente de forma distribuida  $x'_m$  mediante la formula (25) (Cuevas, 2016):

$$x'_m = l_m + x_m(u_m - l_m) \quad (2.25)$$

donde  $u_m$  y  $l_m$  corresponde a los limites del espacio de búsqueda para la parte a mutar, pero que son parte del conjunto del espacio de búsqueda total de forma que:  $u = \{u_1, \dots, u_m, \dots, u_i\}$  y  $l = \{l_1, \dots, l_m, \dots, l_i\}$  [9].

#### 2.7.3.5. Selección del más apto

En los AG se realiza la selección de los especialmente más aptos para pasar a la siguiente generación, generalmente mediante un criterio de competición entre padres y los hijos cruzados y mutados. Utilizando para ello el criterio del mejor resultado en la evaluación de la función objetivo (Cuevas, 2016).

## 2.8. Ingeniería de sistemas basado en modelos

Uno de los principales problemas en la gestión de proyectos de ingeniería, tiene que ver con la evolución del sistema diseñado en las diferentes faces del ciclo de vida del producto, en especial en sistemas complejos donde más de un equipo de ingeniería tienen que intervenir, ya que esto da origen a requerimientos, restricciones y soluciones de naturaleza multi-domino o de dominio cruzado. Normalmente este problema de gestión de la información durante la evolución del proyecto se lleva a cabo mediante el enfoque de la ingeniería de sistemas basada en documentos, es decir que se tiene un documento centralizado basado en hojas de calculo, presentaciones, documentos de texto, etc. a las cuales cada equipo de trabajo de ingeniería tiene acceso, sin embargo ya que este tipo de estructura de documentos no es dinámica, es decir las modificaciones no se actualizan en tiempo real, las actualizaciones son tardadas y propensas a error, y más aún el rastreo de inconsistencias en el diseño se convierten en una tarea que demanda tiempo y conlleva a numerosas iteraciones en el proceso de diseño, al mismo tiempo que no permite una transmisión de información de forma clara entre los diversos equipos de ingeniería involucrados, por tal motivo en las ultimas décadas se ha optado por un cambio de paradigma en la gestión de la ingeniería de sistemas, pasando de un enfoque basado en documentos a la Ingeniería de Sistemas Basado en Modelos (MBSE por sus siglas en inglés) (Casse, 2017) (Delligatti, 2014) (Friedenthal, et al. 2012).

Este paradigma consiste en desarrollar un modelo gráfico centralizado escrito en

un lenguaje común para todos, al cual cada equipo de ingeniería tiene acceso, y que permite la integración de las diversas herramientas de software de dominio específico.

El desarrollo de la ingeniería de sistemas basado en modelos requiere principalmente de tres aspectos fundamentales:

1. Lenguaje.
2. Software especializado.
3. Metodología.

Actualmente existen diversos lenguajes de modelado gráfico para el desarrollo de la ingeniería de sistemas basado en modelos, pero el más comúnmente utilizado es SysML, que es un lenguaje para describir sistemas de naturaleza ciber-física de forma gráfica. SysML surge a partir de UML (Unified Modeling Language) como un subconjunto, para satisfacer las necesidades de la ingeniería de sistemas, por lo que comparte algunos diagramas, sin embargo SysML cuenta con un set de nuevos diagramas extendidos de UML, como se aprecia en la figura 2.22 (Delligatti, 2014).

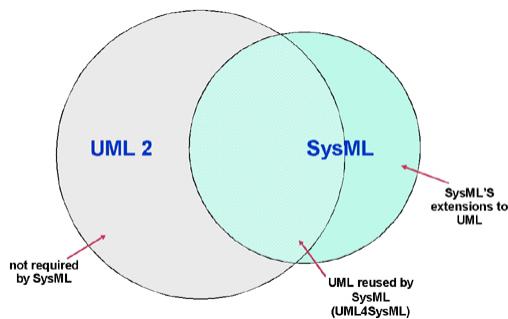


Figura 2.22: Relación entre UML y SysML (Object Management Group, 2018).

De forma similar a UML, SysML también cuenta con una cantidad de diagramas preestablecidos, los cuales se muestran en la figura 2.23. Donde se puede apreciar cuales han sido «heredados» de UML, algunos modificados y otros completamente nuevos.

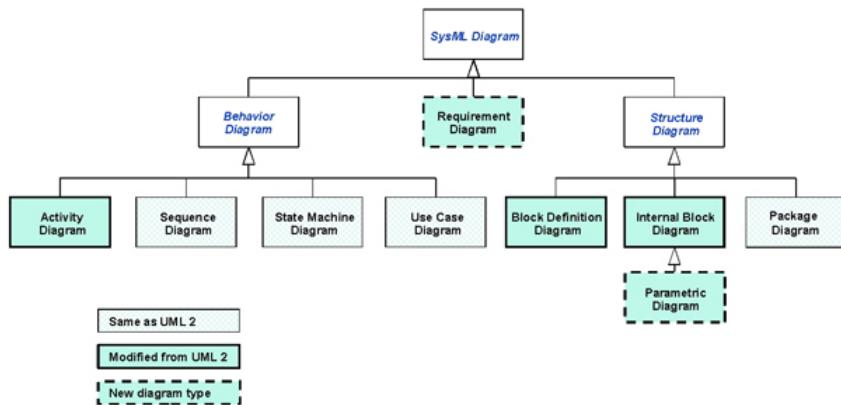


Figura 2.23: Tipos de diagramas de SysML (Object Management Group, 2018).

Por otra parte, para el desarrollo de la Ingeniería de Sistemas Basado en Modelos se requiere de un entorno de trabajo donde sea posible crear y gestionar el modelo central del sistema escrito en SysML, para tal propósito en el mercado existen programas que permiten la creación, gestión y simulación de estos modelos (Delligatti, 2014), los más comunes son por citar algunos son:

De grado comercial:

- Enterprise Architect.
- Cameo system modeler.
- Rhapsody.
- Agilian.

De distribución libre:

- Modelio
- Papyrus

Teniendo en cuenta el lenguaje y la plataforma de desarrollo para la ingeniería del sistema a desarrollar, se requiere del seguimiento de una metodología que en este caso permita su integración con SysML. Existen metodologías estandarizadas para tal propósito (Delligatti, 2014) (Friedenthal, et al. 2012), como lo son:

- INCOSE Object-Oriented Systems Engineering Method (OOSEM).
- Weilkiens System Modeling (SYSMOD) method.

- IBM Telelogic Harmony-SE.

Sin embargo no existe una metodología general para el desarrollo de la ingeniería de sistemas basado en modelos, por tanto cada equipo de ingeniería puede hacer uso del lenguaje de modelado y el entorno de desarrollo en conjunto con la metodología o una adaptación de ella que mejor se ajuste a las necesidades de diseño, en este caso particular, debido a que se va a desarrollar un sistema mechatrónico, se usará el Modelo-V (según las especificaciones del lineamiento VDI-2206), bajo el paradigma de la Ingeniería de Sistemas Basado en Modelos.

# Capítulo 3

## Metodología de diseño mecatrónico

### 3.1. Introducción

La metodología empleada en este proyecto se basa en la implementación del modelo-V en el contexto de la Ingeniería de Sistemas Basado en Modelos, utilizando como entorno de trabajo el software Cameo System Modeler del fabricante NoMagic, en conjunto con el lenguaje gráfico de modelado SysML en todas las etapas del ciclo de vida del producto, sin considerar las etapas de mantenimiento posteriores a la fabricación del producto. De forma que en cada fase de la metodología se emplean determinados diagramas en SysML, la relación de los diagramas empleados en cada fase del modelo-V se observa en la figura 3.1.

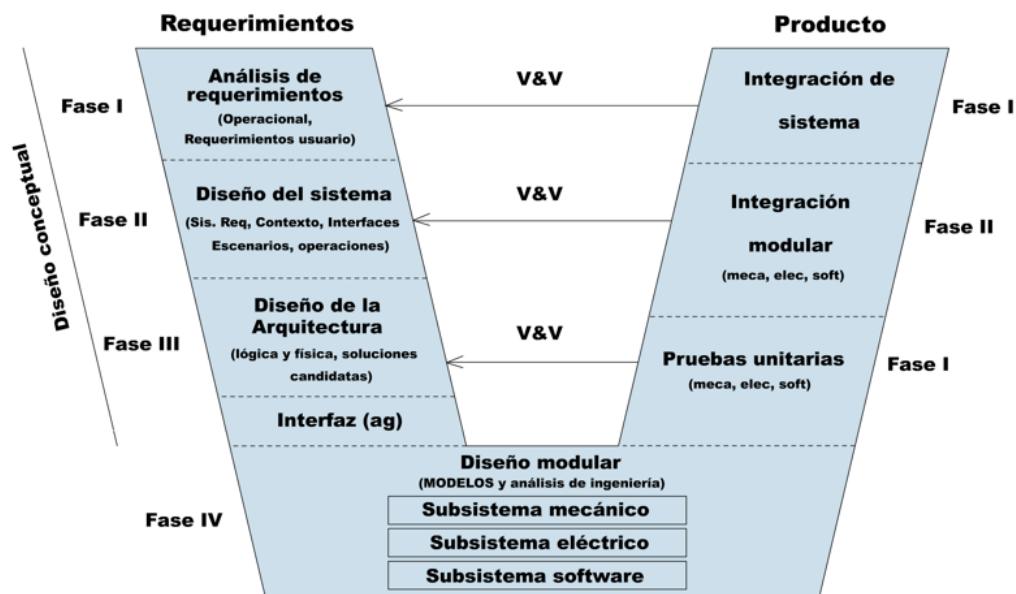


Figura 3.1: Descripción de las fases del modelo-V en la metodología propuesta.

FASE I - Análisis de los requerimientos	
1.1 Análisis operacional	
SubSección	Diagrama de SysML
1.1.1 Casos de uso	uc
1.1.2 Contexto	bbd, ibd
1.1.3 Escenario global	sd
1.1.4 Modos del sistema	stm
1.2 Requerimientos del sistema	
SubSección	Diagrama de SysML
1.2.1 Importación de requerimientos	req table, req

Tabla 3.1: Relación de los diagramas de SysML utilizados en la FASE I-Análisis de los requerimientos.

El modelo-V consta de dos etapas principales, la etapa de verificación, como se observa en la figura corresponde a la descomposición de las necesidades requeridas por el usuario, buscando siempre que las propuestas de diseño del sistema estén siempre justificadas (traceabilidad) con los requerimientos del usuario. La etapa de validación consta de tres fases, donde de forma ascendente se reconstruye el sistema validando cada diseño seleccionado en cada nivel de integración, hasta alcanzar la validación del sistema completamente integrado.

En la primera fase dentro de la etapa de **verificación**, denominada **Análisis de los requerimientos**, se busca primero realizar un **análisis operacional** del sistema, es decir determinar los **actores principales**, delimitar el **contexto del sistema**, identificar los **escenarios globales** posibles, así como los **modos de operación** del mismo, todo esto considerando al sistema como una **caja negra**. También involucra la importación y decomposición de los **requerimientos** obtenidos mediante una lista de necesidades del usuario. Cada una de estas etapas dentro de la fase **Análisis de los requerimientos** involucra el uso de diversos diagramas de SysML, lo cual se ilustra en la tabla 3.1.

En la fase II denominada **Diseño del sistema**, se realiza la **derivación** de los **requerimientos del sistema** cuidando siempre la **traceabilidad** en relación a las **necesidades del usuario** (requerimientos del usuario), se analizan las **interfaces externas** considerando al sistema una **caja negra**. Los **escenarios** particulares del sistema son obtenidos es decir, las situaciones de operación que pueden existir en el sistema (por ejemplo el escenario durante la ejecución de la marcha del robot), lo que ayuda a la obtención de las **operaciones** (acciones) que ejecuta durante los escenarios visualizados, cuidando siempre la **traceabilidad** de las operaciones con los **requeri-**

FASE III - Diseño de la arquitectura	
SubSeccion	Diagrama de SysML
3.1 Arquitectura lógica	act, satisfy requirement matrix
3.2 Arquitectura física	bdd, ibd
3.3 Soluciones candidatas	ibd

Tabla 3.3: Diagramas de SysML utilizados en la FASE III - Diseño de la arquitectura.

mientos del sistema obtenidos en esta fase. La relación de los diagramas utilizados de SysML se describen en la tabla 3.2.

FASE II - Diseño del sistema	
SubSeccion	Diagrama de SysML
2.1 Requerimientos del sistema	req table, derive requirement matrix
2.2 Interfaces externas	ibd
2.3 Escenarios del sistema	sd
2.4 Operaciones	bdd, satisfy requirement matrix

Tabla 3.2: Diagramas de SysML utilizados en la FASE II - Diseño del sistema.

En la fase III de la etapa de verificación del modelo-V se denomina **Diseño de la arquitectura**, se busca la determinación de la **estructura funcional** (lógica) y **arquitectura física** (subsistemas) del sistema estudiado, con lo que es posible determinar las diferentes **soluciones candidatas** propuestas en conjunto con los diferentes equipos de ingeniería. Los diagramas de SysML empleados en esta fase se describen en la tabla 3.3.

La etapa de Diseño conceptual finaliza con el establecimiento la arquitectura física de la solución candidata propuesta, obtenidos en la fase III mediante la generación de las soluciones candidatas, sin embargo a pesar de que la arquitectura física del sistema esta definida (tipos de actuadores, computadoras centrales, tipos de sensores, circuitos de potencia, masas iniciales, etc.), aun no se conoce la medida de sus parámetros que hacen que cada subsistema sea óptimo en relación a los demás (y por tanto todo el sistema), es decir se desconocen las **restricciones de diseño a nivel de subsistemas** que se deben satisfacer en la fase IV Diseño modular de la etapa de verificación, que hacen posible que cada equipo de ingeniería genere una solución óptima en forma local.

Para tal propósito la metodología propuesta especifica en esta **transición a la ingeniería de detalle** (de la fase III a la fase IV), el uso de **algoritmos genéticos** de forma que a través de su implementación se obtengan restricciones de diseño para cada equipo de ingeniería de los diferentes subsistemas, con el objetivo garantizar dentro de

la decomposicion del sistema, una solución optima global en la medida de lo posible.

Por tanto, en la **fase IV - Diseño modular**, se crean los diferentes **modelos** que requiere cada equipo de ingeniería, por ejemplo modelos matemáticos, físicos, eléctricos, lógicos, hidráulicos, etc. los cuales deben estar enlazados con el modelo centralizado construido con SysML en Cameo System Modeler y se debe cuidar que sus parámetros cumplan con las **restricciones de diseño** obtenidas mediante el algoritmo genético, para ello se hace uso de los **diagramas treparriscos** (par), y de **definición de bloques** (bdd).

Cabe mencionar que no es factible programar el algoritmo genético directamente en Cameo System Modeler, sino que mas bien este debe de ser ejecutado en una herramienta de ingeniería apropiada (Matlab, SciLab, etc.) cuyos resultados serán importados en elementos de restriccción (Constraint block) y utilizados dentro de los diagramas parametricos diseñados, para comenzar con la **etapa de validación** del modelo-V.

La **validación** esta especificada en la metodología propuesta en la etapa ascendente del modelo-V y consta de tres fases, la **fase I - Pruebas de unidad**, esta enfocada a validad mediate diagramas parametricos (par) enlazados con las herramientas de software de cada equipo de ingeniería las partes que conforman los subsistemas propuestos, de forma que es posible realizar pruebas con **hardware-in-the-loop** en herramientas de software externas (Matlab-Simulink) cuyos resultados son inyectados en el modelo central en Cameo System Modeler vía diagramas parametricos (Casse, 2017), con lo cual es posible validar la parte diseñada (por ejemplo un motor eléctrico).

La **fase II - Integración modular** y **fase III - Integración del sistema**, siguen la misma lógica que la fase I (en la etapa de validación), es decir se busca validar que la integración hacia arriba del sistema cumpla con las restricciones y requisitos obtenidos en la etapa de Diseño conceptual, ejecutando para ello pruebas en los sistemas físicos construidos y cuyos resultados se ven reflejados en el modelo central en Cameo System Modeler, de forma que al finalizar la validación del sistema integrado globalmente se obtiene el producto final, en una primera corrida del modelo-V. Es posible reiterar en el ciclo, si el primer producto necesita ser refinado, pero este tema queda fuera del alcance de este proyecto.

A continuacion se describe la aplicación de la metodología propuesta en el desarrollo del robot bípedo, utilizando el software Cameo System Modeler.

# Capítulo 4

## Diseño del robot bípedo

### 4.1. Fase I - Análisis de los requerimientos

Antes de comenzar la importación de los requerimientos del usuario, es necesario construir la estructura del modelo, para ello se construyeron paquetes dentro de Cameo System Modeler, que representan cada etapa y fase de la metodología propuesta. En CSM se puede representar la estructura del modelo ya sea mediante la visualización del árbol de estructura o bien mediante un diagrama de paquetes, sin embargo, este ultimo no es conveniente cuando la complejidad del modelo sea muy alta, en estos casos es mejor la visualización de la estructura del modelo mediante el árbol jerarquizado, como se observa en la figura 4.1.

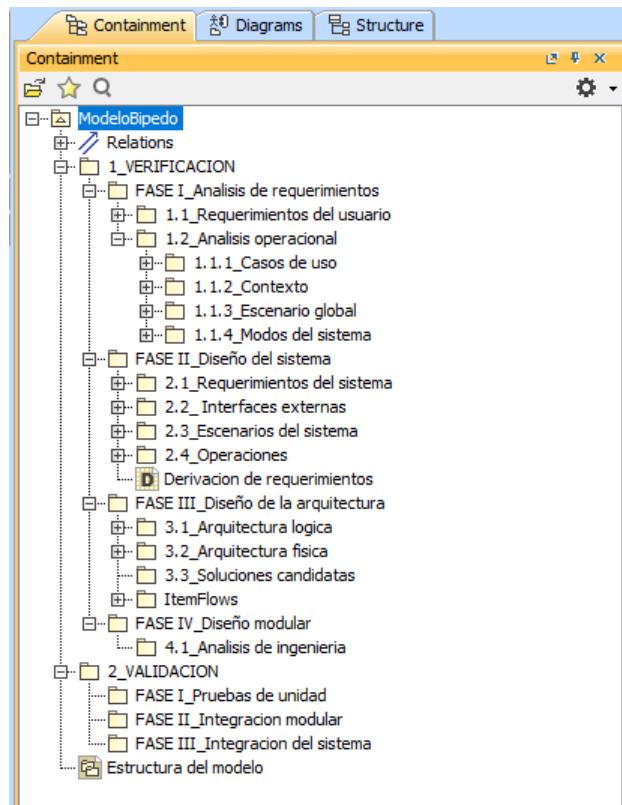


Figura 4.1: Estructura del modelo creado en Cameo System Modeler.

Para propósitos ilustrativos también se presenta el diagrama de paquetes que contiene únicamente la etapa de verificación, sin embargo carece de utilidad práctica ya que el diagrama es menos entendible cuando existe una gran cantidad de paquetes, incluso en este proyecto relativamente simple, ya no es conveniente la visualización de su estructura por este medio (ver figura 4.2).

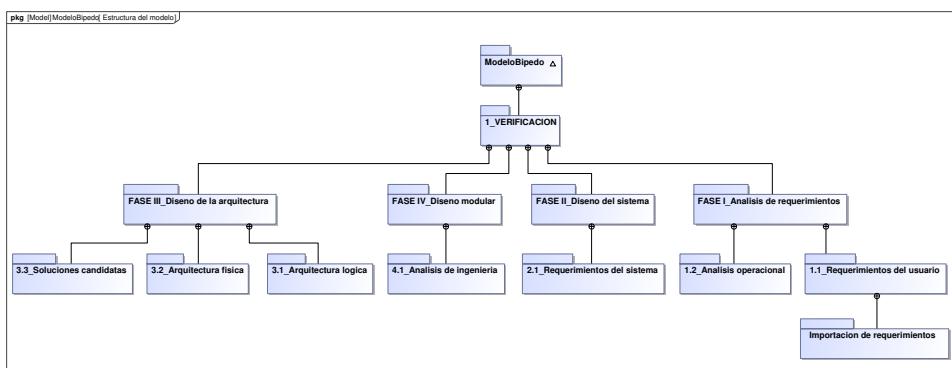


Figura 4.2: Diagrama de paquetes como forma de visualización de la estructura del modelo.

A	B	C
ID	Name	Description
UR.1	Robot bipedo	El usuario debe ser capaz de estudiar la dinámica de la marcha bipeda descrita por los humanos, específicamente en los miembros inferiores, mediante un sistema mecatrónico.
UR.2	Controlar robot	El usuario debe ser capaz de controlar el robot mediante comandos de alto nivel a distancia mediante una computadora externa al robot.
UR.2.1	Coste computacional	Debido al alto costo computacional que se requiere en la evaluación del modelo cinemático inverso para la generación de las trayectorias articulares deseadas, sean obtenidos en una computadora externa mediante software.
UR.2.2	Selección de modos	El usuario debe ser capaz de trabajar con el robot en modo manual o automático.
UR.3	Modularidad	Se espera que el robot sea transportable, modular y que cuente con un número mínimo de partes consistentes.
UR.4	Manufactura asequible	Se requiere que el robot sea construible y replicable con las herramientas de manufactura disponibles en el laboratorio de robótica o en la facultad de Ingeniería.
UR.5	Optimización mecánica	Se requiere que su estructura mecánica del robot esté optimizada.
UR.6	Posición angular	Se espera que los sensores articulares del robot deben permitir la medición de la posición angular.
UR.7	Paro de emergencia	El usuario debe ser capaz de detener la ejecución de la marca del robot en todo momento.
UR.8	Indicadores visuales	El usuario debe ser capaz de determinar visualmente el estado actual del robot

Figura 4.3: Requerimientos del usuario escritos en una hoja de cálculo.

#### 4.1.1. Requerimientos del usuario

Los requerimientos del usuario son las características y comportamientos que espera que el sistema cumpla o alcance, en este caso la lista de requerimientos fue previamente definida fuera del ambiente de CSM, en una hoja de cálculo como se observa en la figura 4.3 ya que se espera que no se espera que el usuario esté familiarizado con el uso de CSM.

Se continuó con la importación a CSM mediante el copiado y pegado de las mismas en un **Requirement Table** ya que este automáticamente convierte cada reglón de la hoja de cálculo en un elemento del modelo de tipo: Requirement, con los parámetros básicos de: ID, nombre y descripción. Existen dos formas de visualizar las dependencias que los requerimientos presentan entre ellos en CSM, una de ellas es mediante una lista en el árbol de estructura del modelo, y otra mediante un diagrama de requerimientos (req), la segunda forma es impráctica cuando existe un número elevado de requerimientos a visualizar. En la figura 4.4 se observa la tabla de requerimientos creada y la estructura de requerimientos en el contenedor.

Los requerimientos del usuario importados se describen a continuación en la tabla 4.1.

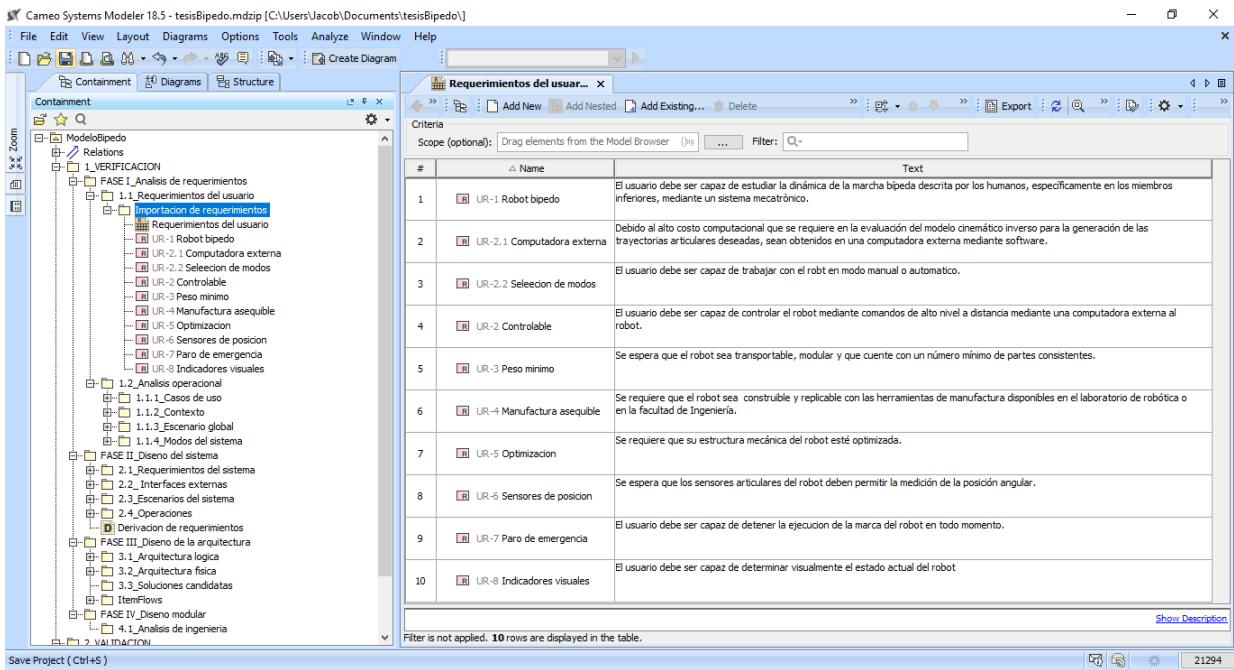


Figura 4.4: Requerimientos importados en el modelo.

Descripción

ID	Nombre	Descripción
UR.1	Robot bípedo	El usuario debe ser capaz de estudiar la dinámica de la marcha bípeda descrita por los humanos, específicamente en los miembros inferiores mediante un sistema mecatrónico.
UR.2	Controlar robot	El usuario debe ser capaz de controlar el robot mediante comandos de alto nivel a distancia mediante una computadora externa al robot.
UR.2.1	Coste computacional	Debido al alto costo computacional que se requiere en la evaluación del modelo cinemático inverso para la generación de las trayectorias articulares deseadas, sean obtenidos en una computadora externa mediante software.
UR.2.2	Selección de modos	El usuario debe ser capaz de trabajar con el robot en modo manual o automático.
UR.3	Modularidad	Se espera que el robot sea transportable, modular y que cuente con un número mínimo de partes consistentes.
UR.4	Manufactura asequible	Se requiere que el robot sea construible y replicable con las herramientas de manufactura disponibles en el laboratorio de robótica o en la facultad de Ingeniería.
UR.5	Optimización mecánica	Se requiere que su estructura mecánica del robot esté optimizada.
UR.6	Posición angular	Se espera que los sensores articulares del robot deben permitir la medición de la posición angular.
UR.7	Paro de emergencia	El usuario debe ser capaz de detener la ejecución de la marca del robot en todo momento.
UR.8	Indicadores visuales	El usuario debe ser capaz de determinar visualmente el estado actual del robot.

Tabla 4.1: Requerimientos del usuario importado.

En la figura 4.5 se aprecia la representación de la derivación de los requerimientos en un diagrama de requerimientos, lo cual es una forma gráfica de observar las relaciones entre los requerimientos del usuario.

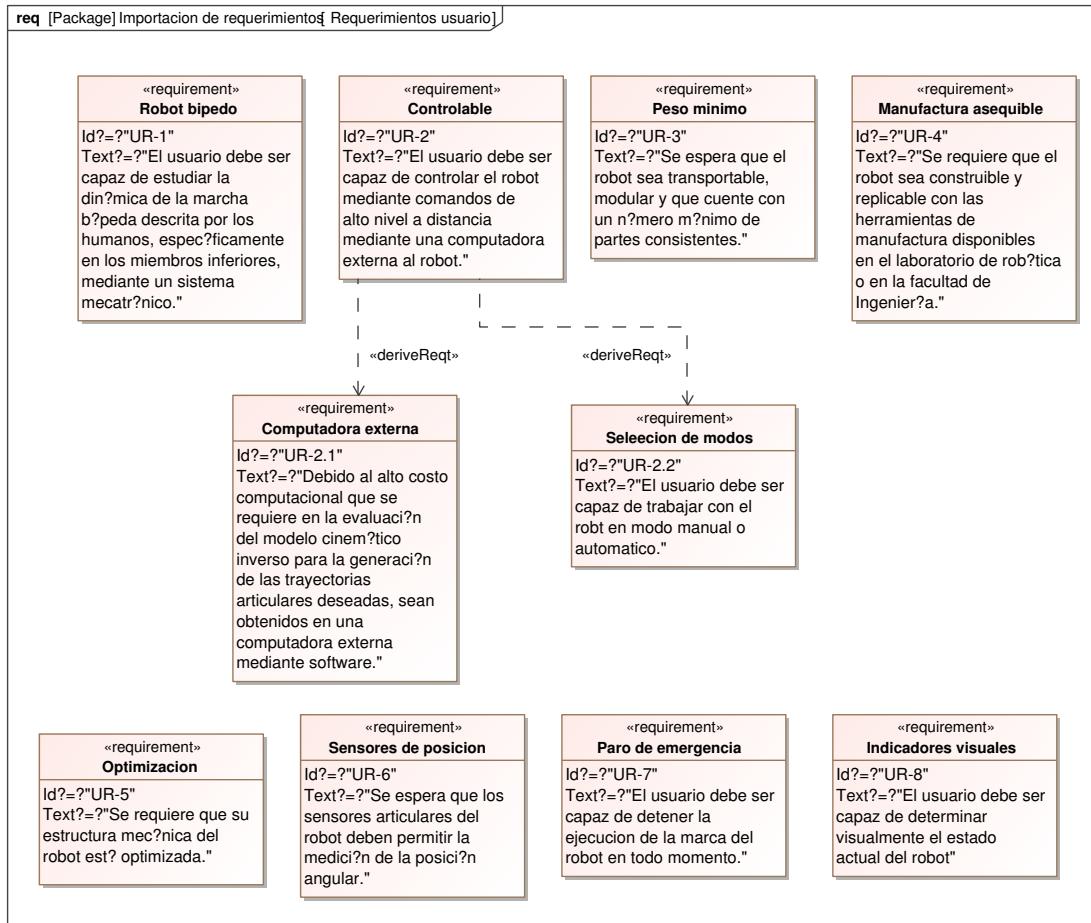


Figura 4.5: Representaci?n de los requerimientos del usuario en un diagrama de requerimientos.

#### 4.1.2. An?lisis operacional

Una vez establecidos los requerimientos del usuario, se procedi? a analizar el contexto operacional del sistema, que es una an?lisis de caja negra del sistema para determinar la misi?n principal y de alto nivel del sistema, que son los casos de uso principales, as? como el contexto del mismo derivando en la obtenci?n de los actores que sirven como interfaces.

#### 4.1.2.1. Casos de uso

El primer caso de uso construido en el modelo tiene que ver con la descripción de la misión global del sistema, en este caso, es evidente que la misión principal es que un usuario (investigador) maneje el robot, es decir interactúe con él, efectúe análisis e investigación. Por tanto se creó el diagrama de casos de uso simple mostrado en la figura, donde el caso de uso recibe el nombre de: ManejarRobot, lo cual indica que el actor: Investigador, usará al robot para tal propósito como misión principal. Por otra parte también es necesario describir el uso nominal del sistema, para ello también se creó un diagrama de casos de uso denominado: Uso nominal el cual se puede apreciar en la figura 4.6.

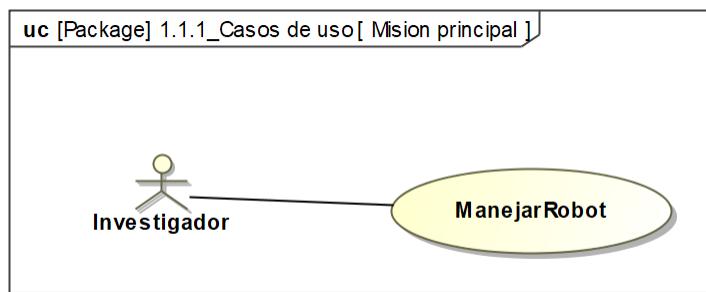


Figura 4.6: Diagrama de casos de uso que muestra la misión principal del sistema.

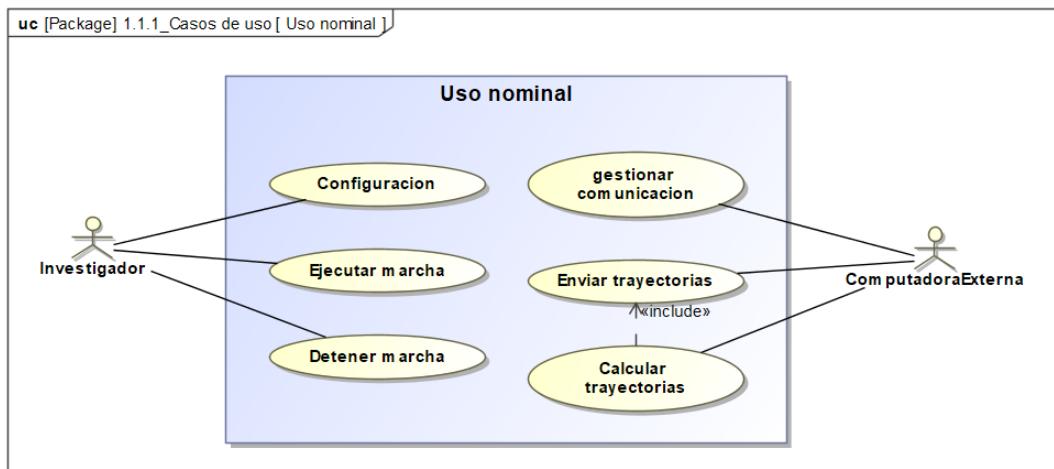


Figura 4.7: Diagrama de casos de uso que describe el uso nominal del sistema, en interacción con los actores.

En el diagrama de uso nominal es posible observar los diferentes casos de uso para el sistema, y un nuevo actor denominado Computadora Externa, el cual fue visualizado realizando una inspección a los requerimientos del usuario importados. Como se aprecia

en la figura 4.7, el usuario investigador es el encargado de gestionar las acciones de alto nivel del robot, ya que es el encargado de configurar, ejecutar y detener la marcha del robot, y por otra parte la ComputadoraExterna se encarga de gestionar la comunicación entre el Investigador y el RobotBipedo, calcula las trayectorias deseadas y las envía al robot con alguna interfaz que aun no esta definida en esta etapa del modelo.

#### 4.1.2.2. Contexto

El contexto puede ser considerado como un sistema del cual el sistema pertenece, es decir si el sistema contexto se denomina MarchaBipeda, es evidente que el sistema RobotBipedo forma parte de el, y a su vez los actores externos al sistema que también forman parte de ese contexto, como son: ComputadoraExterna, FuentePoder y el propio Investigador. Estos actores fueron descubiertos a partir del análisis de los requerimientos del usuario. Para especificar el contexto operacional, se creo un diagrama de definición de bloques (bdd), haciendo uso de bloques especiales para este tipo de análisis, los cuales son el block «system context» y el block «system», utilizados comúnmente en la MBSE, el diagrama se aprecia en la figura 4.8.

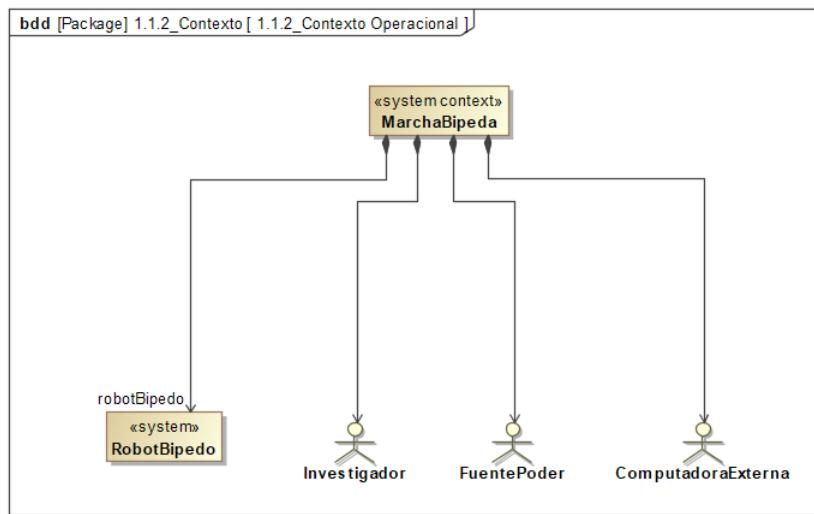


Figura 4.8: Contexto operacional del sistema RobotBipedo.

Otra forma de visualizar el contexto del robot se aprecia en la figura 4.9, donde se ha construido un diagrama de definición interna de bloques (ibd), dentro del bloque «system» es decir del RobotBipedo, de forma que el contexto operacional es implícito y no es necesario describir al robot como parte de otro sistema, sino que ahora este recibe directamente las relaciones de los actores externos.

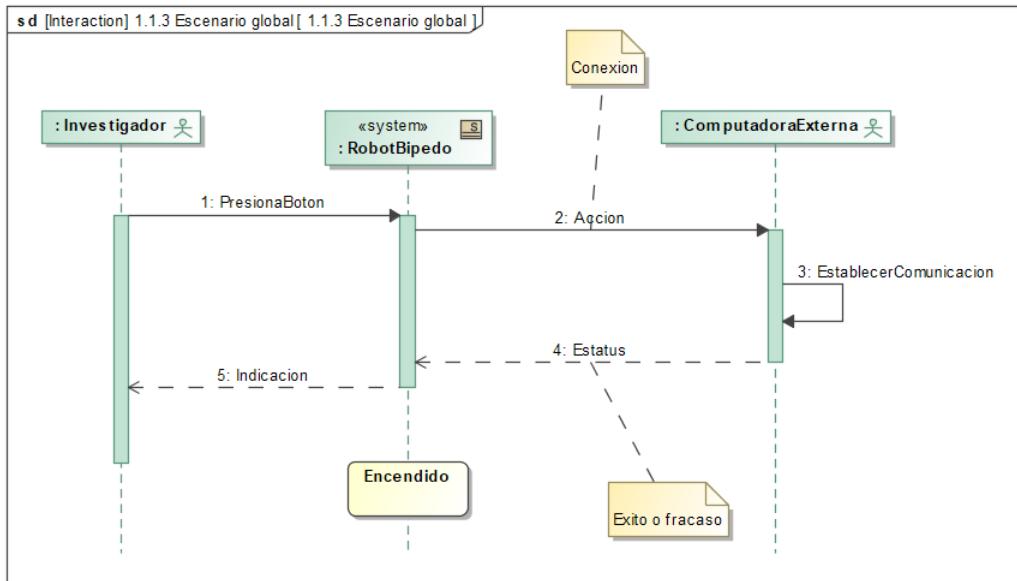


Figura 4.10: Escenario global para el sistema RobotBipedo y su contexto.

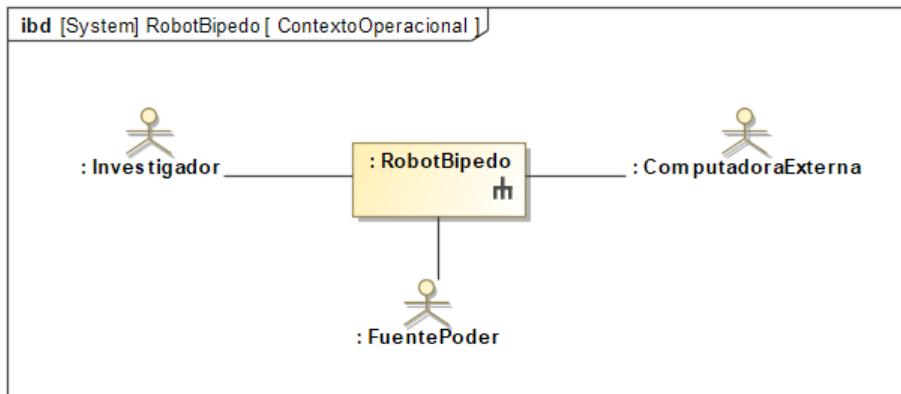


Figura 4.9: Contexto operacional desde el punto de vista del sistema RobotBipedo.

#### 4.1.2.3. Escenario global

Una vez identificados los actores, se puede determinar el escenario global de forma simple, con el objetivo de describir las acciones entre los diferentes actores, en SysML este tipo de interacciones se puede construir en un diagrama de secuencia (sd). El diagrama construido para el sistema se observa en la figura 4.10.

#### 4.1.2.4. Modos del sistema

Se deben identificar de forma simple los diferentes modos del sistema, pero no se tratan de características de comportamiento específicos como por ejemplo: modo manual o

automático, sino más bien los estados en lo que puede encontrarse el RobotBipedo, esto es: Conectado/Desconectado, puesto que estos dos estados son previsibles y gobiernan todo el sistema. El diagrama creado es del tipo maquina de estados (stm) y se aprecia en la figura 4.11.

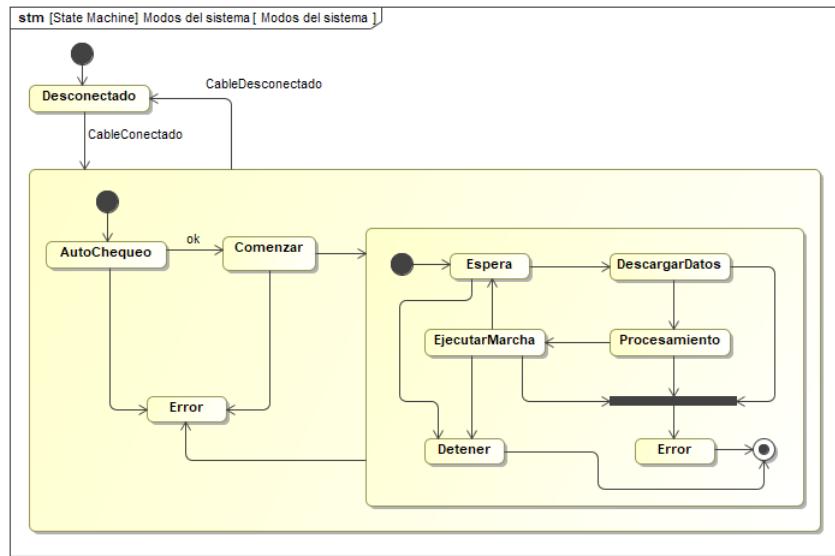


Figura 4.11: Maquina de estados, que modelan los estados básicos del sistema.

Con este análisis concluye el análisis operacional del sistema, cuyo objetivo principal es conocer los actores involucrados que interactúan con el sistema, así también establecer los requerimientos del usuario y su derivación.

## 4.2. FASE II - Diseño del sistema

### 4.2.1. Requerimientos del sistema

Realizando un análisis de los requerimientos del usuario se obtuvieron los siguientes requerimientos del sistema. A continuación se muestran en la tabla 4.2 y la derivación de los requerimientos se aprecia en el diagrama de requerimientos de la figura 4.12.

SR-1.1	Miembros inferiores	El robot debe contar con dos piernas, despreciando el tronco y los miembros superiores.
SR-1	Antropomórfico	Robot bípedo debe ser antropomórficamente consistente.
SR-2.1	IMU	Se requiere la medición de la inclinación y orientación del robot en el centro de masa.
SR-2.2	Actuadores robustos	El robot debe contar con actuadores de respuesta rápida y con suficiente torque.
SR-2	Marcha dinámica	El robot debe permitir desplazarse describiendo una marcha dinámica.
SR-3.1	Control de procesos	La computadora embebida debe gestionar el procesamiento de las señales provenientes desde los sensores y las señales hacia el sistema de potencia que alimenta a los actuadores.
SR-3.2	Puertos entrada salida	La computadora embebida requiere contar con un número suficiente de puertos de entrada y salida.
SR-3	Computadora embebida	El robot debe permitir recibir los comandos desde la computadora extena mediante una computadora embebida.
SR-4.1	Reprogramable	El robot debe permitir ser reprogramable.
SR-4	Estrategias de control	El robot debe permitir la prueba de diferentes estrategias de control.
SR-5	Altura	El robot no debe superar los 70 cm de altura.
SR-6.1	Fuente de poder externa	Para reducir el peso del robot se necesita que la etapa de potencia y la computadora embebida sean alimentados por una fuente de poder externa mediante un cable de potencia, evitando así el uso de baterías eléctricas.
SR-6	Peso	El robot no debe superar los 3 kg de peso.
SR-8	Rigidez	El robot debe permitir la máxima rigidez con la menor masa.
SR-9	Indicadores visuales	El robot debe tener indicadores acerca de su estado interno, permitiendo al usuario conocer si se ha producido un error.

Tabla 4.2: Requerimientos del sistema derivados de los requerimientos del usuario.

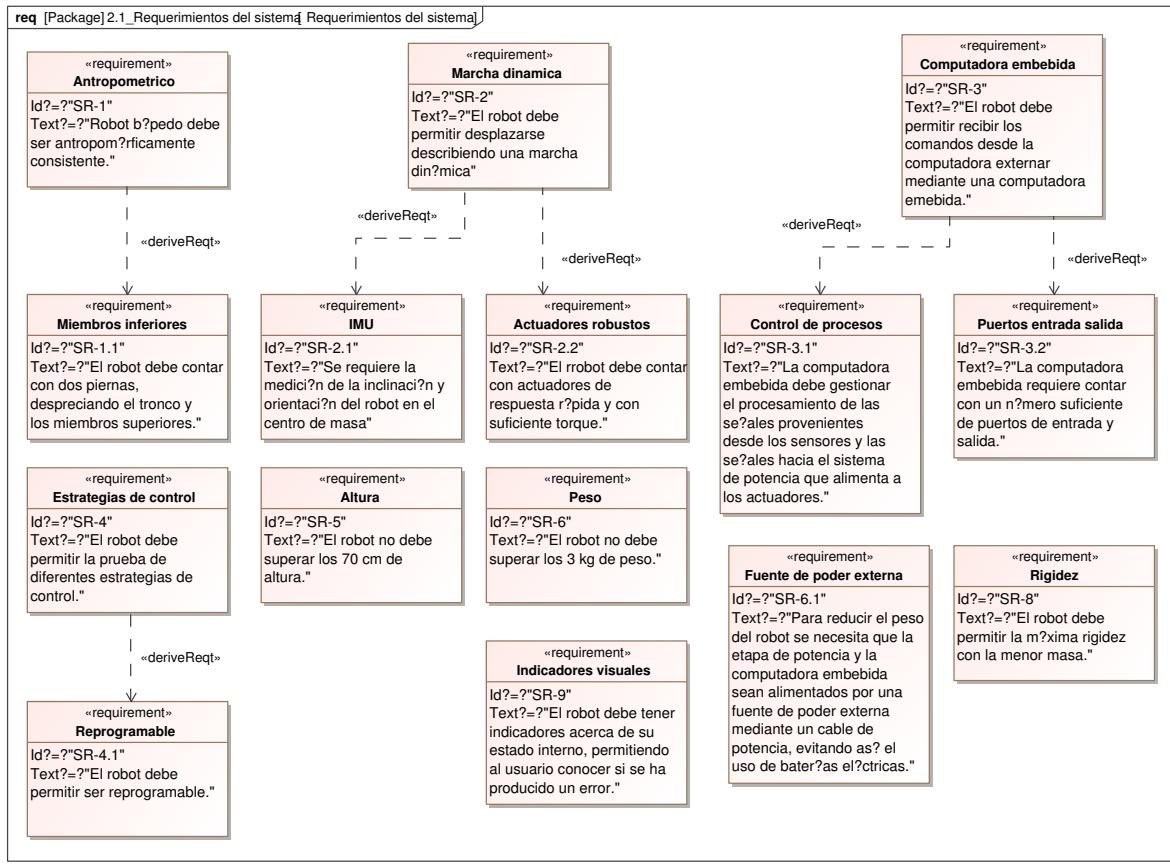


Figura 4.12: Diagrama de derivación de los requerimientos del sistema.

#### 4.2.2. Interfaces externas

La definición de los requerimientos del sistema y al definición de los actores permite definir las interfaces externas del RobotBipedo, considerando al sistema como una caja negra se realizo el diagrama de definición de bloques internos, el cual se muestra en la figura 4.13.

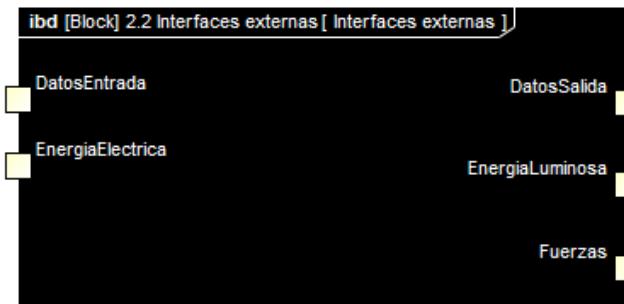


Figura 4.13: Diagramas de interfaces externas del sistema RobotBipedo.

De la figura se observan las diferentes interfaces externas del RobotBipedo, donde las interfaces DatosEntrada y DatosSalida fungen como puerto con el actor ComputadoraExterna, así mismo la interfaz EnergiaElectrica, se creó como puerto con el actor FuentePoder, que es un sistema externo al RobotBipedo.

#### 4.2.3. Escenarios del sistema

Conociendo las interfaces externas del sistema, se procede a diseñar con un mayor detalle los escenarios posibles del sistema, en este caso se presenta el escenario denominado EjecutarMarcha, donde se observan las posibles interacciones (mensajes) de los diferentes actores con el sistema RobotBipedo en el tiempo, este escenario es descrito por medio de una diagrama de secuencias (sd), el cual se observa en la figura 4.14.

De la figura 4.14 se observa que la primera interacción ocurre entre los actores Investigador y ComputadoraExterna, donde Investigador envía un mensaje a la ComputadoraExterna solicitando la configuración de los parámetros de la marcha, a lo que la ComputadoraExterna responde el mensaje enviando un formulario donde el actor Investigador indica cuales son los parámetros que debe considerar el RobotBipedo durante la ejecución de la marcha. Estos parámetros definidos por el actor Investigador son enviados de regreso a la ComputadoraExterna, quien realiza internamente la acciones de CalcularTrayectorias, y una vez finalizada esta acción, envía los datos generados al RobotBipedo, el cual a su vez realiza la acción de procesar dichos datos para que sean usados para el control durante la ejecución de la marcha del mismo.

En este punto pueden ocurrir dos situaciones, por ello se hace uso del marco «alt», que indica que puede ocurrir una de las dos secciones que contiene, es decir si todo el procesamiento de las trayectorias es «Correcto» entonces el RobotBipedo realizará la acción de emitir la SeñalEspera, es decir, el RobotBipedo se encuentra listo para ejecutar la marcha, pero espera la orden por parte del Investigador. Cuando el Investigador envía el comando a la ComputadoraExterna que indica que desea que comience la marcha, la ComputadoraExterna recibe el mensaje y posterior a ello manda un mensaje al RobotBipedo que le indica que ejecute la marcha con los parámetros solicitados por el investigador.

Posterior a ello, el RobotBipedo envía un mensaje al Investigador denominado SeñalCorriendo, y este la interpreta como una indicación de que la ejecución de la marcha está ocurriendo, mientras el sistema RobotBipedo entra en ejecución y se mantiene en ese estado en un ciclo, por ello en el diagrama se observa el uso del marco «loop», que repite la acción de EjecutarMarcha. Al finalizar el RobotBipedo envía al Investigador el mensaje llamado SeñalEspera, lo cual le indica que el sistema ha terminado la ejecu-

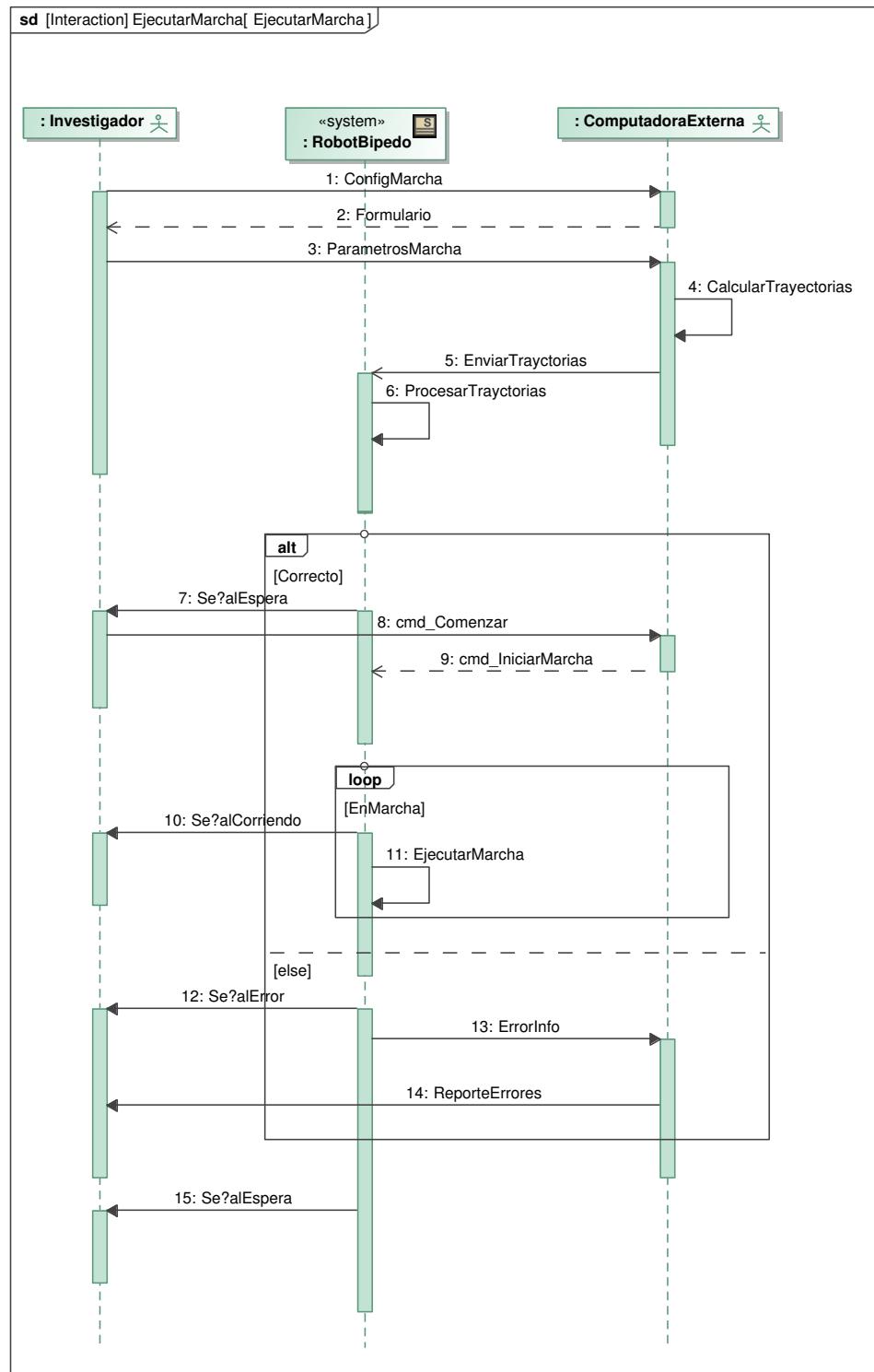


Figura 4.14: Diagrama de secuencia que describe las interacciones de los diferentes actores con el sistema RobotBipedo, en el escenario EjecutarMarcha.

ción de la marcha y espera nuevos mensajes, sin embargo, a partir de ese momento el escenario cambia y por tanto deja de ser relevante en el escenario EjecutarMarcha.

Retomando el marco «alt», el otro caso posible es cuando ocurre un error en el procesamiento de los datos recibidos por el RobotBipedo desde la ComputadoraExterna, en este caso el RobotBipedo se limita a enviar un mensaje al Investigador indicando que ocurrió un error, y posterior a ello envía un mensaje a la ComputadoraExterna denominado ErrorInfo que contiene los datos del error. La ComputadoraExterna a su vez notifica al Investigador la información proporcionada por el RobotBipedo mediante el mensaje ReporteErrores, para finalmente indicar al Investigador mediante el mensaje SeñalEspera que el sistema está listo para recibir nuevos mensajes, que quedan fuera del escenario EjecutarMarcha.

#### 4.2.4. Operaciones

A partir de las acciones propuestas en el diagrama de secuencia del escenario EjecutarMarcha, se obtienen las acciones candidatas (operaciones), que el sistema RobotBipedo debe ser capaz de realizar. Para indicar las operaciones se construye un diagrama de definición de bloques (bdd), el cual se aprecia en la figura 4.15.

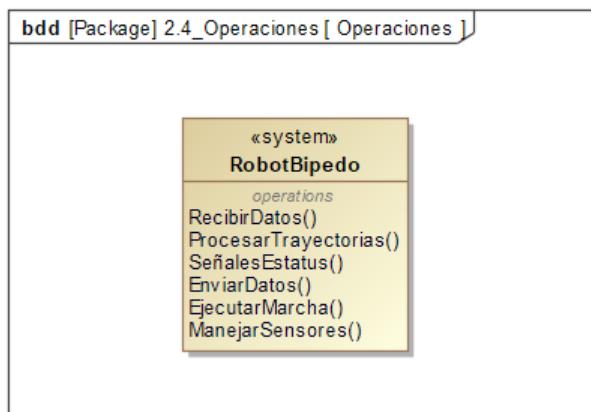


Figura 4.15: Operaciones que el robot bípedo debe realizar.

Antes de continuar con la siguiente fase de la metodología, se debe cuidar la traceabilidad tanto de los requerimientos del sistema respecto de los requerimientos del usuario, como de la traceabilidad de las operaciones definidas del robot bípedo respecto de los requerimientos del sistema. Para cuidar la traceabilidad entre los requerimientos, se hizo uso de la matriz de derivación de requerimientos, el resultado se observa en la figura 4.16.

<b>Legend</b>		Importación de requerimientos									
		UR-1 Robot bipedo	UR-2 Controlable	UR-2.1 Computadora externa	UR-2.2 Selección de modos	UR-3 Peso mínimo	UR-4 Manufactura asequible	UR-5 Optimización	UR-6 Sensores de posición	UR-7 Paro de emergencia	
<b>2.1_Requerimientos del sistema</b>		7	9	11	9	5	2	4	5	4	5
SR-1 Antropometrico		2									
SR-1.1 Miembros inferiores		1									
SR-2 Marcha dinamica		7									
SR-2.1 IMU		4									
SR-2.2 Actuadores robustos		6									
SR-3 Computadora embebida		11									
SR-3.1 Control de procesos		8									
SR-3.2 Puertos entrada salida		4									
SR-4 Estrategias de control		5									
SR-4.1 Reprogramable		3									
SR-5 Altura		3									
SR-6 Peso		2									
SR-6.1 Fuente de poder externa		2									
SR-8 Rigidez		2									
SR-9 Indicadores visuales		1									

Figura 4.16: Matriz de derivación de requerimientos, donde se especifica la relación de derivación entre los requerimientos.

La traceabilidad de las operaciones del sistema RobotBipedo, se indica mediante una matriz de satisfacción de requerimientos, la cual se aprecia en la figura 4.17.

Figura 4.17: Matriz de satisfacción de requerimientos, que justifica las operaciones del RobotBipedo respecto de los requerimientos del sistema.

#### 4.3. Fase III - Diseño de la arquitectura

#### 4.3.1. Arquitectura lógica

A cada operaciones definida en el «block» «system» fue asignada una actividad, para construir con estas acciones la actividad principal del sistema en una diagrama de actividad (act), el cual se muestra en la figura 4.18.

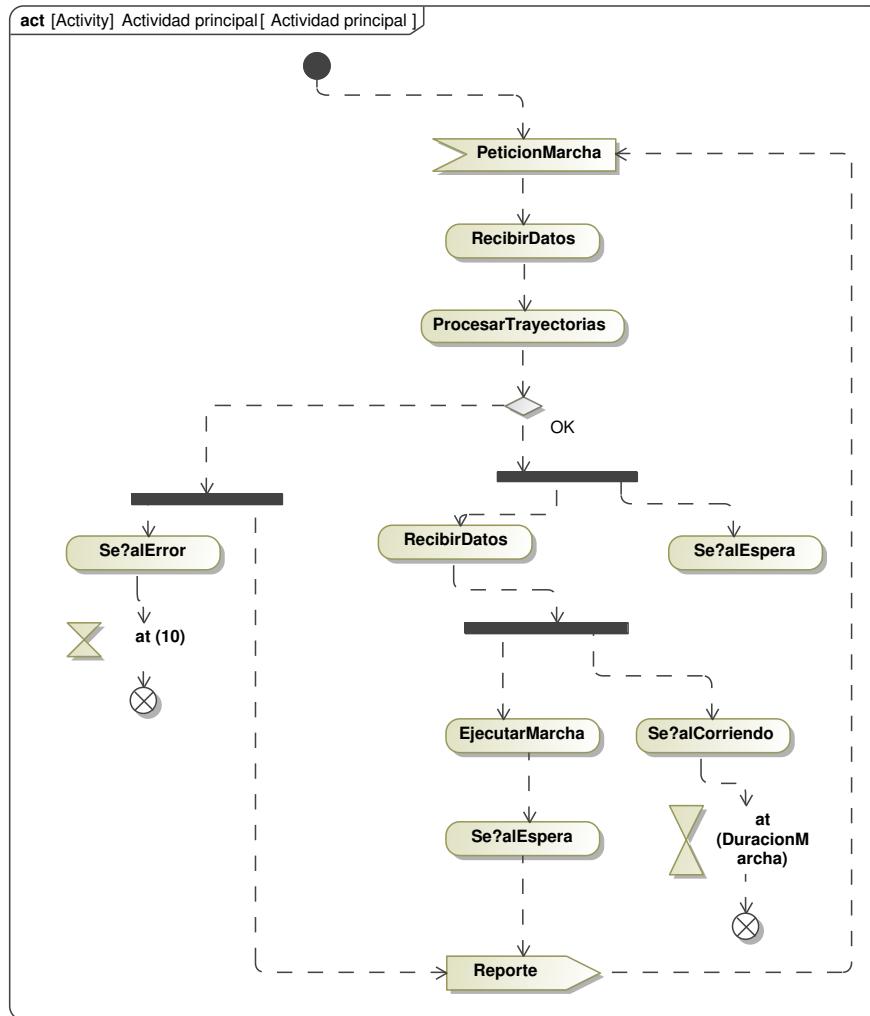


Figura 4.18: Actividad principal del sistema RobotBipedo.

Antes de continuar con la descomposición interna lógica del robot bípedo, se debe cuidar la traceabilidad de las actividades propuestas en el diagrama de actividades de la figura, para tal propósito se creó una matriz de satisfacción de requerimientos, la cual se observa en la figura 4.19.

Legend		2.1 Requerimientos del sistema														
	Satisfy	SR-1 Antropometrico	SR-1.1 Miembros inferiores	SR-2 Marcha dinamica	SR-2.1 IMU	SR-2.2 Actuadores robustos	SR-3 Computadora embobida	SR-3.1 Control de procesos	SR-3.2 Puertos entrada salida	SR-4 Estrategias de control	SR-4.1 Reprogramable	SR-5 Altura	SR-6 Peso	SR-6.1 Fuente de poder externa	SR-8 Rigidz	SR-9 Indicadores visuales
Actividad principal		3	1	1	3	3	1	1	1	1	1	1	1	3		
EjecutarMarcha	2															
ProcesarTrayectorias	2															
RecibirDatos	6															
SeñalCorriendo	1															
SeñalError	1															
SeñalEspera	4															

Figura 4.19: Matriz de satisfacción de requerimientos para verificar las actividades propuestas.

En este punto del modelo se analiza al sistema por primera vez como una «caja blanca», por tanto se creo un diagrama de definición de bloques, para el sistema RobotBipedo, el cual se muestra en la figura 4.20.

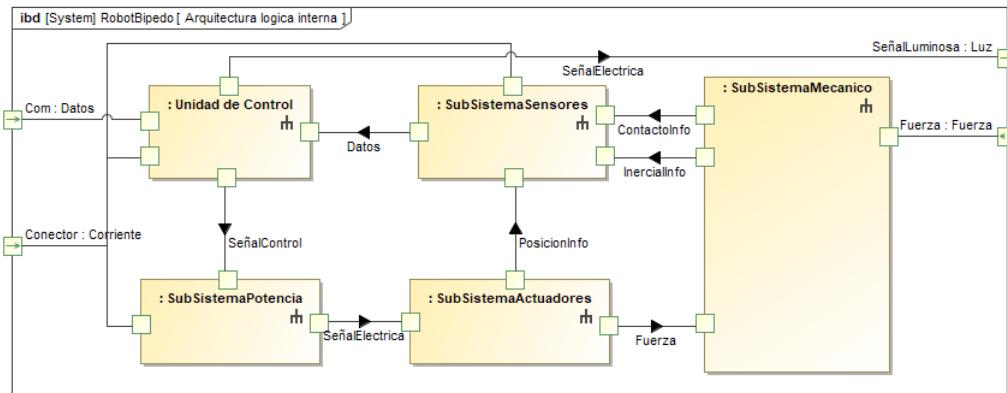


Figura 4.20: Estructura lógica interna del RobotBipedo.

Como se observa en la figura 4.20, se ilustran los diferentes elementos que conforman la arquitectura interna del robot bípedo, la cual esta conformada por unidad de control, subsistema mecánico, subsistema que engloba a los actuadores y de forma similar un subsistema de potencia. Cada una de las partes intercambian flujos de información y energía, en el caso del subsistema mecánico, entran al subsistema una fuerza, que

proviene de algún actuador, y del sistema sale otra fuerza que esta aplicada en su centro de masa, así también, sale del sistema información pertinente a la orientación, inclinación y contacto del mismo.

A continuación se describen la descomposición funcional de cada elemento interno del RobotBipedo, especificadas en diagramas de definicion interna. La composición interna de la «Unidad de Control», se aprecia en la figura 4.21, donde se observa que requiere de tres funciones principales: Se requiere de un sistema que sirva de interfaz entre los datos provenientes de la computadora externa, para poder transferir estos datos a la computadora embebida en el robot, que se encarga de realizar el control, y debido a que para tal propósito se requiere saber información provenientes de los diferentes sensores distribuidos en el sistema, la «Unidad de Control» debe contar con una interfaz analogica-digital, que permita la transferencia de los datos hacia la computadora embebida.

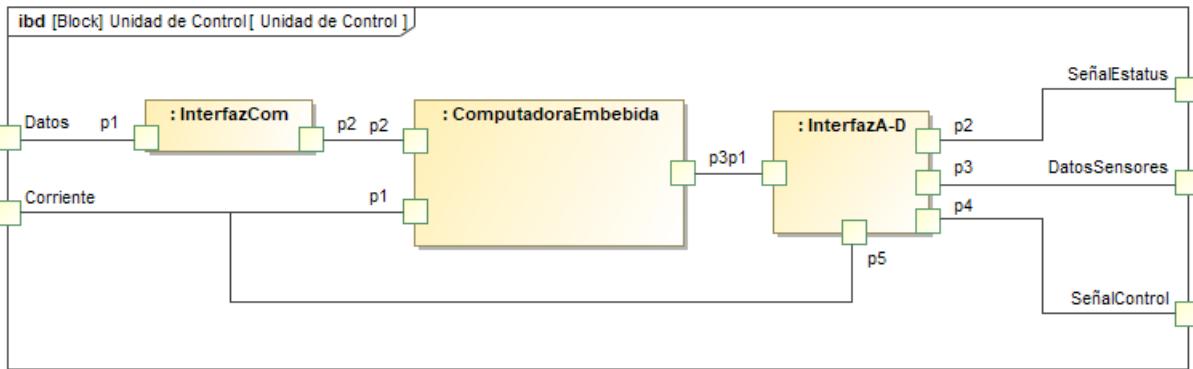


Figura 4.21: Composición interna de la «Unidad de Control»

La composición interna del subsistema «SubsistemaSensores» se aprecia en la figura 4.22, donde se aprecia que se busca que este subsistema sea capaz de obtener datos de tres diferentes tipos, la «UnidadMedicionInercial» debe proporcionar a la computadora embebida, la informacion inercial del centro de masa del robot, por otra parte el bloque «Estrangulamiento» obtiene para el sistema la informacion de posición angular proveniente del subsistema mecánico, de igual forma para la informacion de contacto (finales de carrera), se requiere del bloque «SensorContacto». El bloque «ManejadorBusDatos» hace referencia a la necesidad del sistema de contar con un bus de datos gestionado por algún protocolo de comunicación aun no especificado.

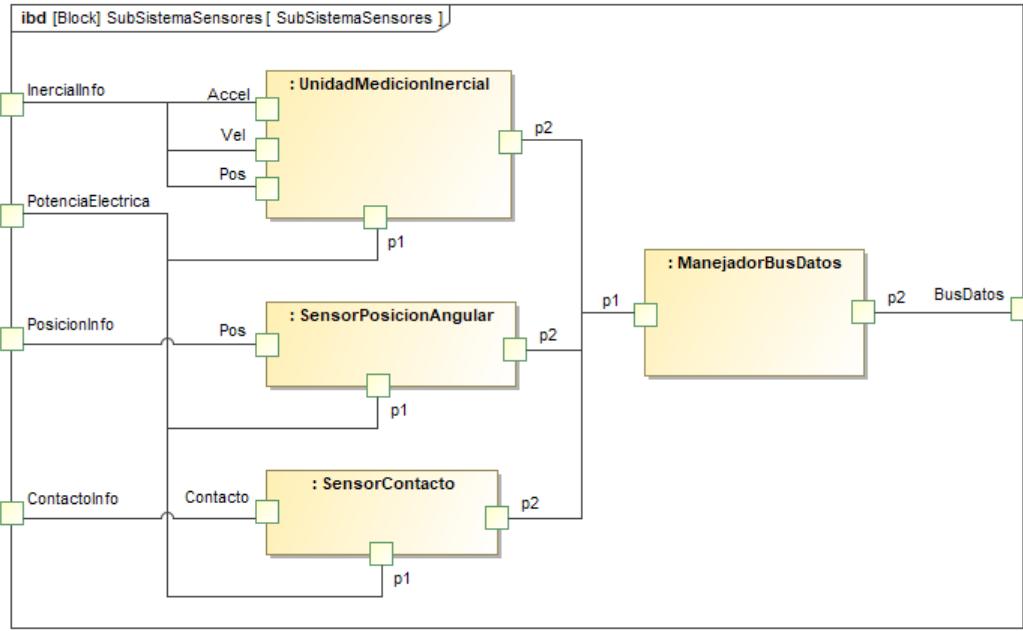


Figura 4.22: Composición interna del «SubSistemaSensores».

Para el «SubSistemaPotencia» de la arquitectura lógica interna del robot bípedo, se tiene los componentes mostrados en la figura 4.23.

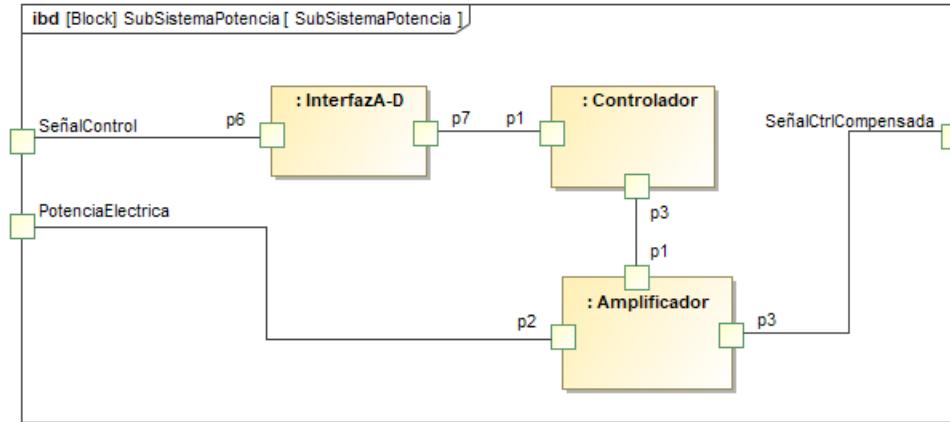


Figura 4.23: Composición interna del «SubSistemaPotencia».

Por otra parte, en la figura 4.24 se aprecia la composición interna del bloque «SubSistemaMecanico», el cual consta de una interfaz mecánica, es decir este bloque indica la función de un elemento que permita el acoplamiento entre dos elementos de naturaleza mecánica, como por ejemplo un eje. El bloque TransmisionPotencia hace referencia a que el sistema debe de ser capaz de cumplir con la transmisión de la potencia mecánica suministrada por la interfaz mecánica, este bloque indica que se debe elegir algún tipo

de transmisión de potencia adecuado. El bloque de tipo MasaInercial, especifica que dentro del subsistema mecánico existe un componente que cumple con la función de generar una masa inercial, y de la cual es posible extraer información relevante a su posición y orientación.

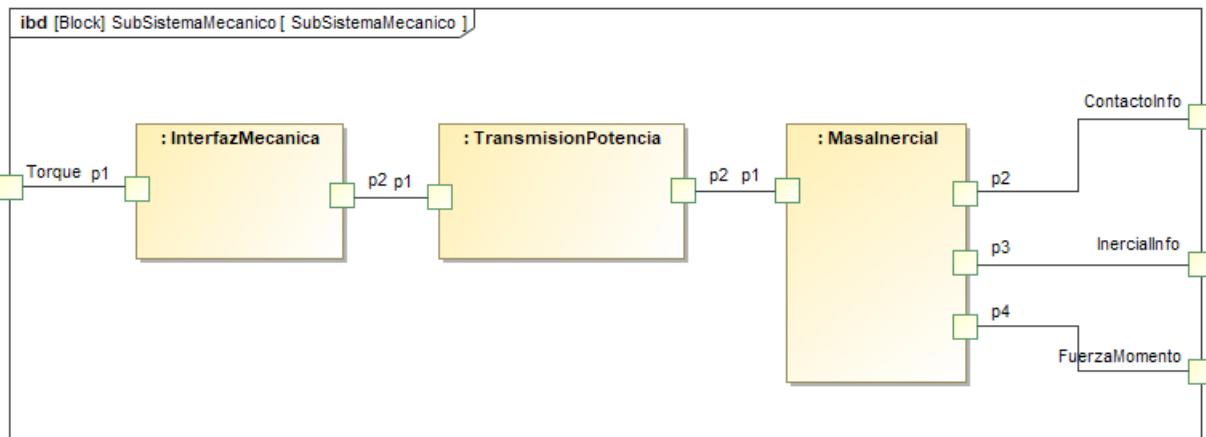


Figura 4.24: Composición interna del «SubSistemaMecanico».

Finalmente se describe en la figura 4.25 la estructura lógica interna del «SubSistemaActuadores».

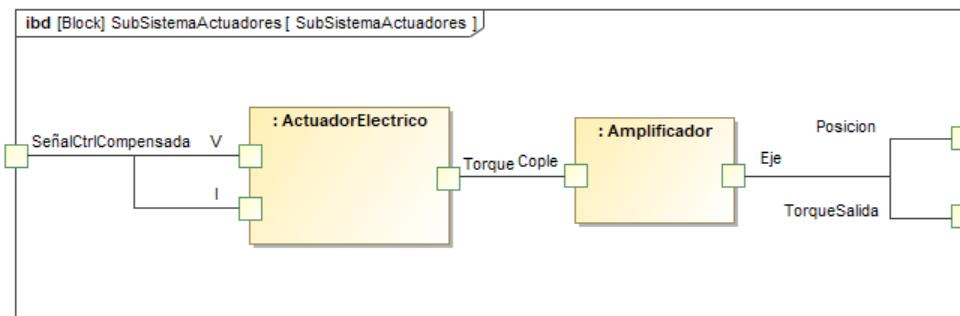


Figura 4.25: Composición interna del «SubSistemaActuadores».

### 4.3.2. Arquitectura física

La arquitectura física representa las diferentes partes que conformaran al robot bípedo en su implementación (ver figura 4.26), cada parte de la estructura esta basada en la descomposición lógica.

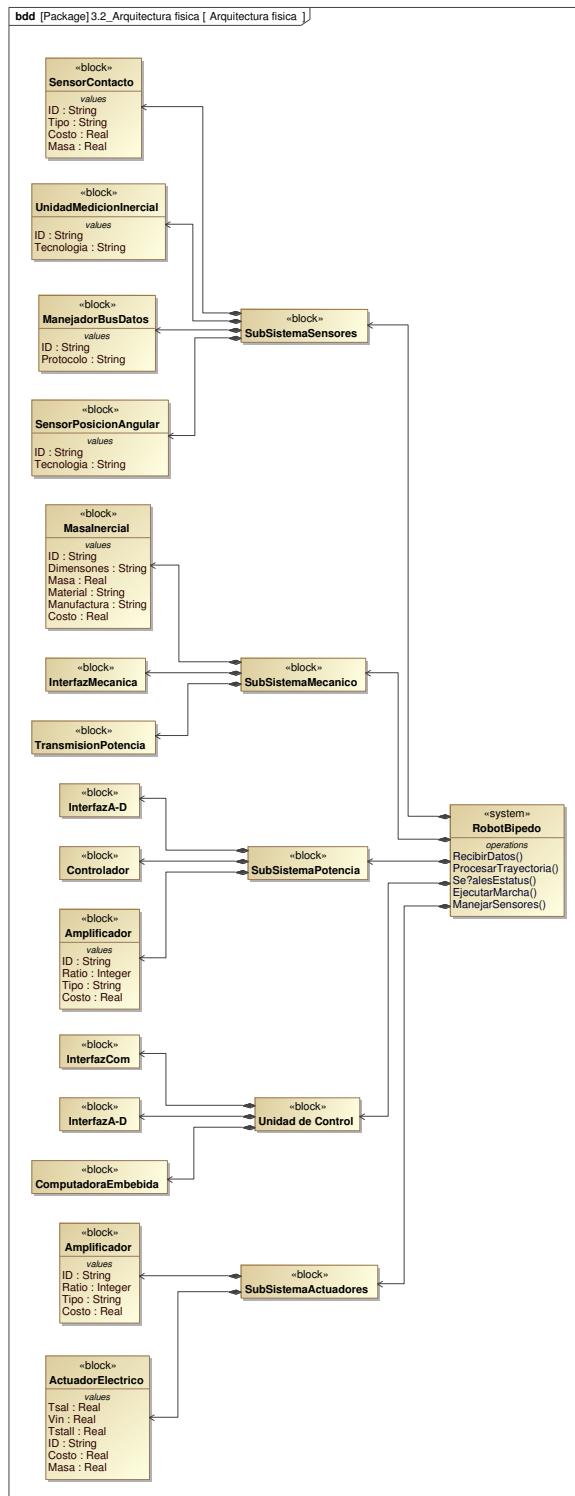


Figura 4.26: Arquitectura física del robot bipedo.

# Capítulo 5

## Análisis de ingeniería

### 5.1. Modelos matemáticos

Predecir el comportamiento cinemático y dinámico del robot bípedo, es de crucial importancia en el diseño mecatronico del mismo, el cual tiene que ser simulado y validado, teniendo como objetivo la detección de errores de dimensionamiento (o de diferente naturaleza), movimientos esperados para el caso del modelo cinemático; por otra parte con el modelo dinámico inverso se buscan conocer las fuerzas y momentos que se presentan en el robot durante el movimiento, siguiendo una determinada trayectoria, lo cual es necesario para la correcta selección de los actuadores del robot.

Para este trabajo se obtuvieron dos modelos matemáticos, el modelo cinemático directo y el modelo dinámico inverso, ambos de forma numérica, puesto que la complejidad de las ecuaciones simbólicas resultantes carecían de utilidad práctica. A continuación se describen los procedimientos para la obtención de ambos.

#### 5.1.1. Modelo cinemático directo

Para la obtención del modelo cinemático directo, y procurando que se llegase a un modelo consistente con la biomecánica humana, se adjuntaron a cada articulación del robot un sistema de coordenadas locales, cuya orientación y ubicación fueran congruente con (Murray, 1984), además también se tomaron en cuenta las convenciones en biomecánica para las mediciones de las articulaciones en los humanos.

Se tomó como eslabón fijo o base del robot bípedo al pie derecho del mismo, de manera que se forma una cadena cinemática abierta cuyo eslabón final es el pie izquierdo del robot bípedo. Con el modelo cinemático directo obtenido mediante la multiplicaron de las transformaciones homogéneas desde  ${}_1^0T$  hasta  ${}_{13}^{12}T$  utilizando el método propuesto

en, se consigue obtener la posición y orientación del pie izquierdo del robot respecto del derecho.

$${}^0_{13}T = {}^0_1T {}^1_2T {}^2_3T {}^3_4T {}^4_5T {}^5_6T {}^6_7T {}^7_8T {}^8_9T {}^9_{10}T {}^{10}_{11}T {}^{11}_{12}T {}^{12}_{13}T \quad (5.1)$$

Las matrices de transformación homogénea que describen la posición entre eslabones, desde la base del robot, hasta el pie izquierdo, se describen como:

$$\begin{aligned} {}^0_1T &= \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & -TOB \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^1_2T = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^2_3T &= \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & -L_1 \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^3_4T = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & -L_2 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \\ {}^4_5T &= \begin{bmatrix} \cos(\theta_5 - \frac{\pi}{2}) & -\sin(\theta_5 - \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_5 - \frac{\pi}{2}) & \cos(\theta_5 - \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^5_6T &= \begin{bmatrix} \cos(\theta_6 - \pi) & -\sin(\theta_6 - \pi) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_6 - \pi) & \cos(\theta_6 - \pi) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^6_7T &= \begin{bmatrix} \cos(\theta_7) & -\sin(\theta_7) & 0 & L_3 \\ \sin(\theta_7) & \cos(\theta_7) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^7_8T = \begin{bmatrix} \cos(\theta_8 - \frac{\pi}{2}) & -\sin(\theta_8 - \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_8 - \frac{\pi}{2}) & \cos(\theta_8 - \frac{\pi}{2}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} {}^8T = \begin{bmatrix} \cos(\theta_9) & -\sin(\theta_9) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_9) & -\cos(\theta_9) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^9_{10}T = \begin{bmatrix} \cos(\theta_{10}) & -\sin(\theta_{10}) & 0 & L_4 \\ \sin(\theta_{10}) & \cos(\theta_{10}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^{10}_{11}T = \begin{bmatrix} \cos(\theta_{11}) & -\sin(\theta_{11}) & 0 & L_5 \\ \sin(\theta_{11}) & \cos(\theta_{11}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad {}^{11}_{12}T = \begin{bmatrix} \cos(\theta_{12}) & -\sin(\theta_{12}) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_{12}) & \cos(\theta_{12}) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ {}^{12}_{13}T = \begin{bmatrix} 1 & 0 & 0 & TOB \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Para efectos de representación visual de los resultados, se considero una rotación de 90° sobre el eje  $y$ , ya que el sistema coordenados global no coincide con el primer sistema de coordenadas correspondiente a la base inmóvil de robot. Por tanto se pre-multiplica ecuación por la por la matriz de rotación:

$$R_y = \begin{bmatrix} \cos\left(\frac{\pi}{2}\right) & 0 & \sin\left(\frac{\pi}{2}\right) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\left(\frac{\pi}{2}\right) & 0 & \cos\left(\frac{\pi}{2}\right) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Por otra parte los parámetros de traslación:  $TOB$ ,  $L_1$ ,  $L_2$ ,  $L_3$ ,  $L_4$  y  $L_5$ , corresponden a las dimensiones de cada eslabón del robot.

Si bien, al efectuar las multiplicaciones de la ecuación 26, se obtendría de forma simbólica la matriz de transformación homogénea, que en sus elementos define la posición y orientación del efecto final respecto de la base del robot, donde la parte del modelo cinemático directo corresponde al componente de traslación de la matriz, en la practica estas expresiones simbólicas son demasiado extensas y complejas, por lo que su definición simbólica como tal no es utilizable, por tal motivo se optó por calcular de forma numérica la posición del efecto final, cuando se conocen las variables articulares y todos los parámetros dimensiones del robot.

El modelo cinemático directo fue obtenido con el objetivo de poder visualizar las posiciones del robot ante determinados ángulos articulares de forma visual, para tal fin se programó en Matlab (Apéndice 1), una función que recibe como parámetros las

variables articulares y genera una forma alámbrica del robot ante determinados valores de entrada. Ejemplo de ello se observa en la figura 5.1.

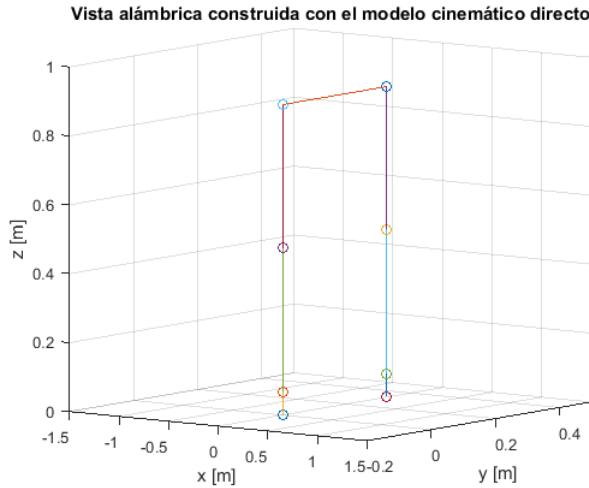


Figura 5.1: Vista alámbrica de la estructura del robot cuando todas las variables articulares son cero.

### 5.1.2. Modelo dinámico inverso

Se obtuvo el modelo dinámico inverso mediante el algoritmo iterativo de Newton-Euler de forma numérica, puesto que la obtención de las expresiones simbólicas para cada torque articular, superaron las capacidades computacionales disponibles, ya que resultaron ecuaciones excesivamente grandes y complejas que no son utilizables ni manejables. Por tal motivo se optó por programar una función en Matlab (Apendice 1), con las ecuaciones recursivas de Newton-Euler, cuya entrada son las trayectorias articulares en posición, velocidad y aceleración, así como todos los parámetros dimensionales e inerciales del robot.

A continuación se muestran los resultados obtenidos con el modelo dinámico inverso numérico del robot bípedo, considerando los parámetros que se muestran en las tablas 5.1 y 5.2.

Eslabón	1	2	3	4	5	6	7	8	9	10	11	12
Masa(kg)	0.22	0.25	0.32	0.15	0.22	0.40	0.22	0.15	0.32	0.25	0.22	0.10

Tabla 5.1: Parámetro de masas de eslabones utilizados en la simulación del modelo dinámico inverso

Eslabón	TOB	L1	L2	L3	L4	L5
Longitud (m)	0.0257	0.1624	0.1617	0.1261	0.1617	0.1624

Tabla 5.2: Dimensiones de los eslabones para la simulación.

Para efectos de simplificación en el modelo, los tensores de inercia para todos los eslabones corresponden a una matriz nula de 3x3, ya que se considera que toda la masa de los eslabones se encuentra concentrada en el centro de masas. Además se indica que el pie izquierdo del robot no está interactuando con el entorno, de modo que:

$${}^{13}f_{13} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}^{13}n_{13} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

de forma similar, el pie derecho del robot se considera la base inmóvil del robot por lo que:

$${}^0\omega_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}^0\dot{\omega}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad {}^0\ddot{v}_0 = \begin{bmatrix} -9.81 \\ 0 \\ 0 \end{bmatrix}$$

Sin embargo la base inmóvil del robot si está afectada por la gravedad, que es una aceleración lineal, por tanto, el efecto de la gravedad en la simulación se implementó mediante el vector de aceleración lineal en la base del robot:  ${}^0\ddot{v}_0$ , la dirección de la gravedad se considera negativa y en dirección del eje  $x$  del sistema coordenado de la base inmóvil del robot, ya que de forma general en la aplicación práctica del algoritmo iterativo de Newton-Euler, el considerar que el robot acelera hacia arriba causa el mismo efecto que la gravedad (Craig, 2006)(Schillig, 1990), aquí el sistema coordenado de la base del robot esta orientado como se muestra en la figura 3.2, de forma que al considerarse que la gravedad aceleraría al robot hacia arriba y debido a que el eje  $x$  positivo apunta hacia abajo del robot en la base, la gravedad debe ser negativa para que surta el efecto mencionado.

Por otra parte las trayectorias articulares en posición  $\theta_i(t)$ , están modeladas mediante polinomios de tercer orden de la forma:

$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (5.2)$$

donde:

$$a_0 = \theta_0,$$

$$a_1 = 0,$$

$$a_2 = \frac{3}{t_f^2} (\theta_f - \theta_0),$$

$$a_3 = \frac{2}{t_f^3} (\theta_f - \theta_0)$$

Se obtuvieron las velocidades  $\dot{\theta}_i(t)$  y aceleraciones  $\ddot{\theta}_i(t)$  derivando el polinomios (), de forma que:

$$\dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 \quad (5.3)$$

$$\ddot{\theta}(t) = 2a_2 + 6a_3 t \quad (5.4)$$

Por lo tanto para construir las trayectorias articulares en posición, velocidad y aceleración:  $\theta_i(t)$ ,  $\dot{\theta}_i(t)$  y  $\ddot{\theta}_i(t)$  respectivamente, es necesario conocer las constantes  $a_0$ ,  $a_2$  y  $a_3$ , que están definidas en términos de el anulo inicial  $\theta_0$  en el tiempo  $t_0$  y el anulo final  $\theta_f$  en el tiempo  $t_f$ .

En la tabla 3.3 se indican los valores para  $\theta_0$ ,  $\theta_f$  y  $t_f$ , que generan los polinomios cúbicos para las trayectorias de los doce grados de libertad rotacionales del robot bípedo, utilizados en la simulación, dichos valores generan en el robot el movimiento de la fase de soporte simple del ciclo de marcha bieldada, donde el pie de apoyo (base inmóvil del robot) corresponde al pie derecho, mientras que el pie izquierdo deja de estar en contacto con el suelo para balancear la píerda izquierda hacia adelante para efectuar el paso de marca y entrar en contacto nuevamente con el piso, como se observa en la figura 5.2.

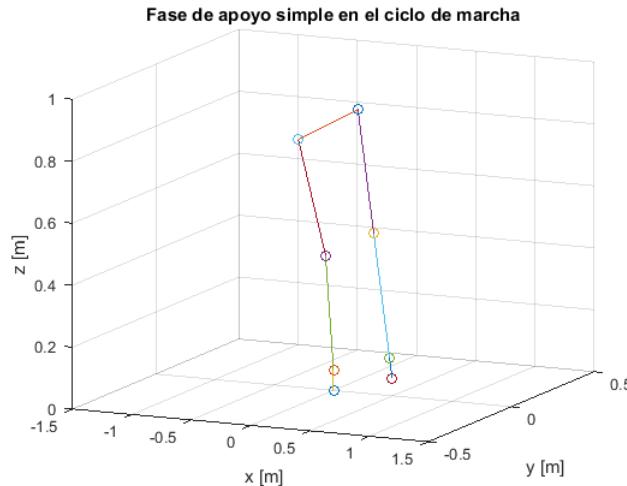


Figura 5.2: Fase de apoyo simple utilizada en la simulación del modelo dinámico inverso.

Trayectoria	$\theta_1(t)$	$\theta_2(t)$	$\theta_3(t)$	$\theta_4(t)$	$\theta_5(t)$	$\theta_6(t)$
$\theta_0[\text{rad}]$	$\frac{\pi}{12}$	$\frac{\pi}{16}$	$-\frac{2\pi}{16}$	$\frac{\pi}{16}$	$-\frac{\pi}{18}$	$\frac{\pi}{18}$
$\theta_f [\text{rad}]$	$\frac{\pi}{12}$	$\frac{\pi}{16}$	$-\frac{2\pi}{16}$	$\frac{\pi}{16}$	$-\frac{\pi}{18}$	$-\frac{\pi}{18}$
$t_0[\text{s}]$	0	0	0	0	0	0
$t_f [\text{s}]$	3	3	3	3	3	3

Trayectoria	$\theta_7(t)$	$\theta_8(t)$	$\theta_9(t)$	$\theta_{10}(t)$	$\theta_{11}(t)$	$\theta_{12}(t)$
$\theta_0[\text{rad}]$	$-\frac{\pi}{16}$	0	$\frac{\pi}{16}$	0	0	0
$\theta_f [\text{rad}]$	$\frac{\pi}{16}$	0	$-\frac{\pi}{16}$	0	0	0
$t_0[\text{s}]$	0	0	0	0	0	0
$t_f [\text{s}]$	3	3	3	3	3	3

Tabla 5.3: Parámetros utilizados para la generación de trayectorias articulares en la simulación del modelo dinámico inverso.

Al aplicar todas estas consideraciones en el algoritmo iterativo de Newton-Euler, se obtuvieron las gráficas mostradas en las figuras 5.3, 5.4 y 5.5.

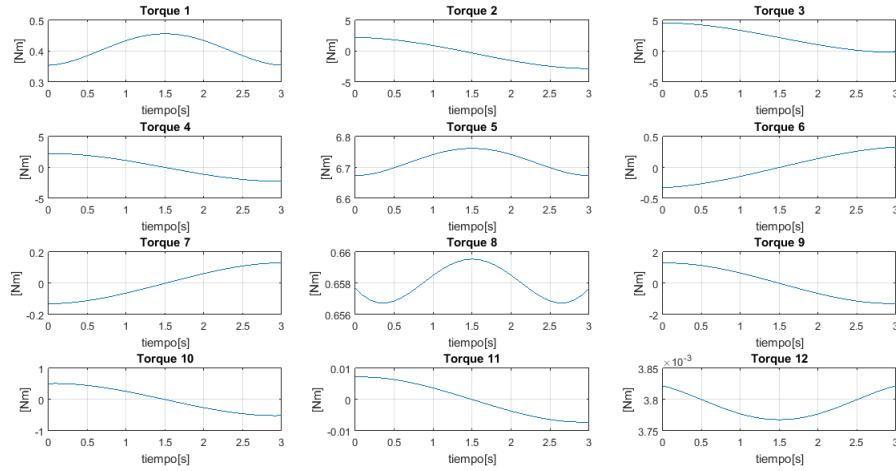


Figura 5.3: Torques articulares requeridos para la trayectoria deseada en cada eslabón del robot.

En la figura 3.4 se aprecia el torque aplicado por cada articulación del robot, en el tiempo en que duran las respectivas trayectorias, es decir son los torques requeridos para que el robot describa el movimiento deseado caracterizado por las trayectorias articulares presentadas en la tabla2. Los torques máximos ocurren en las articulaciones tres y cinco, que corresponden a los movimientos de abdusión y flexión de la cadera y rodilla respectivamente, esto es natural puesto que las trayectorias representadas en la tabla 3.3, describen el ciclo de marca correspondiente a la etapa de apoyo simple, cuando el soporte es el pie derecho, por tanto las fuerzas iniciales mayores ocurren en la articulación de la rodilla de la misma pierna, de igual forma debido a que el torso del robot efectúa un giro para adelantar la pierna izquierda y continuar el ciclo de marcha, este movimiento recae sobre la articulación que representa la abdusión de la cadera, misma que permite que se efectúe dicho movimiento.

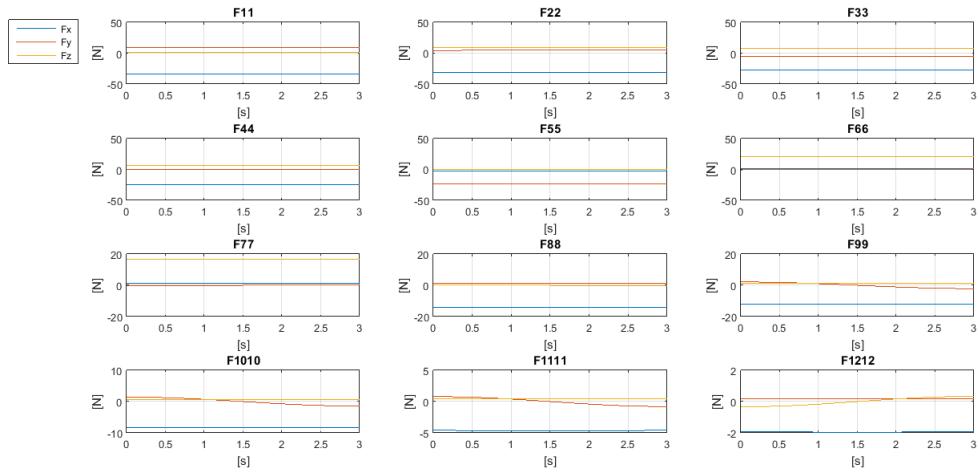


Figura 5.4: Componentes de las fuerzas netas aplicadas en el centro de masas de cada eslabón del robot.

En la figura 3.5 se observan las fuerzas aplicadas en el centro de masas de cada eslabón, en sus tres componentes descritos en el sistema de coordenadas local, solidario al eje de articulación. Las magnitudes de fuerza mayor se presentan en los eslabones de la pierna que actúa de soporte en esta fase de la marca, llegando a los  $30Nm$  en el eslabón 1 del robot.

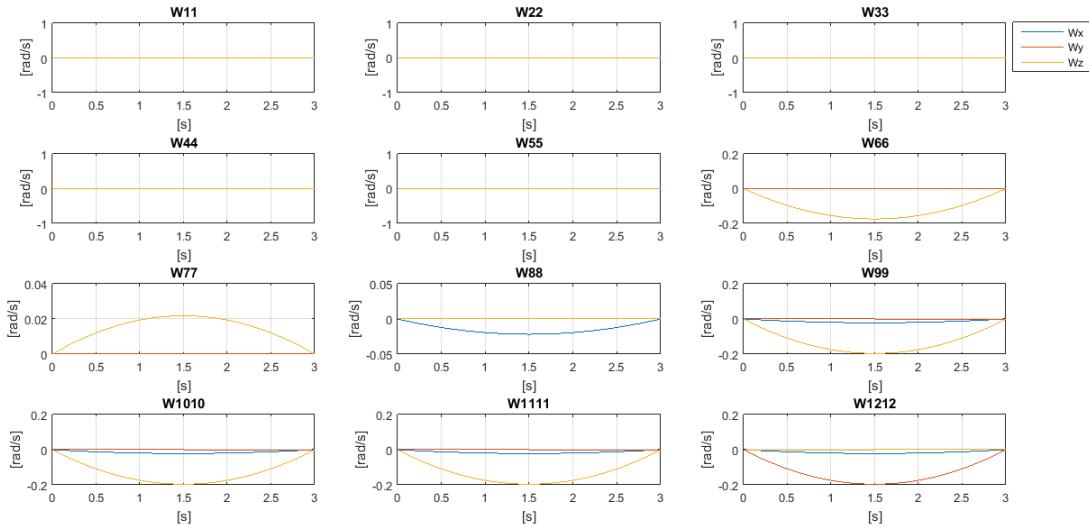


Figura 5.5: Componentes de las velocidades anulares que experimenta cada eslabón del robot.

Las trayectorias articulares provocan que cada eslabón rote sobre su propio eje, sin

embargo, esta rotación no es puramente sobre el eje, sino que presenta dos componentes  $y$   $yz$ , por acción del efecto de las rotaciones de los demás eslabones del robot, lo cual puede apreciarse en la figura 3.6, donde se observan los componentes de velocidad anular expresado en rad/s.

Una vez obtenido el modelo dinámico inverso, mediante el algoritmo iterativo de Newron-Euler de forma numérica, es posible continuar con la metodología de diseño la cual hace uso de este modelo, para realizar una optimización mediante algoritmos genéticos, con el objetivo de brindar restricciones de diseño en la fase de diseño de detalle. A continuación se presenta la optimización realizada usando AG con el uso del software Matlab.

### 5.1.3. Optimización por medio de algoritmos genéticos del sistema

El algoritmo de optimización por medio de A, trabaja como se especificó en el capítulo 2, con un vector de diseño de entrada, el cual contiene las variables que se desean optimizar, una función objetivo a la cual le pertenecen los parámetros que modifican el comportamiento dinámico del robot mediante el vector de diseño, que a su vez está sujeto a restricciones que en la metodología propuesta están basados en los requerimientos del sistema en la búsqueda de la convergencia. Otro parámetro importante es la población y número de generaciones de trabajo del A, los criterios de convergencia y paro del algoritmo, determinados de forma empírica al correr las simulaciones en este caso particular.

#### 5.1.3.1. Función objetivo

El objetivo de la optimización es reducir el torque articular durante la trayectoria de marca en su fase de apoyo simple, en la articulación donde el modelo indica que se encuentra el torque más elevado, siendo la función objetivo el modelo dinámico. Se puede apreciar que la reducción del torque articular depende de los parámetros de masa, velocidades y aceleraciones anulares de cada eslabón durante la marca, así como de las longitudes de los mismos, que están ligados a los requerimientos del sistema. A continuación se presenta el vector de diseño para el algoritmo genético.

#### 5.1.3.2. Vector de diseño

De los requerimientos del sistema se desprenden la selección de las variables del modelo dinámico, ya que por ejemplo se desea que el robot bípedo se desplace a cierta

velocidad, sea liviano, manufacturable, modular, pero a su vez se requiere que el costo del prototipo sea lo más bajo posible, al mismo tiempo que el robot sea congruente con la antroponomía humana y que funciones con el menor torque posible. Por tales motivos las variables de diseño son las seleccionadas son:

1. Altura del robot bípedo  $H$
2. Masa de los eslabones
3. Velocidad de paso

Estas tres variables son descritas en profundidad a continuación.

**Dimensiones sujetas a antroponomía humana** En la figura 5.6 se aprecia un diagrama que representa una parametrización de las longitudes de las diferentes partes del cuerpo humano basado en la altura  $H$  del cuerpo completo, por ejemplo la longitud del suelo a la articulación del tobillo está determinada por  $TOB = 0.039H$  y de forma análoga cada segmento del modelo. Por tanto se eligió este parámetro de altura  $H$  en la optimización de las longitudes de las extremidades inferiores. De forma que la primer variable del vector de optimización es: la altura del robot bípedo  $H$ .

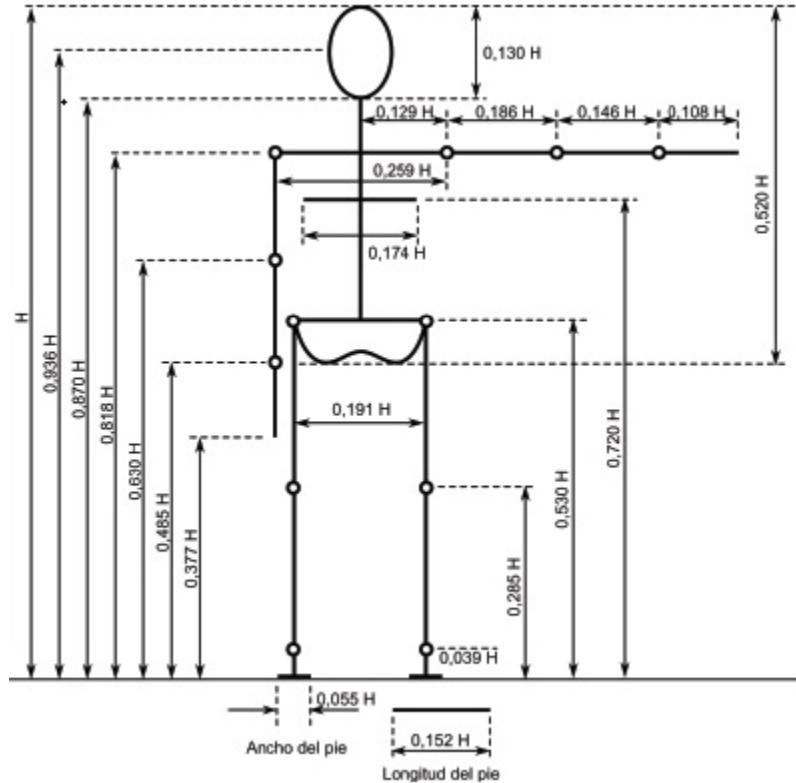


Figura 5.6: Modelo antropomórfico del cuerpo humano (Ruiz, 2014).

De la cual se desprenden las demás longitudes de los eslabones del miembro inferior como:

- Longitud de la pierna:  $L_2 = 0.285H - 0.152H = 0.16H$ .
- Longitud de la antepierna:  $L_3 = 0.530H - 0.285H - 0.152H = 0.37H$ .
- Longitud de la cadera:  $L_4 = 0.191H$ .

Los otros segmentos son equivalentes entre piernas por lo que se realiza una igualación de los parámetros entre las dos, de forma que  $L_2 = L_6$  y  $L_3 = L_5$ , y así se asegura que el robot permanezca simétrico respecto de su vertical.

**Masas de los eslabones** Debido a la simetría existen en el robot las masas de algunos segmentos de ambas piernas son iguales, por tanto basta con considerar las masas para una sola pierna, en el vector de diseño se consideraron las siguientes masas:  $m_1, m_2, m_3, m_4, m_5, m_6$  y  $m_{12}$  puesto que  $m_7 = m_5, m_8 = m_4, m_9 = m_3, m_{10} = m_2, m_{11} = m_1$ .

**Velocidad de paso** El modelo dinámico calcula los terques requeridos para que el robot describa una trayectoria determinada por polinomios cúbicos en cada articulación, las cuales representan la fase de apoyo simple sobre el pie derecho o base inmóvil del robot, un instante después de que el pie izquierdo deja de estar en contacto con el suelo a un instante antes de que se apoye de nuevo en el, el tiempo que tarde en realizar el balanceo de la pierna izquierda es un parámetro importante ya que de este tiempo depende la velocidad de paso, en conjunto con las dimensiones de cada eslabón del robot.

Las trayectorias construidas con polinomios cúbicos requieren un tiempo inicial  $t_0$  y un tiempo final  $t_f$  donde las velocidades son cero, se ha elegido este tiempo como:  $t_0$  es el instante en el que el pie izquierdo deja el suelo y  $t_f$  un instante antes de que entre en contacto de nuevo, con lo que el tiempo total de la trayectoria es:  $T = t_f - t_0$ , si se considera además que  $t_0 = 0, T = t_f$ , el cual es el parámetro que se considera en el vector de diseño. Por tanto variando este parámetro las velocidades articulares se incrementan o disminuyen, de forma que las fuerzas inercias lo hacen y el torque requerido para producir el movimiento, así que es necesario encontrar el valor óptimo de duración del ciclo de apoyo simple que en conjunto con las otras variables de optimización minimicen el torque articular requerido.

### 5.1.3.3. Restricciones

$$c(:,1) = \max(\text{Torques}) - \text{eps}; c(:,2) = \text{eps2-TOB};$$

Al AG sólo se le han indicado dos restricciones, ambas son no-lineales, una de ellas tiene que ver con una restricción dimensional, específicamente en la distancia entre la articulación del tobillo y el suelo, que es representado como la longitud nombrada TOB, esta distancia tiene que ser lo suficientemente grande como para permitir que un sistema de transmisión por bandas sea alcanzable, por lo que:

$$TOB > \varepsilon_1 \quad (5.5)$$

donde  $\varepsilon_1 = 0.03m$  es la distancia que  $TOB$  debe superar para cumplir con la desigualdad. La restricción no lineal puede reescribirse como:

$$c_1 = \varepsilon_1 - TOB \quad (5.6)$$

De forma similar, la segunda restricción tiene su origen el torque máximo para todas las articulaciones, por tanto se tiene:

$$\tau_m < \varepsilon_2 \quad (5.7)$$

donde  $\varepsilon_2 = 2Nm$  es el torque articular máximo permisible para todas las articulaciones, por tanto para efectos del algoritmo genético la segunda restricción es:

$$c_2 = \tau_m - \varepsilon_2 \quad (5.8)$$

Con estas restricciones se asegura que la masa de los eslabones no provoque momentos sobre los ejes de articulación superiores a  $\varepsilon_2$ , parámetro que depende del motor de CD utilizado.

### 5.1.3.4. Resultados

Se realizó en Matlab un script (Apendice 1), utilizando la función  $ga()$  del toolbox de algoritmos genéticos, considerando que el vector de diseño es de nueve variables:

$$V_d = [ H \ m_1 \ m_2 \ m_3 \ m_4 \ m_5 \ m_6 \ m_{12} \ t_f ] \quad (5.9)$$

donde:  $H$  es la altura del robot,  $m_1, m_2, m_3, m_4, m_5, m_6$  y  $m_{12}$  son las masas de los eslabones correspondientes su subíndice, y  $t_f$  es el tiempo de duración de la fase de simple apoyo. Los límites superior e inferior para cada una de las variables está deter-

minado por los vectores UB y LB, que determinan los límites superiores e inferiores de las variables.

$$UB = [ 1.7 \ 0.28 \ 0.38 \ 0.38 \ 0.20 \ 0.28 \ 0.40 \ 0.20 \ 3 ] \quad (5.10)$$

$$LB = [ 0.66 \ 0.22 \ 0.25 \ 0.32 \ 0.15 \ 0.22 \ 0.38 \ 0.10 \ 1 ] \quad (5.11)$$

El tiempo de duración de la trayectoria articular se requiere que sea una variable tipo entero, por lo cual se creó una restricción de variable entera como:  $IntCont = [9]$ , donde se le indica al algoritmo genético que la variable numero nueve del vector de diseño tiene dicha restricción. La función `ga()` se estableció de la forma:

```
ga(@obj_bipedo,9,[],[],[],LB,UB,@nonlcon_bipedo,IntCont,options)
```

donde `@nonlcon_bipedo` hace referencia a un script escrito en Matlab que contiene las restricciones no lineales descritas, y se configuran las opciones del algoritmo genético mediante el parámetro: `options`:

```
options=gaoptimset('Generations',200,'PopulationSize',400,'PlotFcns',{@gaplotbestf,  
                          @gaplotstopping},'TolFun',0.1)
```

En las opciones de configuración del AG, se establece que el número de generaciones es de 200 mediante '`Generations`', el número de individuos a 400 con '`PopulationSize`', y mediante '`PlotFcns`' se genera una ventana de progreso de las generaciones del algoritmo donde se muestra una gráfica con el valor de la función objetivo, y el porcentaje del criterio de paro. Con el parámetro '`TolFun`' con un valor de 0.1 se establece que el algoritmo se detenga cuando el valor del cambio promedio en la evaluación de la función objetivo sea menor o igual a 0.1.

El parámetro `@obj_bipedo`, hace referencia a una función denominada `obj_bipedo(x)` escrita en otro Script de Matlab que contiene el modelo dinámico del robot el cual es la función objetivo. La función tiene como único parámetro la variable `x`, que es donde recibe el vector de diseño en cada nueva generación producida por el algoritmo genético, la función también retorna el valor con el cual se evalúa el desempeño de la función objetivo, en este caso retorna el valor del torque articular crítico, que en la fase de simple apoyo es la articulación de adusión/abducción del tobillo derecho.

A continuación se muestran los resultados obtenidos mediante la optimización. En la figura se aprecia la ventana de progreso que genera el toolbox de AG de Matlab, donde se observa la convergencia de la solución conforme se iteran en las generaciones.

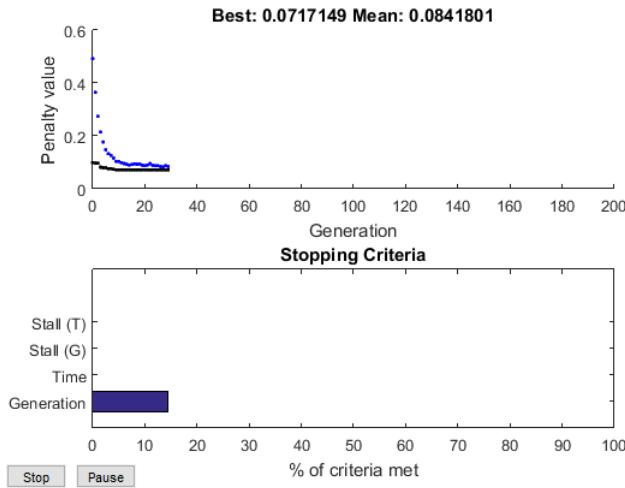


Figura 5.7: Convergencia de la solución del algoritmo genético.

El vector de salida  $x$  con los valores óptimos para cada variable generado por el algoritmo genético es:

$$x = [ \begin{array}{cccccccccc} 0.66 & 0.22 & 0.25 & 0.32 & 0.15 & 0.22 & 0.38 & 0.10 & 3 \end{array} ] \quad (5.12)$$

de donde se desprenden los valores para las masas de los eslabones:  $m_1 = 0.22$ ,  $m_2 = 0.25$ ,  $m_3 = 0.32$ ,  $m_4 = 0.15$ ,  $m_5 = 0.22$ ,  $m_6 = 0.40$ ,  $m_7 = 0.22$ ,  $m_8 = 0.15$ ,  $m_9 = 0.32$ ,  $m_{10} = 0.25$ ,  $m_{11} = 0.22$ ,  $m_{12} = 0.10$ . Estas masas optimizadas producen torques articulares menores que  $\varepsilon_1$ , en conjunto con las dimensiones de los eslabones:  $TOB = 0.0257$ ,  $L_1 = 0.1624$ ,  $L_2 = 0.1617$ ,  $L_3 = 0.1261$ ,  $L_4 = 0.1617$ ,  $L_5 = 0.1624$ .

El modelo dinámico utilizando los parámetros optimizados genera las gráficas de torques articulares mostrados en la figura 3.9, donde se observa que se cumple con la restricción no lineal  $c_2$ , la cual indica que ningún torque debe superar a  $\varepsilon_2 = 2Nm$ .

Se puede apreciar una disminución de torque articular requerido si se compara con los resultados de la figura 5.8, donde alcanzan valores cercanos a los  $5Nm$ , de forma similar también se produce una disminución de las fuerzas aplicadas en el centro de masas de los eslabones, como se aprecia en la figura 5.9.

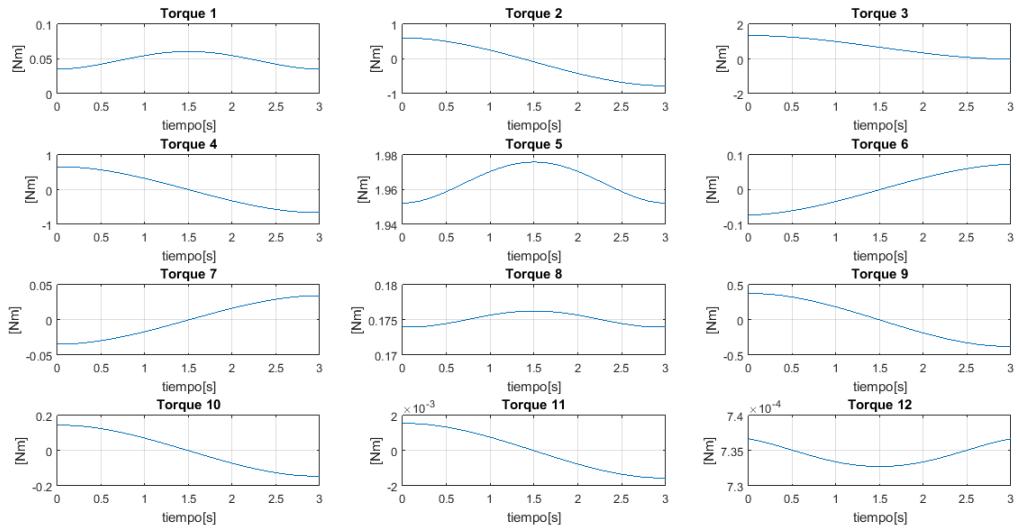


Figura 5.8: Torques articulares obtenidos con los valores óptimos del vector de diseño.

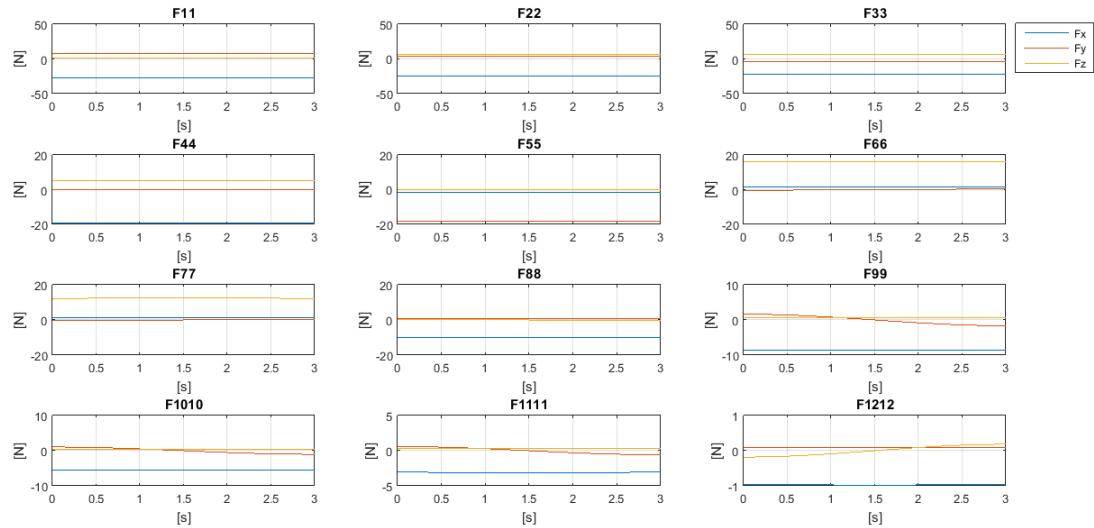


Figura 5.9: Fuerzas inerciales aplicadas en el centro de masas de los eslabones.

Las dimensiones obtenidas para longitudes de eslabones junto con las masas, son los requerimientos de diseño en la siguiente fase de la metodología, concretamente la fase de diseño modular, por lo que fungirán como restricciones en los diferentes sub-sistemas.

## 5.2. Modularización, diseño de detalle y validación a nivel de parte

### 5.2.1. Diseño del sub-sistema mecánico

El sub-sistema mecánico consta de dos módulos: estructura de los eslabones y elementos de máquinas, el primer modulo está enfocado al diseño de la estructura mecánica de soporte del eslabón, es decir la pieza sobre la que van montados diversos elementos como los actuadores, sensores, rodamientos, ejes, poleas y cableado, el segundo modulo engloba de manera específica el diseño de los elementos de maquina que requiere cada eslabón los cuales son tres: ejes, rodamientos y sistemas de transmisión por banda sincrona.

El diseño pertinente a cada modulo está determinado por restricciones de masas y longitudes heredadas de la optimizacion mediante algoritmos genéticos. A continuación se describe el diseño en cada modulo del sub-sistema mecánico.

### 5.2.2. Diseño de elementos de maquinas

En esta sección se presenta el diseño de los elementos de máquinas involucrados en los eslabones del robot, utilizando para ello herramientas de software disponibles por el fabricante, en el caso de la selección de rodamientos, bandas y poleas, se utilizan las fuerzas obtenidas mediante el modelo dinámico inverso resultante de la optimizacion por algoritmos genéticos.

En cuando al diseño de los ejes aquí se presentan dos, puesto que por la similitud de las fuerzas involucradas, estos diseños son validos para algunos eslabones, aquellos cuya carga sobre el eje es ordinariamente radial son de tipo 1 y de tipo 2 los ejes que soportan ordinariamente cargas axiales. De forma similar sólo se presenta la selecciona de los rodamientos para un solo eje, ya que el margen de seguridad para estos es suficiente para ser utilizado en todos los eslabones del robot debido a la similitud de las condiciones de fuerzas.

#### 5.2.2.1. Diseño de ejes

Para determinar las dimensiones correctas de los ejes se parte de las gráficas de fuerza cortante y momento flector, que se construyen con base en las fuerzas aplicadas sobre el eje. Estas fuerzas están basadas en los resultados obtenidos por medio de la optimizacion mediante AG. En la figura 3.11 se observa las dimensiones del eje y los

diámetros de diseño, así como el diagrama de cuerpo libre. La fuerza  $P_1$  corresponde a la fuerza aplicada sobre el centro de masas del eslabón anterior, la fuerza  $P_2$  es la fuerza neta de impulsión producida por la polea cuando ejerce una torsión sobre el eje y el momento  $M_1$  es producido por el eslabón adyacente.

$$P_2 = 1.5 \left( \frac{T}{D/2} \right) \quad (5.13)$$

donde  $T = 2.5 \text{ Nm}$  es el torque generado por la polea,  $D = 0.02 \text{ m}$  es el diámetro estimado de la polea [11][12], por tanto:

$$P_2 = 1.5 \left( \frac{2.5 \text{ Nm}}{0.0254 \text{ m}/2} \right) = 295.27 \text{ N} \quad (5.14)$$

por lo que la fuerza flexionante producida por la polea está determinada por los componentes:  $P_{2x} = P_2 \cos(90)$  y  $P_{2y} = P_2 \sin(90)$ , de forma que la única componente flexionante producida por la polea es la componente  $P_{2y} = 294.27 \text{ N}$ .

Las fuerzas de reacción calculadas son:  $R_B = 195.14 \text{ N} \downarrow$  y  $R_A = 520.41 \text{ N} \uparrow$ .

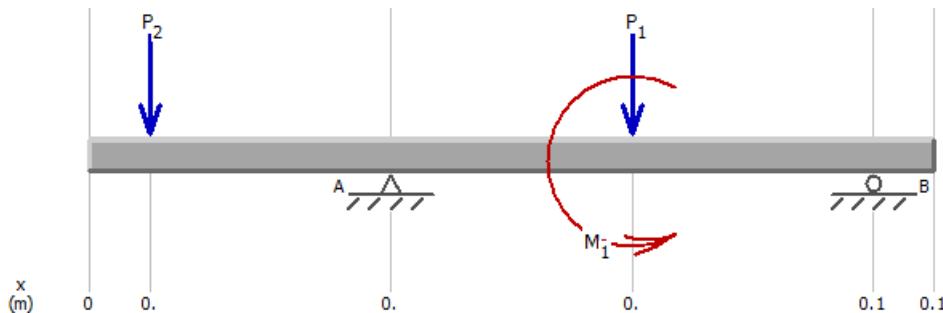


Figura 5.10: Diagrama de cuerpo libre del eje.

El análisis parte de los diagramas de momento flector y fuerza cortante del eje mostrado en la figura 5.10. El diámetro  $D$  de eje está determinado por la ecuación (40).

$$D = \left[ \frac{32(N)}{\pi} \sqrt{\left( \frac{k_t M}{S'_n} \right)^2 + \frac{3}{4} \left( \frac{T}{S_y} \right)^2} \right]^{1/3} \quad (5.15)$$

donde:

- $N$  es el factor de seguridad en el diseño.
- $k_t$  es el factor de concentración de esfuerzos.
- $S'_n$  es el esfuerzo modificado.

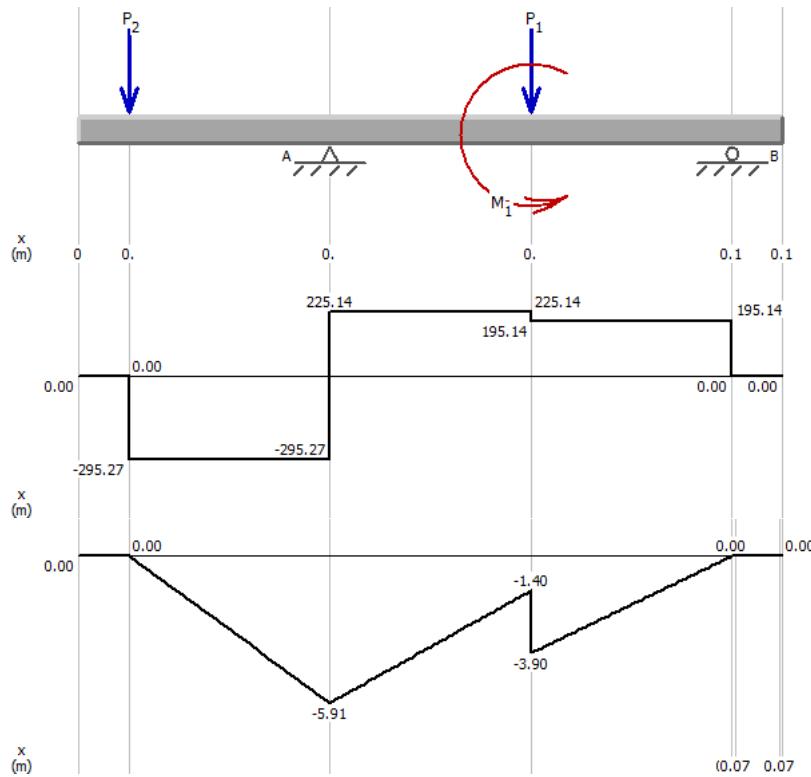


Figura 5.11: Diagrama de fuerza cortante y momento flector para eje tipo 1.

$M[Nm]$	$T[Nm]$	$S'_n[Mpa]$	$S_y[Mpa]$	$k_t$	$N$
0	2.5	97	276	N/A	1.5

Tabla 5.4: Parámetros de diseño para el eje en el punto A.

- $S_y$  es el esfuerzo de fluencia del material.
- $T$  es el torque neto presente en el punto de diseño.

El material de diseño para los ejes es Aluminio 6061, el cual cuenta con las siguientes características:  $S'_n = 97[MPa]$  y  $S_y = 276[MPa]$ , ya que según (Mott, 2006) el  $S'_n$  puede estimarse como  $S'_n = 0.5S_y$ .

**Diámetro para el punto A:** Los datos de diseño del eje en el punto A se presentan en la tabla 5.4. Se observa de la gráfica de momentos flexionantes que en el punto A es cero, por lo que el término  $\left(\frac{k_t M}{S'_n}\right)^2$  se desprecia, resultaron la ecuación de diseño como (41):

$$D_A = \left[ \frac{32(N)}{\pi} \sqrt{\frac{3}{4} \left( \frac{T}{S_y} \right)^2} \right]^{1/3}$$

$M[Nm]$	$T[Nm]$	$S'_n[Mpa]$	$S_y[Mpa]$	$k_t$	$N$
5.91	2.5	97	276	2.5	1.5

Tabla 5.5: Parámetros de diseño para el eje en el punto A.

$M[Nm]$	$T[Nm]$	$S'_n[Mpa]$	$S_y[Mpa]$	$k_t$	$N$
3.90	2.5	97	276	2.5	1.5

Tabla 5.6: Parámetros de diseño para el eje en el punto A.

$$D_A = \left[ \frac{32(1.5)}{\pi} \sqrt{\frac{3}{4} \left( \frac{2.5Nm}{276MPa} \right)^2} \right]^{1/3} = 0.0047m \approx 0.005m \quad (5.16)$$

**Diámetro para el punto B:** Los datos de diseño del eje en el punto A donde es el lugar de un rodamiento se presentan en la tabla. Se observa de la gráfica de momentos flexionantes que en el punto B es  $5.91Nm$ , y como se presenta un cambio de diámetro no redondeado  $k_t = 2.5$ , por tanto la ecuación de diseño resulta en:

$$D_B = \left[ \frac{32(N)}{\pi} \sqrt{\left( \frac{k_t M}{S'_n} \right)^2 + \frac{3}{4} \left( \frac{T}{S_y} \right)^2} \right]^{1/3}$$

$$D_B = \left[ \frac{32(1.5)}{\pi} \sqrt{\left( \frac{2.5(5.91Nm)}{97MPa} \right)^2 + \frac{3}{4} \left( \frac{2.5Nm}{276MPa} \right)^2} \right]^{1/3} = 0.0133m \quad (5.17)$$

**Diámetro para el punto C:** Los datos de diseño del eje en el punto A se presentan en la tabla 5.6. Se observa de la gráfica de momentos flexionantes que en el punto B es  $3.9Nm$ , y como se presenta un cambio de diámetro no redondeado  $k_t = 2.5$  a la derecha según (Mott, 2006), la ecuación de diseño resulta en:

$$D_C = \left[ \frac{32(N)}{\pi} \sqrt{\left( \frac{k_t M}{S'_n} \right)^2 + \frac{3}{4} \left( \frac{T}{S_y} \right)^2} \right]^{1/3}$$

$$D_C = \left[ \frac{32(1.5)}{\pi} \sqrt{\left( \frac{2.5(3.9Nm)}{97MPa} \right)^2 + \frac{3}{4} \left( \frac{2.5Nm}{276MPa} \right)^2} \right]^{1/3} = 0.0115m \quad (5.18)$$

$M[Nm]$	$T[Nm]$	$S'_n[Mpa]$	$S_y[Mpa]$	$k_t$	$N$
0	2.5	97	276	N/A	1.5

Tabla 5.7: Parámetros de diseño para el eje en el punto A.

Si bien el diámetro  $D_C$  resultó ser menor que el diámetro  $D_B$ , en el diseño final del eje se tomará la recomendación del fabricante de los rodamiento como pared lateral hombro, por tanto el diámetro de diseño para  $D_C = 0.012 + d_h$ , donde  $d_h$  es la altura designada por el fabricante para el hombro del rodamiento.

**Diámetro para el punto D:** Los datos de diseño del eje en el punto A donde es el lugar de un rodamiento se presentan en la tabla 5.7. Se observa de la gráfica de momentos flexionantes que en el punto B es  $5.91Nm$ , y como se presenta un cambio de diámetro no redondeado  $k_t = 2.5$  según (Mott, 2006), por tanto la ecuación de diseño resulta en:

$$D_D = \left[ \frac{32(N)}{\pi} \sqrt{\frac{3}{4} \left( \frac{T}{S_y} \right)^2} \right]^{1/3}$$

$$D_D = \left[ \frac{32(1.5)}{\pi} \sqrt{\frac{3}{4} \left( \frac{2.5Nm}{276MPa} \right)^2} \right]^{1/3} = 0.0047m \approx 0.005m \quad (5.19)$$

Sin embargo, a pesar de que en el punto D es el lugar de un rodamiento, al igual que en el punto B, y ambos resultados son diferentes debido a las cargas presentes, se considera mejor opción adoptar para el punto D, el mismo diámetro del rodamiento en B, para disminuir el numero de piezas distintas en el robot, esto es posible ya que el diámetro es menor que en B y por tanto el rodamiento seleccionado para B es más adecuado para el punto D.

El diseño final del eje tipo 1, se aprecia en la figura 5.12. La masa total del eje considerando la densidad del Aluminio 6061 es de 0.0144 kg.

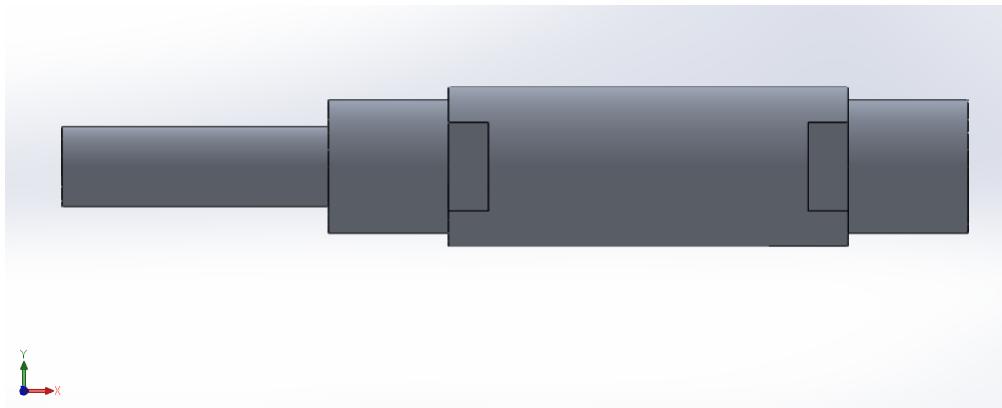


Figura 5.12: Dimensiones del diseño final para el eje para cargas radiales.

#### 5.2.2.2. Selección rodamientos

Para la selección de los rodamientos se requiere del cálculo de la carga dinámica, que depende de las fuerzas tangenciales y axiales a soportar por el rodamiento. Para este caso los rodamientos están sometidos a carga radial principalmente por lo que la carga equivalente  $P_e$  se determina según (Mott, 2006) mediante:

$$P_d = VR \quad (5.20)$$

donde  $V$  es el factor de rotación cuyo valor depende de si lo que rota es la pista interna del rodamiento o la externa, siendo  $V = 1$  si es la pista interna o  $V = 1.2$  si lo que rota es la pista externa. En el caso de los rodamientos para el robot bípedo se considera  $V = 1$ , puesto que en todos los casos la cara interna del rodamiento es la que rota con el eje. En el diseño se considera la carga del modelo dinámico de 30 N, utilizando un factor de seguridad de 2, por lo que  $P_d = 60N$ .

El diámetro interno requerido para el rodamiento es un factor de restricción en la selección del mismo, en este caso se utiliza un diámetro menor  $d = 0.01m$ , ya que este se aproxima a los resultados obtenidos en el cálculo del eje. Como las cargas que afectan al rodamiento son de tipo radial el rodamiento adecuado para el robot es del tipo una hilera de bolas con ranura profunda (Mott, 2006). La duración de diseño  $L_d$  está determinada por la ecuación:

$$L_d = (h)(rpm)(60min/h) \quad (5.21)$$

donde  $h$  es un factor establecido por las características de la máquina, en este caso usa 2000h que corresponde con las horas de trabajo estándar para electrodomésticos (Mott, 2006) y  $rpm$  es el valor de la velocidad de giro de la pista interna del rodamiento,

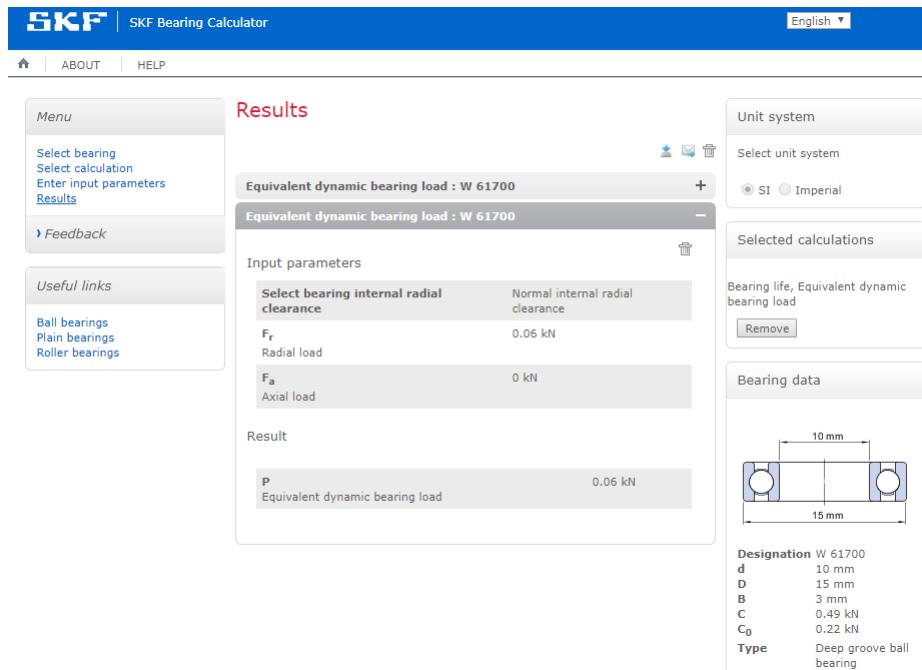


Figura 5.13: Rodamiento seleccionado por medio de la aplicación del fabricante SKF (SKF, 2017).

que en este caso es de 100 rpm. La capacidad de carga dinámica básica  $C$  se calcula mediante:

$$C = P_d (L_d/10^6)^{1/k} \quad (5.22)$$

donde  $k$  es un valor que depende del tipo de rodamiento, para el caso de rodamiento de bolas  $k = 3$ .

Sustituyendo los valores de  $h$  y rpm en se obtiene que:  $L_d = (h)(rpm)(60min/h) = (2000h)(100rpm)(60min/h) = 12 \times 10^6 rev$ , por tanto la capacidad de carga dinámica básica para el rodamiento es:

$$C = P_d (L_d/10^6)^{1/k} = 60 (12 \times 10^6 / 10^6)^{1/3} = 63.75 N$$

Con los datos calculados se seleccionaron los rodamientos con el fabricante SKF, mediante su aplicación para búsqueda de rodamientos, conforma a los parámetros de carga radial y axial. El programa calcula la carga dinámica y selecciona un rodamiento adecuado de la base de datos considerando para ello el rango de valores de diámetro interno y externo. El resultado de la búsqueda en el programa se observa en la figura 5.13.

De la figura 3.14 se observa que la carga dinámica calculada por el programa es

de  $C = 0.06kN$ , lo que coincide con los cálculos analíticos. Cambien se especifica que el rodamiento adecuado es el designado con el código W61700. En la figura 5.14 se aprecian las dimensiones del propio rodamiento, así como las tolerancias de montaje.

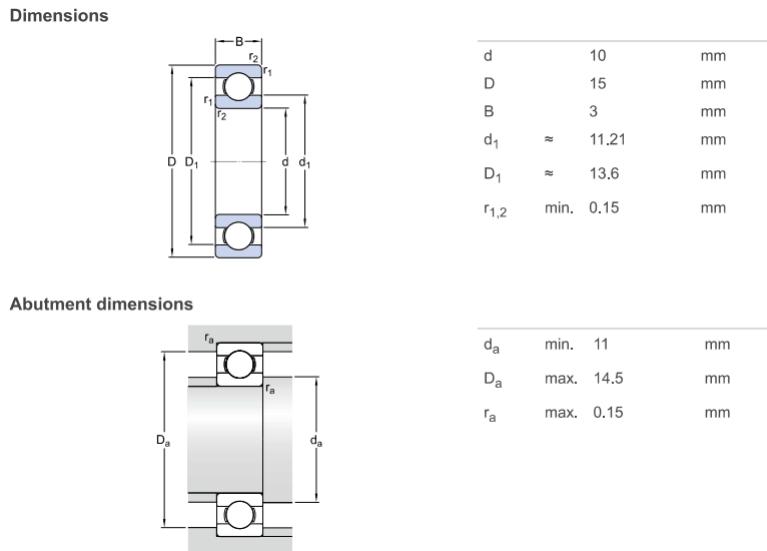


Figura 5.14: Especificaciones del rodamiento W61700, del fabricante SKF (SKF, 2017)

A continuación se en listan las características del rodamiento:

- Carga dinámica básica: 0.488 kN.
- Carga estática básica: 0.22 kN.
- Carga límite de fatiga: 0.009 kN.
- Límite de velocidad: 56000 r/min.
- Masa: 0.0013 kg

Debido a que las cargas que soportan los diferentes eslabones del robot son similares en magnitud, la carga dinámica básica en sus respectivos rodamientos es prácticamente la misma, la selección presentada de rodamiento es válido para todos los eslabones del robot.

#### 5.2.2.3. Selección de sistemas de transmisión por bandas sincronas

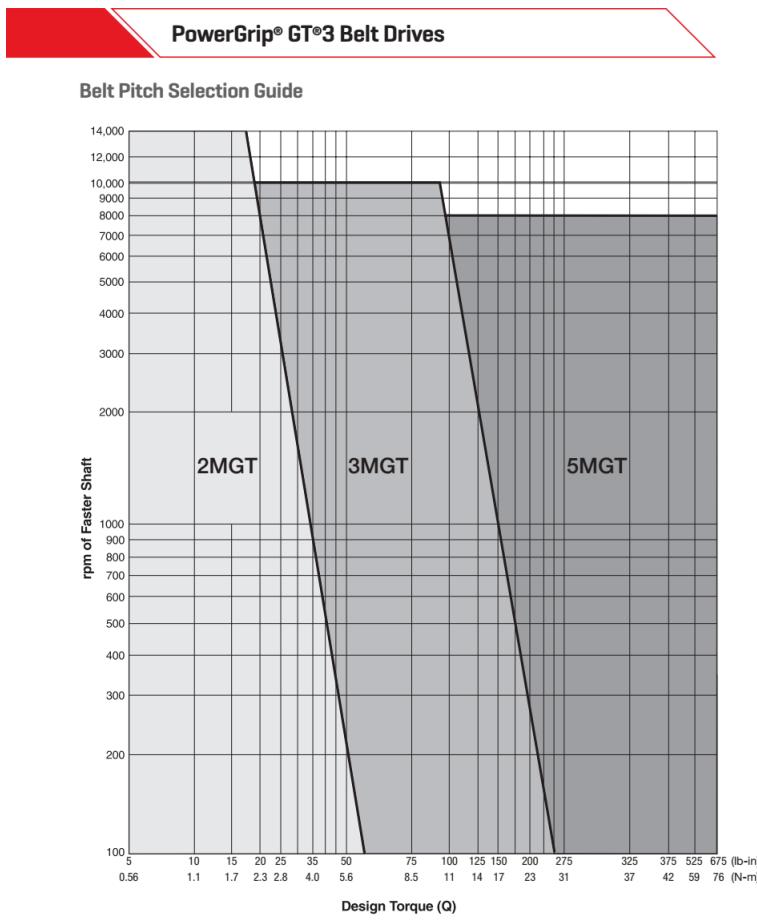
El sistema de transmisión de potencia utilizado en el robot bípedo esta basado en bandas sincronas, permitiendo distanciar al actuador de la articulación que controla, con lo que es posible reducir la inercia del propio eslabón, aunado a que este tipo de

sistemas no requiere de lubrifican, el juego entre dientes es menor y son de fácil reemplazo. A continuación se en listan los requisitos de diseño con bandas sincronas:

- Velocidad del motor de CD.
- Potencia del motor.
- Factor de servicio.
- Potencia de diseño.
- Relación de velocidades entre polea motriz y conducida.
- Distancia entre centros.

En este trabajo se presenta la metodología de selección del sistema de transmisión de potencia mediante bandas sincronas, basado en las recomendaciones del fabricante Gates Corporación. Cabe señalar que el procedimiento es el mismo para todos los sistemas de transmisión involucrados en el robot, por lo que aquí se presenta solo el caso para el eslabón 2.

Considerando que la potencia del actuador es de 5.55 W, el torque suministrado es de 2.1 Nm y que la velocidad de la polea conducida es de 100 rpm, de la gráfica suministrada por el fabricante mostrada en al figura 5.15, se obtiene el paso de la banda, considerando un factor de servicio de 2.5 y que por tanto la potencia de diseño es de  $F_s = 2.1Nm(2.5) = 5.25Nm$ .



 [www.gates.com/drivedesign](http://www.gates.com/drivedesign)

17

Figura 5.15: Gráfica de selección del paso de la banda del tipo PowerGrip G3 del fabricante Gates Corporación (Gates, 2017).

De la figura se observa que para una potencia de diseño de 3.36Nm y una velocidad de eje de 100 rpm, el paso de banda corresponde al tipo 2MGT, es decir un paso de 2mm. Por tanto el diseño del sistema de transmisión es factible seleccionando el tipo de banda PowerGrip G3. A continuación se describe el uso del software Design Flex Pro, suministrado por el fabricante, el cual es una herramienta que permite la rápida selección de los componentes del sistema de banda y polea entre dos puntos, consultando la base de datos del fabricante y los datos de entrada suministrados por el usuario. En la figura 5.16 se aprecia la ventana principal del software.

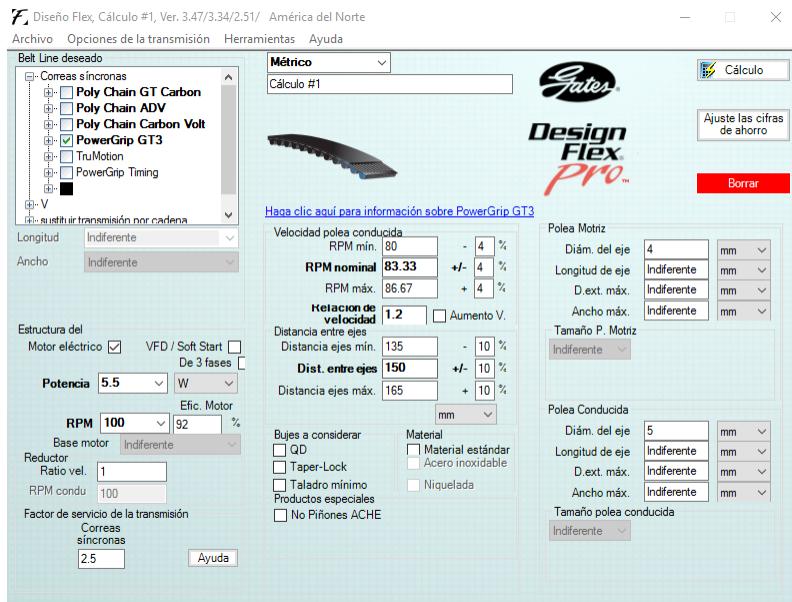


Figura 5.16: Ventana principal del software Design Flex Pro del fabricante Gates Corporation, corriendo en el sistema operativo Windows (Gates, 2017).

Dentro de las opciones del software para la linea de banda deseada se selecciona del arbol de tipo de bandas, dentro de la rama de bandas sincronas el tipo PowerGrip GT3, ya que según la potencia de diseño y velocidad del eje, es la adecuada, como se aprecia en la figura 5.17.

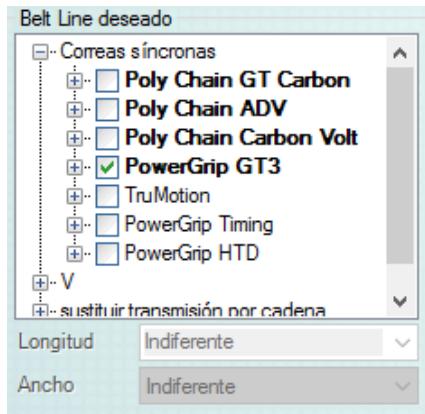


Figura 5.17: a) Tipo de banda seleccionada en el software. b)

Las características para la fuente de suministro de potencia se elige motor de CD y se introduce un valor de potencia de 5.5 W, una velocidad de 100 rpm, una relación de disminución de 1 puesto que se considera la velocidad del eje después de la etapa de reducción del motor, y el factor de servicio de 5.18 según(ver figura 3.19).

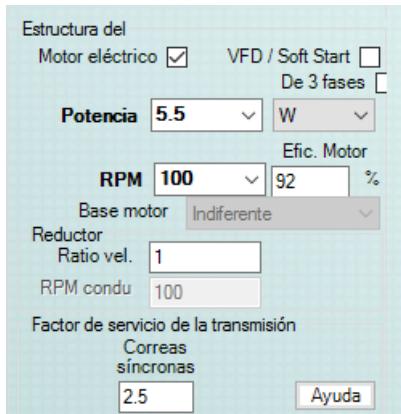


Figura 5.18: Parámetros de configuración del motor de CD introducidos en el software Design Flex Pro.

Dentro de las características para la polea conducida, se desea que esta tenga una relación de reducción de velocidad de 1.2 respecto de la polea motriz, indicándose en el campo correspondiente como se observa en la figura 5.19, por lo que la velocidad de la polea conducida se actualiza a 83.33 rpm nominales, con una tolerancia del  $\pm 4\%$ , también se indica la distancia deseada entre los ejes con una tolerancia de  $\pm 10\%$ .

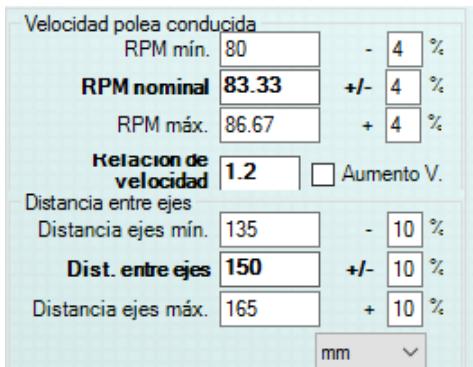


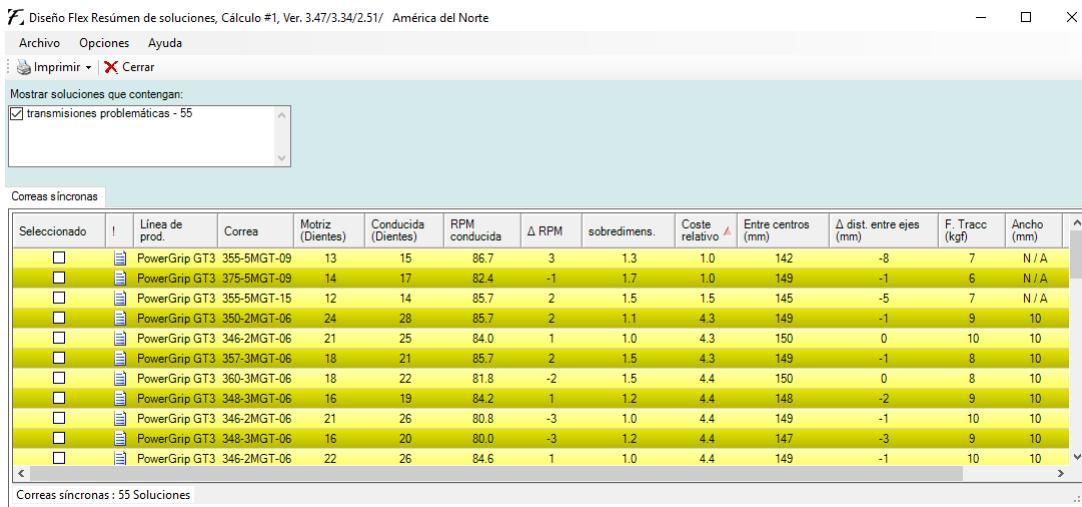
Figura 5.19: Parámetros introducidos en el software respecto de la polea conducida y distancia entre ejes.

finalmente se introducen los parámetros dimensionales de las poleas motriz y conducida, en este caso como solo se conocen los diámetros de los ejes donde irán colocadas las poleas, es el único parámetro que se indica, como se aprecia en la figura 5.20.

Al ejecutar la búsqueda mediante el botón «Cálculo», el software realiza una búsqueda de las posibles soluciones del sistema de transmisión de dos puntos, que satisfacen los requerimientos introducidos. Al finalizar la búsqueda el software entrega una lista de todas las soluciones encontradas, como se observa en la figura 5.21.

Polea Motriz			
Diám. del eje	4	mm	▼
Longitud de eje	Indiferente	mm	▼
D.ext. máx.	Indiferente	mm	▼
Ancho máx.	Indiferente	mm	▼
Tamaño P. Motriz			
Indiferente ▼			
Polea Conducida			
Diám. del eje	5	mm	▼
Longitud de eje	Indiferente	mm	▼
D.ext. máx.	Indiferente	mm	▼
Ancho máx.	Indiferente	mm	▼
Tamaño polea conducida			
Indiferente ▼			

Figura 5.20: Parámetros dimensionales para ambas poleas.



The screenshot shows the 'Diseño Flex' software interface with the title bar 'Diseño Flex Resumen de soluciones, Cálculo #1, Ver. 3.47/3.34/2.51/ América del Norte'. The menu bar includes 'Archivo', 'Opciones', 'Ayuda', 'Imprimir', and 'Cerrar'. A search bar 'Mostrar soluciones que contengan:' with the filter 'transmisiones problemáticas - 55' is visible. Below is a table titled 'Correas síncronas' listing 55 solutions:

Seleccionado	Línea de prod.	Correa	Motriz (Dientes)	Conducida (Dientes)	RPM conducida	Δ RPM	sobredimens.	Coste relativo	Entre centros (mm)	Δ dist. entre ejes (mm)	F. Tracc (kgf)	Ancho (mm)	
<input type="checkbox"/>	PowerGrip GT3	355-5MGT-09	13	15	88.7	3	1.3	1.0	142	-8	7	N/A	
<input type="checkbox"/>	PowerGrip GT3	375-5MGT-09	14	17	82.4	-1	1.7	1.0	149	-1	6	N/A	
<input type="checkbox"/>	PowerGrip GT3	355-5MGT-15	12	14	85.7	2	1.5	1.5	145	-5	7	N/A	
<input type="checkbox"/>	PowerGrip GT3	350-2MGT-06	24	28	85.7	2	1.1	4.3	149	-1	9	10	
<input type="checkbox"/>	PowerGrip GT3	346-2MGT-06	21	25	84.0	1	1.0	4.3	150	0	10	10	
<input type="checkbox"/>	PowerGrip GT3	357-3MGT-06	18	21	85.7	2	1.5	4.3	149	-1	8	10	
<input type="checkbox"/>	PowerGrip GT3	360-3MGT-06	18	22	81.8	-2	1.5	4.4	150	0	8	10	
<input type="checkbox"/>	PowerGrip GT3	348-3MGT-06	16	19	84.2	1	1.2	4.4	148	-2	9	10	
<input type="checkbox"/>	PowerGrip GT3	346-2MGT-06	21	26	80.8	-3	1.0	4.4	149	-1	10	10	
<input type="checkbox"/>	PowerGrip GT3	348-3MGT-06	16	20	80.0	-3	1.2	4.4	147	-3	9	10	
<input type="checkbox"/>	PowerGrip GT3	346-2MGT-06	22	26	84.6	1	1.0	4.4	149	-1	10	10	

Correas síncronas : 55 Soluciones

Figura 5.21: Soluciones factibles para el sistema de transmisión por bandas sincronas tipo PowerGrip GT3.

De las posibles soluciones para el diseño de la transmisión encontrados por el programa, se selecciona el tipo PowerGrip GT3 346-2MGT-06, donde el numero 346 indica la longitud de la polea en mm, 2MGT el paso y el ancho de la banda es de 6 mm, puesto que la distancia entre centros no presenta incrementos, y un bajo incremento en la velocidad de la polea conducida. Elegida la solución el software entrega un reporte con todos los componentes del sistema de transmisión, donde se indican parámetros relativos a la banda y las poleas seleccionadas como se observa en la figura 5.22. En

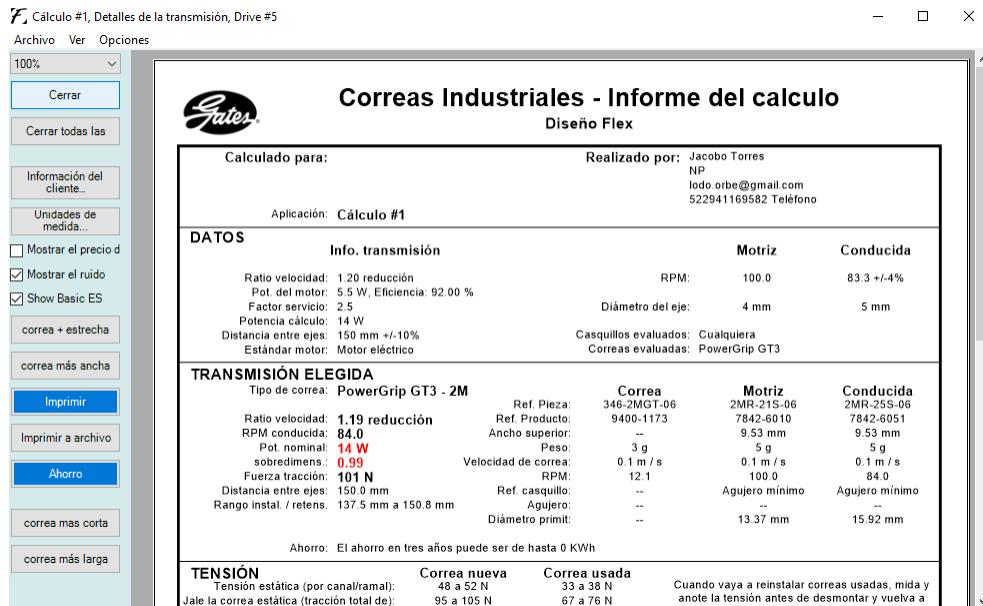


Figura 5.22: Reporte del diseño del sistema de transmisión seleccionado generado por el software Design Flex Pro.

este caso las poleas motriz y conducida corresponden al numero de parte 2MR-21S-06 y 2MR-25S-06 respectivamente.

El peso de la polea motriz 2MR-21S-06 es de 0.009 kg y el de la polea conducida 2MR-25S-06 de 0.0045 kg. Una imagen de la apariencia de las poleas se observa en la figura 5.23.



Figura 5.23: Fotografía de las poleas seleccionadas para polea motriz (2MR-21S-06) y conducida (2MR-25S-06).

### 5.2.3. Optimización topológica local de los eslabones

Para el diseño de los eslabones se tiene dos requerimientos, por una parte las distancia entre los ejes está determinada por los resultados en la optimiazacion mediante

AG, y la masa objetivo asociada a cada eslabón, el problema de diseño surge al intentar hacer converger el diseño del eslabón con la masa deseada maximizando la rigidez, por lo que en la metodología se propone el uso de optimización topológica local para los eslabones donde sea complejo asegurar el criterio de masa objetivo, maximizando la rigidez. Por tanto se requiere de un modelo de eslabón inicial que respete las restricciones de longitud entre ejes, considerando además los orificios de inserción de los rodamientos, con el diámetro específico, los espacios para los ejes, etc. El eslabón inicial es sometido a una optimización topológica con un criterio de reducción de masa para hacerla converger con la solución óptima generada por el AG. Finalizado el proceso de optimización topológica se tiene un nuevo diseño optimizado del eslabón. A continuación se describe el proceso de diseño general mediante optimización topológica para el robot bipedo.

#### 5.2.3.1. Optimización del eslabón L2

Partiendo de las restricción de longitud entre ejes y los resultados dimensionales para los ejes y rodamientos, se presenta en la figura el resultado de la optimización topológica realizada sobre el eslabón del segmento  $L_2$ , utilizando para ello la herramienta de optimización de forma del software Inventor en su versión para estudiantes 2017. La configuración para la optimización topología se en lista a continuación.

- Material de simulación: ABS
  - Comportamiento: Isotrópico
  - Modulo de Young: 2.240 GPa
  - Coeficiente de Poisson: 0.38
  - Modulo cortante: 805.000 MPa
  - Densidad: 1.060 g/cm<sup>3</sup>
  - Limite elástico: 20.000 Mpa.
  - Resistencia a la tracción: 29.600 Mpa.
- Objetivo: Maximizar la rigidez
- Criterio de optimización: Reducción de masa
- Masa inicial: 0.152 kg, porcentaje de reducción de masa: 45 %, masa destino: 0.0837 kg.

- Numero de nodos: 195991
- Numero de elementos: 132955
- Cargas: Fuerza aplicada en rodamiento de -60 N.
- Tipo de sujeciones: Restricción de pasador sin movimiento axial, radial y tangencial.

Además de las restricciones en listadas, también se designaron regiones de preservación de material, es decir las zonas donde el algoritmo de optimización topología no trabajaría, esto puede apreciarse en la figura 5.24. La fuerza aplicada al eslabón es obtenida mediante el modelo dinámico inverso.

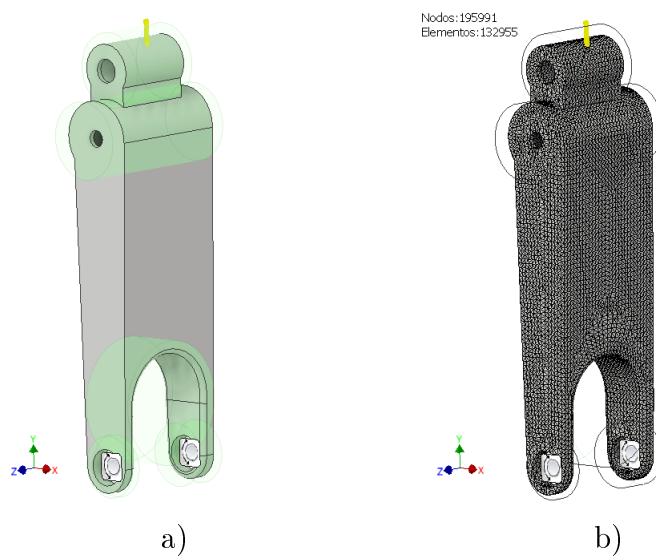


Figura 5.24: a) Cargas, sujetos y zonas de preservación introducidas al modelo para la optimización. b) Malla generada por el software.

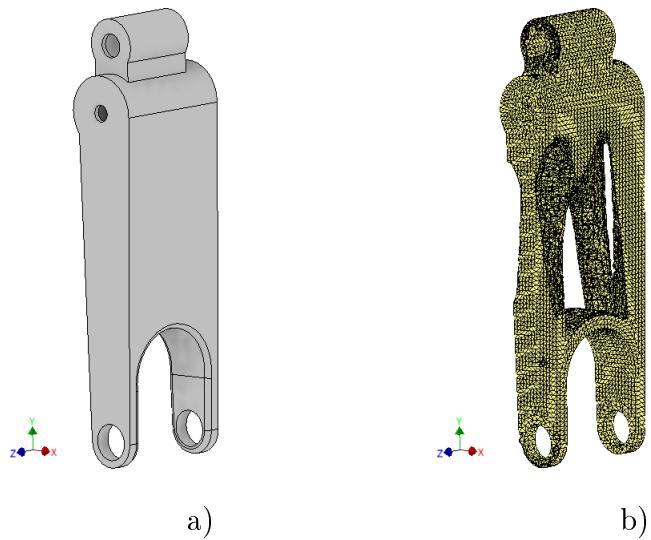


Figura 5.25: Etapas de diseño y optimización del eslabón para el segmento  $L_2$ . a) Diseño inicial sin optimizar, b) Forma generada por la optimización topológica.

Como se observa en la figura 5.25 el programa genera una nueva forma óptima en forma de malla, en función de los parámetros iniciales, esta malla funge como guía para la modificación de la pieza original, el resultado final del eslabón optimizado se observa en la figura 5.26.

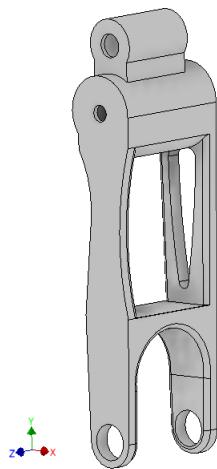


Figura 5.26: Re diseño del eslabón basado en la forma optima.

La figura 5.27 presenta el modelo de eslabón modificado a partir de la optimización topológica, la validación de la pieza se verifica mediante un análisis estructural estático por medio de elementos finitos, evaluando los esfuerzos y deformaciones presentes en el modelo optimizado. Los resultados de simulación realizados en inventor muestran a continuación.

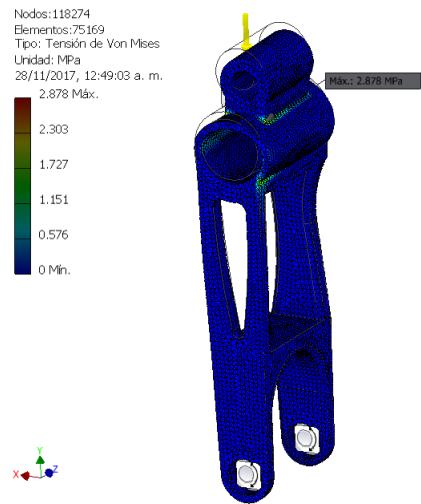


Figura 5.27: Esfuerzos de Vom Mises en el eslabón 2, sometido a las fuerzas calculadas por el modelo dinámico inverso.

Si bien el material utilizado para la simulación del análisis estático es plástico ABS para moldeo de partes plásticas, y el robot está pensado para ser construido mediante métodos de deposición térmica (impresión 3D) y que por lo tanto las propiedades mecánicas del material a pesar de ser ABS varían por la densidad y naturaleza de la impresión 3D, estudios realizados en (Lee, 2013), sugiere que el esfuerzo del límite elástico para piezas impresas 3D en ABS es de 8 Mpa, en el caso más critico, que es cuando los filamentos de impresión del material son perpendiculares a la linea de acción de la fuerza, y cuando los filamentos son paralelos a la fuerza el esfuerzos de límite elástico es de 22 Mpa como se observa en la figura 5.28.

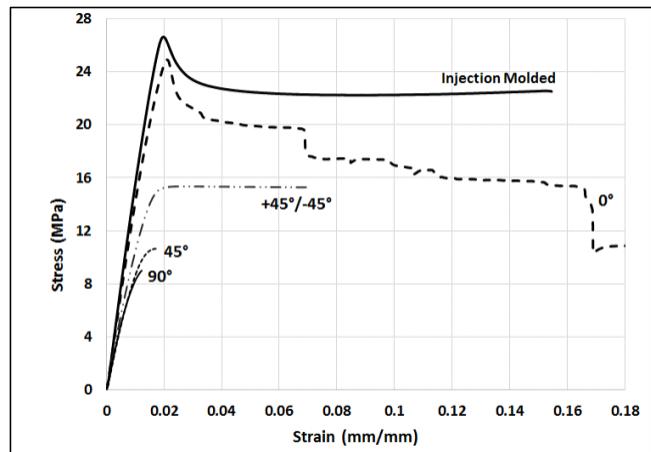


Figura 5.28: Curvas esfuerzo-deformación determinadas experimentalmente para especímenes impresos en 3D de material ABS (Lee, 2013).

Por tanto, bajo este criterio se puede observar en la figura 5.29 que la resistencia de la pieza es suficiente ya que el esfuerzo máximo es de 2.878 Mpa. Por otra parte las deformaciones presenten en el eslabón y el factor de seguridad calculado por el software pueden observarse en la figura 5.29.

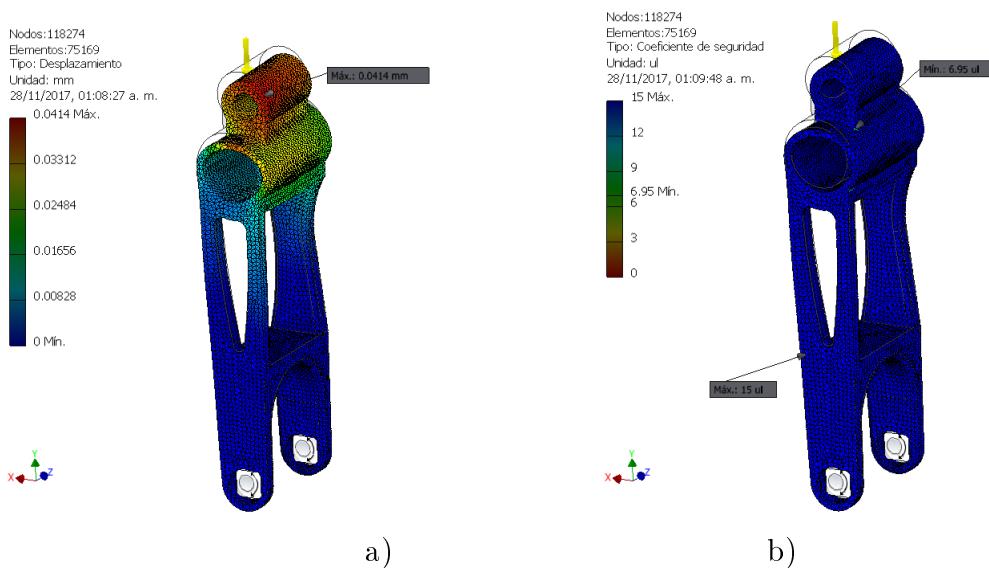


Figura 5.29: a) Desplazamiento máximo y mínimo calculados para el eslabón 2. b) Factor de seguridad calculado.

El desplazamiento presente en la pieza optimizada máximo es de 0.0414 mm, por lo que es despreciable, y por tanto la pieza puede resistir las fuerzas ejercidas sobre el, de igual forma puede verificarse esta afirmación observando el resultado de simulación para el factor de seguridad que en el peor de los casos es de 6.95.

### 5.2.3.2. Optimizacion del eslabón L2

La fuerza aplicada en los rodamientos en el eslabón 2, es obtenida del modelo dinámico inverso durante la trayectoria de soporte simple multiplicado por un factor de seguridad igual a 2. Los criterios de optimizacion para el eslabón dos se en listan a continuación.

- Material de simulación: ABS
  - Comportamiento: Isotrópico
  - Modulo de Young: 2.240 GPa
  - Coeficiente de Poisson: 0.38
  - Modulo cortante: 805.000 MPa
  - Densidad: 1.060 g/cm<sup>3</sup>
  - Limite elástico: 20.000 Mpa.
  - Resistencia a la tracción: 29.600 Mpa.
- Objetivo: Maximizar la rigidez
- Criterio de optimizacion: Reducción de masa
- Masa inicial: 0.178 kg, porcentaje de reducción de masa: 40 %, masa destino: 0.107 kg.
- Numero de nodos: 527006
- Numero de elementos: 363751
- Cargas: Fuerza aplicada en rodamiento de -60 N.
- Tipo de sujeciones: Restricción de pasador sin movimiento axial, radial y tangencial.

En la figura 5.30 se observa las regiones preservadas de material, así como la malla generada, si bien el software no es flexible en cuando al refinado de la malla, cabe mencionar que el análisis es realizado de forma paralela mediante el software de análisis de elementos finitos NASTRAN, el cual realiza automáticamente los ajustes pertinentes en el mallado, con las especificaciones realizadas en Inventor.

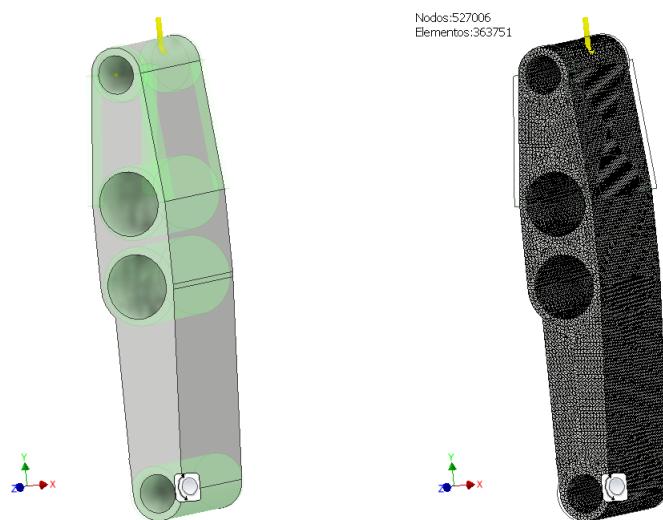


Figura 5.30: a) Regiones preservadas en la forma inicial de la pieza para el eslabón 3.  
b) Mallado generado por el software NASTRAN.

Considerando las restricciones y regiones preservadas el algoritmo de optimización topológica generado se observa en la figura 5.31, en comparación con el forma inicial.

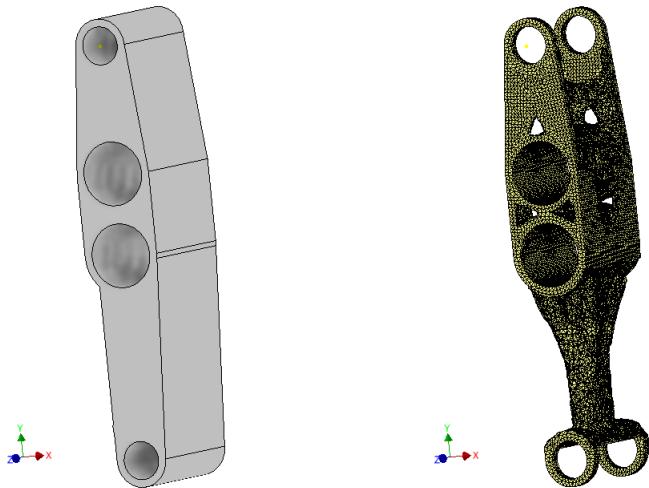


Figura 5.31: a) Pieza inicial diseñada. b) Forma optimizada.

Basado en la forma generada por el algoritmo, se re diseño el eslabón 3, resultando en la pieza mostrada en la figura 5.32. Este procedimiento es aplicado al resto de piezas del robot bípedo, considerando las fuerzas y momentos aplicados en sus respectivos centros de masa. De esta forma se asegura que el eslabón cumpla con el requerimiento de masa obtenidos mediante la optimizacion por algoritmos genéticos.

La pieza optimizada mostrada en la figura 5.32, es validada en la metodología propuesta mediante un análisis de esfuerzo y deformación de manera análoga al eslabón 2. Los resultados obtenidos referentes a los esfuerzos de Von Mises se muestran en la figura 5.33, de donde se aprecia que el diseño es correcto, puesto que el esfuerzo máximo es de 4.451 MPa lo cual es inferior al esfuerzo de límite elástico de las piezas impresa en ABS. También se aprecia en la figura que el desplazamiento es despreciable, ya que el máximo ocurrido es de 0.0358mm.

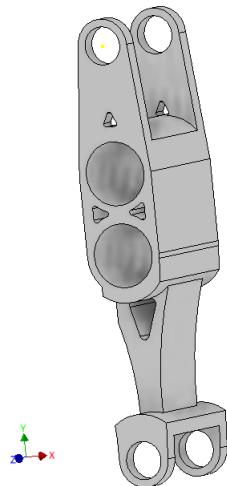


Figura 5.32: Forma final optimizada para el eslabón 3, generada por la optimización topológica.

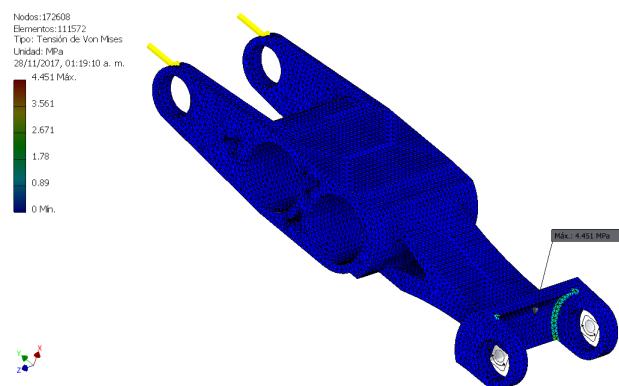


Figura 5.33: Esfuerzos de Von Mises en el eslabón 3.

Por otra parte el factor de seguridad calculado mostrado en la figura, indica que la pieza diseñada a partir de la optimizacion topologica, cuyo criterio de reducción de masa es dictado por la optimizacion por medio de algoritmos genéticos es lo suficientemente rígida para soportar las fuerzas y momentos generados durante la trayectoria de simple soporte según los datos obtenidos mediante el modelo dinámico inverso.

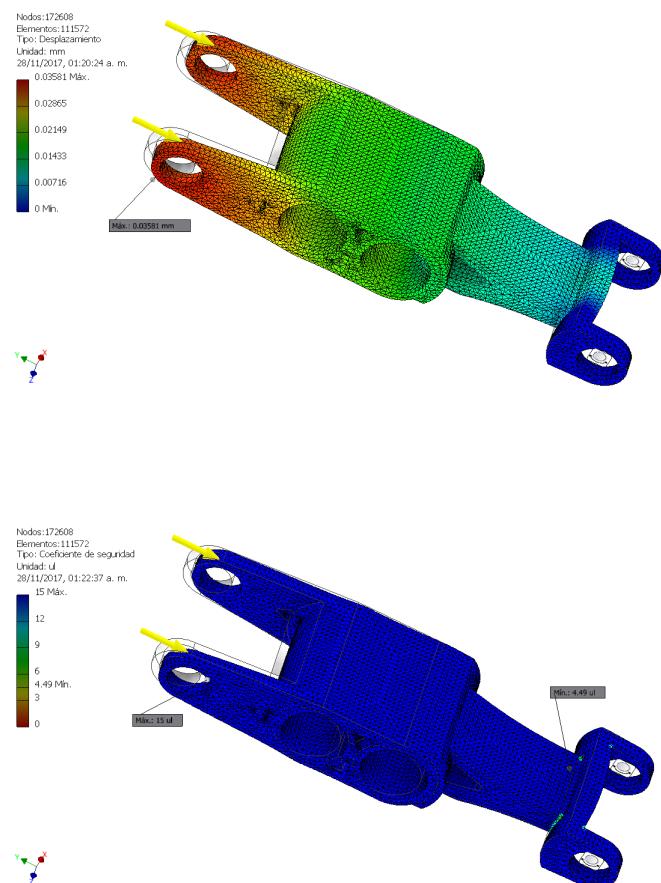


Figura 5.34: a) Desplazamientos máximos y mínimos en el eslabón 3. b) Factor de seguridad calculado por el software.

# Capítulo 6

## Resultados

La reintegración de los diferentes sub-sistemas, principalmente el sub-sistema mecánico, el cual fue validado en todas sus partes mediante análisis estático de esfuerzo-deformación por medio del método de los elementos finitos (FEM por sus siglas en inglés), se produjo de forma satisfactoria, ya que las masas objetivo dictadas por el análisis de optimización por medio de algoritmos genéticos fue mayor a la que se obtuvo al finalizar el diseño.

<i>Eslabón</i>	1	2	3	4	5	6
<i>objetivo(kg)</i>	0.22	0.25	0.32	0.15	0.22	0.40
<i>diseño(kg)</i>	0.2	0.18	0.3	0.12	0.189	0.380
<i>Eslabón</i>	7	8	9	10	11	12
<i>objetivo(kg)</i>	0.22	0.15	0.32	0.25	0.22	0.10
<i>diseño(kg)</i>	0.189	0.13	0.258	0.215	0.189	0.068

Tabla 6.1: Tabla de resultados de las masas obtenidas mediante el diseño basado en optimización topológica y las masa objetivo por medio de algoritmos genéticos.

La tabla 4.1 muestra los resultados de masas objetivos con las masas totales de los eslabones diseñadas bajo la metodología propuesta, donde aprecia que se cumplió con el objetivo de diseño.

# Capítulo 7

## Conclusiones

El uso de algoritmos de optimización, aceleran el proceso de diseño de sistemas mecatrónicos, al mismo tiempo que aseguran con un margen de error razonable la convergencia del diseño al modelo ideal, agilizando principalmente la fase de diseño conceptual y la fase de diseño de detalle.

Las herramientas computacionales de CAD/CAE fueron de gran utilidad en la validación en las fases ascendentes del modelo V, en especial en la validación de esfuerzos y deformaciones por medio de elementos finitos. También aceleraron el proceso de diseño de las piezas mecánicas en el entorno de ingeniería basado en modelos.

Los resultados obtenidos mediante optimización topológica en la metodología propuesta que parte del modelo V, fueron un componente vital en el existo del producto obtenido, ya que gracias a ello el proceso de diseño de los diferentes eslabones del robot, se redujo en sólo tres fases: diseño inicial, optimización topológica y validación, agilizando el proceso de diseño, que mediante métodos clásicos de diseño mecánico hubiera requerido tiempo e iteraciones sobre el re-diseño de las piezas.

Los modelos matemáticos son parte central en la metodología propuesta ya que en el caso del modelo dinámico inverso, este fungieron como función objetivo, lo que condujo a una optimización multi-objetivo dentro de las restricciones consideradas. Por tanto la utilización de Algoritmos genéticos en el diseño mechatrónico es una solución aceptable en proceso de la fase de diseño conceptual, facilitando la transformación de ideas conceptuales a requerimientos de diseño en los sub-sistemas modulares de dominio específico.

Finalmente cabe destacarse que la implementación del método iterativo de Newton-Euler es preferible a otros métodos para la obtención del modelo dinámico de manipuladores, puesto que su estructura algorítmica facilita su implementación mediante lenguajes de computo.

Es posible desarrollar una metodología basada en el modelo V, en el ámbito de la ingeniería de sistemas basada en modelos. El uso de algoritmos genéticos como medio para optimizar sistemas mecatrónicos, agiliza la obtención de requerimientos a nivel de parte. La optimización topológica es útil como forma de alcanzar objetivos de máxima rigidez con la menor masa posible. El método iterativo de Newton-Euler representa una ventaja para sistemas complejos, ya que no se requiere de la obtención de expresiones matemáticas analíticas.

# Referencias

- Schilling, R. J. (1990). FUNDAMENTALS OF ROBOTICS Analysis and Control. New Delhi: N.Y: Prentice-Hall.
- Casse, O. (2017). SysML in Action with Cameo Systems Modeler. ISTE Press - Elsevier.
- Friedenthal, S; Moore, A; Steiner, R. (2012). A Practical Guide to SysML The Systems Modeling Language. New York. Elsevier.
- Delligatti, L. (2013). SysML Distilled A Brief Guide to the Systems Modeling Language. New York. Addison-Wesley.
- Lee J. Huang A. (2013). Fatigue analysis of FDM materials. Rapid Prototyping Journal, 19, 291–299.
- Colomer R. V. (2013). Propuesta Metodológica En El Ámbito De Meca trónica: Vehículos De Competición De Alta Frecuencia (Tesis Doctoral). Valencia: Universidad Politécnica de Valencia.
- Armada M; González de S. P. (2004). Climbing nd Walking Robots. Países Bajos: Springer Science Business Media.
- Ramhuzaini R; Pulm U; Stetter R; 2007). Systematic Mechatronic Design Of A Piezo-Electric Brake. Francia: International Conference on Engineering Design.
- Craig J. J (2006). Robótica. México: Pearson Prentice Hall.
- Kumar, S.S. (2008). Introducción a la robótica. México: Mc Graw Hill
- Kapandji, A.I. (2012). Fisiología Articular Tomo 2. México: Editorial Panamericana
- Cuevas J.E; Osuna E.J; Oliva N.D; Díaz C.N0 (2016). Optimización: Algoritmos Programados Con Matlab. México: Alfaomega

- Sell R; Tamre M. (2005). Integration Of V-Model And Sysml For Advanced Mechatronics System Design. REM, (22), 17-21 .
- Mott R.L; P.E. (2006). Diseño de Elementos de Máquinas. México: Pearson Education
- Ruiz H.O. (2014) Análisis Cinemático y Dinámico de un Robot Bípedo de 12 GDL Internos Utilizando la Formulación Newton-Euler: Tesis. México: UNAM
- Bendsoe M.P; Sigmund O. (2004) Topology Optimization: Theory, Methods and Applications. New York: Springer-Verlag
- Gausemeier J; Moehringer S. (2002) A New Guideline For The Desing Of Mechatronic Systems. California USA: IFAC Publications

## Anexo 1. Programas para simulación de modelos matemáticos

### 1.1 Algoritmo escrito en Matlab para el cálculo del modelo dinámico inverso mediante la metodología de Newton-Euler de forma numérica

```

function obj_bipedo_plot(x)
    W11_Aux = [ ]; W22_Aux = [ ];
    W33_Aux = [ ]; W44_Aux = [ ];
    W55_Aux = [ ]; W66_Aux = [ ];
    W77_Aux = [ ]; W88_Aux = [ ];
    W99_Aux = [ ]; W1010_Aux = [ ];
    W1111_Aux = [ ]; W1212_Aux = [ ];

    F11_Aux = [ ]; F22_Aux = [ ];
    F33_Aux = [ ]; F44_Aux = [ ];
    F55_Aux = [ ]; F66_Aux = [ ];
    F77_Aux = [ ]; F88_Aux = [ ];
    F99_Aux = [ ]; F1010_Aux = [ ];
    F1111_Aux = [ ]; F1212_Aux = [ ];

    g = -9.81;
    TOB = 0.039*x(1)
    L1 = 0.285*x(1)-TOB      %0.15;
    L2 = 0.530*x(1)-TOB-L1  %0.15;
    L3 = 0.191*x(1)          %0.11;
    L4 = L2                  %0.15;
    L5 = L1                  %0.15;

    %H M1 M2 M3 M4 M5 M6 M12 TF]
    m1 = x(2)      %0.085;     m2 = x(3)      %0.350;
    m3 = x(4)      %0.350;     m4 = x(5)      %0.085;
    m5 = x(6)      %0.200;     m6 = x(7)      %0.380;
    m7 = m5        %0.200;     m8 = m4        %0.085;
    m9 = m3        %0.350;     m10 = m2       %0.350;

```

```

m11 = m1      %0.085;      m12 = x(8)      %0.200;

Tf = x(9)      %%;      ti1 = pi/12;
ti2 = pi/16;      ti3 = -2*pi/16;
ti4 = pi/16;      ti5 = -pi/18;
ti6 = pi/18;      ti7 = -pi/16;
ti8 = 0;          ti9 = pi/16;
ti10 = 0;         ti11 = 0;
ti12 = 0;

tf1 = pi/12;      tf2 = pi/16;
tf3 = -2*pi/16;    tf4 = pi/16;
tf5 = -pi/18;      tf6 = -pi/18;
tf7 = pi/16;       tf8 = 0;
tf9 = -pi/16;      tf10 = 0;
tf11 = 0;          tf12 = 0;

```

**figure (1)**

```

cd_bipedo(ti1,ti2,ti3,ti4,ti5,ti6,ti7,ti8,ti9,ti10,ti11,
           ti12,x);      cd_bipedo(tf1,tf2,tf3,tf4,tf5,tf6,tf7,tf8
           ,tf9,tf10,tf11,tf12,x);

```

```

[Theta1 , Thetap1 , Thetapp1] = trayectoria(ti1,tf1,Tf);
[Theta2 , Thetap2 , Thetapp2] = trayectoria(ti2,tf2,Tf);
[Theta3 , Thetap3 , Thetapp3] = trayectoria(ti3,tf3,Tf);
[Theta4 , Thetap4 , Thetapp4] = trayectoria(ti4,tf4,Tf);
[Theta5 , Thetap5 , Thetapp5] = trayectoria(ti5,tf5,Tf);
[Theta6 , Thetap6 , Thetapp6] = trayectoria(ti6,tf6,Tf);
[Theta7 , Thetap7 , Thetapp7] = trayectoria(ti7,tf7,Tf);
[Theta8 , Thetap8 , Thetapp8] = trayectoria(ti8,tf8,Tf);
[Theta9 , Thetap9 , Thetapp9] = trayectoria(ti9,tf9,Tf);
[Theta10 , Thetap10 , Thetapp10] = trayectoria(ti10,tf10,
           Tf);
[Theta11 , Thetap11 , Thetapp11] = trayectoria(ti11,tf11,
           Tf);
[Theta12 , Thetap12 , Thetapp12] = trayectoria(ti12,tf12,
           Tf);

```

Tf);

t = 0:0.1:Tf; [r, c] = size(t);  
*%\* Coordenadas centro de masa del eslabon i respecto de Si*  
*\*)*  
Pc11 = [TOB/3, 0, 0].';  
Pc22 = [-2\*L1/3, 0, 0].';  
Pc33 = [-2\*L2/3, 0, 0].';  
Pc44 = [0, 0, 0].';  
Pc55 = [0, 0, 0].';  
Pc66 = [L3/2, 0, 0].';  
Pc77 = [0, 0, 0].';  
Pc88 = [0, 0, 0].';  
Pc99 = [2\*L4/3, 0, 0].';  
Pc1010 = [2\*L5/3, 0, 0].';  
Pc1111 = [0, 0, 0].';  
Pc1212 = [TOB/3, 0, 0].';

*%\* Momentos de inercia\*)*

Ic11 = [0, 0, 0; 0, 0, 0; 0, 0, 0];  
Ic22 = [0, 0, 0; 0, 0, 0; 0, 0, 0];  
Ic33 = [0, 0, 0; 0, 0, 0; 0, 0, 0];  
Ic44 = [0, 0, 0; 0, 0, 0; 0, 0, 0];  
Ic55 = [0, 0, 0; 0, 0, 0; 0, 0, 0];  
Ic66 = [0, 0, 0; 0, 0, 0; 0, 0, 0];  
Ic77 = [0, 0, 0; 0, 0, 0; 0, 0, 0];  
Ic88 = [0, 0, 0; 0, 0, 0; 0, 0, 0];  
Ic99 = [0, 0, 0; 0, 0, 0; 0, 0, 0];  
Ic1010 = [0, 0, 0; 0, 0, 0; 0, 0, 0];  
Ic1111 = [0, 0, 0; 0, 0, 0; 0, 0, 0];  
Ic1212 = [0, 0, 0; 0, 0, 0; 0, 0, 0];

*%\* Fuerzas en el efecto final\*)*

f1313 = [0, 0, 0].';  
n1313 = [0, 0, 0].';

*%\* Coordenadas de Si respecto de Si-1\*)*

```

P01 = [-TOB, 0, 0].'; P12 = [0, 0, 0].';
P23 = [-L1, 0, 0].'; P34 = [-L2, 0, 0].';
P45 = [0, 0, 0].'; P56 = [0, 0, 0].';
P67 = [L3, 0, 0].'; P78 = [0, 0, 0].';
P89 = [0, 0, 0].'; P910 = [L4, 0, 0].';
P1011 = [L5, 0, 0].'; P1112 = [0, 0, 0].';
P1213 = [TOB, 0, 0].';

```

*(\* Variables articulares para eslabon fijo \*)*

```

W00 = [0, 0, 0].'; Wp00 = [0, 0, 0].';
Vp00 = [g, 0, 0].'; Z0 = [0, 0, 1].';

```

```

for i=1:c
(* Matrices de rotacion Ri y sus inversas *)
R01 = [cos(Theta1(i)), -sin(Theta1(i)), 0; sin(Theta1(i)),
        cos(Theta1(i)), 0; 0, 0, 1];
R10 = R01.';

R12 = [cos(Theta2(i)), -sin(Theta2(i)), 0; 0, 0, 1; -sin(Theta2(i)),
        -cos(Theta2(i)), 0];
R21 = R12.';

R23 = [cos(Theta3(i)), -sin(Theta3(i)), 0; sin(Theta3(i)),
        cos(Theta3(i)), 0; 0, 0, 1];
R32 = R23.';

R34 = [cos(Theta4(i)), -sin(Theta4(i)), 0; sin(Theta4(i)),
        cos(Theta4(i)), 0; 0, 0, 1];
R43 = R34.';

R45 = [cos(Theta5(i) - (pi/2)), -sin(Theta5(i) - (pi/2)),
        0; 0, 0, -1; sin(Theta5(i) - (pi/2)), cos(Theta5(i) - (pi/2)),
        0];
R54 = R45.';

R56 = [cos(Theta6(i) - pi), -sin(Theta6(i) - pi), 0; 0, 0,
        -1; sin(Theta6(i) - pi), cos(Theta6(i) - pi), 0];
R65 = R56.';

R67 = [cos(Theta7(i)), -sin(Theta7(i)), 0; sin(Theta7(i)),
        cos(Theta7(i)), 0; 0, 0, 1];
R76 = R67.';

```

```

R78 = [cos(Theta8(i) - (pi/2)), -sin(Theta8(i) - (pi/2)),
        0; 0,0, -1; sin(Theta8(i) - (pi/2)), cos(Theta8(i) - (
pi/2)), 0];
R87 = R78 .';
R89 = [cos(Theta9(i)), -sin(Theta9(i)), 0; 0, 0,1; -sin(
Theta9(i)), -cos(Theta9(i)), 0];
R98 = R89 .';
R910 = [cos(Theta10(i)), -sin(Theta10(i)), 0; sin(Theta10
(i)),cos(Theta10(i)), 0; 0, 0, 1];
R109 = R910 .';
R1011 = [cos(Theta11(i)), -sin(Theta11(i)), 0; sin(
Theta11(i)),cos(Theta11(i)), 0; 0, 0, 1]; R1110
= R1011 .';
R1112 = [cos(Theta12(i)), -sin(Theta12(i)), 0; 0,0, -1;
sin(Theta12(i)), cos(Theta12(i)), 0]; R1211 =
R1112 .';
R1213 = [1,0,0;0,1,0;0,0,1];
R1312 = [1,0,0;0,1,0;0,0,1];

```

*%NE adelante desde 0 – 11*

```

%PARA I = O
W11 = R10*W00 + Thetap1(i)*Z0;
W11_Aux = horzcat(W11_Aux,W11);
Wp11 = R10*Wp00 + cross((R10*W00),(Thetap1(i)*Z0)) +
Thetapp1(i)*Z0;
Vp11 = R10*(cross(Wp00,P01) + cross(W00,cross(W00,P01)) +
Vp00);
Vpc11 = cross(Wp11,Pc11) + cross(W11,cross(W11,Pc11)) +
Vp11;
F11 = m1*Vpc11;
Nn11 = Ic11*Wp11 + cross(W11,(Ic11*W11));

```

*%PARA I = 1*

```

W22 = R21*W11 + Thetap2(i)*Z0 ;
W22_Aux = horzcat(W22_Aux,W22);
Wp22 = R21*Wp11 + cross((R21*W11),(Thetap2(i)*Z0)) +

```

```

Thetaapp2( i )*Z0 ;
Vp22 = R21*( cross( Wp11, P12 ) + cross( W11, cross( W11, P12 ) ) +
Vp11 ) ;
Vpc22 = cross( Wp22, Pc22 ) + cross( W22, cross( W22, Pc22 ) ) +
Vp22 ;
F22 = m2*Vpc22 ;
Nn22 = Ic22*Wp22 + cross( W22, ( Ic22*W22 ) ) ;

%PARA I = 2
W33 = R32*W22 + Thetaapp3( i )*Z0 ;
W33_Aux = horzcat( W33_Aux, W33 ) ;
Wp33 = R32*Wp22 + cross( ( R32*W22 ), ( Thetaapp3( i )*Z0 ) ) +
Thetaapp3( i )*Z0 ;
Vp33 = R32*( cross( Wp22, P23 ) + cross( W22, cross( W22, P23 ) ) +
Vp22 ) ;
Vpc33 = cross( Wp33, Pc33 ) + cross( W33, cross( W33, Pc33 ) ) +
Vp33 ;
F33 = m3*Vpc33 ;
Nn33 = Ic33*Wp33 + cross( W33, ( Ic33*W33 ) ) ;

%PARA I = 3
W44 = R43*W33 + Thetaapp4( i )*Z0 ;
W44_Aux = horzcat( W44_Aux, W44 ) ;
Wp44 = R43*Wp33 + cross( ( R43*W33 ), ( Thetaapp4( i )*Z0 ) ) +
Thetaapp4( i )*Z0 ;
Vp44 = R43*( cross( Wp33, P34 ) + cross( W33, cross( W33, P34 ) ) +
Vp33 ) ;
Vpc44 = cross( Wp44, Pc44 ) + cross( W44, cross( W44, Pc44 ) ) +
Vp44 ;
F44 = m4*Vpc44 ;
Nn44 = Ic44*Wp44 + cross( W44, ( Ic44*W44 ) ) ;

%PARA I = 4
W55 = R54*W44 + Thetaapp5( i )*Z0 ;
W55_Aux = horzcat( W55_Aux, W55 ) ;
Wp55 = R54*Wp44 + cross( ( R54*W44 ), ( Thetaapp5( i )*Z0 ) ) +

```

```

    Thetapp5( i )*Z0 ;
    Vp55 = R54*( cross( Wp44, P45 ) + cross( W44, cross( W44, P45 ) ) +
              Vp44) ;
    Vpc55 = cross( Wp55, Pc55 ) + cross( W55, cross( W55, Pc55 ) ) +
              Vp55 ;
    F55 = m5*Vpc55 ;
    Nn55 = Ic55*Wp55 + cross( W55, ( Ic55*W55 ) ) ;

    %PARA I = 5
    W66 = R65*W55 + Thetap6( i )*Z0 ;
    W66_Aux = horzcat( W66_Aux, W66 ) ;
    Wp66 = R65*Wp55 + cross( ( R65*W55 ) , ( Thetap6( i )*Z0 ) ) +
              Thetapp6( i )*Z0 ;
    Vp66 = R65*( cross( Wp55, P56 ) + cross( W55, cross( W55, P56 ) ) +
              Vp55 ) ;
    Vpc66 = cross( Wp66, Pc66 ) + cross( W66, cross( W66, Pc66 ) ) +
              Vp66 ;
    F66 = m6*Vpc66 ;
    Nn66 = Ic66*Wp66 + cross( W66, ( Ic66*W66 ) ) ;

    %PARA I = 6
    W77 = R76*W66 + Thetap7( i )*Z0 ;
    W77_Aux = horzcat( W77_Aux, W77 ) ;
    Wp77 = R76*Wp66 + cross( ( R76*W66 ) , ( Thetap7( i )*Z0 ) ) +
              Thetapp7( i )*Z0 ;
    Vp77 = R76*( cross( Wp66, P67 ) + cross( W66, cross( W66, P67 ) ) +
              Vp66 ) ;
    Vpc77 = cross( Wp77, Pc77 ) + cross( W77, cross( W77, Pc77 ) ) +
              Vp77 ;
    F77 = m7*Vpc77 ;
    Nn77 = Ic77*Wp77 + cross( W77, ( Ic77*W77 ) ) ;

    %PARA I = 7           W88 = R87*W77 + Thetap8( i )*Z0;
    W88_Aux = horzcat( W88_Aux, W88 ) ;
    Wp88 = R87*Wp77 + cross( ( R87*W77 ) , ( Thetap8( i )*Z0 ) ) +
              Thetapp8( i )*Z0 ;

```

```

Vp88 = R87*(cross(Wp77,P78) + cross(W77,cross(W77,P78)) +
Vp77);

Vpc88 = cross(Wp88,Pc88) + cross(W88,cross(W88,Pc88)) +
Vp88;

F88 = m8*Vpc88;

Nn88 = Ic88*Wp88 + cross(W88,(Ic88*W88));

%PARA I = 8

W99 = R98*W88 + Thetap9(i)*Z0; W99_Aux = horzcat(W99_Aux,
W99);

Wp99 = R98*Wp88 + cross((R98*W88),(Thetap9(i)*Z0)) +
Thetapp9(i)*Z0;

Vp99 = R98*(cross(Wp88,P89) + cross(W88,cross(W88,P89)) +
Vp88);

Vpc99 = cross(Wp99,Pc99) + cross(W99,cross(W99,Pc99)) +
Vp99;

F99 = m9*Vpc99;

Nn99 = Ic99*Wp99 + cross(W99,(Ic99*W99));

%PARA I = 9

W1010 = R109*W99 + Thetap10(i)*Z0;
W1010_Aux = horzcat(W1010_Aux,W1010);

Wp1010 = R109*Wp99 + cross((R109*W99),(Thetap10(i)*Z0)) +
Thetapp10(i)*Z0;

Vp1010 = R109*(cross(Wp99,P910) + cross(W99,cross(W99,
P910)) + Vp99);

Vpc1010 = cross(Wp1010,Pc1010) + cross(W1010,cross(W1010,
Pc1010)) + Vp1010;

F1010 = m10*Vpc1010;

Nn1010 = Ic1010*Wp1010 + cross(W1010,(Ic1010*W1010));

%PARA I = 10

W1111 = R1110*W1010 + Thetap11(i)*Z0;
W1111_Aux = horzcat(W1111_Aux,W1111);

Wp1111 = R1110*Wp1010 + cross((R1110*W1010),(Thetap11(i)*
Z0)) + Thetapp11(i)*Z0;

```

```

Vp1111 = R1110*(cross(Wp1010,P1011) + cross(W1010,cross(W1010,P1011)) + Vp1010);
Vpc1111 = cross(Wp1111,Pc1111) + cross(W1111,cross(W1111,Pc1111)) + Vp1111;
F1111 = m11*Vpc1111;
Nn1111 = Ic1111*Wp1111 + cross(W1111,(Ic1111*W1111));

```

*%PARA I = 11*

```

W1212 = R1211*W1111 + Thetap12(i)*Z0;
W1212_Aux = horzcat(W1212_Aux,W1212);
Wp1212 = R1211*Wp1111 + cross((R1211*W1111),(Thetap12(i)*Z0)) + Thetapp12(i)*Z0;
Vp1212 = R1211*(cross(Wp1111,P1112) + cross(W1111,cross(W1111,P1112)) + Vp1111);
Vpc1212 = cross(Wp1212,Pc1212) + cross(W1212,cross(W1212,Pc1212)) + Vp1212;
F1212 = m12*Vpc1212;
Nn1212 = Ic1212*Wp1212 + cross(W1212,(Ic1212*W1212));

```

*%N\_E hacia atrás*

*%Torque en 12*

```

f1212 = (R1213*f1313) + F1212;
F1212_Aux = horzcat(F1212_Aux,f1212);
n1212 = Nn1212 + (R1213*n1313) + cross(Pc1212,F1212) + cross(P1213,(R1213*f1313));
T12(i) = n1212.'*Z0;

```

*%Torque en 11*

```

f1111 = (R1112*f1212) + F1111;
F1111_Aux = horzcat(F1111_Aux,f1111);
n1111 = Nn1111 + (R1112*n1212) + cross(Pc1111,F1111) + cross(P1112,(R1112*f1212));
T11(i) = n1111.'*Z0;

```

*%Torque 10*

```

f1010 = (R1011*f1111) + F1010;

```

```

F1010_Aux = horzcat(F1010_Aux, f1010);
n1010 = Nn1010 + (R1011*n1111) + cross(Pc1010, F1010) +
cross(P1011, (R1011*f1111));
T10(i) = n1010.*Z0;

```

*% Torque 9*

```

f99 = (R910*f1010) + F99;
F99_Aux = horzcat(F99_Aux, f99);
n99 = Nn99 + (R910*n1010) + cross(Pc99, F99) + cross(P910
,(R910*f1010));
T9(i) = n99.*Z0;

```

*% Torque 8*

```

f88 = (R89*f99) + F88;
F88_Aux = horzcat(F88_Aux, f88);
n88 = Nn88 + (R89*n99) + cross(Pc88, F88) + cross(P89, (R89
*f99));
T8(i) = n88.*Z0;

```

*% Torque 7*

```

f77 = (R78*f88) + F77;
F77_Aux = horzcat(F77_Aux, f77);
n77 = Nn77 + (R78*n88) + cross(Pc77, F77) + cross(P78, (R78
*f88));
T7(i) = n77.*Z0;

```

*% Torque 6*

```

f66 = (R67*f77) + F66;
F66_Aux = horzcat(F66_Aux, f66);
n66 = Nn66 + (R67*n77) + cross(Pc66, F66) + cross(P67, (R67
*f77));
T6(i) = n66.*Z0;

```

*% Torque 5*

```

f55 = (R56*f66) + F55;
F55_Aux = horzcat(F55_Aux, f55);

```

```

n55 = Nn55 + (R56*n66) + cross(Pc55,F55) + cross(P56,(R56
*f66));
T5( i ) = n55.*Z0;

% Torque 4
f44 = (R45*f55) + F44;
F44_Aux = horzcat(F44_Aux,f44);
n44 = Nn44 + (R45*n55) + cross(Pc44,F44) + cross(P45,(R45
*f55));
T4( i ) = n44.*Z0;

% Torque 3
f33 = (R34*f44) + F33;
F33_Aux = horzcat(F33_Aux,f33);
n33 = Nn33 + (R34*n44) + cross(Pc33,F33) + cross(P34,(R34
*f44));
T3( i ) = n33.*Z0;

% Torque 2
f22 = (R23*f33) + F33;
F22_Aux = horzcat(F22_Aux,f22);
n22 = Nn22 + (R23*n33) + cross(Pc22,F22) + cross(P23,(R23
*f33));
T2( i ) = n22.*Z0;

% Torque 1
f11 = (R12*f22) + F11;
F11_Aux = horzcat(F11_Aux,f11);
n11 = Nn11 + (R12*n22) + cross(Pc11,F11) + cross(P12,(R12
*f22));
T1( i ) = n11.*Z0;

end

figure(2)
subplot(4,3,1); plot(t,T1), title('Torque_1'), xlabel
('tiempo [s]'), ylabel('[Nm]'), grid; subplot(4,3,2)

```

```

;      plot(t,T2), title('Torque_2'), xlabel('tiempo [s]'),
) , ylabel('[Nm]') , grid; subplot(4,3,3); plot(t
,T3), title('Torque_3'), xlabel('tiempo [s]'), ylabel('[
Nm]') , grid; subplot(4,3,4); plot(t,T4), title(
'Torque_4'), xlabel('tiempo [s]'), ylabel('[Nm]') , grid;
subplot(4,3,5); plot(t,T5), title('Torque_5'),
xlabel('tiempo [s]'), ylabel('[Nm]') , grid; subplot
(4,3,6); plot(t,T6), title('Torque_6'), xlabel('
tiempo [s]'), ylabel('[Nm]') , grid; subplot(4,3,7);
plot(t,T7), title('Torque_7'), xlabel('tiempo [s]'),
ylabel('[Nm]') , grid; subplot(4,3,8); plot(t,
T8), title('Torque_8'), xlabel('tiempo [s]'), ylabel('[
Nm]') , grid; subplot(4,3,9); plot(t,T9), title(
'Torque_9'), xlabel('tiempo [s]'), ylabel('[Nm]') , grid;
subplot(4,3,10); plot(t,T10), title('Torque_10'),
xlabel('tiempo [s]'), ylabel('[Nm]') , grid;
subplot(4,3,11); plot(t,T11), title('Torque_11'),
xlabel('tiempo [s]'), ylabel('[Nm]') , grid; subplot
(4,3,12); plot(t,T12), title('Torque_12'), xlabel('
tiempo [s]'), ylabel('[Nm]') , grid; figure(3)
subplot(4,3,1); plot(t,W11_Aux), title('W11'), xlabel
('[s]'), ylabel('[rad/s]') , grid; subplot(4,3,2);
plot(t,W22_Aux), title('W22'), xlabel('[s]'),
ylabel('[rad/s]') , grid; subplot(4,3,3); plot(t
,W33_Aux), title('W33'), xlabel('[s]'), ylabel('[ rad/s ]
') , grid; subplot(4,3,4); plot(t,W44_Aux),
title('W44'), xlabel('[s]'), ylabel('[ rad/s ]') , grid;
subplot(4,3,5); plot(t,W55_Aux), title('W55'),
xlabel('[s]'), ylabel('[ rad/s ]') , grid; subplot
(4,3,6); plot(t,W66_Aux), title('W66'), xlabel('[ s ]
') , ylabel('[ rad/s ]') , grid; subplot(4,3,7);
plot(t,W77_Aux), title('W77'), xlabel('[s]'), ylabel('[ [
rad/s ] ') , grid; subplot(4,3,8); plot(t,W88_Aux)
, title('W88'), xlabel('[s]'), ylabel('[ rad/s ]') , grid;
subplot(4,3,9); plot(t,W99_Aux), title('W99'),
xlabel('[s]'), ylabel('[ rad/s ]') , grid; subplot

```

```

(4,3,10); plot(t,W1010_Aux), title('W1010'), xlabel(['s']),
ylabel(['rad/s']), grid; subplot(4,3,11);
plot(t,W1111_Aux), title('W1111'), xlabel(['s']),
ylabel(['rad/s']), grid; subplot(4,3,12); plot(
t,W1212_Aux), title('W1212'), xlabel(['s']),
ylabel(['rad/s']), grid; figure(4)

subplot(4,3,1); plot(t,F11_Aux), title('F11'), xlabel(
['s']), ylabel(['N']), grid; subplot(4,3,2);
plot(t,F22_Aux), title('F22'), xlabel(['s']),
ylabel(['N']), grid; subplot(4,3,3); plot(t,F33_Aux),
title('F33'), xlabel(['s']), ylabel(['N']), grid;
subplot(4,3,4); plot(t,F44_Aux), title('F44'),
xlabel(['s']), ylabel(['N']), grid; subplot(4,3,5);
plot(t,F55_Aux), title('F55'), xlabel(['s']),
ylabel(['N']), grid; subplot(4,3,6); plot(t,
F66_Aux), title('F66'), xlabel(['s']),
ylabel(['N']), grid; subplot(4,3,7); plot(t,F77_Aux),
title('F77'), xlabel(['s']),
ylabel(['N']), grid; subplot(4,3,8);
plot(t,F88_Aux), title('F88'), xlabel(['s']),
ylabel(['N']), grid; subplot(4,3,9); plot(t,
F99_Aux), title('F99'), xlabel(['s']),
ylabel(['N']), grid; subplot(4,3,10); plot(t,F1010_Aux),
title('F1010'), xlabel(['s']),
ylabel(['N']), grid;
subplot(4,3,11); plot(t,F1111_Aux), title('F1111'),
xlabel(['s']), ylabel(['N']), grid; subplot
(4,3,12); plot(t,F1212_Aux), title('F1212'), xlabel(
['s']), ylabel(['N']), grid;

end

```

## 1.2 Algoritmo de optimización por algoritmos genéticos escrito en Matlab

```

options=gaoptimset('Generations',200,'PopulationSize',400,
'PlotFcns',{[@gaplotbestf,@gaplotstopping],'TolFun',0.1});

%Vector de diseño:[H M1 M2 M3 M5 M6 TF] %LB=[0.75 0.085 0.250
0.300 0.150 0.300 1];
% UB=[1.7 0.150 0.380 0.380 0.220 0.380 3];
% [ H      M1      M2      M3      M4      M5      M6      M12     TF]
LB=[0.66 0.220 0.250 0.320 0.150 0.220 0.380 0.100 1];
UB=[1.7 0.280 0.380 0.380 0.200 0.280 0.400 0.200 3];
% IntCont = [7]; IntCont = [9];
[x,fval,exitflag]=ga(@obj_bipedo,9,[],[],[],[],LB,UB,
@nonlcon_bipedo,IntCont,options);
close all
obj_bipedo_plot(x);
TOB = 0.039*x(1)
L1 = 0.285*x(1)-TOB
L2 = 0.530*x(1)-TOB-L1 L3 = 0.191*x(1)
L4 = L2
L5 = L1

```

## Apéndice 1. Registro de proyecto



Universidad Veracruzana  
 Dirección General de Vinculación  
 Departamento de Fomento a la Vinculación  
 Sistema de Información para la Vinculación Universitaria

---

### REPORTE DE PROYECTO

29/11/2017 13:14:55

---

**CAMPUS:** VERACRUZ  
**AREA ACADEMICA:** TECNICA  
**FACULTAD DE INGENIERIA**

#### DATOS GENERALES DE LA ENTIDAD O DEPENDENCIA

**TIPO DE ENTIDAD:** AC  
**DIRECCION:** CZA ADOLFO RUIZ CORTINEZ 455

#### 1. DATOS GENERALES DEL PROYECTO

**PROPOSITO ESENCIAL:** LA IMPLEMENTACION SISTEMATICA DE ACCIONES PARA PROPONER ALTERNATIVAS DE SOLUCION A LOS PROBLEMAS DEL ENTORNO MEDIANTE LA INVESTIGACION, LA INNOVACION Y EL DESARROLLO TECNOLOGICO.

**TIPO DE PROYECTO:** DISEÑO, INNOVACION O DESARROLLO DOCUMENTAL

**NATURALEZA DEL PROYECTO:** DISCIPLINARIO

**CON LA PARTICIPACION DE:** FACULTAD DE INGENIERIA / REGION VERACRUZ

**AMBITO:** ESTATAL

**TEMPORALIDAD:** PERMANENTE

**FECHA DE INICIO:** 1/1/2017

**FECHA DE TERMINO:** INDEFINIDA

**SECTORES BENEFICIADOS:** CIENCIA Y TECNOLOGIA

#### 2. DATOS ESPECIFICOS DEL PROYECTO

##### PLANEACION

**TITULO DEL PROYECTO:** METODOLOGIA INTEGRAL PARA EL DISEÑO DE UN SISTEMA ROBÓTICO DE LOCOMOCIÓN BÍPEDA  
**JUSTIFICACION:** EN LA CONSTRUCCIÓN DE PROTOTIPOS PARA FINES ACADÉMICOS E INCLUSO DE APLICACIÓN ESPECÍFICA EXTRA, NORMALMENTE SE SIGUEN PROCEDIMIENTOS NO SUJETOS A METODOLOGÍA INTEGRALES, DE TAL MODO QUE CONVENCIONALMENTE SE PRESENTAN DIVERSOS ESCENARIOS DE AJUSTES PARA LLEGAR A UN PRODUCTO FINAL, INCLUSO CUANDO SE DISPONE DE UNA PLATAFORMA FÍSICA APARENTEMENTE FUNCIONAL. LA NECESIDAD DE UNA METODOLOGÍA FORMAL QUE EVITE RETROCESOS EN EL PROCESO DE DISEÑO Y CONSTRUCCIÓN ES RAZÓN PARA DEFINIR UNA SECUENCIA DE PASOS QUE GARANTICEN UN PROTOTIPO FUNCIONAL Y EFICIENTE. EN ESTE TRABAJO SE CONSIDERA EL PLANTEAMIENTO DE UNA METODOLOGÍA PARA EL DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE LOCOMOCIÓN BÍPEDA PAR FINES ACADÉMICOS, LA CUAL INCORPORA TEMÁTICAS QUE CUBREN SOBRE TODO LA PARTE CONCEPTUAL Y DE DETALLE DE UN MODO VIRTUAL, ANTES DE PROCEDER, COMO PARTE FINAL, A LA CONSTRUCCIÓN.

**OBJETIVO GENERAL:** DISEÑO Y CONSTRUCCIÓN MEDIANTE EL USO DE UNA METODOLOGÍA MECATRÓNICA, DE UN PROTOTIPO FUNCIONAL EXPERIMENTAL DE ROBOT BÍPEDO.

##### OBJETIVOS ESPECIFICOS:

Figura 7.1:

## Apéndice 2. Apoyos recibidos durante la maestría



CONSEJO NACIONAL DE CIENCIA Y TECNOLOGÍA  
DIRECCIÓN ADJUNTA DE POSGRADO Y BECAS  
DIRECCIÓN DE BECAS

México D.F. a 05 de septiembre de 2016

### A QUIEN CORRESPONDA:

PRESENTE

Por medio de la presente hacemos constar que el Consejo Nacional de Ciencia y Tecnología, CONACYT, ha otorgado una beca a el ( la ) C. **JACOBO TORRES FIGUEROA** con número ( CVU/Becario ): **733851/605894**, a partir de la fecha **01 de agosto de 2016** hasta el **31 de enero de 2018** \*, para realizar sus estudios de **MAESTRÍA** en el programa **MAESTRÍA EN INGENIERÍA APLICADA** en UNIVERSIDAD VERACRUZANA.

#### La beca incluye:

Manutención Mensual 4.5 veces el Salario Mínimo Vigente en el Distrito Federal, equivalente a \$ 9,860.40 a la fecha de expedición de este documento

Servicio Médico proporcionado por el ISSSTE

Extendemos esta carta a petición de el(la) interesado(a) para los fines que considere pertinentes.

ATENTAMENTE



Mtro. Pablo Rojo Calzada  
DIRECTOR DE BECAS.

Esta Carta de Asignación de Beca (Anexo uno) forma parte del Convenio de Asignación de Beca celebrado en la misma fecha de emisión.

Figura 7.2:

## Apéndice 3. Publicaciones durante la maestría

### 3.1 Diseño de controlador digital para servomecanismo de seguimiento solar para aplicaciones en celdas fotovoltaicas

#### DISEÑO DE CONTROLADOR DIGITAL PARA SERVOMECANISMO DE SEGUIMIENTO SOLAR PARA APLICACIONES EN CELDAS FOTOVOLTAICAS

Ing. Jacobo Torres Figueroa<sup>1</sup>, Ing. Ulises Gabriel García<sup>2</sup>,  
Dr. José Alejandro Vásquez Santacruz<sup>3</sup>, Dr. Agustín Leobardo Herrera May<sup>4</sup>, Dr. Luis Felipe Marín Urias<sup>5</sup>

**Resumen**—Cada día el uso racional de las energías renovables cobra mayor interés, entre ellas destacan los sistemas de captación solar para producir energía eléctrica utilizando paneles fotovoltaicos. En éste trabajo se realiza el diseño de un controlador digital proporcional-integral-derivativo (PID) implementado en una tarjeta Arduino MEGA 2560, para el control de un servomecanismo que oriente a una celda fotovoltaica y realice el seguimiento solar en dos grados de libertad. Las señales de referencia son generadas mediante un algoritmo en Matlab que calcula la posición del sol (azimut y elevación) en función de la fecha, hora, latitud y longitud de un lugar específico y enviadas a Arduino vía puerto serial. También se implementa una interfaz para el graficado de la potencia, corriente y voltaje en el panel solar, dichas señales son adquiridas en Arduino y enviadas a Matlab vía puerto serial. Se evalúa la potencia del diseño propuesto y se compara con un sistema estático, resaltando una mejoría en la potencia de salida, y sugiriendo profundizar en el estudio de un modelo dinámico para tal fin.

**Palabras clave**— Celdas fotovoltaicas, Servomecanismo, PID, Arduino, Matlab.

#### Introducción

En la actualidad hay un incremento en la utilización de energías renovables, por sus ventajas respecto de las energías no renovables más utilizadas como los combustibles fósiles, que inciden en el impacto ambiental y la independencia energética. Dentro de las energías renovables, los sistemas de captación solar son una alternativa viable ya sea para sistemas de calefacción o para producir energía eléctrica, lo cual es el caso de los paneles fotovoltaicos y sistemas eólicos. En los sistemas de paneles fotovoltaicos cuyo funcionamiento se basa en las propiedades fotosensibles de algunos semiconductores (germanio o silicio previamente tratados), la incidencia de la radiación solar sobre dichos materiales induce una corriente eléctrica directa que se traduce en potencia eléctrica (Reda et al 2008). Sin embargo la generación de la energía está ligada al ángulo de incidencia sobre la superficie del panel solar, es decir mientras más perpendicular se encuentre el panel solar respecto de la radiación, la producción de energía será mayor (ver Figura 1).

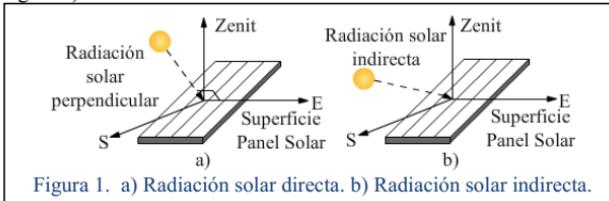


Figura 1. a) Radiación solar directa. b) Radiación solar indirecta.

Para compensar dicho inconveniente, el panel se fija a una estructura con una posición determinada, que maximice la incidencia de la radiación a lo largo del recorrido del sol. Otro modo es que la estructura sea capaz de orientar al panel para que éste siempre se encuentre en posición perpendicular a la radiación solar, consiguiendo generar el máximo de energía posible. Además, sabiendo que la eficiencia que en los paneles solares ronda el 15%, el caso de seguimiento puede abordarse ya sea utilizando un sistema sensorizado (Lazarou et al 2015) o bien utilizar

<sup>1</sup> El Ing. Jacobo Torres Figueroa es estudiante de maestría en la Universidad Veracruzana, Boca del río, Veracruz. México  
lodo.orbe@gmail.com

<sup>2</sup> El Ing. Ulises Gabriel García es estudiante de maestría en la Universidad Veracruzana, Boca del río, Veracruz. México.  
ulises3@hotmail.com

<sup>3</sup> El Dr. José Alejandro Vásquez Santacruz es profesor investigador en la Universidad Veracruzana, Boca del río, Veracruz. México. alejanvasquez@uv.mx

<sup>4</sup> El Dr. Agustín Leobardo Herrera May es profesor investigador en la Universidad Veracruzana, Boca del río, Veracruz. México. leherrera@uv.mx

<sup>5</sup> El Dr. Luis Felipe Marín Urias es profesor investigador en la Universidad Veracruzana, Boca del río, Veracruz. México. luismarin@uv.mx

una ecuación que prediga la posición del sol a lo largo del día (Pintos 2010). En este trabajo se aborda una metodología que contempla un sistema de control para el seguimiento solar, obteniéndose una mejora en la potencia de salida del panel, con ecuaciones predictivas propuestas por Cooper (1969) y Spencer (1971) para la obtención de los ángulos de azimut y la elevación que determinan la posición del sol respecto de un sistema de coordenadas fijo, cuyo origen es la ubicación del lugar geográfico específico (Guo L. et al 2013), dichas magnitudes son calculadas con base en parámetros astronómicos que se describen en la siguiente sección.

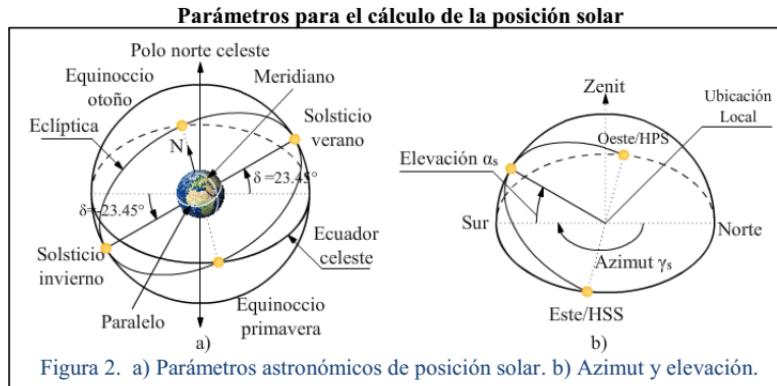


Figura 2. a) Parámetros astronómicos de posición solar. b) Azimut y elevación.

**Ángulo de declinación de la tierra.** Es el ángulo  $\delta$  que se crea entre el plano de la eclíptica (plano de la órbita terrestre alrededor del sol) y el plano ecuatorial de la tierra (ver Figura 2a). Su valor oscila entre  $23.45^\circ$  en el solsticio de verano y  $-23.45^\circ$  en el solsticio de invierno. Durante los equinoccios de primavera y otoño el valor de  $\delta$  es  $0^\circ$ . La ecuación (1) describe el cálculo de  $\delta$ , siendo  $n$  el día transcurrido del año.

$$\delta = 23.45 \times \operatorname{sen}\left(\frac{360 \times 284 + n}{365}\right) \quad (1)$$

**Latitud y longitud.** La latitud es la distancia angular, al norte o sur del ecuador, desde un punto cualquiera (paralelo). La longitud es la distancia angular desde un punto cualquiera con referencia al meridiano cero, Greenwich,  $\lambda_{ms}$  (ver Figura 2a) ambas son representadas por líneas imaginarias horizontales y verticales respectivamente.

**Ecuación del tiempo.** La ecuación del tiempo (ET) (Spencer 1971), cuantifica la diferencia entre la hora solar (HS) y la hora civil (HC) medida en minutos y está descrita por la ecuación (2) en términos de la ecuación (3).

$$ET = 229.2(0.000075 + 0.001868 \cos(\beta) - 0.032077 \operatorname{sen}(\beta) - 0.014615 \cos(2\beta) - 0.04089 \operatorname{sen}(2\beta)) \quad (2)$$

$$\beta = \frac{(n-1) \times 360}{365} \quad (3)$$

Donde  $\beta$  es el ángulo descrito por la tierra sobre la órbita eclíptica y  $n$  el número de día del año.

**Hora solar.** La hora solar (HS) es una medida del tiempo en función del movimiento aparente del sol a lo largo del plano horizontal de un lugar en específico. Se obtiene HS a partir de la ecuación (2), el valor de  $\lambda_{ms}$ , la longitud del meridiano del lugar en estudio  $\lambda_{mlc}$  y la hora civil (HC), basándose en la ecuación (4).

$$HS = HC + ET + 4 \times (\lambda_{ms} - \lambda_{mlc}) \quad (4) \qquad \omega = HS - 12 \times 15^\circ \quad (5)$$

**Ángulo horario.** El ángulo horario  $\omega$  está definido como el ángulo que se forma a partir del punto zenit (medio día solar) y la posición del sol en cualquier instante de la hora solar. Este parámetro toma valores de  $15^\circ$  por cada hora, siendo su valor es de  $0^\circ$  cuando es el medio día solar, positivo antes y negativo después de dicho momento; siendo determinado por la ecuación (5).

**Horas luz.** Depende del valor de  $\delta$  y de  $\varphi$  del lugar en estudio. Para obtener las horas luz (HL) se hace uso de la ecuación (6). Por otra parte la hora de salida (HSS) y la hora de puesta del sol (HPS) se pueden determinar mediante el valor de HL, haciendo uso de las ecuaciones (7) y (8) respectivamente.

$$HL = \frac{2}{15} \operatorname{cos}^{-1}(-\operatorname{tan}(\varphi) \operatorname{cos}(\delta)) \quad (6), \quad HSS = 12 - \frac{HL}{2} \quad (7), \quad HPS = 12 + \frac{HL}{2} \quad (8)$$

**Azimut y elevación.** Determinan la posición del sol a lo largo del día: ángulo de elevación solar  $\alpha_s$  y el ángulo de azimut solar  $\gamma_s$  (ver Figura 2b), están en función de la ubicación, la fecha y hora del lugar específico, dichos parámetros están determinados por las ecuaciones (9) y (10).

$$\operatorname{sen}(\alpha_s) = \operatorname{cos}(\varphi) \operatorname{cos}(\delta) \operatorname{cos}(\omega) + \operatorname{sen}(\varphi) \operatorname{sen}(\delta) \quad (9)$$

$$\gamma_s = \text{sign}(\omega) \left[ \cos^{-1} \left( \frac{\cos(\theta_z) \sin(\varphi) - \sin(\delta)}{\cos(\theta_z) \cos(\varphi)} \right) \right] \quad (10), \quad \text{donde } \theta_z \text{ es el ángulo zenith}$$

### Diseño de sistema de control

#### Diseño de los controladores

Para operar la celda fotovoltaica de forma dinámica y así obtener una mejora en la cantidad de potencia obtenida con respecto a un sistema estático (Lazarou et al 2015), se agrega un sistema de control de lazo cerrado con el fin de gobernar el servomecanismo (ver Figura 3) respecto de la posición del sol, donde las señales de referencia son el azimut y la elevación. La ley de control utilizada fue el controlador proporcional-integral-derivativo (PID), el cual se basa en la ecuación (10) en el dominio del tiempo. Donde la compensación proporcional  $K_p$  ayuda a la respuesta del sistema a llegar a la referencia con mayor rapidez, la compensación integral  $K_i$  ayuda a disminuir el error en estado estable, y finalmente se atenúa el sobre impulso mediante la acción derivativa  $K_d$ . Debido a que se trata de un sistema digital (Ogata 2002), la ley de control, ecuación (10), se reescribe en tiempo discreto como se muestra en la ecuación (11).

$$y(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (10)$$

donde  $K_p$ : ganancia proporcional,  $K_i$ : ganancia integral,  $K_d$ : ganancia derivativa.

$$U(z) = K_p \left[ 1 + \frac{T}{T_i(1-z^{-1})} + T_d \frac{(1-z^{-1})}{T} \right] E(z) \quad (11)$$

donde  $T$ : periodo de muestreo,  $T_i$ : tiempo integral y  $T_d$ : tiempo derivativo.

Además se utiliza el método experimental propuesto por Ziegler-Nichols (Ogata 2002) para la sintonización del controlador, el cual se basa en parámetros obtenidos de la respuesta del sistema ante una entrada escalón, para calcular  $K_p$ ,  $K_i$  y  $K_d$ , haciendo usos de las ecuaciones (12) (13) (14).

$$K_p = 1.2K_0 \quad (12), \quad K_i = \frac{K_p}{T_i} \quad (13), \quad K_d = K_p T_d \quad (14), \text{ utilizando:}$$

$$T_i = 2T_1, \quad T_d = 0.5T_1, \quad K_0 = \frac{(\Delta x)(T_2)}{(\Delta y)(T_1)}$$

donde  $K_0$ : Constante del sistema,  $T_1$ : Tiempo muerto,  $T_2$ : Tiempo de subida,  $\Delta x$ : variación de la señal escalón,  $\Delta y$ : variación de la respuesta del sistema (ver Figura 2b). Aplicando el método Ziegler-Nichols a los dos motores de C.D. se obtuvieron las siguientes ganancias del controlador PID:  $K_p = 0.2$ ,  $K_i = 0.35$ ,  $K_d = 0.15$  y se fijó el periodo de muestreo en 10 ms.

#### Implementación del sistema de control

Para la implementación de las leyes de control se utilizó la tarjeta electrónica Arduino Mega 2560, fue elegida como plataforma para el sistema electrónico puesto que es de código abierto y las características de la tarjeta son las adecuadas para los requerimientos del sistema (ver Figura 4a), además de que resulta más económico que utilizar un PLC. Por otra parte los actuadores en el prototipo son dos motores de C.D. los cuales tiene acoplados codificadores ópticos para la medición de la posición angular, 12 ppr de resolución y un sistema de engranajes con un factor de reducción de 64:1, por lo que la resolución de los codificadores ópticos se incrementa a 768 ppr (ver Figura 4b).



Figura 3. Prototipo de seguidor solar.

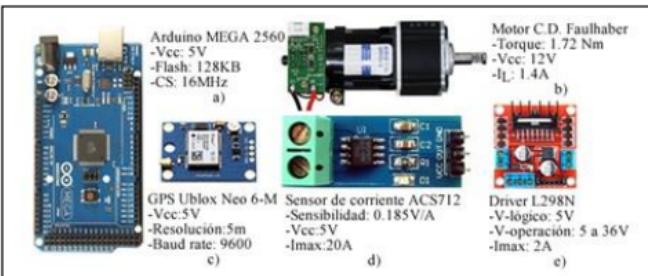


Figura 4. Componentes electrónicos y electromecánicos utilizados.

Existen foto-sensores que, expuestos a la luz del sol, pueden determinar su posición (Lazarou et al 2015), pero se incrementa el costo del sistema. Por lo anterior, se implementa un módulo GPS modelo Ublox Neo-6M (ver Figura 4c), con el cual se obtienen parámetros geográficos como: latitud, longitud y hora de la ubicación geográfica particular, para calcular los parámetros de referencia utilizando las ecuaciones (9) y (10). El módulo GPS es conectado vía puerto USB a una computadora (haciendo uso de un conversor TTL-USB), el cual entrega los datos en formato *National Marine Electronics Association* (NMEA), que es el estándar de la marina norteamericana para

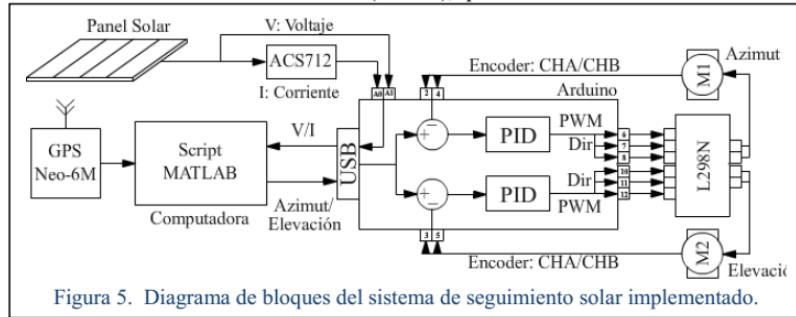


Figura 5. Diagrama de bloques del sistema de seguimiento solar implementado.

la representación de información de geolocalización. Los datos del GPS son leídos mediante un algoritmo implementado en Matlab cada quince minutos, los procesa y obtiene el comando NMEA que contiene la información sobre longitud, altitud y hora local en formato UTC, también calcula las ecuaciones (9) y (10) que generan el ángulo de azimut y elevación, son enviados a Arduino vía puerto serial (USB) y representan la actualización de la posición solar, por lo que Arduino calcula la compensación de error requerida conforme a las leyes de control PID diseñadas y genera dos señales de control codificadas en modulación por ancho de pulso (PWM) que se envían a los actuadores, sin embargo dichas señales no cuentan con la suficiente corriente para accionar a los motores de CD, por lo que se implementó una etapa de potencia basada en un driver para motores, el cual consta de dos puentes H contenidos en el circuito integrado L298N (ver Figuras 4e y 5) y tiene la capacidad de manejar dos cargas de hasta 2A. La potencia en el panel solar utilizado (12 x 12 cm), el cual genera un máximo de 5 V y 300 mA, con una potencia de 1.3W en condiciones ideales, se calculó midiendo el voltaje y la corriente instantáneas. La medición de la corriente se llevó a cabo con el sensor de corriente ACS712 (ver Figuras 4d y 5), el cual se conecta a un puerto analógico de la tarjeta Arduino y a su vez el voltaje es medido haciendo uso de otro puerto analógico al cual se conecta directamente las terminales del panel solar. Con la corriente y el voltaje instantáneo se calcula la potencia y se envían los datos (voltaje, corriente y potencia) vía puerto serial a la computadora a una tasa de transferencia datos de 9600 baudios, donde son recibidos por el algoritmo implementado en Matlab y almacenados para su posterior graficado y análisis. Un diagrama completo del sistema implementado se muestra en la figura 5.

### Resultados experimentales

#### Análisis comparativo de generación de energía

Se realizaron mediciones en dos condiciones diferentes A y B sobre el prototipo, la primera condición (A) es una configuración fija del servomecanismo, cuyos parámetros de azimut y elevación se fijaron en: azimut 0°, es decir con el panel solar orientado hacia el sur, y elevación correspondiente a latitud+10°. La latitud y longitud utilizadas en el experimento son correspondientes a la localidad de San Andrés Tuxtla, Veracruz. Latitud: 18.441389°, longitud: -95.214167°, altitud: 320 msnm. Se midió la potencia instantánea durante el día 21 de junio de 2016, desde

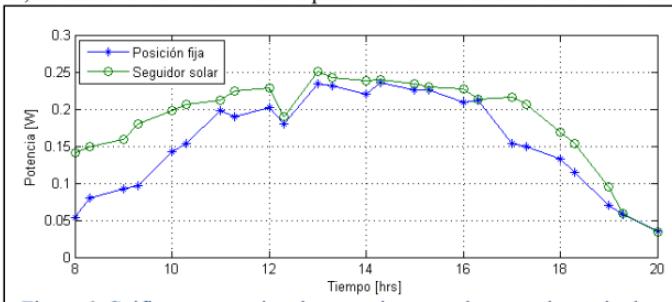


Figura 6. Gráfico comparativo de potencia generada por ambos métodos.

las 8 am, hasta las 8 pm, a intervalos de 30 minutos. Para la configuración B, se puso en marcha el sistema de seguimiento solar, justo después de realizar la medición de potencia medida en la configuración para cada intervalo A, dejando al sistema ubicarse en la posición calculada por la ecuación predictiva, para proceder a realizar nuevamente la medición de la potencia. Se calculó el promedio de ambas mediciones obteniéndose 0.15 W para la configuración fija A y 0.19 W para la dinámica B. Lo anterior implica una mejoría en la salida de potencia de un 22% de la configuración dinámica con respecto a la estática. En la Figura 6 se aprecia el comportamiento de los resultados obtenidos experimentalmente.

Para verificar este supuesto se realizó una prueba Z de comparación de medias considerando las hipótesis:  $H_0: \mu_1 = \mu_2$  y  $H_1: \mu_1 > \mu_2$  y un  $\alpha = 0.05$ , con los datos mostrados en el cuadro 1. Lo que arrojó como resultado con una confianza del 95% el rechazo de la hipótesis  $H_0$  que indica que la producción de energía es mayor cuando se sigue la trayectoria del sol durante el día.

Datos	$\mu$ : Media	$s^2$ : Varianza	n
Seguidor solar	$\mu_1: 0.1888$	0.0033	25
Posición fija	$\mu_2: 0.1566$	0.0043	25

$Z_0 = 1.84 > Z_{0.05} = 1.64$

Cuadro 1. Parámetros estadísticos utilizados en la comparación de medias.

*Verificación de los algoritmos de control*

Se realizaron pruebas de seguimiento de trayectoria para verificar el funcionamiento de los algoritmos de control, con los datos de posición solar (azimut y elevación) se eligieron valores al azar para introducirlos como señal de referencia y enviarlos a la tarjeta Arduino, para que se generaran las acciones de control.

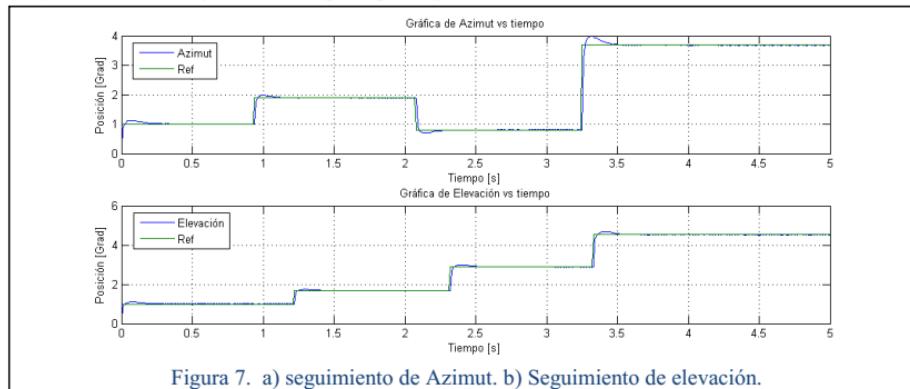


Figura 7. a) seguimiento de Azimut. b) Seguimiento de elevación.

La figura 7 muestra los resultados obtenidos al probar el sistema de control tanto del posicionamiento de azimut como elevación. Se observa que los algoritmos de control si cumplieron con el objetivo de seguimiento. Además se aprecia un sobre impulso en ambas señales, esto es debido a la inercia generada por el mecanismo, donde en el ángulo de azimut se genera más puesto que el actuador carga tiene que mover más masa, caso contrario en el actuador para la elevación.

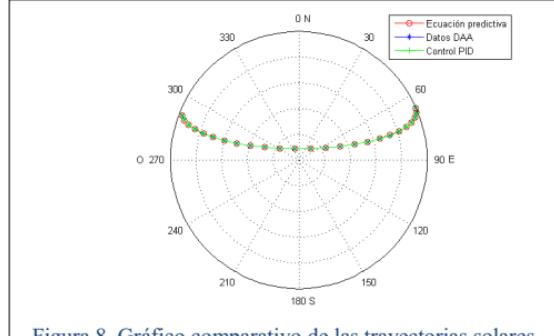


Figura 8. Gráfico comparativo de las trayectorias solares.

**Prueba de precisión**

Mediante un algoritmo en Matlab se obtuvieron las posiciones de azimut y elevación durante el día 21 de junio de 2016, desde las 8 am hasta las 8 pm en intervalos de 30 min, los resultados obtenidos se compararon con datos obtenidos del Departamento de Aplicaciones Astronómicas (DAA), de la Marina de los Estados Unidos, los datos calculados por las ecuaciones predictivas utilizadas y además con los resultados de posición calculados por los algoritmos de posición PID. En la figura 7 se muestra la gráfica de los datos superpuestos.

De la figura 8 se observa que las tres trayectoria de recorrido del sol el día 21 de junio de 2016 generados por: DAA, ecuación predictiva y PID, presentan un error de una respecto de otra de alrededor de  $\pm 1^\circ$ , lo que es suficiente para un buen desempeño en un sistema de seguimiento solar, puesto que una desviación de tal magnitud no es lo suficientemente significativa para marcar una diferencia en la generación de energía.

**Comentarios Finales****Conclusiones**

Los resultados demuestran que se presenta una mejoría de un 22%, en la producción de energía eléctrica, cuando el sistema de seguimiento solar está activo, sugiriendo así que una incidencia de radiación perpendicular mejora el desempeño del sistema de captación solar. El controlador PID sintonizado bajo el método experimental propuesto por Ziegler-Nichols, dio buenos resultados en cuanto a compensación de error, arrojando diferencias de  $\pm 1^\circ$  respecto de la referencia.

La investigación demuestra que es posible construir un sistema básico de seguimiento solar con componentes de bajo costo, lo que pondría a un mayor alcance la construcción de sistemas de seguimiento solar a una mayor parte de la población, lo que se traduciría en una mayor eficiencia en los sistemas de producción de energía eléctrica por medio de paneles solares domésticos y un ahorro económico considerable.

**Recomendaciones**

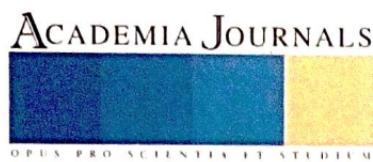
Los investigadores interesados en continuar nuestra investigación podrían abaratar el sistema no utilizando software de terceros como Matlab, en vez de ello se puede hacer una migración de código en Matlab a otros lenguajes orientados a objetos como Python, C# o Java por citar a algunos, y evitar el uso de una computadora personal de escritorio o *lap-top* mediante el uso de otros sistemas computacionales de menor tamaño y consumo energético como lo es la computadora *Raspberry Pi* cuyas ventajas es principalmente la filosofía *open-source* y la comunidad en internet que lo respalda, además de correr bajo sistemas operativos basados en Linux, lo cual se complementa muy bien con la filosofía de los sistemas Arduino. Además existen tecnologías como los sensores de radiación solar, que en combinación de un panel solar podría fungir como parámetro de referencia para el estudio de mejoramiento de la producción de energía eléctrica, pero que eleva los costos en equipamiento por lo que se tendría que hacer un estudio económico al respecto.

**Referencias**

- Reda, I., Andreas, A. "Solar position algorithm for solar radiation applications," Solar Energy, Vol. 76, No. 5, 2008, pp.577-589.
- Lazarou G. C., Longo M., Roscia M., Pagano M. "Comparative analysis of fixed and sun tracking low power PV systems considering energy consumption," Energy Conversion and Management, Vol 92, No. 1, 2015, pp 143-148.
- Pintos L. C. "Seguidor Solar Ligero para Cubiertas". Universidad Pontificia Comillas, 2010.
- Cooper, P.I "The absorption of radiation in solar stills". Solar Energy, Vol. 12, No. 15, 1969, pp.333-346.
- Spencer, J. W. "Fourier series representation of the position of the Sun", Vol. 2, No. 5, 1971, pp.172.
- Tiwari A., Vora M., Shewate M., Waghmare V. "Sun Tracking Solar Panel with Maximum Power Point Tracking," IJESC, Vol. 5, 2016, pp.2451-2454.
- Crutchik N. M. "Dispositivo de Seguimiento Solar Dedicado". Universidad del Norte Antofagasta, Chile. 1985
- Guo L., Ham J., W. Otieno A. "Design and Simulation of a Sun Tracking Solar Power System". 120th ASEE Annual Conference & Exposition, 2013.
- Ogata K. "Sistemas de control en tiempo discreto". Prentice Hall, Hispanoamericana, 2002.
- NMEA. National Marine Electronics Association. Consultado el 10 de junio de 2016. Disponible en: <http://www.nmea.org/>

### 3.2 Evidencia de participación en el congreso

Congreso Internacional de Investigación Academia Journals en Tecnologías Estratégicas Colima 2016



## Certificado

OTORGADO A

Ing. Jacobo Torres Figueroa

Ing. Ulises Gabriel García

Dr. Jose Alejandro Vasquez Santacruz

Dr. Agustín Leobardo Herrera May

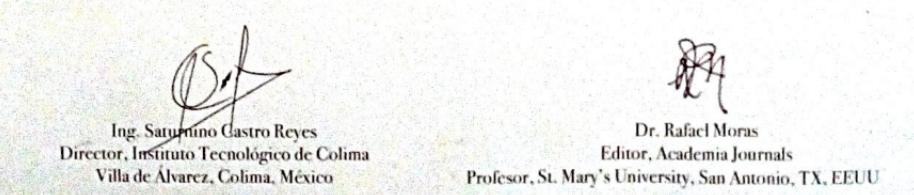
Dr. Luis Felipe Marin Urias

POR SU ARTÍCULO INTITULADO

Diseño de controlador digital para servomecanismo de seguimiento solar para aplicaciones en celdas fotovoltaicas

(Artículo Col130)

El artículo fue presentado en el congreso desarrollado los días 3 al 5 de octubre de 2016 en las instalaciones del Instituto Tecnológico de Colima, Villa de Álvarez, Colima, México, y publicado en el portal de Internet Academiacjournals.com (ISSN 2380-503X)  
y en el libro CDROM *Compendio de Investigación en Tecnologías Estratégicas Colima 2016*  
(ISBN 978-1-939982-25-4 con código de barras)



### 3.3 Estudio hidrodinámico de brazo manipulador de dos grados de libertad para robot submarino

#### ESTUDIO HIDRODINÁMICO DE BRAZO MANIPULADOR DE DOS GRADOS DE LIBERTAD PARA ROBOT SUBMARINO

Ing. Jacobo Torres Figueroa<sup>1</sup>, Dr. José Alejandro Vásquez Santacruz<sup>2</sup>, Dr. Marco Osvaldo Vigueras Zúñiga<sup>3</sup>

**Resumen**—En este trabajo se presenta el análisis hidrodinámico para el cálculo de las fuerzas de arrastre presentes en los eslabones de un brazo manipulador de dos grados de libertad, con aplicaciones en robótica submarina, para determinar los torques articulares máximos presentes en configuraciones de posición críticas del manipulador y realizar una selección adecuada de los actuadores electromecánicos ante condiciones específicas determinadas por el fluido (agua). Se obtienen los modelos dinámicos del manipulador y resultados analíticos de fuerzas de arrastre, verificados por medio de CFD, para lo cual se realiza un análisis de diseño de malla con forme a la teoría de  $y^+$  y capa límite. También se presenta una simulación del manipulador en Matlab Simulink como medio de validación.

**Palabras clave**—ROV, Fuerza de arrastre, Brazo manipulador.

##### Introducción

El constante avance tecnológico sucedió desde un principio y ha generado grandes cambios a nivel social y demográfico, entre estos avances, uno de los más conocidos y esperados es la robótica, la cual se encarga de diseñar, construir y operar toda clase de robots, los cuales constan de una entidad virtual y otra mecánica, y se encargan de realizar distintos procesos o actividades de manera automatizada, necesitando para ello la acción de un operador el cual le indica sus tareas o bien; ajusta las variables requeridas para que éste tenga el menor margen de error posible (Barrientos A. et al, 2007).



Figura 1. a) ROV Hércules a punto de ser sumergido en el océano. b) Brazo manipulador acoplado al ROV Magnum Plus.

Hoy en día el desarrollo de la robótica se encuentra enfocado en la creación de Robots Inteligentes, siendo éstos capaces de operar por sí mismos, donde la participación del Ser Humano sea únicamente para control, reportes y detección de fallas o anomalías en el sistema. Actualmente, existen también ciertos sectores donde aún no es preciso conseguir una elevada productividad, son lugares donde las tareas a realizar no son repetitivas y no existe un conocimiento detallado del entorno, es aquí donde la aplicación de la robótica tiene una gran oportunidad, debido a que son entornos peligrosos, caracterizados por su falta de estructuración en los que no es posible el acceso a personas, tal es el caso del mar, donde existe la necesidad de usar equipos especialmente diseñados para este medio; a partir de la detección de éstas necesidades surge la Robótica Submarina, enfocada en el diseño de Vehículos operados a distancia Subacuáticos (ROV por sus siglas en inglés) (ver figura 1a) (Rafael K y Santibáñez V, 2003).

Los ROV subacuáticos generalmente cuentan con brazos manipuladores en su exterior (ver figura 1b), los cuales le ayudan a interactuar con las estructuras externas al mismo, este tipo de brazos robóticos manipuladores deben de ser capaces de resistir las altas presiones que se generan al estar sumergidos en el agua, además de ser poder

<sup>1</sup> El Ing. Jacobo Torres Figueroa es estudiante de la maestría en ingeniería aplicada de la Universidad Veracruzana Campus Boca del Río, Ver. [jodo.orbe@gmail.com](mailto:jodo.orbe@gmail.com)

<sup>2</sup> El Dr. José Alejandro Vásquez Santacruz es catedrático de la maestría en ingeniería aplicada de la Universidad Veracruzana Campus Boca del Río, Ver. [alejanvasquez@uv.mx](mailto:alejanvasquez@uv.mx)

<sup>3</sup> El Dr. Marco Osvaldo Vigueras Zúñiga es catedrático y coordinador de la maestría en ingeniería aplicada de la Universidad Veracruzana Campus Boca del Río, Ver. [vzmo@hotmail.uk](mailto:vzmo@hotmail.uk)

contrarrestar los momentos articulares provocados por los efectos dinámicos del agua, dichos efectos están directamente relacionados con un problema hidrodinámico (White F, 2004). Es por ello que en el presente proyecto se realizó el estudio y análisis de la dinámica de fluidos cuando incide sobre la superficie de un brazo robótico, empleado en aplicaciones submarinas, con el objetivo de caracterizar la fuerza de arrastre producida y efectuar una selección adecuada de actuador articular capaz de contrarrestar esta fuerza.

#### Metodología

Para realizar el presente proyecto se propuso de manera simplificada un brazo robótico utilizado en aplicaciones submarinas de dos grados de libertad, esto con el fin de llevar a cabo el análisis del comportamiento de la dinámica de fluidos de manera computacional y analítica, para lo cual el diseño presenta las siguientes medidas: [1.25 m x 0.20 m x 0.15 m], así también, en la figura 2a podemos apreciar la geometría del modelo propuesto.

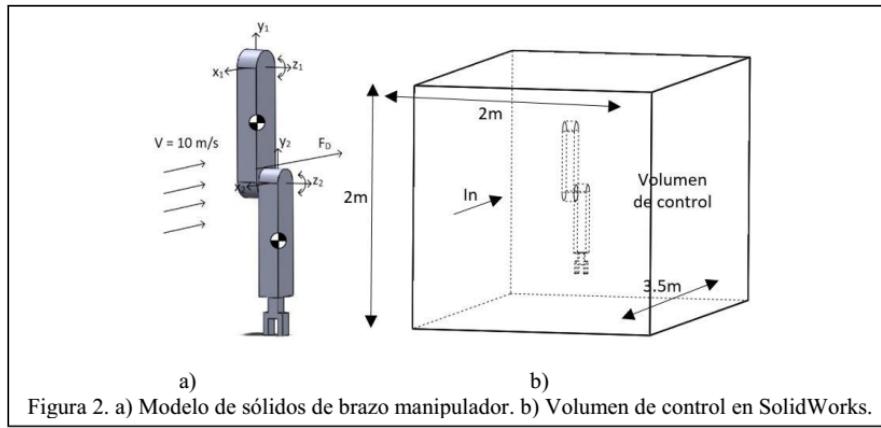


Figura 2. a) Modelo de sólidos de brazo manipulador. b) Volumen de control en SolidWorks.

La geometría fue realizada mediante el uso del software SolidWorks, el cual es una herramienta de diseño asistido por computadora para modelado mecánico en 3D. Habiendo realizado la geometría se procedió al cálculo del espesor de la capa límite donde fue necesario calcular el tipo de régimen del flujo, conociendo la longitud característica del brazo y la velocidad de entrada del fluido, también se resolvió el valor de  $y^+$  el cual es requerido para calcular la distancia entre la superficie y el primer nodo de la capa límite. Partiendo de éstos resultados se procedió a generar el mallado de la geometría, considerando el espesor de capa límite (ver figura 3a). En la figura 3 podemos ver el mallado generado para el cual se hizo uso del software ANSYS, el cual es un ecosistema de programas para diseño, análisis y simulación de partes por elementos finitos. Como podemos apreciar en la figura 3a, se optó por un tipo de mallado fino.

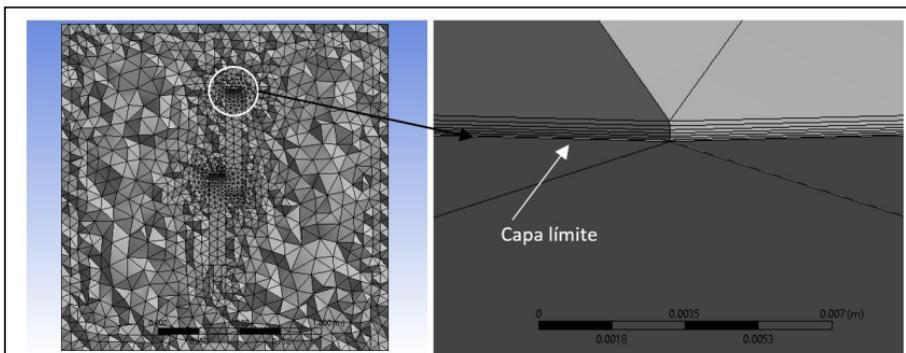


Figura 3. a) Vista de corte del mallado generado en ANSYS CFX. b) Detalle de la capa límite, calculada conforme a la teoría  $v^+$  con los valores del cuadro I.

Previo a realizar la simulación, se establecieron también las condiciones de frontera para nuestro caso de estudio, las cuales fueron establecidas como: velocidad de entrada = 10 m/s, presión de salida = 0 y paredes. Las ecuaciones (1), (2), (3) y (4) para resolver los cálculos de la sensibilidad de malla. Para conocer el número de Reynolds el cual determina el régimen del flujo (1), para determinar el espesor de la capa límite (2), cálculo de  $y^+$  (3), y para determinar la distancia del primer nodo a la superficie (4). Para conocer la fuerza de arrastre se utilizó (5) (White F, 2004).

$$Re = \frac{(\rho)(V)(Lc)}{\mu} \quad (1), \quad \delta = \frac{5x}{\sqrt[2]{Re}} \quad (2), \quad y^+ = \frac{U}{U^*} \quad (3), \quad \Delta s = \frac{y^+ \mu}{\rho U^*} \quad (4), \quad F_D = \frac{1}{2} C_D \rho U_\infty^2 A_D \quad (5)$$

Por otra parte el modelo dinámico del manipulador está determinado por las ecuaciones (6) y (7), donde  $\tau_1$  y  $\tau_2$  son los torques articulares requeridos para mover al brazo en el caso de que no esté diseñado con flotabilidad neutra.

$$\begin{aligned} \tau_1 &= [m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 + 2m_2 l_1 l_{c2} \cos(q_2) + I_1 + I_2] \ddot{q}_1 + [m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2] \ddot{q}_2 \\ &\quad - 2m_2 l_1 l_{c2} \operatorname{sen}(q_2) \dot{q}_1 \dot{q}_2 - m_2 l_1 l_{c2} \operatorname{sen}(q_2) \dot{q}_2^2 + [m_1 l_{c1} + m_2 l_1] g \operatorname{sen}(q_1) \\ &\quad + m_2 g l_{c1} \operatorname{sen}(q_1 + q_2), \quad (1) \end{aligned}$$

$$\tau_2 = [m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2] \ddot{q}_1 + [m_2 l_{c2}^2 + I_2] \ddot{q}_2 + m_2 l_1 l_{c2} \operatorname{sen}(q_2) \dot{q}_1^2 + m_2 g l_{c2} \operatorname{sen}(q_1 + q_2) \quad (2)$$

Si el robot se diseña con flotabilidad neutra entonces los términos de pares gravitacionales:  $[m_1 l_{c1} + m_2 l_1] g \operatorname{sen}(q_1) + m_2 g l_{c1} \operatorname{sen}(q_1 + q_2)$  de (6) y  $m_2 g l_{c2} \operatorname{sen}(q_1 + q_2)$  (7) se desprecian, con lo que el sistema se simplifica. El objetivo de este trabajo determinar el torque adicional en  $\tau_1$  y  $\tau_2$  provocadas por la fuerza de arrastre máximo, es decir cuando el brazo manipulador se encuentra extendido con flotabilidad no neutra.

A continuación se procede a realizar la simulación en CFX para conocer la fuerza de arrastre provocada por el fluido sobre el brazo manipulador, la cual es la fricción entre el objeto y el fluido opuesta al movimiento de éste. La velocidad de entrada del fluido fue de 10m/s, se ocupó la densidad del agua a 25[°C] de 1000[m^3/s], un coeficiente de arrastre  $C_D$  de 0.0167993, y el modelo de turbulencia utilizado fue el SST (Shear Stress Transport). Así también se llevaron a cabo soluciones analíticas, para conocer la fuerza de arrastre.

### Comentarios finales

#### Resumen de resultados

El análisis para la determinación del  $\Delta s$ , arrojaron los resultados mostrados en el cuadro 1 para su obtención se utilizaron las ecuaciones (1), (2), (3) y (4).

Parámetros	Magnitud
Re (Reynolds)	3.87E+06
$\delta$ (Blasius)	6.32E-03
Cf (Coef. Fricción)	0.002985366
Tw (Esfuerzo en la pared)	149.2683242
U* (Velocidad de fricción)	0.386352591
$y^+$	25.88309292
$\Delta s$	5.96242E-05

Cuadro 1. Valores obtenido para el cálculo de  $\Delta s$ .

Los resultados analíticos para la fuerza de arrastre considerando un coeficiente de arrastre de  $C_D = 0.0167993$  se aprecian en (8).

$$F_D = \frac{1}{2} C_D \rho U_\infty^2 A_D = \frac{1}{2} (0.0167993)(1000)(10^2)(0.11) = 92.3961 \text{ [N]} \quad (8)$$

Por tanto el momento generado en la articulación  $q_1$  corresponde al producto vectorial (en este caso es escalar) de la distancia al CoM del manipulador respecto del eje de  $q_1$ :

$$M_{q1} = (92.39 \text{ N})(0.52\text{m}) = 48.045998 \text{ [Nm].} \quad (9)$$

De (9) se deduce que el actuador en la articulación  $q_1$  tiene que ser capaz de otorgar el máximo de (6) más el momento generado por la fuerza de arrastre:  $\max(6) + M_{q1}$ , sin considerar la fuerza de carga (PL) y el factor de seguridad FS, por tanto el momento requerido total para  $q_1$  se aprecia en (10).

$$\tau_1 = (\max(6) + M_{q1} + PL) * FS \quad (10)$$

Los resultados de simulación ofrecieron valores similares a los analíticos, para la fuerza de arrastre se utilizó la expresión mostrada en la figura 4, donde además se observa el resultado calculado por CFX para el coeficiente de arrastre y la fuerza de arrastre.

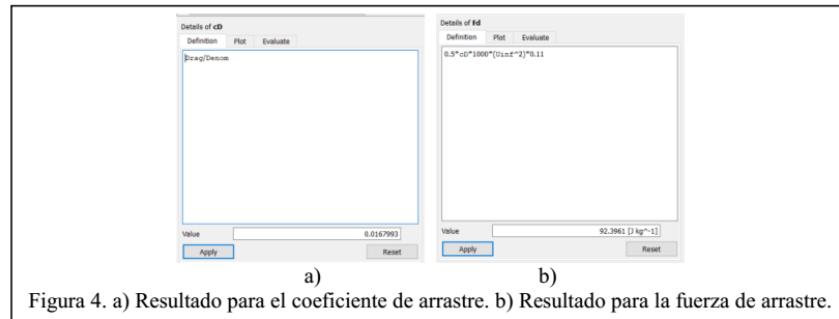


Figura 4. a) Resultado para el coeficiente de arrastre. b) Resultado para la fuerza de arrastre.

De la figura 4 se observa que CFX calculó el coeficiente de arrastre en: 0.0167993, y la fuerza de arrastre en: 92.3961 [N]. Por otra parte en la figura 5 se observan los resultados de presión y velocidades de la simulación de fluido computacional en CFX.

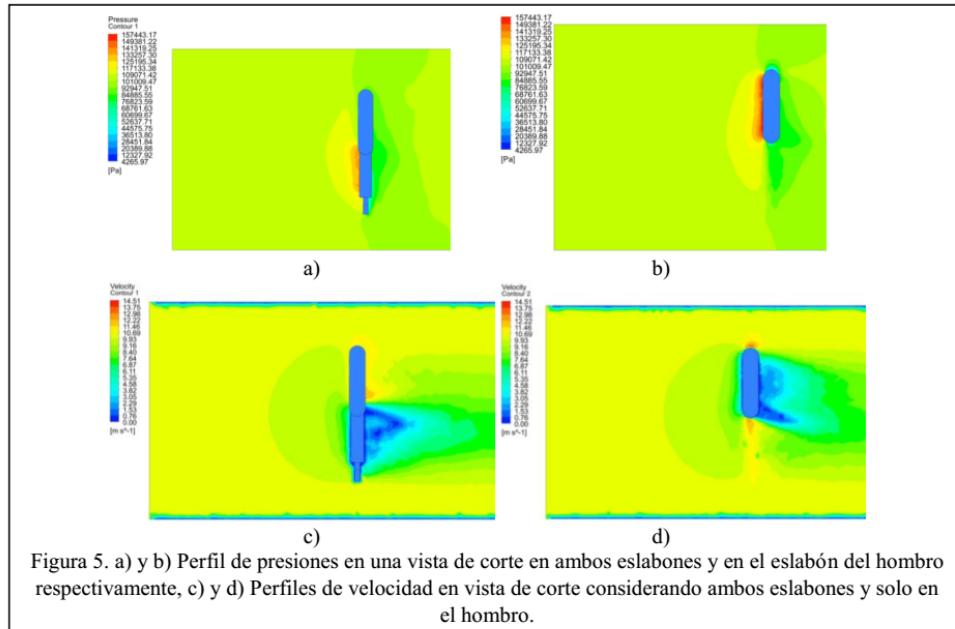
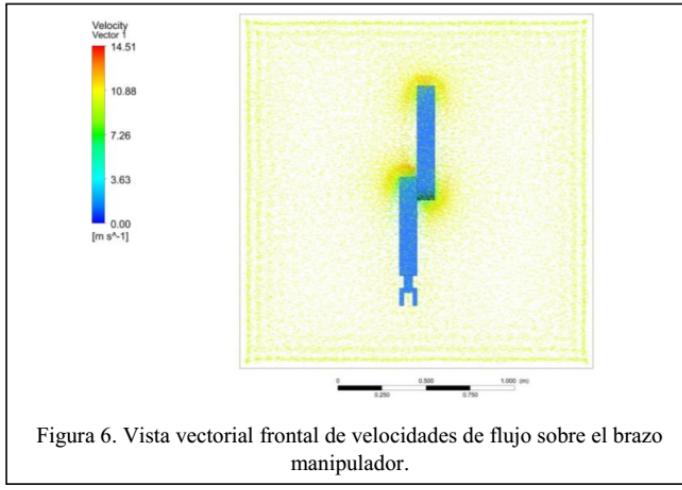


Figura 5. a) y b) Perfil de presiones en una vista de corte en ambos eslabones y en el eslabón del hombro respectivamente, c) y d) Perfiles de velocidad en vista de corte considerando ambos eslabones y solo en el hombro.

De la figura 5 se aprecia que la máxima presión es de 157 [KPa] y se observa un poco de turbulencia en la cara contraria al flujo, con velocidades máximas de 14.51 [m/s]. Por otra parte el perfil de velocidades vectorial se observa en la figura 6, las cuales pueden apreciarse en la figura 5.



De la figura 6 se aprecia que las zonas críticas son en las uniones de los eslabones en  $q_2$  por el cambio de geometría, además en estas zonas se presenta turbulencias en la parte posterior.

Debido a que los resultados analíticos comparados con la simulación en CFX son prácticamente iguales con un error del 0.1% en magnitud (ver cuadro 2), puede estimarse que el torque de diseño para la selección del actuador de la articulación  $q_1$ , está determinada por la ecuación (10). Sin embargo la simplificación del modelo 3D conlleva errores acarreados y también es posible refinar aún más la malla para obtener resultados aún más cercanos a los analíticos.

$F_D$ analítico [N]	$F_D$ CFX [N]
92.3961	92.3961

Cuadro 2. Tabla comparativa de los resultados obtenidos para fuerza de arrastre.

Sin embargo, los resultados obtenidos para el torque articular  $q_1$  son suficientes para determinar un criterio de diseño.

#### Conclusiones

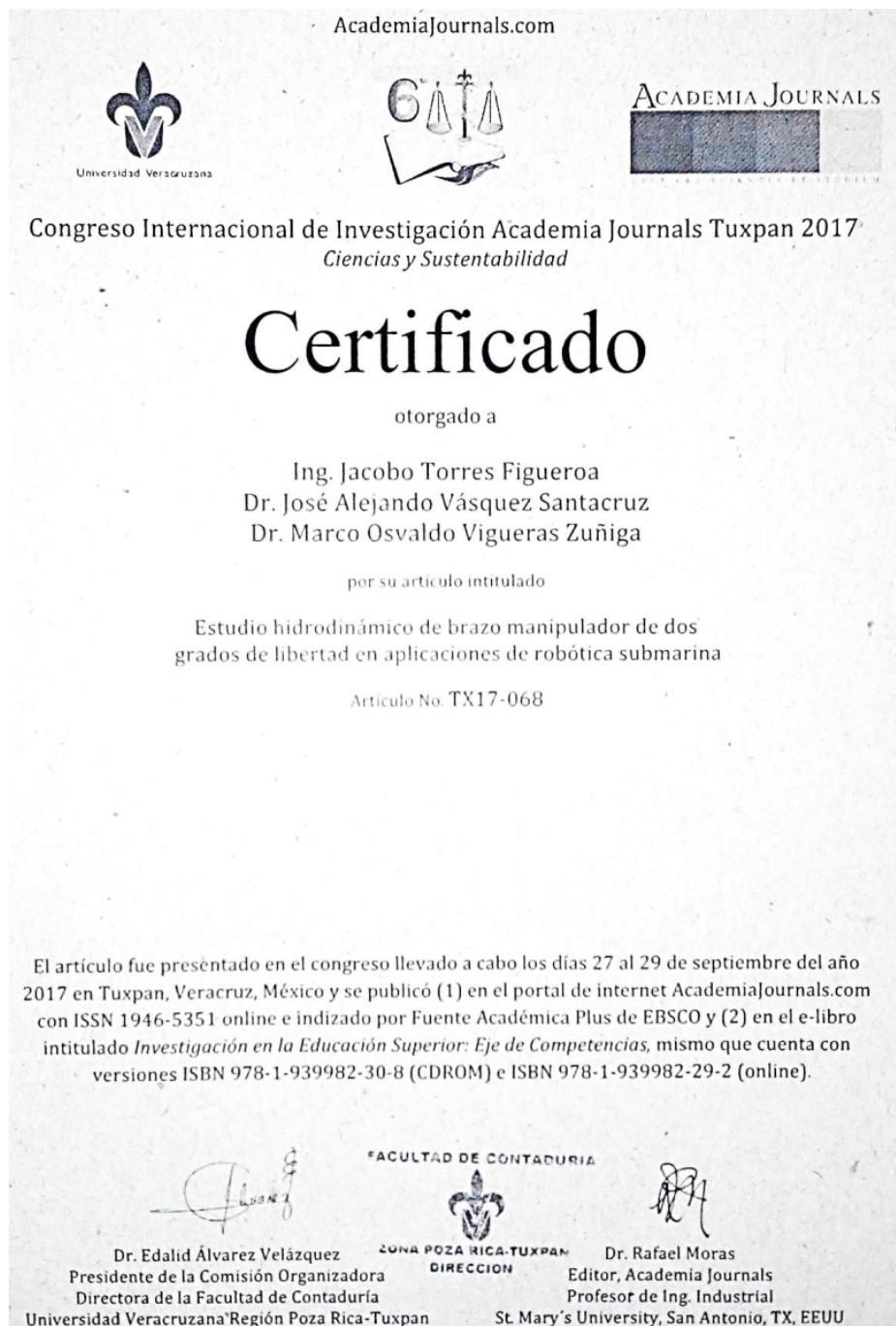
De los resultados de simulación y analíticos puede concluirse que las simulaciones numéricas pueden ofrecer un medio confiable de validación para diseños mecatrónicos sometidos a dinámica de fluidos. El objetivo del trabajo fue cumplido satisfactoriamente, con lo que se tiene una referencia para futuros análisis en cuanto para el diseño de brazos manipuladores en ROV subacuático. Sin embargo el presente trabajo puede extenderse para analizar otras configuraciones problemáticas del robot manipulador, debido a que el alcance y objetivo fue determinar únicamente el torque articular  $q_1$  puesto que la articulación que presenta mayor problema, aún falta considerarse la determinación del torque articular para  $q_2$ .

La utilización de perfiles romos en el diseño del manipulador ayudó a disminuir el coeficiente de arrastre y por ende la fuerza de arrastre, lo que conduce a un menor torque articular requerido. También es posible efectuar un diseño de brazo manipulador que presente flotabilidad neutra y utilizar un modelo simplificado para (6) y (7).

#### Referencias

- Barrientos A, Peñín L. F. y Balaguer C. (2007). "Fundamentos de Robótica", España: Mc Graw Hill.
- Ollero A. (2001). "Robótica; Manipuladores y Robots Móviles". España: Marcombo.
- Rafael K, Santibáñez V. (2003). "Control de Movimiento de Robots Manipuladores". Pearson Educación.
- White F. (2004). "Mecánica de Fluidos". Mc Graw Hill.

### 3.4 Evidencia de participación en el congreso



## Apéndice 4. CVU Actualizado

 																					
<b>Reporte CVU Completo</b>																					
Datos generales																					
CURP: TOFJ880814HVZRC04	Fecha de nacimiento: 14/ago/1988	RFC:																			
Nombre: JACOBO	Primer apellido: TORRES	Segundo apellido: FIGUEROA																			
Sexo: Masculino	Estado conyugal: Soltero(a)	País de nacimiento: México																			
Entidad federativa: VERACRUZ DE IGNACIO DE LA LLAVE		CVU: 733851																			
Contacto principal: lodo.orbe@gmail.com		Nacionalidad: Mexicana																			
Medios de contacto																					
Medio de contacto	Categoría de contacto	Correo / Teléfono	Principal																		
Correo electrónico	Oficial	lodo.orbe@gmail.com	SI																		
Grados académicos																					
Título: INGENIERO																					
Nivel de escolaridad: Licenciatura	Estatus: Grado obtenido																				
Cédula profesional:	Opciones de titulación: Tesis																				
Título de tesis: Estrategias de control de movimiento en robot pendular																					
Fecha de obtención: 30/ene/2015																					
País de obtención de grado: México																					
Participación en congresos																					
<p>Nombre del congreso: Academia Journals Colima</p> <p>Título del trabajo: DISEÑO DE CONTROLADOR DIGITAL PARA SERVOMECHANISMO DE SEGUIMIENTO SOLAR PARA APLICACIONES EN CELDAS</p> <p>Título de participación congreso: Ponencia</p> <p>Fecha: 03/oct/2016 País: México</p> <p>Palabra clave 1: Celdas fotovoltaicas Palabra clave 2: Servomecanismo Palabra clave 2: PID</p>																					
Colaboradores																					
<p>Nombre: Ulises Gabriel Garcia</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Sexo: Masculino</td> <td>Nivel de escolaridad: Licenciatura</td> <td>Nacionalidad: Mexicana</td> </tr> <tr> <td>Área:</td> <td>Campo:</td> <td></td> </tr> <tr> <td>Disciplina:</td> <td>Subdisciplina:</td> <td></td> </tr> <tr> <td>¿Pertenece al S. N. I.? No</td> <td>Producto generado: Artículo científico</td> <td></td> </tr> <tr> <td>Tiempo de colaboración: 6</td> <td>Tipo de colaborador: Otro</td> <td></td> </tr> <tr> <td>Medio de contacto: Correo electrónico</td> <td>Categoría de medio de contacto: Personal</td> <td>Correo: ulises3@hotmail.com</td> </tr> </table>				Sexo: Masculino	Nivel de escolaridad: Licenciatura	Nacionalidad: Mexicana	Área:	Campo:		Disciplina:	Subdisciplina:		¿Pertenece al S. N. I.? No	Producto generado: Artículo científico		Tiempo de colaboración: 6	Tipo de colaborador: Otro		Medio de contacto: Correo electrónico	Categoría de medio de contacto: Personal	Correo: ulises3@hotmail.com
Sexo: Masculino	Nivel de escolaridad: Licenciatura	Nacionalidad: Mexicana																			
Área:	Campo:																				
Disciplina:	Subdisciplina:																				
¿Pertenece al S. N. I.? No	Producto generado: Artículo científico																				
Tiempo de colaboración: 6	Tipo de colaborador: Otro																				
Medio de contacto: Correo electrónico	Categoría de medio de contacto: Personal	Correo: ulises3@hotmail.com																			
<p>Nombre: José Alejandro Vásquez Santacruz</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Sexo: Masculino</td> <td>Nivel de escolaridad: Doctorado</td> <td>Nacionalidad: Mexicana</td> </tr> <tr> <td>Área:</td> <td>Campo:</td> <td></td> </tr> <tr> <td>Disciplina:</td> <td>Subdisciplina:</td> <td></td> </tr> <tr> <td>¿Pertenece al S. N. I.? Si</td> <td>Producto generado: Ponencia</td> <td></td> </tr> <tr> <td>Tiempo de colaboración: 6</td> <td>Tipo de colaborador: Investigador SNI</td> <td></td> </tr> </table>				Sexo: Masculino	Nivel de escolaridad: Doctorado	Nacionalidad: Mexicana	Área:	Campo:		Disciplina:	Subdisciplina:		¿Pertenece al S. N. I.? Si	Producto generado: Ponencia		Tiempo de colaboración: 6	Tipo de colaborador: Investigador SNI				
Sexo: Masculino	Nivel de escolaridad: Doctorado	Nacionalidad: Mexicana																			
Área:	Campo:																				
Disciplina:	Subdisciplina:																				
¿Pertenece al S. N. I.? Si	Producto generado: Ponencia																				
Tiempo de colaboración: 6	Tipo de colaborador: Investigador SNI																				
<b>MÉXICO</b> <small>GOBIERNO DE LA REPÚBLICA</small>   																					
<b>Contacto:</b> Avenida Insurgentes Sur, 1582, Benito Juárez, Crédito Constructor, 03940 Ciudad de México, D.F. Tel: (55) 5322-7700																					
Página 2 de 3																					

Reporte CVU Completo		
Medio de contacto: Correo electrónico	Categoría de medio de contacto: Oficial	Correo: alejanvasquez@uv.mx
Participación en congresos		
<b>Nombre del congreso:</b> Academia Journals Tuxpan <b>Título del trabajo:</b> ESTUDIO HIDRODINÁMICO DE BRAZO MANIPULADOR DE DOS GRADOS DE LIBERTAD PARA ROBOT SUBMARINO <b>Título de participación congreso:</b> Ponencia <b>Fecha:</b> 27/sep/2017 <b>País:</b> México <b>Palabra clave 1:</b> ROV <b>Palabra clave 2:</b> Fuerza de arrastre <b>Palabra clave 2:</b> Brazo manipulador		
Proyectos de investigación		
<b>Nombre del proyecto:</b> Metodología integral para el diseño de un sistema mecatrónico de locomoción bipeda <b>Tipo de proyecto:</b> Investigación <b>Inicio:</b> 02/feb/2016 <b>Fin:</b> 31/ene/2018 <b>Institución:</b> Universidad Veracruzana		
Áreas de conocimiento		
<b>Área:</b> Ingeniería y tecnología	<b>Campo:</b> Ingeniería	
<b>Disciplina:</b> Ingeniería mecánica	<b>Subdisciplina:</b> Robótica	
Colaboradores		
<b>Nombre:</b> JOSÉ ALEJANDRO VASQUEZ SANTACRUZ		
<b>Sexo:</b> Masculino	<b>Nivel de escolaridad:</b> Doctorado	<b>Nacionalidad:</b> Mexicana
<b>Área:</b> Ingeniería y tecnología	<b>Campo:</b> Ingeniería	
<b>Disciplina:</b> Ingeniería mecánica	<b>Subdisciplina:</b> Robótica	
<b>¿Pertenece al S.N.I.?:</b> Sí	<b>Producto generado:</b> Artículo científico	
<b>Tiempo de colaboración:</b> 18	<b>Tipo de colaborador:</b> Investigador SNI	
<b>Medio de contacto:</b> Correo electrónico	<b>Categoría de medio de contacto:</b> Oficial	Correo: alejanvasquez@uv.mx
<b>Nombre:</b> JACOB TORRES FIGUEROA		
<b>Sexo:</b> Masculino	<b>Nivel de escolaridad:</b> Licenciatura	<b>Nacionalidad:</b> Mexicana
<b>Área:</b> Ingeniería y tecnología	<b>Campo:</b> Ingeniería	
<b>Disciplina:</b> Ingeniería mecánica	<b>Subdisciplina:</b> Robótica	
<b>¿Pertenece al S.N.I.?:</b> No	<b>Producto generado:</b> Artículo científico	
<b>Tiempo de colaboración:</b> 18	<b>Tipo de colaborador:</b> Otro	
<b>Medio de contacto:</b> Móvil	<b>Categoría de medio de contacto:</b> Personal	Teléfono: 2941169582