



INAOE

**Control en Modo Deslizante con Estimación de la
Perturbación Aplicado a la Marcha
de un Robot Bípedo.**

Por:

Marlon Fernando Velásquez Lobo

Tesis sometida como requisito parcial
para obtener el grado de

**MAESTRO EN CIENCIAS EN LA
ESPECIALIDAD DE ELECTRÓNICA**

en el

**Instituto Nacional de Astrofísica,
Óptica y Electrónica**
Tonantzintla, Puebla
Octubre del 2013

Supervisado por:

**Dr. José Luis Vázquez González
Dr. Juan Manuel Ramírez Cortés**

©INAOE 2013

Derechos Reservados

El autor otorga al INAOE el permiso de reproducir y
distribuir copias de esta tesis en su totalidad o en partes.



RESUMEN

En el estudio de prótesis inteligentes de las extremidades inferiores es común usar modelos de robots bípedos para diseñar el controlador. En la literatura se han reportado numerosos esquemas de control para el seguimiento de las trayectorias deseadas de un robot bípedo. En el presente trabajo se presenta el diseño de un controlador en modo deslizante con estimación de la perturbación, como una alternativa viable y novedosa en el control de robots bípedos. Este esquema de control soluciona algunos de los problemas presentes en el control de modo deslizante convencional, como las ganancias altas y las oscilaciones de alta frecuencia de la acción de control, que afectan negativamente la estabilidad del robot bípedo y aumentan el desgaste en los actuadores. Además se obtiene un controlador robusto a perturbaciones e incertidumbres en los parámetros.

En el capítulo 2 se presentan los conceptos fundamentales y ecuaciones cinemáticas y dinámicas que rigen el comportamiento de un robot bípedo de 5 eslabones en el plano sagital. Después el diseño de las trayectorias de referencia es explicado.

En el capítulo 3, el algoritmo del control en modo deslizantes convencional es explicado brevemente. Posteriormente, el concepto de estimación de la perturbación es incluido. Luego, se explica el diseño del controlador y se muestran los resultados en ausencia de perturbaciones.

En el capítulo 4 se describen dos pruebas de robustez realizadas mediante simulación. La primera prueba estudia el desempeño del controlador a desplazamientos abruptos del robot bípedo. En la segunda prueba es analizado el comportamiento frente a incertidumbres en los parámetros. En ambas pruebas se propone medir tres índices de desempeño basados en el error de seguimiento para demostrar la robustez del controlador frente a perturbaciones. Las pruebas anteriores apoyan la conclusión de un buen desempeño del control en modo deslizante del robot bípedo descrito.

ABSTRACT

Nowadays, biped robot models for simulation and design of controllers in the field of intelligent lower limb prosthesis, are widely used. A number of control schemes for tracking the trajectory of references in biped robots are currently reported in the literature. This thesis presents the design of a sliding mode controller with disturbance estimation as a feasible and new alternative for the control of biped robots. The proposed control scheme solves some of the existing problems in conventional sliding mode control, such as high gains and high frequency oscillations of the control action, which adversely affect the stability of the biped robot and increase the wear on the actuators.

Chapter 2 presents the fundamental concepts on kinematic and dynamic equations which govern the behavior of a 5-link biped robot in the sagittal plane. Design and requirements of reference trajectories is explained.

In Chapter 3, the algorithm of the conventional sliding mode control is briefly explained. Further, the concept of perturbation estimation is included. Design of the controller and display of results in the absence of disturbances is described.

Chapter 4 describe two simulation test aiming to evaluate robustness of the model. The first test examines performance of the controller to abrupt displacements on the biped robot trajectory. In the second test, performance against uncertainties in parameters is evaluated.

Three performance indexes based on the tracking error are proposed to quantitatively evaluate robustness of the controller on some disturbances. These tests support the conclusion of good performance of the proposed sliding mode control used on the biped robot.

Agradecimiento

A Dios por ser la guía en la realización de todos los triunfos obtenidos en el transcurso de todo este camino.

A mi familia por su apoyo incondicional, especialmente a mis padres y hermanas.

A los doctores José Luis Vázquez y Juan Manuel Ramírez, por haber ofrecido una excelente orientación y colaboración oportuna durante el desarrollo de este trabajo.

Al Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) por la formación profesional ofrecida.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por brindarme el apoyo financiero para poder culminar mis estudios de maestría.

Contenido

1	INTRODUCCIÓN	10
1.1	Algoritmos de control en robots bípedos.....	12
2	MODELADO DEL ROBOT BÍPEDO	18
2.1	Fundamentos de la caminata bípeda.....	18
2.1.1	Conceptos básicos	18
2.2	Descripción del sistema	20
2.3	Modelo cinemático	20
2.4	Modelo dinámico.....	22
2.4.1	Fase de soporte simple	23
2.4.2	Fase de doble soporte.....	24
2.4.3	Impacto y conmutación	25
2.5	Generación de trayectorias.....	26
2.6	Modelado en SimMechanics.....	31
2.6.1	Modelado del torso.....	32
2.6.2	Modelado de la pierna.....	33
2.6.3	Modelado de la superficie de contacto	35
3	TEORÍA Y DISEÑO DEL CONTROLADOR EN MODO DESLIZANTE CON ESTIMACIÓN DE LA PERTURBACIÓN (SMCPE).....	37
3.1	Estructuras Variables.....	38
3.2	Fundamentos de control en modo deslizante	38
3.3	Fundamentos de control en modo deslizante con estimación de la perturbación.....	45
3.4	Diseño del controlador	48

3.4.1	Consideraciones de diseño	50
4	RESULTADOS Y ANÁLISIS DE DESEMPEÑO	53
4.1	Definición de los índices y criterios de desempeño	54
4.1.1	Integral del error absoluto (IAE)	54
4.1.2	Integral del error cuadrático (ISE)	54
4.1.3	Integral del valor absoluto del error ponderado en el tiempo (ITAE)	55
4.2	Pruebas de robustez.....	55
4.2.1	Fuerza externa actuando sobre el torso.....	55
4.2.2	Incertidumbre en los parámetros de la planta.	60
5	CONCLUSIONES Y OBSERVACIONES	65
	BIBLIOGRAFÍA.....	69
6	APENDICES.....	77
6.1	Generación de trayectorias.....	77
6.2	Función S en <i>Simulink</i>	81
6.2.1	Función S para el modelo del robot bípedo.....	82
6.2.2	Función S para el controlador SMCPE.....	86

Lista de Figuras

Figura 1. Evolución del robot ASIMO. Fuente: http://world.honda.com/ASIMO/	14
Figura 2. Robot bípedo Lucy. [31].	16
Figura 3. Planos anatómicos del cuerpo humano. Fuente: W. M. Boothby. An Introduction to Differentiable Manifolds and Rie-mannian Geometry.	19
Figura 4. Fase de simple y doble soporte.	20
Figura 5. Modelo del robot bípedo de 5 eslabones	21
Figura 6. Parámetros para la planeación de la caminata en el plano sagital.	27
Figura 7. Trayectorias de la caminata en el plano sagital.	29
Figura 8. Posición y velocidad angular para un ciclo.	31
Figura 9. Diagrama de bloques general en Simulink del robot bípedo.	32
Figura 10. Diagrama de bloques para el torso (SimMechanics).	33
Figura 11. Diagrama de bloque para la pierna.	34
Figura 12. Diagrama de bloque para el subsistema actuador.	34
Figura 13. Diagrama de bloques para el modelo de la superficie.	36
Figura 14. Región deslizante en el plano fase con chattering para un sistema de tercer orden.	41
Figura 15. Diagrama de bloques del sistema completo.	49
Figura 16. Comportamiento del torque generalizado, el error y la dinámica de la superficie s	51

Figura 17. Resultado de la animación con <i>SimMechanics</i> en diferentes tiempos.	52
Figura 18. Índices de desempeño para cada ángulo al aplicar diferentes torques de perturbación en el tiempo 0.38 [s] y duración de 0.2 [s].....	57
Figura 19. Torque y error al aplicar una perturbación de 480 [Nm] en el torque $T_{\theta 3}$	58
Figura 20. Torque y error al aplicar una perturbación de 500 [Nm] en el torque $T_{\theta 3}$	59
Figura 21. Secuencia del robot bípedo al aplicar una perturbación de 500 [Nm] en el tiempo 0.38 [s] y duración de 0.2 [s]	59
Figura 22. Máxima perturbación permitida contra duración.	60
Figura 23 Resultado de la prueba de incertidumbre en los parámetros.....	62
Figura 24. Índices de desempeño para variaciones porcentuales en la inercia.	63
Figura 25. Torque y error para cada ángulo al variar -50[%] la inercia de cada eslabón.	64

Lista de Tablas

Tabla 1. Valor numérico de los parámetros para el diseño de la trayectoria.	28
Tabla 2. Parámetros para el torso.....	33
Tabla 3. Parámetros para la pierna.....	34
Tabla 4. Índices de desempeño para cada ángulo al aplicar diferentes torque de perturbación en el tiempo 0.38 [s] y duración de 0.2 [s].....	56
Tabla 5. Tareas a ser ejecutadas de acuerdo a la bandera en la función S ..	82

1 INTRODUCCIÓN

El tema de la discapacidad física ha tenido mayor atención por parte de la sociedad científica en los últimos años. Las dificultades y barreras que enfrentan las personas con discapacidades físicas han sido expuestas y analizadas públicamente y cada vez hay más personas, instituciones y asociaciones comprometidas con la tarea de mejorar la calidad de vida de este grupo poblacional.

En el censo del 2010 el Instituto Nacional de Estadística y Geografía (INEGI) contó dos millones 400 mil mexicanos con impedimentos para moverse por cuenta propia. El Consejo Nacional para las Personas con Discriminación calcula que al menos la cuarta parte ha sufrido la pérdida de un pie, una pierna, una mano o un brazo. La cifra parece discreta frente al número de mexicanos que cada año debe someterse a una amputación por enfermedad (50 por ciento de los casos), accidente (30 por ciento) o padecimiento congénito (20 por ciento), de acuerdo con la clasificación oficial.

La prótesis es una extensión artificial que reemplaza o provee una parte del cuerpo que falta por diversas razones, cumpliendo las mismas funciones que la parte faltante. Particularmente, las prótesis pasivas (sin actuadores) de miembros inferiores requieren que sus usuarios realicen esfuerzos importantes al realizar movimientos coordinados de varias articulaciones. Al caminar una persona con prótesis puede llegar a gastar 60% más energía metabólica en comparación con una persona sana. Las limitaciones crecen al usar escaleras y caminar sobre superficies con pendiente [1].

Con la motivación de superar los anteriores inconvenientes el estudio de prótesis robóticas o inteligentes se ha incrementado en los últimos años. Uno de los principales temas de investigación es la estrategia de control. Básicamente al aplicar control sobre prótesis robóticas el objetivo es emular la marcha natural del ser humano, luego el controlador está enfocado en el

seguimiento de trayectorias que pueden ser obtenidas en tiempo real o previamente generadas.

Para el diseño y simulación del control de prótesis del miembro inferior es necesario contar con un modelo matemático, que permita realizar análisis teóricos y simulaciones antes de ser probado en seres humanos. En este tipo de investigaciones es común trabajar con robots bípedos [2] y [3]. Generalmente el modelo dinámico de robots bípedos es obtenido a partir de las ecuaciones de Lagrange, cuya complejidad crece dependiendo del número de grados de libertad (DoF¹) y eslabones que se consideren [4].

El principal problema de control en robots bípedos surge de la complejidad de la dinámica no lineal y su alta inestabilidad. Numerosas técnicas han sido propuestas con el propósito de tener un buen seguimiento de las trayectorias de referencias sin perder estabilidad. Entre las más relevantes se pueden mencionar: controladores PID ([5] [6] y [7]), método de torque computado (CTM²) ([8], [9] y [10]) y otras más robustas como el control en modo deslizante (SMC³) [11]. El SMC presenta algunas ventajas, puesto que a diferencia del CTM no necesita conocer perfectamente la dinámica del sistema, además es robusto a incertidumbres en los parámetros y perturbaciones externas. Sin embargo, esta técnica presenta un fenómeno conocido como “*chattering*” debido al proceso de conmutación en la acción de control, por lo tanto se afecta negativamente la estabilidad de los robots bípedos, además de aumentar el desgaste en los actuadores.

Algunos autores han usado modificaciones del SMC ([3] y [12]), las cuales consisten en regular los parámetros principales del controlador, con el

¹ Degree of freedom

² Computed Torque Method

³ Sliding mode control

objetivo de eliminar o reducir considerablemente el “*chattering*”. Estas técnicas han presentado buenos resultados, pero requieren del conocimiento de los límites de las perturbaciones y modelado de incertidumbre para asegurar robustez.

El presente trabajo propone evaluar una nueva alternativa conocida como control en modo deslizante con estimación de la perturbación (SMCPE⁴), la cual ofrece un control por realimentación robusto y con ganancias menores que el SMC tradicional. Aplicada a un robot bípedo de 5 eslabones en el plano sagital. Primero se presenta una descripción del modelo cinemático y dinámico, generación de trayectorias y el desarrollo del modelo del robot bípedo usando la herramienta *SimMechanics* de Simulink. Luego se introducen los fundamentos del control en modo deslizante con estimación de la perturbación y el diseño del controlador. Por último, se presenta el análisis de los resultados de dos pruebas de robustez.

1.1 Algoritmos de control en robots bípedos

El desarrollo de prótesis robóticas ha tenido un creciente interés en la comunidad científica. Particularmente las prótesis para el miembro inferior están relacionadas directamente con el estudio de robots bípedos, los cuales permiten evaluar las prótesis diseñadas y desarrollar nuevos algoritmos de control antes de ser probada en seres humanos [13] [14]. El modelo dinámico de los robots bípedos es determinado a partir de las ecuaciones de Lagrange [15]. Generalmente se considera que las piernas son idénticas, pero en [16] proponen una metodología para modelar un robot bípedo con piernas heterogéneas, es decir, una integración de un robot bípedo común y una

⁴ Sliding mode control with perturbation estimation

pierna biónica inteligente. Este tipo de modelado puede ser usado para estudiar la coordinación entre un ser humano y su extremidad artificial [17].

El control de robots bípedos requiere de una planificación adecuada de la marcha para poder asegurar la estabilidad. La tendencia a caer puede ser determinada por el criterio del punto de momento cero (ZMP⁵) [18], [19] obtenido como un punto de cruce del vector de fuerza resultante inercial y gravitacional con el suelo. Muchos estudios acerca del control y generación de trayectoria de la caminata bípeda utilizan dicho criterio [2], [20], [21], [22] y [23]. Otro concepto usado con frecuencia es el indicador de rotación del pie (FRI⁶) [24], [25] el cual corresponde al punto en la superficie donde la reacción neta del piso tendría que actuar para que el pie de apoyo permanezca estacionario.

El ZMP como esquema de control, fue introducido hace muchos años [19], y es actualmente uno de los más extendidos. Este concepto fue aplicado con éxito por primera vez en 1984 en *Waseda University, laboratory of Ichiro Kato*, en el robot *WL-10RD* el primero equilibrado dinámicamente, y desde entonces en otros múltiples robots. ASIMO⁷ es un robot humanoide creado por la empresa HONDA (Figura 1), y es tal vez el robot más representativo basado en el algoritmo ZMP. El funcionamiento se basa en generar un patrón de movimiento ideal para las articulaciones que cumplan con el concepto de estabilidad del ZMP. ASIMO aplica tres estrategias de control para prevenir una caída y lograr estabilizar la locomoción. La primera consiste en el control de la reacción del suelo, se controla la fuerza ejercida en los pies para detectar irregularidades en el terreno y poder corregir a tiempo la posición

⁵ Zero Moment Point

⁶ Foot Rotation Indicator

⁷ Advanced Step in Innovative Mobility

deseada. La segunda consiste en el control del ZMP objetivo, el cual mantiene la posición por la aceleración del torso en la dirección en la que tiende a caer cuando los pies no brindan un soporte firme. La tercera estrategia consiste en el control del siguiente paso, de tal manera que el siguiente paso sea el más apropiado para compensar la perturbación. Una de las desventajas principales de este esquema de control es que siempre busca que las trayectorias de locomoción sean dinámicamente estables, lo cual no es necesario para imitar la locomoción humana, porque los humanos no cumplen la condición de ZMP en todas las fases de la marcha.



Figura 1. Evolución del robot ASIMO [26].

El controlador más simple que puede ser usado para el control de robots bípedos es el PID, sin embargo, sus ganancias son difíciles de seleccionar debido a las no linealidades del robot ([5], [27]). Las ecuaciones de movimiento se linealizan alrededor de una postura vertical asumiendo que no existe un desvío considerable de esta posición, por lo tanto este tipo de control tiene muy bajo desempeño en presencia de perturbaciones o incertidumbre en los parámetros. Para solucionar lo anterior en [6] proponen un nuevo concepto de un modelo no lineal de referencia PID para un robot bípedo con 7 DoF. El controlador consiste de 5 ecuaciones diferenciales, cada una en término del error de la trayectoria de cada articulación. Se presentan simulaciones donde el nuevo enfoque permite tener un buen

seguimiento en presencia de perturbaciones. Por otro lado, una técnica conocida como refuerzo de aprendizaje es aplicada en conjunto con un controlador PD [7] (usado en los actuadores neumáticos del robot) a un robot bípedo de 2 eslabones. Los resultados de las simulaciones demuestran que el robot bípedo aprende rápidamente a superar las perturbaciones de la superficie, sin comprometer el estilo del caminar natural.

Algunos métodos de control óptimo han sido usados por los investigadores para regular la fase dinámica de los sistemas de locomoción bípeda. Dos enfoques se han adoptado para el problema de optimización. El primer método se basa en el cálculo de los valores de los parámetros seleccionados en las funciones objetivo que reduzcan al mínimo las funciones de costo basados en la energía [28]. El segundo enfoque se basa en métodos variacionales para obtener controladores que reduzcan al mínimo las funciones de costo [29]. Ambos enfoques buscan la generación óptima de trayectorias y se olvidan del problema de estabilidad frente a perturbaciones.

Los métodos híbridos inteligentes gracias a su capacidad complementaria, han encontrado lugar en la investigación de la síntesis y control de la marcha de los robots bípedos. En [30], un control adaptativo de la marcha conocido como *neural-fuzzy* es propuesto. Este esquema usa una red neuronal basada en regresión lineal para construir la base del controlador. En [31] combinan *neural-fuzzy* con el concepto ZMP para controlar la marcha de un robot bípedo capaz de ser sometido a empujones en el plano sagital, usando una red neuronal de primer orden tipo Takagi-Sugeno-Kang, la cual es entrenada con datos de simulación y experimentales.

Un controlador muy popular es el método de torque computado (CTM) que ha sido aplicado a modelos de locomoción bípeda con varios niveles de complejidad [8]. En [9] y [32] aplican CTM para accionar los músculos artificiales del robot bípedo conocido como Lucy (Figura 2), el cual está

restringido al plano sagital. El diseño de control es específico para las dos principales fases de la marcha (apoyo simple y doble) donde el torque calculado es diferente. Utilizan un modelo de simulación dinámica para evaluar la arquitectura del control. Para los resultados de rendimiento y pruebas de robustez tienen en cuenta el error generado por el modelo y las limitaciones reales del hardware. Finalmente reportan resultados satisfactorios al incorporar el controlador diseñado en el robot real.

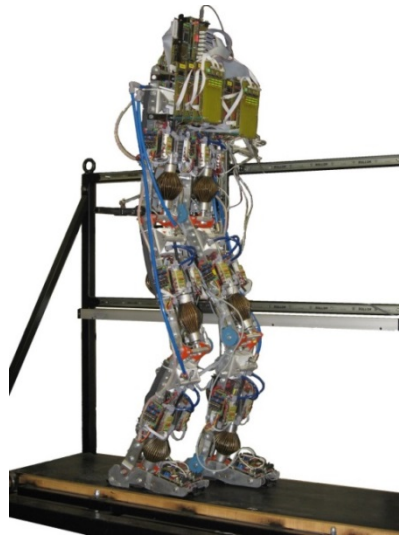


Figura 2. Robot bípedo Lucy. [32].

Chevallerau [10] combinó CTM con escalamiento en el tiempo de las trayectorias deseadas, lo que permite la convergencia de tiempo finito del estado del sistema hacia el movimiento deseado. El robot estudiado es de 5 eslabones. La ley de control la definen de tal manera que sólo la evolución geométrica del robot se controla, no la evolución temporal. Cabe resaltar que el control por CTM presenta la desventaja del deterioro del desempeño debido a incertidumbres en los parámetros, ya que esta técnica requiere un perfecto conocimiento de la dinámica del sistema.

El control en modo deslizante (SMC) ha sido aplicado a un modelo de robot bípedo de 5 eslabones en el plano sagital y comparado con CTM [11] los

resultados obtenidos a través del SMC fueron mejores que los obtenidos con CTM, incluso en la presencia de perturbaciones e incertidumbre en los parámetros. Pero uno de los problemas presentados con el tradicional SMC es la presencia de '*chattering*' el cual son oscilaciones de alta frecuencia en la acción de control, que producen inestabilidad y desgaste de los actuadores en los robots bípedos. Como una alternativa para solucionar los anteriores inconvenientes en [3] presentan el diseño de trayectorias suaves para aliviar los impactos debidos al contacto con el suelo, luego aplican SMC regulando sus principales parámetros. Los resultados reportados muestran que el rendimiento en presencia de incertidumbres y perturbaciones es aceptable y además cumplen con el objetivo de eliminar el *chattering*. Por otro lado en [12] a diferencia del anterior trabajo, utilizan un sistema difuso junto con SMC reportando resultados similares.

Como una alternativa para reducir los problemas del SMC clásico en [33] se propone el control en modo deslizante con estimación de la perturbación (SMCPE), el cual permite reducir el desgaste en los actuadores, al evitar la sobrestimación de la perturbación presente en los métodos no lineales convencionales como el SMC. Esta técnica estima de forma precisa las perturbaciones en los estados presentes, obteniendo como beneficio ganancias de control menos rígidas y más suaves que con acciones equivalentes en el SMC. El SMCPE ha sido aplicado con éxito a un manipulador de dos eslabones [33], al sistema de altitud de un satélite [34] y un micro satélite [35] y al control de velocidad del rotor de una turbina hidráulica [36]. En robots bípedos aún no ha sido reportada su aplicación, por lo tanto en el presente trabajo el SMCPE se presenta como una alternativa viable y novedosa para el control de seguimiento de una trayectoria deseada, consiguiendo como resultado un controlador no lineal robusto a perturbaciones e incertidumbres en los parámetros.

2 MODELADO DEL ROBOT BÍPEDO

En este capítulo se describen los aspectos relacionados con el modelo del robot bípedo a usar. Primero se realiza una breve descripción de los principales conceptos básicos. Seguido a esto se presentan el modelo cinemático y dinámico. Luego se describe una manera sencilla de generar las trayectorias deseadas, las cuales son necesarias para el funcionamiento del controlador. Por último, se describe una metodología para crear el modelo del robot bípedo con la herramienta *SimMechanics* de *Simulink*.

2.1 Fundamentos de la caminata bípeda

Los robots bípedos forman una subclase de los robots con piernas. El estudio de la locomoción de piernas mecánicas ha sido motivado por el uso potencial en terrenos difíciles o con discontinuidades como los peldaños de una escalera. El tema de robot bípedos ha generado gran interés por diversas razones, que van desde el deseo de sustituir a los humanos en tareas peligrosas (minas, plantas nucleares, intervenciones militares, etc.) a la restauración del movimiento en personas con discapacidades (prótesis controladas dinámicamente, rehabilitación robótica y estimulación neurofuncional) [37]. A continuación se presentan los conceptos básicos en el estudio de robots bípedos para luego poder definir el modelo dinámico a usar.

2.1.1 Conceptos básicos

Para poder describir el modelado de un robot bípedo es necesario estar familiarizado con los siguientes conceptos:

- **Planos anatómicos:** Son los planos que dividen a un cuerpo. El *plano sagital* es el plano longitudinal que divide el cuerpo en izquierda y derecha. El *plano frontal* es el plano paralelo al eje largo del cuerpo y

perpendicular al plano sagital. El *plano transversal* es perpendicular al plano sagital y al plano frontal [38]. La siguiente figura ilustra lo anterior.

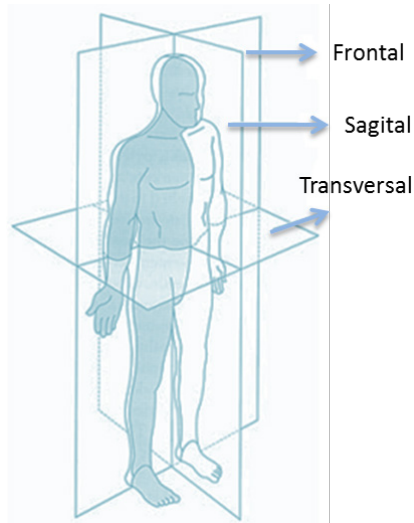


Figura 3. Planos anatómicos del cuerpo humano [38].

- **Centro de masa (COM):** Es un punto dentro del cuerpo humano que puede ser considerado como un punto de masa concentrada.
- **Fases de la marcha humana:** La fase de soporte simple (SSP) o de oscilación se define como la fase de locomoción donde sólo un pie está en el suelo. Por el contrario, durante la fase de doble apoyo (DSP) ambos pies están en el suelo (Figura 4). Caminar se define entonces como fases alternantes de apoyo simple y doble, con el requisito de que el desplazamiento de la componente horizontal del centro de masa del robot es estrictamente monótona. Implícito en esta descripción es la suposición de que los pies no se deslicen cuando están en contacto con el suelo.
- **Punto de momento cero (ZMP):** Es obtenido como un punto de cruce del vector de fuerza resultante inercial y gravitacional con el suelo

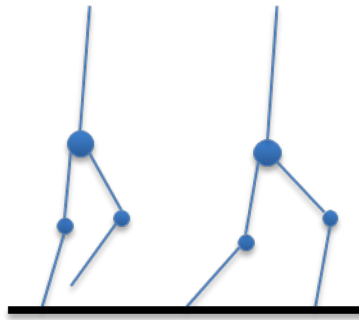


Figura 4. Fase de simple y doble soporte.

2.2 Descripción del sistema

El robot bípedo bajo estudio es de 5 eslabones, restringido al plano sagital. Consiste de un torso y dos piernas idénticas, cada pierna está compuesta por 2 eslabones. Todas las uniones son consideradas rotacionales. La fricción entre el pie y la superficie es considerada suficiente para prevenir deslizamientos. El robot tiene 4 actuadores dos en la cadera y uno para cada rodilla.

2.3 Modelo cinemático

El modelo cinemático describe el movimiento del robot bípedo en términos de posiciones, velocidades y aceleraciones basado en las uniones angulares, sin considerar las fuerzas externas que causan el movimiento, como las fuerzas de impacto con la superficie.

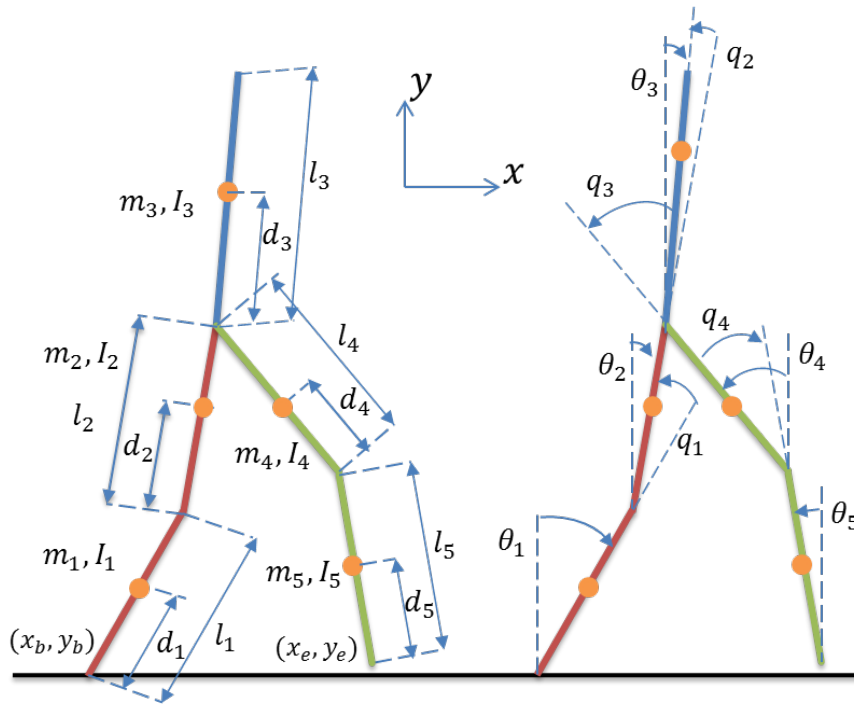


Figura 5. Modelo del robot bípedo de 5 eslabones.

Los parámetros mostrados en la Figura 5 son los siguientes:

- m_i : masa del eslabón i .
- l_i : longitud del eslabón i .
- d_i : distancia entre el centro de masa del eslabón i y su unión más baja.
- I_i : momento de inercia con respecto al eje que pasa a través del centro de masa del eslabón i y siendo perpendicular al plano del movimiento.
- θ_i : ángulo del eslabón i con respecto a la vertical.
- q_i : las deflexiones angulares relativas de las articulaciones correspondientes.
- (x_b, y_b) : posición del punto de soporte.
- (x_e, y_e) : posición del extremo de la pierna en balanceo.

De lo anterior se puede obtener [39]:

$$x_e = x_b + l_1 \sin \theta_1 + l_2 \sin \theta_2 + l_4 \sin \theta_4 + l_5 \sin \theta_5 \quad (2.1)$$

$$y_e = x_b + l_1 \cos \theta_1 + l_2 \cos \theta_2 - l_4 \cos \theta_4 - l_5 \cos \theta_5 \quad (2.2)$$

Si (cgx, cgy) son las coordenadas del centro de masa del robot bípedo, (x_{ci}, y_{ci}) las coordenadas del centro de masa para cada eslabón, se obtiene:

$$\begin{aligned} x_{c1} &= d_1 \sin \theta_1 \\ y_{c1} &= d_1 \cos \theta_1 \\ x_{c2} &= l_1 \sin \theta_1 + d_2 \sin \theta_2 \\ y_{c2} &= l_1 \cos \theta_1 + d_2 \cos \theta_2 \\ x_{c3} &= l_1 \sin \theta_1 + l_2 \sin \theta_2 + d_3 \sin \theta_3 \\ y_{c3} &= l_1 \cos \theta_1 + l_2 \cos \theta_2 + d_3 \cos \theta_3 \\ x_{c4} &= l_1 \sin \theta_1 + l_2 \sin \theta_2 + (l_4 - d_4) \sin \theta_4 \\ y_{c4} &= l_1 \cos \theta_1 + l_2 \cos \theta_2 - (l_4 - d_4) \cos \theta_4 \\ x_{c5} &= l_1 \sin \theta_1 + l_2 \sin \theta_2 + l_4 \sin \theta_4 + (l_5 - d_5) \sin \theta_5 \\ y_{c5} &= l_1 \cos \theta_1 + l_2 \cos \theta_2 - l_4 \cos \theta_4 - (l_5 - d_5) \cos \theta_5 \end{aligned} \quad (2.3)$$

$$\begin{aligned} cgx &= \frac{(m_1 x_{c1} + m_2 x_{c2} + m_3 x_{c3} + m_4 x_{c4} + m_5 x_{c5})}{(m_1 + m_2 + m_3 + m_4 + m_5)} \\ cgy &= \frac{(m_1 y_{c1} + m_2 y_{c2} + m_3 y_{c3} + m_4 y_{c4} + m_5 y_{c5})}{(m_1 + m_2 + m_3 + m_4 + m_5)} \end{aligned} \quad (2.4)$$

2.4 Modelo dinámico

El modelo dinámico es usado para calcular los torques que actúan en cada unión, por lo tanto todas las fuerzas exteriores se incluyen en el modelo dinámico. Se asume que la fuerza de fricción con la superficie es suficientemente grande para asegurar que no haya deslizamiento en la pierna de soporte. El modelo dinámico se presenta para las fases de simple y doble soporte.

2.4.1 Fase de soporte simple

Dado que el movimiento del robot bípedo es realizado en el plano sagital (Figura 5), los ángulos θ_i ($i = 1, 2, \dots, 5$) son suficientes para describir completamente su configuración. De acuerdo a Lagrange el modelo dinámico para SSP es el siguiente [39]:

$$D(\theta)\ddot{\theta} + H(\theta)\dot{\theta}^2 + G(\theta) = T_\theta \quad (2.5)$$

Donde $D(\theta)$ es la matriz de inercia de 5x5 definida positiva y simétrica, la cual representa el cambio de estado de movimiento del robot; $H(\theta)$ es una matriz de 5x5 que agrupa los términos de inercia centrífuga y Coriolis, la fuerza centrífuga son fuerzas radiales y la fuerza de Coriolis representa una desviación del movimiento de traslación debido a su componente de rotación. La matriz $G(\theta)$ es de 5x1 que representa el término de la gravedad. A continuación se presenta la forma detallada [40]:

$$\begin{aligned} D_{ij} &= p_{ij} \cos(\theta_i - \theta_j) \\ H_{ij} &= p_{ij} \sin(\theta_i - \theta_j) \\ G_i &= g_i \sin \theta_i \end{aligned} \quad (2.6)$$

Donde $i, j = 1, 2, \dots, 5$, p_{ij} y g_i definidos como:

$$p_{ij} = \begin{cases} I_i + m_i d_i^2 + a_i \left(\sum_{j=i+1}^5 m_j \right) & j = i \\ a_i m_j d_j l_i + a_i a_j \left(\sum_{k=j+1}^5 m_k \right) l_i l_j & j > i \\ p_{ji} & j < i \end{cases} \quad (2.7)$$

$$g_i = m_i d_i g + a_i \left(\sum_{j=i+1}^5 m_j \right) l_i g \quad (2.8)$$

Donde a_i es un número definido por $a_i = 0$ si $i = 3$ y $a_i = 1$ si $i = 1, 2, 4, 5$.

2.4.2 Fase de doble soporte

En la fase de doble soporte, la posición entre las puntas de las extremidades y la superficie es fija [40], con lo cual el conjunto de ecuaciones algebraicas que representan las restricciones en la posición es:

$$\Phi(\theta) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} x_e - x_b - L \\ y_e - y_b \end{bmatrix} = 0 \quad (2.9)$$

Donde L es la longitud del paso. Ahora si aplicamos Lagrange con las restricciones, la ecuación dinámica para DSP se puede expresar como:

$$D(\theta)\ddot{\theta} + H(\theta)\dot{\theta}^2 + G(\theta) = J^T(\theta)\lambda + T_\theta \quad (2.10)$$

Donde D, H, G y T_θ son las mismas de la ecuación (2.5), λ es un vector 2×1 de multiplicadores de Lagrange, y J es la matriz Jacobiana ($J = \partial\Phi/\partial\theta$) de 2×5 .

$$J = \begin{pmatrix} l_1 \cos \theta_1 & l_3 \cos \theta_3 & 0 & l_4 \cos \theta_4 & l_5 \cos \theta_5 \\ -l_1 \sin \theta_1 & -l_3 \sin \theta_3 & 0 & -l_4 \sin \theta_4 & -l_5 \sin \theta_5 \end{pmatrix} \quad (2.11)$$

Por último de acuerdo con [41], la ecuación (2.10) puede ser reescrita como:

$$\begin{aligned} \ddot{\theta} &= D^{-1}(T_\theta - H(\theta)\dot{\theta}^2 - G(\theta) + J^T\lambda) \\ \lambda &= -(JD^{-1}J^T)^{-1} \left(JD^{-1} \left(T_\theta - H(\theta)\dot{\theta}^2 - G(\theta) \right) + j\dot{\theta} \right) \end{aligned} \quad (2.12)$$

2.4.3 Impacto y conmutación

Para asegurar que el modelo dinámico funcione en el próximo ciclo, se necesita conmutar el rol de la pierna de soporte y la pierna de balanceo. Los desplazamientos físicos y velocidades en realidad no cambian, pero la articulación, la cual es usada en el modelo dinámico necesita ser renombrada [40]. Los estados inmediatamente antes y después de la conmutación pueden ser escritos como:

$$\begin{bmatrix} \theta^+ \\ \dot{\theta}^+ \end{bmatrix}_{Conmutación} = \begin{bmatrix} T & 0_{5 \times 5} \\ 0_{5 \times 5} & T \end{bmatrix} \begin{bmatrix} \theta^- \\ \dot{\theta}^+_{Impacto} \end{bmatrix} \quad (2.13)$$

Donde

$$T = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.14)$$

Para aplicar el torque al actuador de cada articulación, la dinámica del bípido durante SSP y DSP descritos por (2.5) y (2.10) necesitan ser reformulados en términos de los ángulos relativos (ver Figura 5).

$$D_q(q)\ddot{q} + H_q(q)\dot{q}^2 + G_q(q) = T_q \quad (2.15)$$

$$D_q(q)\ddot{q} + H_q(q)\dot{q}^2 + G_q(q) = J_q^T(q)\lambda + T_q \quad (2.16)$$

Donde $T_q = [\tau_1, \tau_2, \tau_3, \tau_4, \tau_5]^T$ representan los torque de cada articulación.

Por otro lado una matriz de transformación puede ser encontrada para expresar q por θ [40].

$$q = M_{q\theta}\theta$$

$$M_{q\theta} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \quad (2.17)$$

Luego, la relación entre los torques parciales y los torques de cada unión pueden ser expresados como:

$$T_{\theta_i} = \sum_{j=1}^5 \tau_j \frac{\partial q_j}{\partial \theta_i} = \sum_{j=1}^5 \tau_j (M_{q\theta})_{ji} \quad (2.18)$$

$$T_{\theta} = (T_q^T M_{q\theta})^T = M_{q\theta}^T T_q$$

2.5 Generación de trayectorias

En esta sección se describe el procedimiento usado para generar las trayectorias en el plano sagital del robot bípedo de 5 eslabones. Primero obtenemos la trayectoria de la cadera y del extremo de la pierna en balanceo, para luego poder obtener los ángulos de cada articulación. La Figura 6, muestra los parámetros necesarios para la planeación de la caminata en el plano sagital. Donde T_p es la duración de un paso, l_p es la longitud del paso, h_{cmax} es la máxima altura de la cadera, h_{cmin} es la mínima altura de la cadera, h_c es el rizado máximo de la altura de la cadera, h_p es la altura máxima del pie, l_m la distancia medida desde que el pie deja la superficie hasta que obtiene la altura máxima y T_m es el tiempo en el que se obtiene la altura máxima del pie.

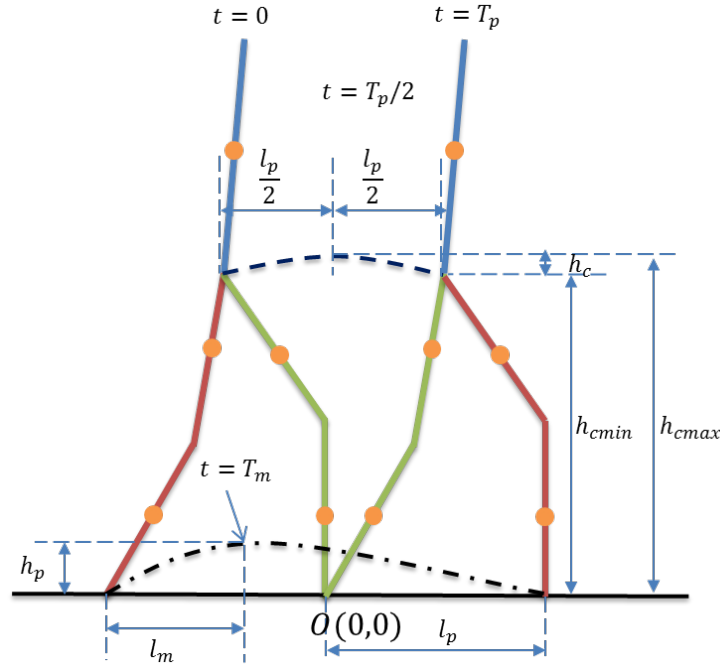


Figura 6. Parámetros para la planeación de la caminata en el plano sagital.

La trayectoria del pie es denotada por $(x_e(t), y_e(t))$, el pie parte del punto $(-l_p, 0)$ para llegar al punto $(-l_p + l_m, h_p)$ en el tiempo T_m , luego impacta la superficie en el punto $(l_p, 0)$ en el tiempo T_p . Por otro lado, la trayectoria de la cadera es denotada por $(x_c(t), y_c(t))$, la cadera parte del punto $(-\frac{l_p}{2}, h_{cmin})$ para llegar al punto $(0, h_{cmax})$ en el tiempo $T_p/2$, por último llega al punto $(\frac{l_p}{2}, h_{cmin})$ en el tiempo T_p .

Para minimizar los efectos del impacto con la superficie, el cual causa cambios instantáneos en la velocidad angular de las articulaciones, se lleva la velocidad a cero en el momento del impacto. Además, para asegurar la repetibilidad de la marcha, la velocidad en el inicio de la trayectoria también debe ser cero.

Un polinomio para cada trayectoria, cumpliendo todos los requisitos de la posición y velocidad, tendría un orden muy alto, aparte de necesitar muchas

condiciones. Por esta razón, se elige utilizar “*cubic splines*” en un esquema de interpolación, que son ampliamente utilizados para crear las trayectorias de las articulaciones en robótica. “*Cubic splines*” consisten de polinomios de tercer orden interpolados, y pueden asegurar la continuidad de la velocidad y aceleración [42]. Usando las anteriores restricciones se obtiene:

$$x_e(t) = \frac{-4l_p}{T_p^3}t^3 + \frac{6l_p}{T_p^2}t^2 - l_p \quad 0 < t \leq T_p \quad (2.19)$$

$$y_e(t) = \begin{cases} \frac{-16h_p}{T_p^3}t^3 + \frac{12h_p}{T_p^2}t^2 & 0 < t \leq T_p/2 \\ \frac{16h_p}{T_p^3}t^3 - \frac{36h_p}{T_p^2}t^2 + \frac{24h_p}{T_p}t - 4h_p & T_p/2 < t \leq T_p \end{cases} \quad (2.20)$$

$$x_c(t) = \frac{-2l_p}{T_p^3}t^3 + \frac{3l_p}{T_p^2}t^2 - \frac{l_p}{2} \quad 0 < t \leq T_p \quad (2.21)$$

$$y_c(t) = \begin{cases} \frac{-16h_c}{T_p^3}t^3 + \frac{12h_c}{T_p^2}t^2 + h_{cmin} & 0 < t \leq T_p/2 \\ \frac{16h_c}{T_p^3}t^3 - \frac{36h_c}{T_p^2}t^2 + \frac{24h_c}{T_p}t + h_{cmin} - 4h_c & T_p/2 < t \leq T_p \end{cases} \quad (2.22)$$

Para poder obtener las trayectorias mostradas en la Figura 7, se usan los datos de la Tabla 1, los cuales son valores derivado de la caminata humana [43].

Parámetro	T_p	T_m	l_p	h_p	l_{max}	h_{cmin}	h_{cmax}
Valor	0.76 [s]	0.38 [s]	0.4 [m]	0.07 [m]	0.2 [m]	0.96 [m]	0.97 [m]

Tabla 1. Valor numérico de los parámetros para el diseño de la trayectoria.

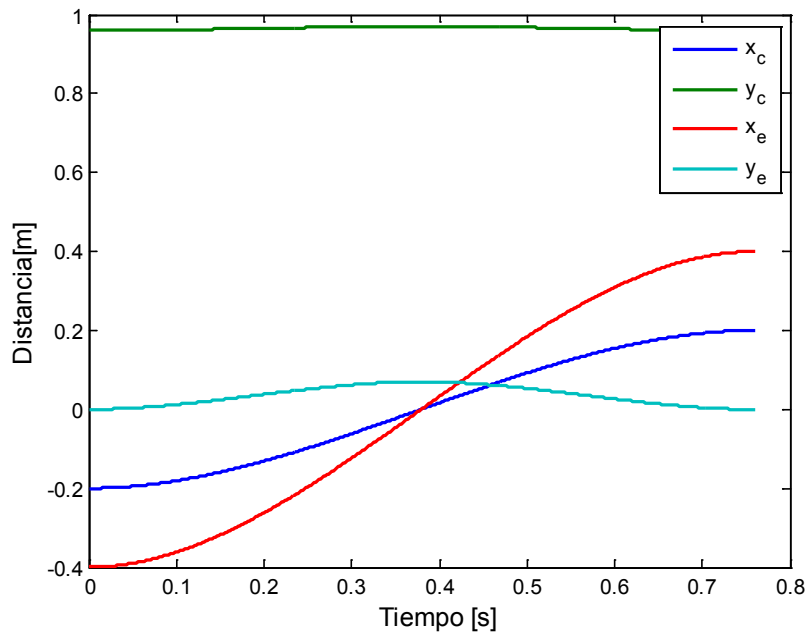


Figura 7. Trayectorias de la caminata en el plano sagital.

Una vez que se tienen las coordenadas de cada trayectoria necesitamos encontrar los ángulos de cada articulación. Para este propósito podemos obtener las ecuaciones en términos de los ángulos mostrados en la Figura 5 usando el modelo cinemático. A continuación se muestran las ecuaciones para los ángulos de cada articulación. El ángulo θ_3 , es considerado constante y dependerá de cada diseñador elegir su valor (por lo general es tomado como cero [44]).

$$\begin{aligned}
\theta_1(t) &= 2 \tan^{-1} \frac{2l_1 x_c(t) + \sqrt{-(l_1^2 - l_2^2)^2 + 2A(l_1^2 + l_2^2) - A^2}}{l_1^2 - l_2^2 + 2l_1 y_c(t) + A} \\
\theta_2(t) &= 2 \tan^{-1} \frac{2l_2 x_c(t) - \sqrt{-(l_1^2 - l_2^2)^2 + A((l_1^2 + l_2^2)^2 - A)}}{-l_1^2 + l_2^2 + 2l_2 y_c(t) + A} \\
\theta_3(t) &= 0.04 \\
\theta_4(t) &= 2 \tan^{-1} \frac{2l_2 B + \sqrt{-(l_5^2 - l_4^2)^2 + D((l_5^2 + l_4^2)^2 - D)}}{-l_5^2 + l_4^2 + 2l_2 C + D} \\
\theta_5(t) &= 2 \tan^{-1} \frac{2l_1 B - \sqrt{-(l_5^2 - l_4^2)^2 + 2D(l_5^2 + l_4^2) - D^2}}{l_5^2 - l_4^2 + 2l_5 C + D}
\end{aligned} \tag{2.23}$$

Donde:

$$\begin{aligned}
A &= x_c^2(t) + y_c^2(t) \\
B &= x_e(t) - l_1 \sin \theta_1(t) - l_2 \sin \theta_2(t) \\
C &= -y_e(t) + l_1 \cos \theta_1(t) + l_2 \cos \theta_2(t) \\
D &= B^2 + C^2
\end{aligned}$$

La Figura 8, muestra las posiciones y velocidades de cada articulación, usando una longitud de 0.5 [m] para cada eslabón. En la gráfica para las velocidades notamos que alrededor del tiempo T_p y $2T_p$ la velocidad es cero, por lo tanto, se verifica que las trayectorias diseñadas disminuyen los efectos del impacto.

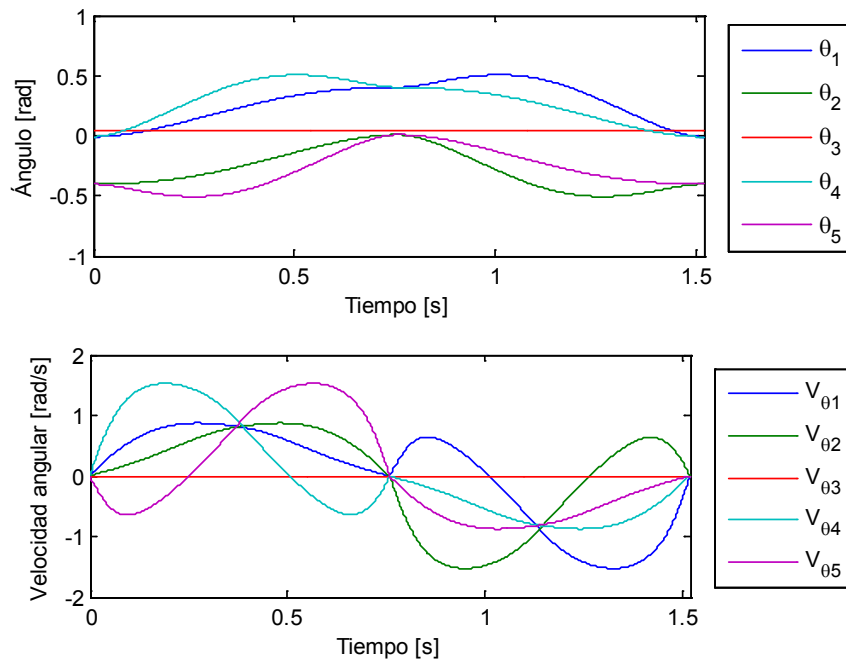


Figura 8. Posición y velocidad angular para un ciclo.

2.6 Modelado en SimMechanics

Usualmente, un modelo en *Simulink* es representado por ecuaciones algebraicas y diferenciales que predicen un estado futuro del sistema con los estados actuales [45]. Por otro lado, un modelo en *SimMechanics* es una representación de la estructura física de una máquina, especificada a través de algunas variables como masa, geometría y relaciones cinemáticas entre sus componentes. *SimMechanics* convierte esta representación en un modelo matemático equivalente [46]. Los pasos necesarios para generar y ejecutar un modelo en *SimMEchanics* son los siguientes:

- Especificar propiedades inerciales del cuerpo, grados de libertad, restricciones y las coordenadas del cuerpo para medir posiciones y velocidades.
- Configurar sensores y actuadores para registrar e iniciar movimientos del cuerpo, así como aplicar las fuerzas/torques.

- Ejecutar la simulación, y visualizar la animación del modelo usando *'Handle Graphics'* o la herramienta de realidad virtual.

El modelado de un robot bípedo puede ser dividido en tres partes: los eslabones, los actuadores y el modelo de la superficie de contacto. En la sección 2.3, se describe el robot bípedo. La Figura 9 muestra el diagrama del robot bípedo. Tres subsistemas pueden ser identificados, ambas piernas y el torso, y una señal de entrada que permite aplicar el torque a cada actuador generada por un controlador externo. Los valores usados para el modelado buscan asemejar el robot bípedo a una persona adulta de 75 kg y 1.70 m de altura.

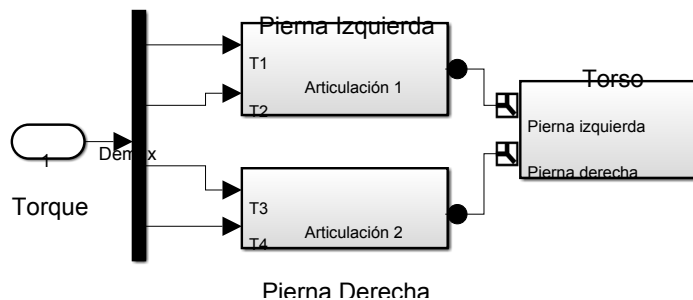


Figura 9. Diagrama de bloques general en Simulink del robot bípedo.

2.6.1 Modelado del torso

El modelo del torso es compuesto de un cuerpo rígido con conexiones a ambas piernas. La Figura 10 muestra el diagrama de bloques usando la herramienta SimMechanics. El bloque “Tierra” es un marco de referencia para todo el sistema. El bloque “plano” define dos grados de libertad traslacionales, ya que sólo nos interesa el comportamiento del robot en el plano sagital. El eslabón del torso es considerado como un paralelepípedo de lados a, b y c alineado a lo largo de los ejes x, y, z respectivamente. Con la ayuda de la ecuación (2.8) encontramos la matriz de inercia correspondiente.

La Tabla 2 especifica el valor de cada parámetro usado para el torso para configurar el bloque “Eslabón”.

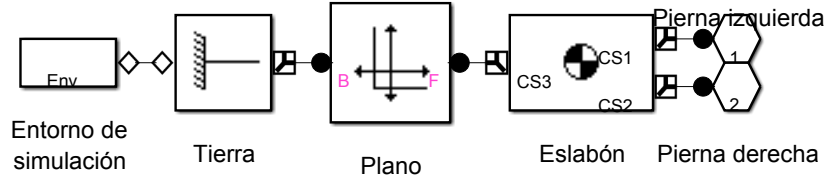


Figura 10. Diagrama de bloques para el torso (SimMechanics).

Parámetros del torso	a (m)	b (m)	c (m)	Masa (kg)
Valor	0.15	0.7	0.4	35

Tabla 2. Parámetros para el torso

$$I = \begin{bmatrix} \frac{m(b^2 + c^2)}{12} & 0 & 0 \\ 0 & \frac{m(a^2 + c^2)}{12} & 0 \\ 0 & 0 & \frac{m(a^2 + b^2)}{12} \end{bmatrix} \quad (2.24)$$

2.6.2 Modelado de la pierna

El modelo de la pierna está compuesto por dos eslabones (el muslo y el pie). La Figura 11 muestra el diagrama de bloque para la pierna. El bloque “Superficie de contacto” simula las fuerzas de reacción con el suelo. Los eslabones de la pierna son representados como un cilindro de radio R y altura h alineado a lo largo del eje y , la ecuación (2.8) muestra la forma de encontrar la matriz de inercia para un cilindro. Ambos eslabones son considerados iguales, la Tabla 3 resume el valor usado para cada parámetro.

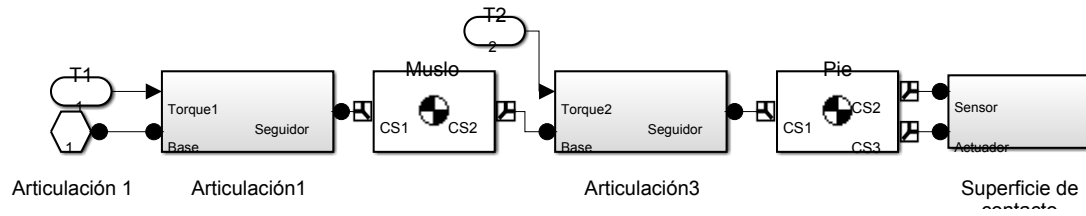


Figura 11. Diagrama de bloque para la pierna.

Parámetros del muslo y el pie	R (m)	h (m)	Masa (kg)
Valor	0.15	0.7	8.75

Tabla 3. Parámetros para la pierna.

$$I = \begin{bmatrix} \left(\frac{m}{4}\right)\left(\frac{3R^2}{5} + h^2\right) & 0 & 0 \\ 0 & \left(\frac{3mR^2}{10}\right) & 0 \\ 0 & 0 & \left(\frac{m}{4}\right)\left(\frac{3R^2}{5} + h^2\right) \end{bmatrix} \quad (2.25)$$

La Figura 12 expone el diagrama de bloques para el subsistema “Articulación1”. La articulación en este caso es representada por una revoluta, el bloque “Sensor” mide la posición y velocidad, las cuales son enviadas a una variable global (Estados1). El bloque “IC” establece la condición inicial y el bloque “Actuador” permite aplicar el torque a la unión.

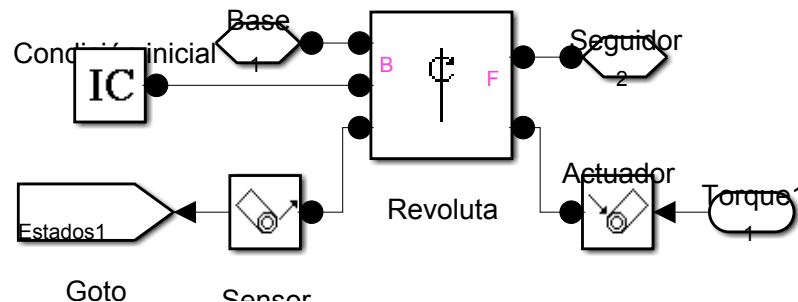


Figura 12. Diagrama de bloque para el subsistema actuador

2.6.3 Modelado de la superficie de contacto

La caracterización de la superficie de contacto es importante durante el proceso de simulación de un robot bípedo. El robot es soportado por las fuerzas de reacción del suelo pero no atraído a ella [47]. Cuando la punta de la pierna toca el suelo, las fuerzas normal y tangencial son aplicadas. La fuerza normal, alineada a lo largo del eje y , es modelada como un resorte con amortiguamiento [48]. La ecuación (2.26) representa la fuerza dinámica, donde y es la coordenada de la punta de la pierna, k_y es la constante elástica de la superficie, y b_y es el coeficiente de amortiguamiento. La fuerza normal es limitada a valores positivos para prevenir que la pierna quede sujeta al suelo. Por otra parte, la fuerza tangencial está alineada al eje x , esta incluye la fricción del sistema, representada por la ecuación (2.8), donde x es la coordenada actual y μ_x la constante de fricción.

$$F_n = -k_y y - b_y \dot{y} \quad (2.26)$$

$$F_t = -\mu_x \dot{x} \quad (2.27)$$

La Figura 13 ilustra el diagrama de bloques para el modelo de la superficie; la posición en el eje y es comparada con el nivel de la superficie, la señal resultante selecciona el valor de la fuerza normal y tangencial, la cual es cero si la posición del pie es mayor al nivel del piso. Cabe señalar que la fuerza en la coordenada z es cero porque en el experimento descrito solo se consideran los movimientos en el plano sagital.

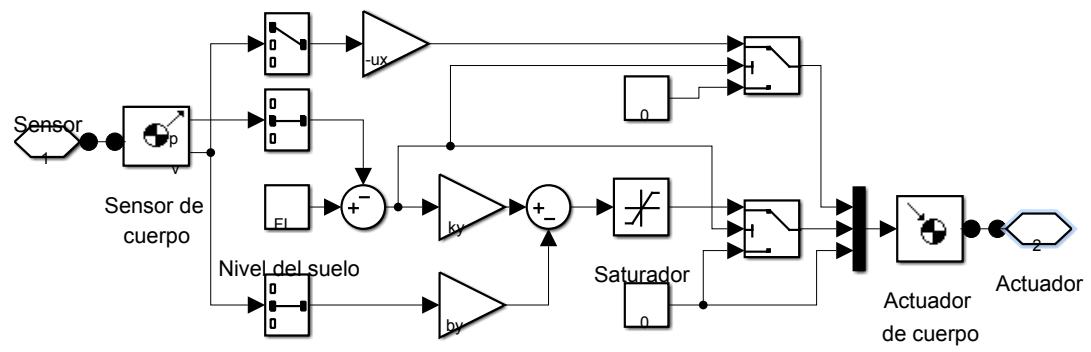


Figura 13. Diagrama de bloques para el modelo de la superficie.

3 TEORÍA Y DISEÑO DEL CONTROLADOR EN MODO DESLIZANTE CON ESTIMACIÓN DE LA PERTURBACIÓN (SMCPE)

Los sistemas reales presentan comportamientos no lineales, los cuales debido a su complejidad son usualmente tratados mediante algunas simplificaciones hacia aproximaciones lineales. Sin embargo, dichas aproximaciones no son viables cuando se presentan grandes variaciones de parámetros, resultando en grandes errores en el cálculo de las variables.

Debido a esto, se presenta el control en modos deslizantes con estimación de la perturbación como una alternativa robusta y novedosa para efectuar el control de sistemas no lineales. El SMCPE permite reducir el desgaste en los actuadores, al evitar la sobrestimación de la perturbación presente en los métodos no lineales convencionales tales como el control en modo deslizante (SMC por sus siglas en inglés), el cual estima el ancho de la superficie para asegurar robustez ante la máxima perturbación permitida. Esta técnica estima de forma precisa las perturbaciones en los estados presentes, permitiendo así cambiar el ancho de dicha superficie con base en las perturbaciones actuales. Como beneficio, las ganancias de control son menos rígidas y más suaves que con acciones equivalentes en el SMC.

En este capítulo, primero se tratan las bases teóricas del control en modo deslizante con estimación de la perturbación propuesta por Elmali y Olgac [49]. Además, los conceptos de control en tiempo retardado y control robusto son tratados por su gran importancia para entender el algoritmo de control propuesto. Por último se describe el algoritmo usado para el diseño del controlador y las diferencias al variar los parámetros de diseño elegidos arbitrariamente.

3.1 Estructuras Variables

Un sistema de estructuras variables se compone de dos o más subsistemas continuos y una lógica que realiza la conmutación entre ellos en función de los estados del sistema. La acción de control resultante de esta ley de conmutación es una función discontinua de los estados. El modo de operación llamado régimen deslizante se obtiene cuando las conmutaciones ocurren a frecuencias muy elevadas, restringiendo la trayectoria de los estados del sistema [50]. Los sistemas de estructuras variables tienen sus raíces en la teoría de control *Bang-Bang* (conocido como controlador de histéresis), el cual cambia bruscamente entre dos estados [51].

Los sistemas de estructuras variables están afectados por el efecto *chattering*, que consiste en oscilaciones indeseadas que se produce debido a la conmutación de alta frecuencia de la señal de control, el cual es perjudicial en aplicaciones prácticas donde el actuador presenta desgaste [52].

3.2 Fundamentos de control en modo deslizante

El control en modo deslizante es una técnica de control robusto que permite llevar y mantener al sistema en un estado de estabilidad, conocido como superficie deslizante, sin importar las perturbaciones que actúen de forma directa o indirecta sobre la planta.

Las leyes de control para sistemas no lineales presentan varias propiedades atractivas, entre ellas la robustez frente a incertidumbres de los parámetros y dinámicas no modeladas, inmunes a perturbaciones y límites de conmutación dependientes de los estados. Esta última característica permite disminuir la magnitud del *chattering* asociado con los regímenes deslizantes, mejorando así la regulación sobre la salida del sistema que se desea controlar [53].

La idea principal del SMC, es llevar el sistema a un espacio de control en donde los estados se mantengan dentro de un rango aceptable, conocido como superficie deslizante, la cual representa la variación del error en la salida y sus derivadas en función de un polo de desempeño deseado. Uno de los aspectos más interesantes del modo deslizante es la naturaleza discontinua de la acción de control. Esta característica del sistema da como resultado un rendimiento magnífico, ya que incluye la baja sensibilidad a las variaciones de los parámetros y el completo rechazo a perturbaciones.

El diseño se compone de dos pasos: el primero consiste en diseñar una superficie que limita el máximo valor de la acción de control en función de las incertidumbres de la planta y las perturbaciones externas. El segundo paso consiste en diseñar una ley de control para proporcionar la convergencia de la trayectoria del sistema hacia la superficie deslizante, la cual debe ser alcanzada en un tiempo finito. Así mismo, esta ley de control está dividida en dos partes: la ley de control del modo deslizante y la del modo de alcance. La primera es responsable de mantener la dinámica del sistema controlado sobre la superficie deslizante (comportamiento en lazo cerrado), mientras la segunda se diseña para alcanzar a la superficie deseada desde cualquier punto en el espacio de estados [54]

Para entender el comportamiento del algoritmo de control, se considera un sistema dinámico no-lineal de orden 'n' [49]:

$$\begin{aligned} \dot{x}^{(n)} = & f(X_1, \dots, X_m) + \Delta f(X_1, \dots, X_m) \\ & + [B(X_1, \dots, X_m) + \Delta B(X_1, \dots, X_m)]u + d(t) \end{aligned} \quad (3.1)$$

Donde $X_i = [x_i, \dot{x}_i, \dots, x_i^{(n_i-1)}]^T \in \mathbb{R}^{n_i}, i = 1, \dots, m$ es el vector de sub-estados el cual forma el vector de espacio de estados $[X_1^T, X_2^T, \dots, X_m^T]^T \in \mathbb{R}^r$, $r = \sum_{i=1}^m n_i, x_i, i = 1, \dots, m$ para m coordenadas independientes; $f =$

$[f_1, f_2, \dots, f_m]^T \in \mathbb{R}^m$ y $\Delta \mathbf{f} = [\Delta f_1, \Delta f_2, \dots, \Delta f_m]^T \in \mathbb{R}^m$ son vectores de campo correspondientes a los términos no lineales y sus incertidumbres respectivamente; $\mathbf{B} = [b_{ij}] \in \mathbb{R}^{m \times m}$ y $\Delta \mathbf{B} = [\Delta b_{ij}] \in \mathbb{R}^{m \times m}$, $i, j = 1, \dots, m$ son matrices que representan las ganancias de control y sus incertidumbres, respectivamente; $\mathbf{d} = [d_1, d_2, \dots, d_m]^T \in \mathbb{R}^m$ es el vector de las perturbaciones del sistema y $\mathbf{u} = [u_1, u_2, \dots, u_m]^T \in \mathbb{R}^m$ es el vector de entradas de control; $\mathbf{x}^{(n)} = [x_1^{(n_1)}, x_2^{(n_2)}, \dots, x_m^{(n_m)}]^T \in \mathbb{R}^m$, $x_i^{(n_i)} \in \mathbb{R}$ con $x_i^{(k)} = \frac{d^k(x_i)}{dt^k}$, $\dot{x}_i = \frac{d(x_i)}{dt}$. Las incertidumbres $\Delta \mathbf{f}$ y $\Delta \mathbf{B}$, y el vector de perturbaciones \mathbf{d} no se conocen pero se asume que están limitadas por funciones continuas \mathbf{X}_i , $i = 1, \dots, m$ y t . Los límites superiores de están dados por:

$$|\Delta f_j| \leq \mathbf{F}_j(X_1, \dots, X_m, t), \quad j = 1, \dots, m \quad (3.2)$$

$$|\Delta b_{ij}| \leq \mathbf{B}_{ij}(X_1, \dots, X_m, t), \quad i, j = 1, \dots, m \quad (3.3)$$

$$|d_j| \leq \mathbf{D}_j(X_1, \dots, X_m, t), \quad j = 1, \dots, m \quad (3.4)$$

El primer objetivo del control deslizante es satisfacer la condición de atracción, la cual se basa en una conmutación discontinua de la acción de control y se representa matemáticamente mediante:

$$\mathbf{s}^T \dot{\mathbf{s}} \leq 0 \quad (3.5)$$

Donde \mathbf{s} representa el plano de deslizamiento, el cual es seleccionado como un polinomio tipo Hurwitz de los errores asociados con los estados del controlador. Un polinomio de Hurwitz es un polinomio con raíces reales negativas, es decir, se encuentran en el lado izquierdo del plano complejo.

La tarea del proceso de control es lograr que el vector de estado x siga un vector deseado del sistema x_r , luego el error de seguimiento puede definirse como:

$$e_n = x - x_r = [e_n, \dot{e}_n, \dots, e_n^{n-1}]^T \quad (3.6)$$

La superficie deslizante está definida en el espacio de estado R^n por la ecuación escalar dada por:

$$s(e_n, t) = \left(\frac{d}{dt} + c \right)^{n-1} \cdot e_n \quad (3.7)$$

Donde n es el orden del sistema y c es una constante real estrictamente positiva correspondiente a la pendiente de la superficie deslizante. El trabajar con un polinomio de grado reducido $(n - 1)$ permite además de controlar la salida, suavizar la respuesta de sus derivadas, lo cual es de gran importancia para sistemas de control con variables derivadas.

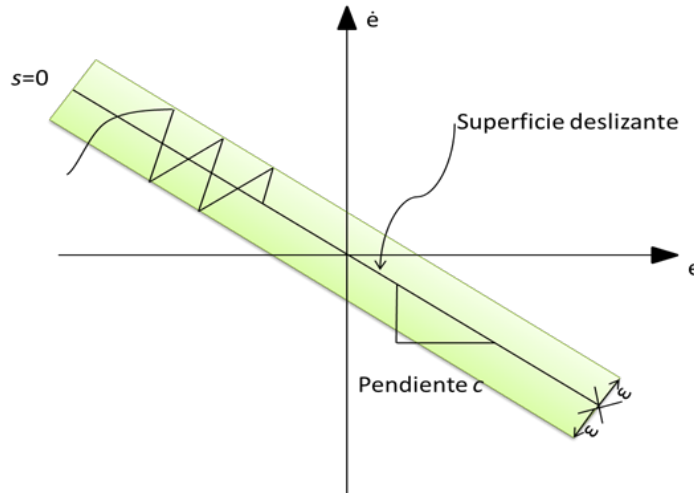


Figura 14. Región deslizante en el plano fase con chattering para un sistema de tercer orden.

Para el caso de la figura el orden $n = 3$,

$$s(e_n, t) = \left\langle \frac{d}{dt} + c \right\rangle^{n-1} \cdot e_n = \ddot{e}_n + 2c\dot{e}_n + c^2 e_n \quad (3.8)$$

El parámetro de diseño c interpretado como la pendiente de la superficie deslizante tiene gran influencia en el desempeño del sistema. Al ser un parámetro elegido *ad hoc* sus posibles valores deben ser validados mediante simulación, teniendo en cuenta que se presentará *chattering* en la señal de control y en la salida del sistema, cuando se exige un comportamiento mayor a la respuesta natural de la planta.

La condición 1, (3.5), no es posible en la práctica debido a que la conmutación de los componentes reales no es instantánea, por lo cual se presentaría un movimiento en la dirección de la superficie deslizante $s(t) = 0$, como se observa en la Figura 14. Con el fin de eliminar el problema en la región deslizante, se considera un margen de oscilación por lo cual la nueva condición se puede expresar como [54]:

$$s \cdot \dot{s} \leq -k \cdot \text{sgn}(s) \quad (3.9)$$

Donde k es una constante real positiva. A continuación se muestra la forma de la acción de control y la dinámica deseada en ausencia de perturbaciones.

$$\mathbf{u} = \mathbf{B}^{-1}[-P \cdot \mathbf{s} - \mathbf{K} \cdot \text{sgn}(\mathbf{s}) - \mathbf{f} - \dot{\boldsymbol{\xi}} + \mathbf{x}_d^{(n)}] \quad (3.10)$$

$$\dot{\mathbf{s}} = -P \cdot \mathbf{s} - \mathbf{K} \cdot \text{sgn}(\mathbf{s}) \quad (3.11)$$

En (3.10) P es una constante positiva real, y:

$$\mathbf{s} = [s_1, s_2, \dots, s_m]^T \quad (3.12)$$

$$\mathbf{K} = \text{diag}(k_j), \quad k_j > 0, \quad j = 1, \dots, m \quad (3.13)$$

$$\mathbf{sgn}(s) = [\text{sgn}(s_1), \dots, \text{sgn}(s_m)]^T, \quad \text{sgn}(s_j) = \text{signum}(s_j) \quad (3.14)$$

$$\boldsymbol{\xi} = [\xi_1, \xi_2, \dots, \xi_m]^T, \quad \xi_j = \sum_{k=1}^{n_j-1} c_{j,k} e_j^{(k)} \quad (3.15)$$

$$\mathbf{x}_d^{(n)} = [x_{d_1}^{(n_1)}, x_{d_2}^{(n_2)}, \dots, x_{d_m}^{(n_m)}]^T \quad (3.16)$$

En presencia de perturbaciones, la dinámica de la superficie deslizante toma la siguiente forma:

$$\dot{s} = -P \cdot s - \mathbf{K} \cdot \mathbf{sgn}(s) + \Delta \mathbf{f} + \mathbf{d} + \Delta \mathbf{B} \mathbf{u} \quad (3.17)$$

Para mantener la condición de atracción, la ganancia (k_j) debe asegurar control para el peor de los casos:

$$k_j \geq (\Delta \mathbf{f} + \mathbf{d} + \Delta \mathbf{B} \mathbf{u})_{j, \text{para el peor de los casos}} \quad (3.18)$$

Con el fin de eliminar el problema de causalidad, donde se encuentran acciones de control desconocidas, el valor actual de dicha acción, \mathbf{u} , se sustituye por una más reciente $\mathbf{u}(t - \delta)$, donde δ es el tiempo de muestreo el cual debe ser muy pequeño para que la aproximación sea válida [55]. Este reemplazo reduce el efecto *chattering* en la acción de control. En consecuencia, la nueva constante para la acción discontinua que garantiza la robustez frente a las perturbaciones actuales es:

$$k_j \geq F_j + D_j + \sum_{k=1}^m B_{jk} |u_k(t - \delta)|, \quad j = 1, \dots, m \quad (3.19)$$

En (3.17) y (3.18) la función signo se define como:

$$\mathbf{sgn}(s) = \frac{|s|}{s} = \begin{cases} 1; & \text{si } s > 0 \\ 0; & \text{si } s = 0 \\ -1; & \text{si } s < 0 \end{cases} \quad (3.20)$$

Con el fin de suavizar la conmutación, la función discontinua signo se reemplaza por una función tangente hiperbólica [34], sin embargo por simplicidad es utilizada la función de saturación:

$$\mathbf{K} \cdot \mathbf{sgn}(s) = \mathbf{K} \mathbf{sat}\left(\frac{s}{\varepsilon}\right) \quad (3.21)$$

La cual está definida como:

$$\mathbf{sat}\left(\frac{s}{\varepsilon}\right) = \begin{cases} \mathbf{sgn}\left(\frac{s}{\varepsilon}\right); & \text{cuando } |s| > \varepsilon \\ \frac{s}{\varepsilon}; & \text{cuando } |s| \leq \varepsilon \end{cases} \quad (3.22)$$

Donde ε es el espesor de la superficie de deslizamiento.

Las características de filtro pasa bajas deseado discutido anteriormente pueden ser alcanzadas al seleccionar la frecuencia de corte mediante:

$$\mathbf{eig}\left\{\left(\mathbf{I} + \Delta \mathbf{B} \mathbf{B}^{-1}\right) \left(\mathbf{P} \mathbf{I} + \frac{1}{\varepsilon} \mathbf{K}\right)\right\}_{\max} \leq \omega_{\max} \quad (3.23)$$

Usando propiedades de las matrices se llega a una expresión para el espesor de la superficie:

$$\varepsilon = \frac{2k_{max}}{\omega_{max} - 2P} \omega_{max} - 2P > 0 \quad (3.24)$$

La selección de este límite es un equilibrio entre la precisión y la robustez. La expresión para calcular el espesor de la capa límite (3.24) debe cumplirse rigurosamente.

3.3 Fundamentos de control en modo deslizante con estimación de la perturbación.

El SMC básico presenta dos grandes problemas: requiere conocer previamente los límites superiores de todas las perturbaciones (lo cual no siempre es posible) y tiene una limitada integridad de respuesta (es poco usual que un sistema opere en el peor de los casos) a causa de la rigidez de los parámetros. Para el caso de SMC (3.19) la ganancia debe ser estrictamente calculada con base en las peores condiciones del sistema para mantener la atracción de la superficie en todo momento, dando como resultado ganancias relativamente altas para aplicaciones prácticas.

En esta sección, se intentan corregir estos problemas introduciendo el concepto de ‘Control en tiempo retardado’ [55], según el cual la continuidad de la acción de control permite aproximar el efecto implícito de la misma con su valor inmediatamente anterior.

Partiendo de lo anterior, la perturbación del sistema $\Psi(t)_{actual}$ puede calcularse al restar la máxima derivada de la entrada calculada junto a las relaciones f y B , las cuales son funciones de los estados del controlador:

$$\Psi(t)_{actual} = \Delta f + \Delta B u + d = x^{(n)} - f - B u \quad (3.25)$$

Si todos los componentes presentan variaciones lentas con respecto al muestreo en lazo cerrado, la ecuación (2.8) se puede escribir con $\mathbf{u}(t - \delta)$ en lugar de $\mathbf{u}(t)$. Es importante aclarar que $\mathbf{x}^{(n)}$ se calcula a partir de $\mathbf{x}^{(n-1)}$ mediante derivación numérica. Si las mediciones de $\mathbf{x}^{(n-1)}$ son ruidosas, lo cual es común, deben filtrarse para evitar grandes variaciones al momento de calcular $\mathbf{x}^{(n)}$.

La ecuación (3.10), correspondiente a la acción de control obtenida, puede reescribirse incluyendo la estimación de las perturbaciones:

$$\mathbf{u} = \mathbf{B}^{-1}[-\mathbf{P}s - \mathbf{K}sgn(s) - \mathbf{f} - \boldsymbol{\xi} + \mathbf{x}_d^{(n)} - \boldsymbol{\Psi}(t)_{Estimado}] \quad (3.26)$$

donde:

$$\begin{aligned} \boldsymbol{\Psi}(t)_{Estimado} = & \mathbf{x}_{Calculado}^{(n)} - \mathbf{f}(X_1(t), \dots, X_m(t)) \\ & - \mathbf{B}(X_1(t), \dots, X_m(t))\mathbf{u}(t - \delta) \end{aligned} \quad (3.27)$$

Las correspondientes dinámicas de s corresponden a:

$$\dot{s} = -\mathbf{P}s - \mathbf{K}sgn(s) + \boldsymbol{\Psi}(t)_{actual} - \boldsymbol{\Psi}(t)_{Estimado} \quad (3.28)$$

Si la diferencia de $|\boldsymbol{\Psi}(t)_{actual} - \boldsymbol{\Psi}(t)_{Estimado}|$ se mantiene dentro de un límite $\eta|\boldsymbol{\Psi}(t)_{Estimado}|$, con $\eta > 0$, la condición de atracción de (3.5) se puede asegurar seleccionando los k_j como:

$$[k_1, \dots, k_m]^T = \eta|\boldsymbol{\Psi}(t)_{Estimado}| \geq |\boldsymbol{\Psi}(t)_{actual} - \boldsymbol{\Psi}(t)_{Estimado}| \quad (3.29)$$

Nótese que en comparación con las altas ganancias del control en modo deslizante ahora los k_j son más pequeños, de donde radica la gran ventaja del control SMCPE. Esta reducción se debe a la forma de calcular el término

de la acción discontinua mediante la estimación de la perturbación actual en lugar de la máxima perturbación posible [49].

Utilizando la función saturación para remplazar la función signo, la dinámica s asume el siguiente comportamiento dentro de la frontera.

$$\dot{s} + (PI + \bar{K})s = \Psi(t)_{actual} - \Psi(t)_{Estimado} \quad (3.30)$$

Donde $\bar{K} = \text{diag}(\frac{k_j}{\varepsilon_j})$. La dinámica de s se reduce a una forma similar a (3.17), salvo con una amplitud mucho menor de las condiciones de conducción. Por lo tanto, se obtienen espesores ε_j más pequeños.

La dinámica de s en (3.30) es bastante complicada debido a los términos $\Psi(t)_{actual}$ y $\Psi(t)_{Estimado}$, los que contienen influencias de s que son imposibles de aislar. En consecuencia, todas las contribuciones de s en los términos $\Psi(t)_{actual}$ y $\Psi(t)_{Estimado}$ deben ser filtrados para eliminar dinámicas no deseadas.

El comportamiento deseado del filtro pasa baja de s se obtiene mediante la limitación de su frecuencia de corte máxima a ω_{max} , la cual se puede lograr seleccionando el espesor de la frontera como:

$$\varepsilon_j = \frac{k_j}{\omega_{max} - P} \quad (3.31)$$

Todas estas consideraciones permiten que el control en modo deslizante mejore el desempeño en función de las incertidumbres y perturbaciones no conocidas. El cálculo en línea de los límites de la superficie mejora el desempeño y duración de los actuadores mecánicos los cuales son sometidos a la acción de control proveniente de estos controladores. Sin embargo, un amplio conocimiento de la planta y de su comportamiento

dinámico debe ser conocido ya que las funciones f y B deben ser precisas para garantizar la estabilidad del controlador [34].

3.4 Diseño del controlador

El algoritmo de control presentado en [49] define varios pasos claves requeridos durante el diseño de un controlador en modo deslizante con estimación de la perturbación. Primero, se computan los errores mediante representación numérica de los estados del controlador, los cuales son utilizados posteriormente en el cálculo de las variables s^8 y X_i , según las ecuaciones (3.7) y (3.15) respectivamente.

Luego, se debe expresar el comportamiento dinámico en espacio de estados en la forma canónica controlable, identificando como estados la posición y velocidad de cada articulación. Para simplificar el diseño solo se tendrá en cuenta la dinámica de la fase de soporte simple, dado que la fase de doble soporte es de corta duración y la fase de impacto puede ser despreciada por la forma en que se diseñaron las trayectorias. La ecuación (2.5) puede ser reescrita de la siguiente forma:

$$\ddot{\theta} = D^{-1}(\theta)(-H(\theta)\dot{\theta}^2 - G(\theta) + T_{\theta}) \quad (3.32)$$

De lo anterior es fácil determinar las ecuaciones de f y B , que son usadas junto a la acción de control inmediatamente anterior para estimar la perturbación en el sistema; según la ecuación (3.27).

⁸ Representa la distancia al plano de las variables del sistema a la superficie deslizante en el plano de fase.

$$\begin{aligned} \mathbf{f} &= \mathbf{D}^{-1}(\theta)(-\mathbf{H}(\theta)\dot{\theta}^2 - \mathbf{G}(\theta)) \\ \mathbf{B} &= \mathbf{D}^{-1}(\theta) \end{aligned} \quad (3.33)$$

Posteriormente se calcula el espesor variable ε de la superficie con (3.31). La función de saturación descrita en (3.22) es computada mediante comparadores lógicos. Por último se obtiene el torque de control u definido en (3.26).

El algoritmo de control es introducido mediante una función S^9 (Ver Anexo 6.2). Los bloques correspondientes a la planta y el controlador son modificados con base en el modelo dinámico utilizado. La Figura 15 presenta la distribución de bloques para realizar la simulación del sistema.

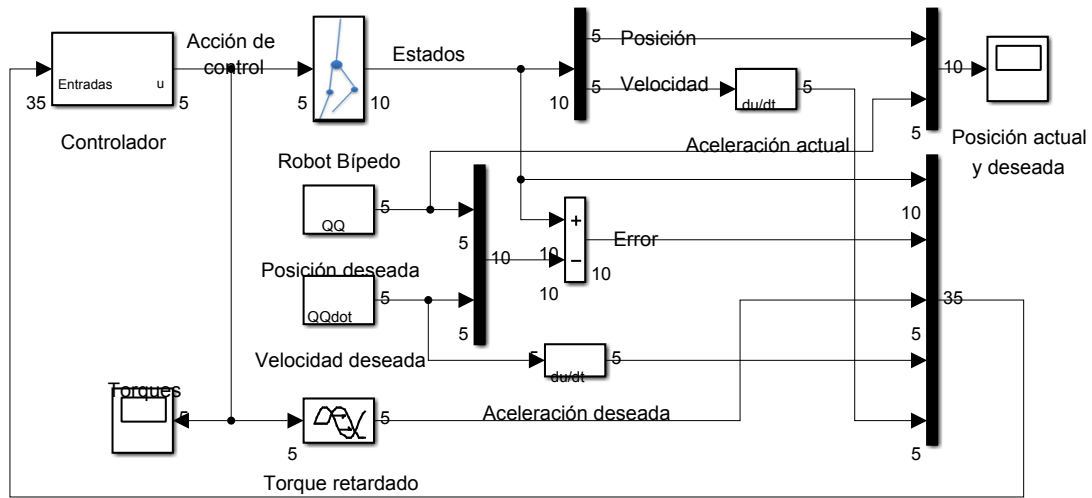


Figura 15. Diagrama de bloques del sistema completo.

⁹ Una función S es un lenguaje de descripción computacional para un bloque de *Simulink* escrito en MATLAB, C, C++ o Fortran.

3.4.1 Consideraciones de diseño

En el diseño de un controlador SMCPE, uno de los principales problemas es la elección de los parámetros *ad-hoc* mencionados anteriormente; la pendiente c de la superficie deslizante, el parámetro η que limita la estimación de la perturbación y ayuda a reducir el *chattering*, el polo P y la frecuencia de corte máximo ω_{max} que ayudan a determinar el ancho de la superficie. Encontrar un equilibrio entre estas constantes brindará una precisión y robustez aceptable. Teniendo en cuenta, cada uno de los efectos que produce las constantes, se realizaron diferentes simulaciones, donde se estudia la acción de control generada y el error, para modificar dichas constantes. A continuación, se muestra el resultado de la simulación para $c = 15$, $\eta = 0.2$, $\omega_{max} = 12$, $P = 10$, durante un ciclo de la marcha.

La Figura 16 muestra el torque generalizado, el error y la dinámica de la superficie s , para el controlador SMCPE diseñado. En el torque se puede observar un cambio brusco en el tiempo 0.759 [s], como consecuencia del impacto con la superficie de la pierna en balanceo. De la gráfica del error se puede resaltar que el pico máximo es de 0.097 grados para θ_2 . El comportamiento de la variable s no llega exactamente a cero, pero se mantiene dentro de la banda de tolerancia definida por el espesor de la superficie, por lo cual se puede asegurar que el controlador cumple las condiciones de diseño. Para visualizar mejor lo anterior la Figura 17 muestra la secuencia del robot bípedo durante un ciclo, obtenida con la herramienta de *SimMechanics*.

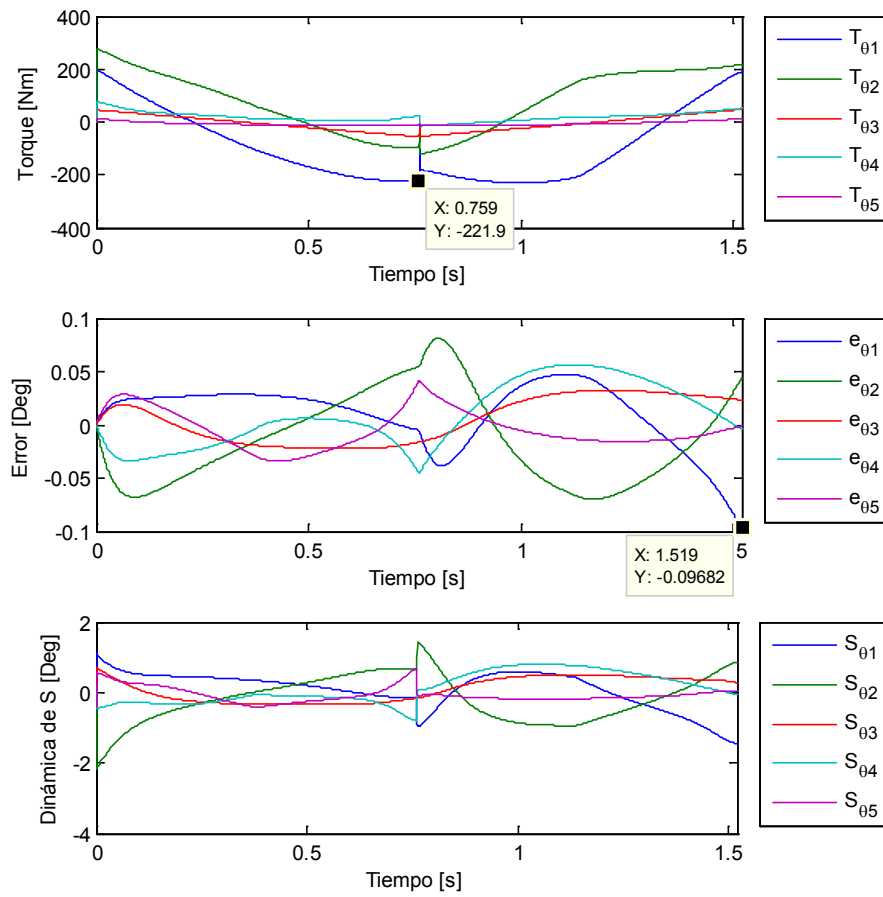


Figura 16. Comportamiento del torque generalizado, el error y la dinámica de la superficie s .

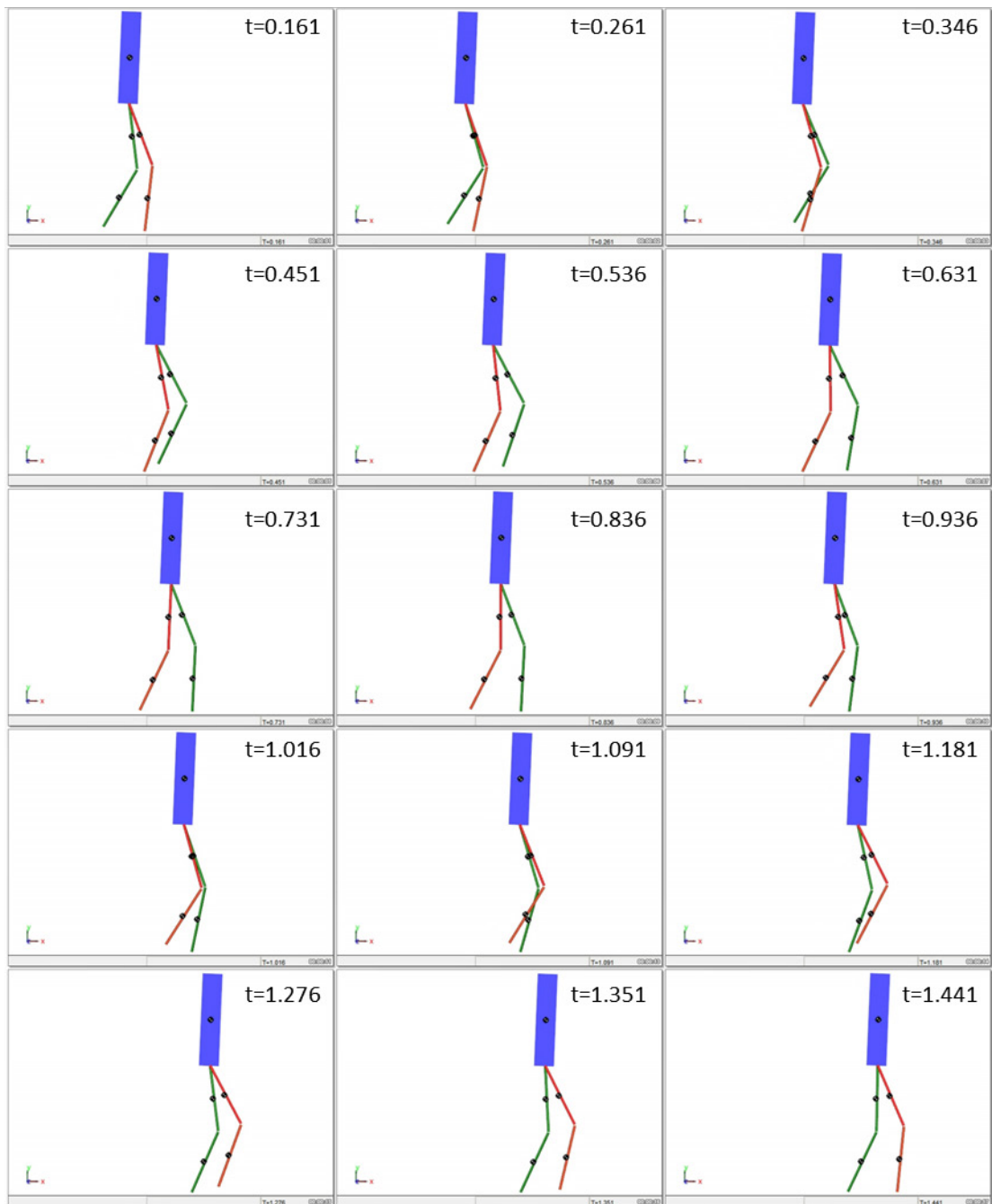


Figura 17. Resultado de la animación con *SimMechanics* en diferentes tiempos.

4 RESULTADOS Y ANÁLISIS DE DESEMPEÑO

En el diseño de controladores robustos es necesario medir cuantitativamente su comportamiento frente a diferentes perturbaciones. En este capítulo se presentan los resultados para dos diferentes pruebas de robustez, perturbación externa en el torque del torso e incertidumbre en los parámetros donde el desempeño del controlador diseñado es evaluado con la ayuda de tres índices de desempeño.

Durante la marcha un robot que se mueve en un entorno desconocido puede ser sometido a fuerzas externas que implican un desequilibrio y si esta fuerza es demasiado grande, entonces el robot puede caer [56]. Por consiguiente, es esencial que el controlador tenga la capacidad de reaccionar rápidamente con el fin de encontrar el equilibrio. Por lo tanto en esta simulación se aplicó al robot un torque de perturbación en el torso que lo empuja hacia adelante, como una forma de medir el desempeño del controlador frente a desplazamientos abruptos. Los resultados son presentados en la sección 4.2.1.

En cuanto a las incertidumbres en los parámetros es común encontrarse con que se desconoce algunos de los parámetros del sistema o que su valor fue estimado a través de métodos aproximados. En consecuencia el objetivo del controlador diseñado es conseguir ciertas prestaciones respecto a la estabilidad, error de posición u otras especificaciones a pesar de presentar incertidumbre en sus parámetros, interferencias externas y comportamientos dinámicos no modelados (flexibilidad de las articulaciones y enlaces, dinámica de los actuadores, fricción, ruido provocado por los sensores y dinámicas desconocidas del entorno). En esta prueba se busca evaluar el desempeño del controlador frente a variaciones porcentuales negativas y positivas de la masa y la longitud. Los resultados son presentados en la sección 4.2.2.

4.1 Definición de los índices y criterios de desempeño

Los índices de desempeño son medidas cuantitativas que especifican el costo de la operación del sistema en función del error y de la energía, con el fin de evaluar el comportamiento dinámico, la calidad de la respuesta transitoria y el esfuerzo del controlador [57]. A continuación se describen los índices de desempeños usados para evaluar el comportamiento del controlador diseñado.

4.1.1 Integral del error absoluto (IAE)

El IAE representa la cantidad de material fuera de especificaciones, energía perdida u otra característica indeseada. Se emplea en sistemas que no son excesivamente oscilatorios ni excesivamente lentos. Para un control perfecto el IAE es cero.

$$IAE = \int_0^{\infty} |e(t)| dt \quad (4.1)$$

4.1.2 Integral del error cuadrático (ISE)

Al igual que el IAE, este índice trata que el sobrepaso máximo sea lo menor posible de manera que el valor de la integral sea mínimo. La diferencia entre el criterio *IAE* y el *ISE*, consiste básicamente en que con el *ISE* se tiene más ponderación para errores grandes, los cuales se presentan generalmente al inicio de la respuesta, y menor ponderación para errores pequeños presentados al final de la respuesta.

$$ISE = \int_0^{\infty} [e(t)]^2 dt \quad (4.2)$$

4.1.3 Integral del valor absoluto del error ponderado en el tiempo (ITAE)

El ITAE es insensible a los errores iniciales, y a veces inevitables, pero penaliza fuertemente los errores que permanecen a lo largo del tiempo. Por utilizar como base la multiplicación del error y el tiempo es un buen indicador de la rapidez del sistema.

$$ITAE = \int_0^{\infty} t|e(t)| dt \quad (4.3)$$

4.2 Pruebas de robustez

La robustez del controlador SMCPE es investigada simulando dos situaciones diferentes: una fuerza externa actuando sobre el torso e incertidumbre en los parámetros de la planta.

4.2.1 Fuerza externa actuando sobre el torso

Como se mencionó anteriormente en esta prueba se busca estudiar el desempeño del controlador frente a desplazamientos abruptos en el torso, para ello se aplica un torque de perturbación en el ángulo θ_3 . El torque es aplicado en la fase de soporte simple, debido a que el robot bípedo es más propenso a caer al aplicar una perturbación. La Tabla 4 y la Figura 18 resume las pruebas realizadas para diferentes perturbaciones, la cuales fueron aplicadas a los 0.38 [s] de empezar la marcha con una duración de 0.2 [s], el torque aplicado por el controlador es saturado a ± 500 [Nm] para modelar la respuesta limitada de un actuador; durante esta prueba se observa que el controlador compensa la perturbación de forma adecuada hasta que empieza a entrar en los límites de saturación, con una perturbación de 490 y 500 [Nm] los índices de desempeño aumentan

considerablemente en la mayoría de los casos, donde la estabilidad del robot bípedo ya no puede ser asegurada.

	Torque [Nm]	θ_1	θ_2	θ_3	θ_4	θ_5
ISE	0	9.84e-07	1.36e-06	2.18e-07	3.09e-07	3.32e-07
	80	9.53e-07	1.51e-06	4.02e-07	4.64e-07	3.27e-07
	160	9.32e-07	1.79e-06	1.02e-06	7.46e-07	3.33e-07
	320	9.06e-07	2.72e-06	3.54e-06	1.69e-06	3.96e-07
	400	9.00e-07	3.37e-06	5.44e-06	2.34e-06	4.53e-07
	450	8.98e-07	3.84e-06	6.84e-06	2.82e-06	4.98e-07
	480	8.99e-07	4.38e-06	1.83e-05	5.46e-06	1.79e-06
	490	1.10e-05	4.45e-04	1.92e-03	4.76e-04	1.36e-04
	500	1.23e-04	1.51e-03	8.19e-03	1.75e-03	7.31e-04
IAE	0	1.01e-03	1.24e-03	5.27e-04	5.58e-04	5.56e-04
	80	9.83e-04	1.27e-03	6.63e-04	6.92e-04	5.58e-04
	160	9.62e-04	1.36e-03	9.59e-04	8.43e-04	5.59e-04
	320	9.29e-04	1.63e-03	1.57e-03	1.16e-03	5.79e-04
	400	9.21e-04	1.79e-03	1.89e-03	1.32e-03	6.17e-04
	450	9.17e-04	1.88e-03	2.08e-03	1.42e-03	6.47e-04
	480	9.26e-04	2.00e-03	3.21e-03	1.91e-03	1.10e-03
	490	2.68e-03	1.41e-02	2.82e-02	1.43e-02	7.36e-03
	500	7.49e-03	2.54e-02	5.77e-02	2.70e-02	1.69e-02
ITAE	0	8.18e-04	9.89e-04	4.66e-04	4.18e-04	4.75e-04
	80	8.03e-04	1.02e-03	5.59e-04	4.94e-04	4.73e-04
	160	7.93e-04	1.08e-03	7.37e-04	5.81e-04	4.71e-04
	320	7.78e-04	1.25e-03	1.11e-03	7.64e-04	4.79e-04
	400	7.76e-04	1.34e-03	1.30e-03	8.59e-04	5.02e-04
	450	7.75e-04	1.40e-03	1.42e-03	9.18e-04	5.20e-04
	480	7.80e-04	1.46e-03	2.18e-03	1.25e-03	8.03e-04
	490	1.92e-03	9.73e-03	1.92e-02	9.69e-03	4.98e-03
	500	5.19e-03	1.73e-02	3.93e-02	1.82e-02	1.14e-02

Tabla 4. Índices de desempeño para cada ángulo al aplicar diferentes torque de perturbación en el tiempo 0.38 [s] y duración de 0.2 [s]

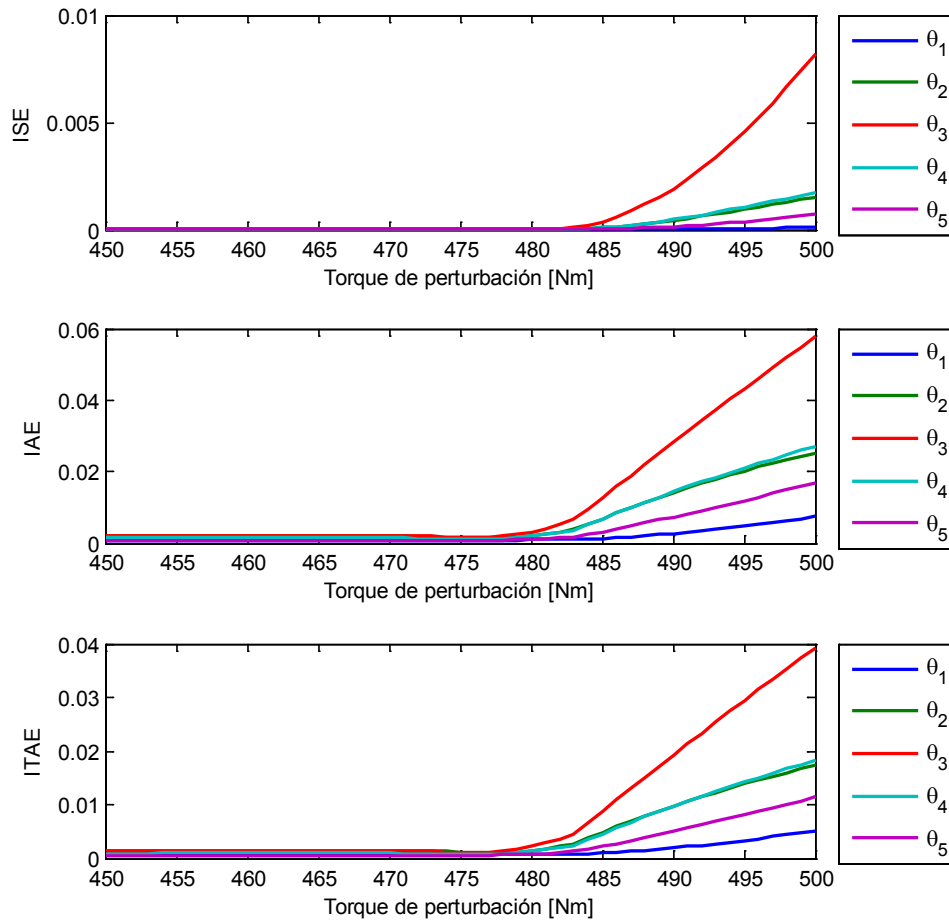


Figura 18. Índices de desempeño para cada ángulo al aplicar diferentes torques de perturbación en el tiempo 0.38 [s] y duración de 0.2 [s]

En las siguientes gráficas se muestra el torque y el error para cada ángulo al aplicar diferentes valores de perturbación. La Figura 19 muestra que el error máximo se presenta en θ_3 y no supera los 0.6 [Deg], a diferencia de la Figura 20 donde el error máximo es de 12.82 [Deg] con una perturbación de 500 [Nm]. De los resultados obtenidos se puede concluir que el controlador tiene un buen desempeño hasta una perturbación máxima de 480 [Nm], para perturbaciones superiores el torque aplicado entra en los límites del actuador, por lo cual el controlador no tiene la capacidad de llevar los

estados de forma rápida al espacio de control, comprometiendo el seguimiento adecuado de la trayectoria y por lo tanto la estabilidad del robot bípedo. Para visualizar lo anterior en la Figura 21 se observa la secuencia del robot bípedo al aplicar un torque de perturbación de 500 [Nm].

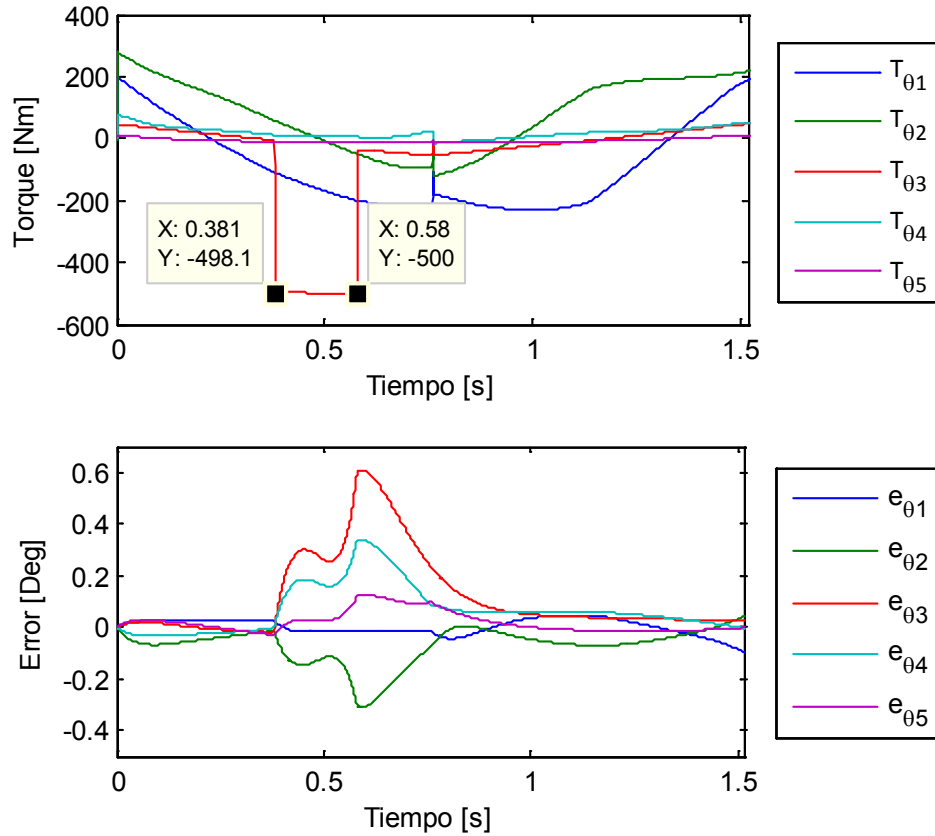


Figura 19. Torque y error al aplicar una perturbación de 480 [Nm] en el torque $T_{\theta 3}$.

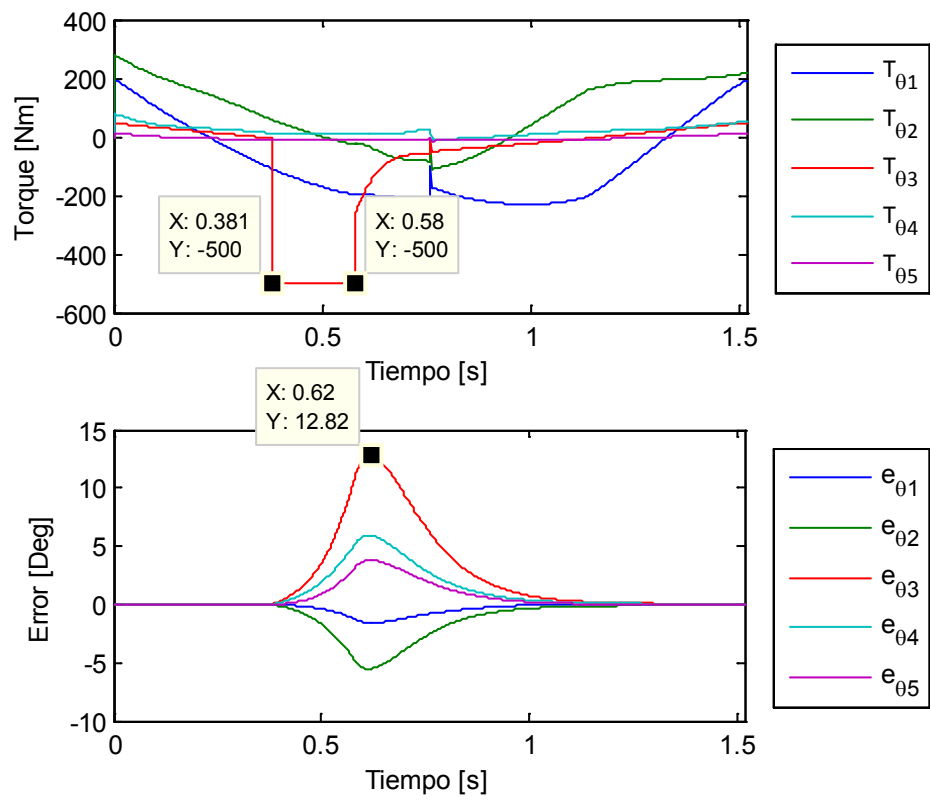


Figura 20. Torque y error al aplicar una perturbación de 500 [Nm] en el torque $T_{\theta 3}$.

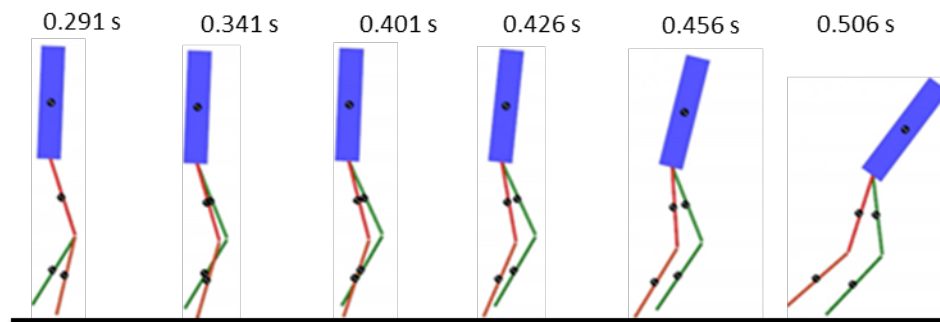


Figura 21. Secuencia del robot bípodo al aplicar una perturbación de 500 [Nm] en el tiempo 0.38 [s] y duración de 0.2 [s]

Como prueba final se analizó la dependencia de la duración y el valor máximo de perturbación para el cual el robot bípedo pierde su estabilidad. La Figura 22 muestra los resultados obtenidos, donde es claro que para duraciones menores a 0.1 [s] el valor máximo de la perturbación aumenta de manera exponencial, y para duraciones mayores la perturbación máxima aceptada empieza a disminuir lentamente.

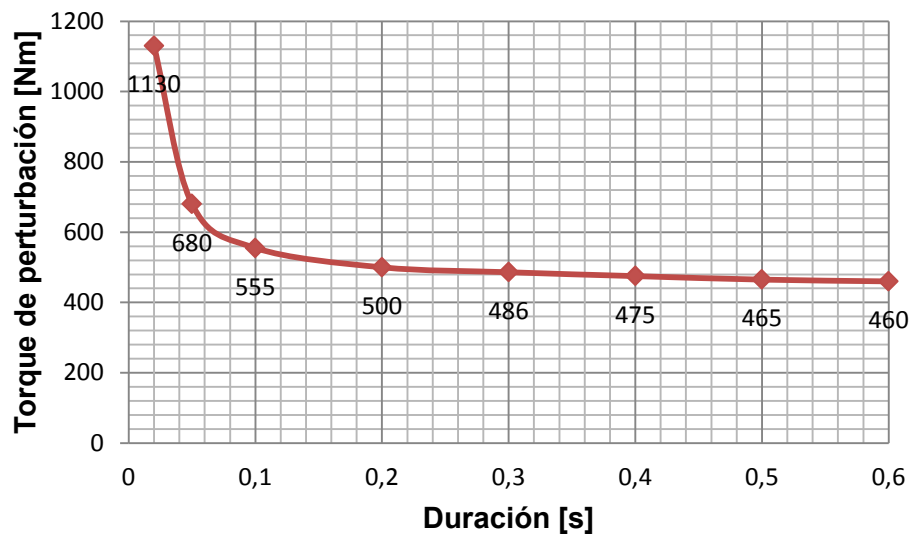


Figura 22. Máxima perturbación permitida contra duración.

4.2.2 Incertidumbre en los parámetros de la planta.

Esta prueba consiste en variar los parámetros de la planta en diferentes valores de porcentajes con los cuales fue diseñado el controlador, con el propósito de evaluar su desempeño. El modelo del robot bípedo usado para el diseño del controlador tiene cuatro parámetros importantes: la masa, la inercia, la longitud y la distancia al centro de masa de cada eslabón, para realizar esta prueba se varía la masa y la longitud, pero las inercias y el centro de masa son recalculados de acuerdo a las variaciones realizadas.

Se propone variar la longitud y la masa en el intervalo de -40% a 40% con aumentos de 1%, obteniendo de esta forma una cuadrícula de puntos para

calcular los índices de desempeño para cada ángulo, lo que finalmente representa 6581 simulaciones. La Figura 23 muestra los resultados obtenidos; el eje z representa el valor del desempeño, el eje y el porcentaje de variación en la masa y el eje x el porcentaje de variación en la longitud. Se puede inducir que el índice IAE presenta mayor sensibilidad a la incertidumbre de parámetros, ya que su valor es mayor en todos los casos.

El desempeño del controlador diseñado decae cuando la incertidumbre de la masa y la longitud se acercan al 40%, pero para los demás casos se presenta un buen desempeño mostrando la robustez del controlador SMCPE. Desde otra perspectiva con esta prueba se muestra la versatilidad del controlador diseñado para ser usado en un modelo de robot bípedo con parámetros completamente diferentes.

Como prueba complementaria se propone variar sólo las inercias de cada eslabón del robot en el intervalo de -50 a 50 [%] con aumentos de 2%; los resultados son resumidos en la Figura 24 donde el peor de los casos es presentado con una variación del -50 [%], por eso en la Figura 25 es mostrado el torque y el error para esta variación donde es claro que el error no supera el $\pm 1 [Deg]$, por lo tanto se puede concluir que el controlador presenta un buen desempeño para las variaciones de incertidumbres de la inercia en el rango estudiado.

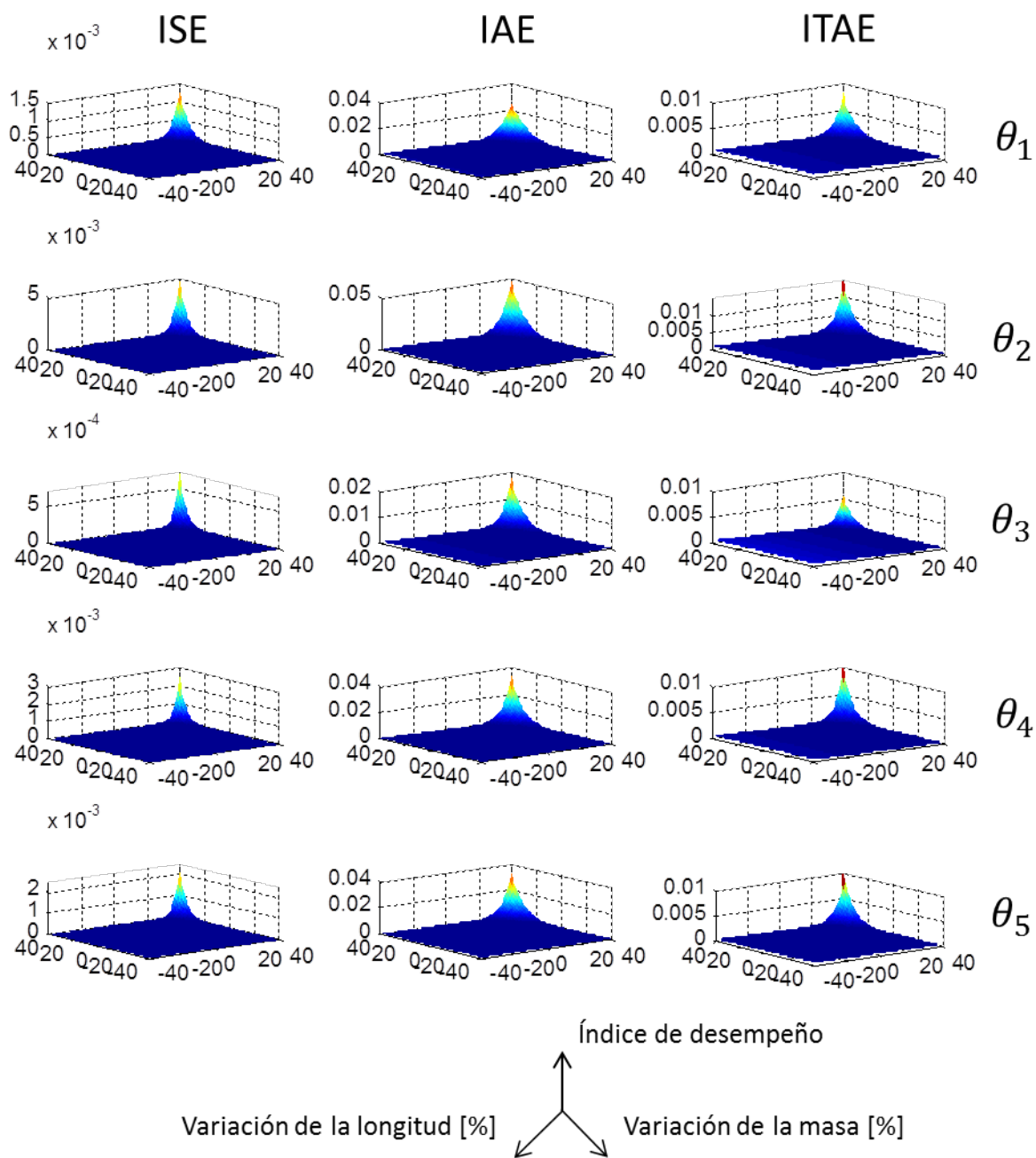


Figura 23 Resultado de la prueba de incertidumbre en los parámetros.

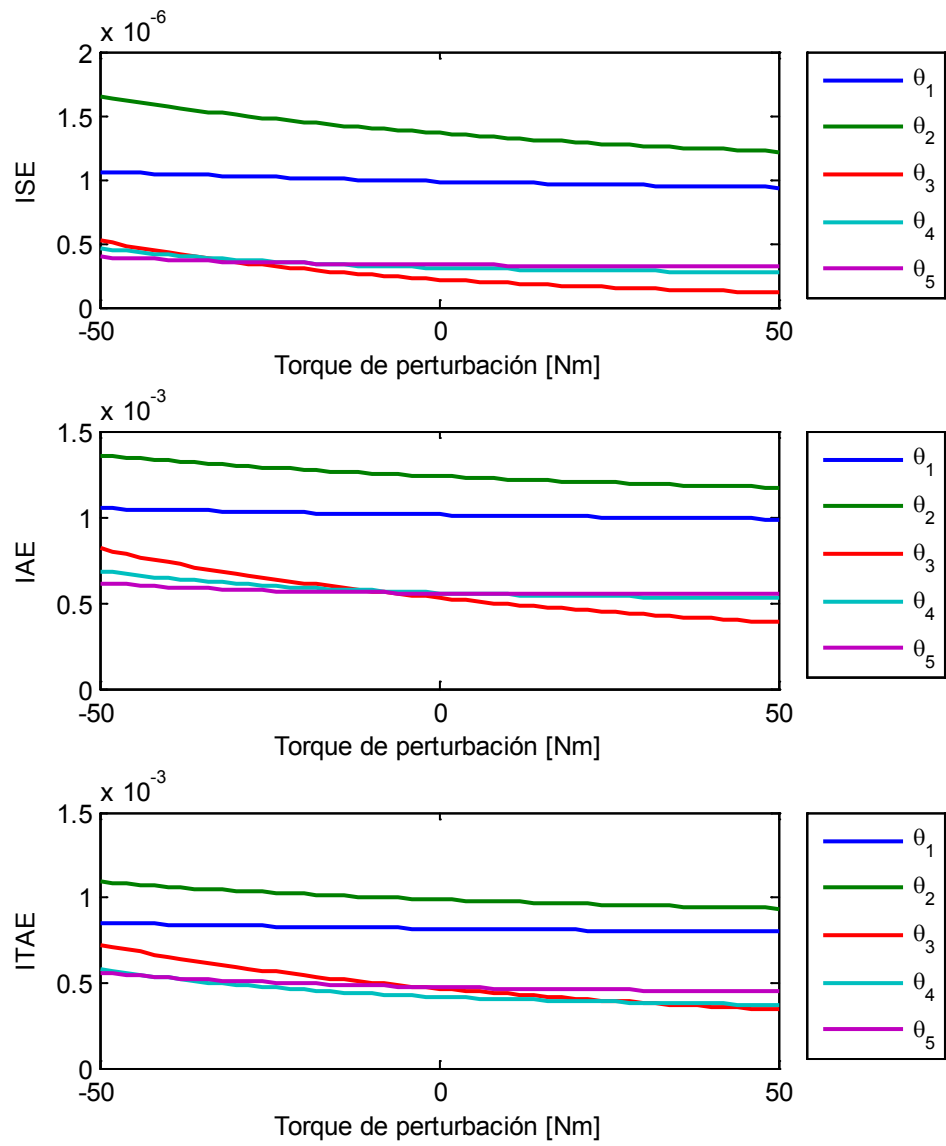


Figura 24. Índices de desempeño para variaciones porcentuales en la inercia.

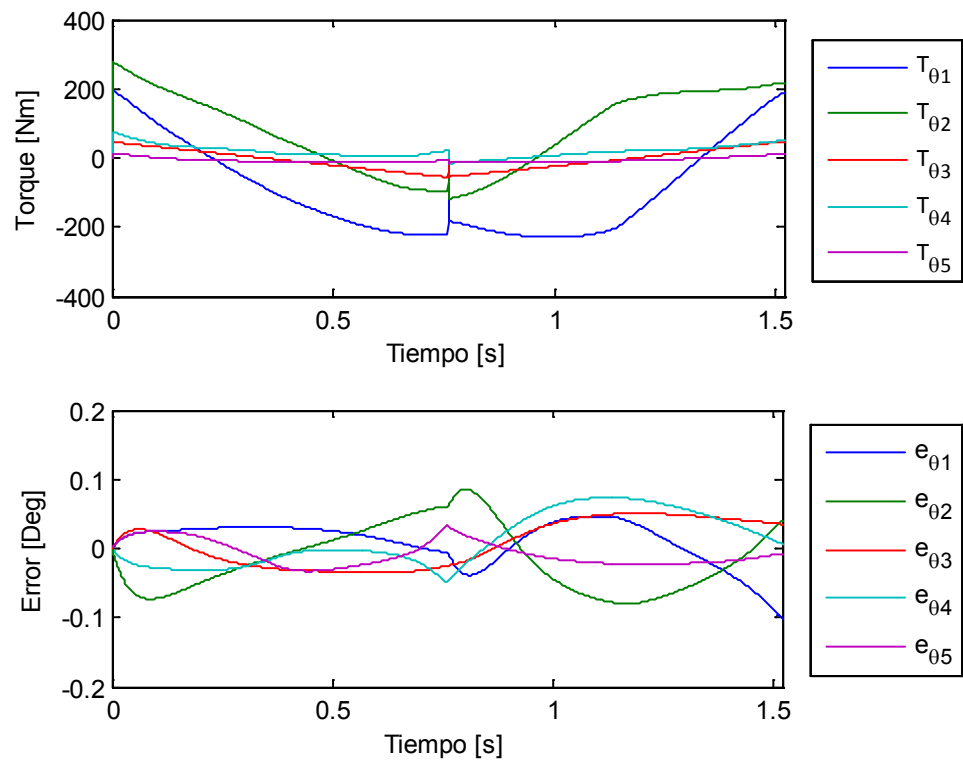


Figura 25. Torque y error para cada ángulo al variar -50[%] la inercia de cada eslabón.

5 CONCLUSIONES Y OBSERVACIONES

El método usado para la generación de trayectorias permitió la repetibilidad de los pasos generados y minimizar los efectos del impacto con la superficie, el cual causa cambios instantáneos en la velocidad angular de las articulaciones. Las trayectorias de la cadera y del pie en balanceo son aproximados con polinomios de tercer orden y sus coeficientes son calculados a partir de las restricciones. Además se describen las ecuaciones correspondientes a cada ángulo a partir de la cinemática del robot bípedo.

Las ecuaciones dinámicas del robot bípedo mediante el método de Lagrange son una formulación que permite obtener las ecuaciones necesarias para describir el movimiento del sistema igual al número de coordenadas generalizadas. Tienen la ventaja de que permiten determinar y explotar las propiedades estructurales de la dinámica del robot, lo que es útil para realizar análisis y diseño de controladores de alto nivel.

Se describe una forma fácil de modelar un robot bípedo con *SimMechanics*, el cual constituye una herramienta de soporte durante el desarrollo y diseño de modelos matemáticos complejos, permitiendo ahorrar esfuerzo y tiempo en el proceso. Durante el desarrollo del presente trabajo, se pretendía usar el modelo de *SimMechanics* para las simulaciones presentadas, pero no fue posible por problemas con la transformación de torques del modelo dinámico utilizado en el diseño del controlador, a los torques esperados por *SimMechanics*. En consecuencia, el modelo dinámico del robot bípedo fue programado como una función S de *MATLAB*, donde sus salidas son los estados del robot bípedo (posiciones y velocidades) las cuales fueron usados para actuar las articulaciones del modelo en *SimMechanics* en vez de usar los torque directamente, y así poder obtener las animaciones de la marcha del robot bípedo presentadas.

Se diseñó un controlador SMCPE para un robot bípedo de 5 eslabones en el plano sagital. Esta técnica estima de forma precisa las perturbaciones de los estados presentes, permitiendo así cambiar el ancho de la superficie de deslizamiento con base en las perturbaciones actuales. Como beneficio, las ganancias de control son menos rígidas y más suaves, lo cual elimina el problema de *chattering* o castañedo presentado en el SMC tradicional. Se mostró que esta técnica brinda un seguimiento con errores menores al $0.1 [Deg]$ para las trayectorias diseñadas. Los torques generados en ausencia de perturbaciones solo presentan un pequeño sobresalto debido al impacto con la superficie de contacto. Gracias a la herramienta *SimMechanics* se pudo verificar gráficamente que el robot bípedo sigue una trayectoria suave y estable.

Un robot que se mueve en un ambiente desconocido puede ser sometido a fuerzas externas que desequilibren su postura y algunas veces puede llegar a derribarlo. Por lo tanto la robustez ante presencia de perturbaciones en el torque fue evaluada para estudiar la resistencia a desplazamientos súbitos del robot bípedo con el controlador SMCPE. Los resultados obtenidos muestran que se tiene un buen desempeño hasta una perturbación máxima de $480 [Nm]$ de $0.2 [s]$ de duración. Para perturbaciones superiores el torque aplicado entra en los límites del actuador, por lo cual el controlador no tiene la capacidad de llevar los estados de forma rápida al espacio de control, comprometiendo el seguimiento adecuado de la trayectoria y por lo tanto la estabilidad del robot bípedo. Como prueba complementaria se estudió el efecto de la duración en la perturbación aplicada; los resultados obtenidos muestran el valor máximo de la perturbación aumenta de manera exponencial para duraciones menores a $0.1 [s]$. Para duraciones mayores la perturbación máxima aceptada empieza a disminuir lentamente hasta $460 [Nm]$ para una duración de $0.6 [s]$.

En los sistemas mecánicos algunas veces se desconoce ciertos parámetros o su valor fue estimado a través de métodos aproximados. El objetivo del controlador diseñado es seguir teniendo prestaciones aceptables en el error de seguimiento de las trayectorias diseñadas. Por lo tanto, se estudió el comportamiento del controlador diseñado a incertidumbres en los parámetros, aplicando variaciones porcentuales positivas y negativas, tanto a la longitud como a la masa, y actualizando los valores de inercia y de la distancia al centro de masa de cada eslabón. Los resultados muestran que el desempeño decae cuando la incertidumbre de la masa y la longitud se acercan al 40%, pero para los demás casos el valor de los índices analizados son aceptables, mostrando la robustez del controlador SMCPE. Como prueba complementaria se estudia el desempeño del controlador sólo a variaciones porcentuales de la inercia en el rango de -50 a 50 [%], los resultados muestran que el controlador tiene un buen desempeño en todo el rango de variación.

El diseño del controlador SMCPE presenta una desventaja importante, debido a que se deben elegir arbitrariamente algunos de sus parámetros basados en las observaciones de variables como el error y la acción de control. Una alternativa aún no estudiada, puede ser la sintonización de dichos parámetros al minimizar el valor de los índices de desempeño. En la prueba de incertidumbre en los parámetros, se concluye que el índice de mayor sensibilidad fue el IAE, por lo tanto este índice es una alternativa atractiva para ser usada como función de costo, y así obtener un controlador robusto y óptimo para este tipo de perturbación.

En el control robusto los parámetros son fijos, mientras que en el control adaptativo se realiza una estimación de los parámetros con cierta frecuencia. La investigación acerca de combinar la técnica de control SMCPE con la teoría de control adaptativo es un enfoque aún no investigado.

La expansión del algoritmo de control para el modelo del robot bípedo en tres dimensiones, teniendo en cuenta actuadores reales y otras dinámicas no modeladas es un enfoque atractivo para la continuación de esta investigación.

BIBLIOGRAFÍA

- [1] H. Varol, F. Sup y M. Goldfarb, «Multiclass Real-Time Intent Recognition of a Powered Lower Limb Prosthesis,» *Biomedical Engineering, IEEE Transactions on* , vol. 57, nº 3, pp. 542 - 551, March 2010.
- [2] K. Mitobe, M. Satoh y G. Capi, «A ZMP control of a powered passive dynamic walking robot,» *World Automation Congress (WAC), 2010*, pp. 1 - 7, 19-23 Sept. 2010.
- [3] S. Moosavian, A. Takhmar y M. Alghooneh, «Regulated Sliding Mode Control of a Biped Robot,» *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, pp. 1547 - 1552, 5-8 Aug. 2007.
- [4] Y. Hurmuzlua, F. Génot y B. Brogliato, «Modeling, stability and control of biped robots—a general framework,» *Automatica, A Journal of IFAC*, vol. 40, nº 10, pp. 1647-1664, October 2004.
- [5] L. A. Khan, J. Naeem, U. Khan y S. Z. Hussain, «PID Control of a biped robot,» *8th WSEAS Int. Conf. on Robotics, Control and Manufacturing Technology (ROCOM '08), Hangzhou, China.*, 6-8 April 2008.
- [6] K. Sukvichai y A. Fish Jr., «The Development of Nonlinear Model Reference PID Controller for a bipedal robot,» *Journal of Computer Science, Informatics & Electrical Enginneering*, vol. 3, nº 1, 2011.
- [7] S. Wang, J. Braaksma, R. Babuska y D. Hobbelen, «Reinforcement Learning Control for Biped Robot Walking on Uneven Surfaces,» *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pp. 4173 - 4178, 16-21 July 2006.

- [8] J. H. Park, «Impedance control for biped robot locomotion,» *Robotics and Automation, IEEE Transactions on* , vol. 17, nº 6, pp. 870 - 882, Dec 2001.
- [9] B. Vanderborght, B. Verrelst, R. Van Ham, J. Vermeulen y D. Lefeber, «Dynamic Control of a Bipedal Walking Robot actuated with Pneumatic Artificial Muscles,» *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 1 - 6, 18-22 April 2005.
- [10] C. Chevallereau, «Time-scaling control for an underactuated biped robot,» *Robotics and Automation, IEEE Transactions on*, vol. 19, nº 2, pp. 362 - 368, Apr 2003.
- [11] M. Raibert, S. TZAFESTAS y C. Tzafestas, «Comparative simulation study of three control techniques applied to a biped robot,» *Systems, Man and Cybernetics, 1993. 'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on*, vol. 1, pp. 494 - 502, 17-20 Oct 1993.
- [12] S. Moosavian, M. Alghooneh y A. Takhmar, «Stable trajectory planning, dynamics modeling and fuzzy regulated sliding mode control of a biped robot,» *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pp. 471 - 476, Nov. 29 2007-Dec. 1 2007.
- [13] H. Hermeni, J. Rosario y E. Cassemiro, «Proposal of modeling, simulation and implementation of robotics leg prosthesis,» *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE* , vol. 2, pp. 1415 - 1418, 2001.
- [14] M. Roa, D. Garzón y R. Ramírez, «Design Methodology for Biped Robots: Applications in Robotics and Prosthetics.,» *Climbing & Walking*

Robots, Towards New Applications, Houxiang Zhang (Ed.), Oct. 2007.

- [15] C. Chevallereau, G. Bessonnet, G. Abba y Y. Aoustin, *Bipedal Robots: Modeling, Design and Walking Synthesis*, Wiley, 2009.
- [16] X. H. Xu, «A new kind of model Biped Robot with Heterogeneous Legs,» *Communication of High Technolog*., vol. 12, pp. 38-41, 2004.
- [17] F. Wang, C. Wu, Y. Zhang y X. Xu, «Design and Implementation of Coordinated Control Strategy for Biped Robot with Heterogeneous Legs,» *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*, pp. 1559 - 1564, 5-8 Aug. 2007.
- [18] M. Vukobratovic, B. Borovac y D. Su, *Biped Locomotion: Dynamics, Stability, Control and Application*, Springer, 1990.
- [19] M. Vukobratović y B. Borovac, «Zero-Moment Point - Thirty Five Years of its Life,» *International Journal of Humanoid Robotics*, vol. 1, nº 1, pp. 157-173, 2005.
- [20] I. Satoshi, A. Shinya, S. Minoru y K. Pasan, «A ZMP Feedback Control for Biped Balance and its Application to In-Place Lateral Stepping Motion,» *Journal of Computers*, vol. 3, nº 8, 2008.
- [21] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara y H. Hirukawa, «Biped Walking Pattern Generator allowing Auxiliary ZMP Control,» *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 2993 - 2999, 9-15 Oct. 2006.
- [22] W. Suleiman, F. Kanehiro, K. Miura y E. Yoshida, «Improving ZMP-based control model using system identification techniques,» *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pp.

74 - 80, 7-10 Dec. 2009.

- [23] J. Ferreira, M. Crisostomo y A. Coimbra, «ZMP trajectory reference for the sagittal plane control of a biped robot based on a human CoP and gait,» *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 1588 - 1593, 10-15 Oct. 2009.
- [24] A. Goswami, «Foot rotation indicator (FRI) point: a new gait planning tool to evaluate postural stability of biped robots,» *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1, pp. 47 - 52, 1999.
- [25] A. Goswami y V. Kallem, «Rate of change of angular momentum and balance maintenance of biped robots,» *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 4, pp. 3785 - 3790, April 26-May 1, 2004.
- [26] HONDA, «Honda Worldwide site,» Honda Motor Co, 2013. [En línea]. Available: <http://world.honda.com/ASIMO/history/>. [Último acceso: 1 Junio 2013].
- [27] O. Haavisto y H. Hyötyniemi, *Simulation Tool of a Biped Walking Robot Model*, Finland: Control Engineering Laboratory, Helsinki University of Technology., 2004.
- [28] T. Saidouni y G. Bessonnet, «Generating globally optimised sagittal gait cycles of a biped robot,» *Robotica*, vol. 21, nº 2, pp. 199-210, 2003.
- [29] P. H. Channon, S. H. Hopkins y D. T. Pham, «A Variational Approach To The Optimization of Gait For a Bipedal Robot,» *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical*

Engineering Science, pp. 177-186, 1996.

- [30] J. Ferreira, T. Amaral, V. Pires, M. M. Crisostomo y A. Coimbra, «A neural-fuzzy walking control of an autonomous biped robot,» *Automation Congress, 2004. Proceedings. World*, vol. 15, pp. 253 - 258, June 28 2004-July 1 2004.
- [31] J. Ferreira, M. Crisostomo y A. Coimbra, «Neuro-fuzzy control of a biped robot able to be subjected to an external pushing force in the sagittal plane,» *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, p. 4191, 22-26 Sept. 2008.
- [32] B. Vanderborght, Dynamic stabilisation of the biped Lucy powered by actuators with controllable stiffness. PhD thesis., Brussels: Department of Applied Mechanics, 2007.
- [33] H. Elmali y N. Olgac, «Theory and implementation of sliding mode control with perturbation estimation (SMCPE),» *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, vol. 3, pp. 2114 - 2119 , 12-14 May 1992.
- [34] H. Elmali y N. Olgac, «Satellite attitude control via sliding mode with perturbation estimation,» *Control Theory and Applications, IEE Proceedings*, vol. 143, nº 3, pp. 276-282, May 1996.
- [35] Y. Cheng, B. Jiang y X. Zhang, «A micro-satellite attitude system using sliding mode observer and sliding mode controller with perturbation estimation,» *Systems and Control in Aerospace and Astronautics, 2008. ISSCAA 2008. 2nd International Symposium on*, pp. 1 - 4, 10-12 Dec. 2008.
- [36] A. Bernal, D. Pineda y M. Velásquez, Control en Modo Deslizante con

Estimación de la Perturbación Aplicado a un Modelo de Turbina Hidráulica, Tesis de pregrado, Bucaramanga, Colombia.: Universidad Industrial de Santander, 2011.

- [37] E. R. Westervelt, . J. W. Grizzle, . C. Chevallereau, J. H. Choi y . B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*, Taylor & Francis, 2007.
- [38] W. M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, 2002.
- [39] S. Tzafestas, . M. Raibert y C. Tzafestas, «Robust sliding-mode control applied to a 5-link biped robot,» *Journal of Intelligent and Robotic Systems*, vol. 15, nº 1, pp. 67-133, Jan 1996.
- [40] X. Mu y Q. Wu, «Development of a Complete Dynamic Model of a Planar Five-Link Biped and Sliding Mode Control of Its Locomotion During the Double Support Phase,» *International Journal of Control*, vol. 77, nº 8, pp. 2609-2614, June 2004.
- [41] X. Mu y Q. Wu, «A complete dynamic model of five-link bipedal walking,» *American Control Conference, 2003. Proceedings of the 2003*, vol. 6, pp. 4926-4931, June 2003.
- [42] Y. Guan, K. Yokoi, O. Stasse y A. Kheddar, «On robotic trajectory planning using polynomial interpolations,» *Robotics and Biomimetics (ROBIO). 2005 IEEE International Conference on*, pp. 111-116, 2005.
- [43] J. Christensen , J. L. Nielsen, M. S. Svendsen y P. F. Ørts, *Development, Modeling and Control of a Humanoid Robot. Master' S Thesis*, Denmark: Aalborg University, 2007.

- [44] X. Mu y Q. Wu, «Sagittal gait synthesis for a five-link biped robot,» *American Control Conference, 2004. Proceedings of the 2004* , vol. 5, pp. 4004-4009, 2004.
- [45] The MathWorks.Inc, Simulink User's Guide March, 2012.
- [46] The MathWorks.Inc, SimMechanics User's Guide March, 2012.
- [47] M. Sobotka, Hybrid Dynamical system methods for legged robot locomotion with variable ground contact. M.S. thesis, Múnich, Alemania: Technischen Universität München, 2006.
- [48] M. F. Silva, J. A. Tenreiro Machado y I. S. Jesus, «Modeling and simulation of walking robots with 3 dof legs,» *Proceedings of the 25th IASTED international conference on Modeling, identification, and control*, pp. 271-276 , 2006.
- [49] H. Elmali y N. Olgac, «Sliding mode control with perturbation estimation (SMCPE): a new approach,» *International Journal of Control*, nº 56, pp. 923-954, 1992.
- [50] F. Garelli, «Sistemas de Estructura Variable: Aplicación al Control Multivariable con restricciones,» Argentina, 2007.
- [51] S. J. Dodds, «Bang-Bang Control Law for Single-Input Time-Invariant Plant,» *Control Theory and Applications, IEE Proceedings D*, vol. 128, nº 5, pp. 227-232, September 1981.
- [52] V. Utkin y H. Lee, «Chattering Problem in Sliding Mode Control Systems,» *Variable Structure Systems, 2006. VSS'06. International Workshop on*, pp. 346-350, June 2006.

- [53] F. Valencia, «Control de Estructura Variable en Sistemas Híbridos de Generación de Energía,» Buenos Aires, Argentina, 2001.
- [54] Ž. Šitum, J. Petrić y M. Crneković, «Sliding Mode Control Applied to Pneumatic Servo Drive,» *Control & Automation (MED), 2011 19th Mediterranean Conference on*, 20-23 June 2001.
- [55] K. Youcef-Toumi y O. Ito, «A Time Delay Controller for Systems with Unknown Dynamics,» *American Control Conference*, pp. 904-913, 15-17 June 1988.
- [56] C. Sabourin, W. Yu y K. Madani, «Gait Pattern Based on CMAC Neural Network for Robotic Applications,» *Neural Processing Letters*. Springer, 2012.
- [57] V. M. Alfaro Ruíz, «Métodos de Sintonización de Controladores PID que Operan como Reguladores,» *Ingeniería, San José, Costa Rica*, vol. 2, n° 12, pp. 21-36, 2002.
- [58] F. Li y X. Xinhe, «Locomotion Modeling and Simulation of the Biped Robot with Heterogeneous Legs,» *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, pp. 1314 - 1317, 23-25 May 2007.
- [59] R. Waters, J. Perry, D. Antonelli y H. Hislop, «Energy cost of walking of amputees: the influence of level of amputation.,» *J. Bone Joint Surg.-Amer*, vol. 58, pp. 42-46, 1976.
- [60] F. Plestan, J. Grizzle, E. Westervelt y G. Abba, «Stable Walking of a 7-DOF Biped Robot,» *Robotics and Automation, IEEE Transactions on* , vol. 19, n° 4, pp. 653 - 668, Aug. 2003.

6 APENDICES

6.1 Generación de trayectorias

El siguiente código fue desarrollado en MATLAB para generar las trayectorias de la cadera y el pie en balanceo (Figura 7), también se obtienen los ángulos mostrados en la Figura 8.

```
%% Determinación de la trayectoria para un robot bípedo de 5 eslabones %%
%% Parámetros de diseño
Lp=0.4;      % Longitud del paso [m]
Tp=0.76;     % Duración de un paso [s]
hcm=0.97;    % Altura máxima de la cadera [m]
hcm=0.96;    % Altura mínima de la cadera [m]
hp=0.07;     % Altura máxima del pie [m]
lm=0.5*Lp;   % Distancia medida desde que el pie deja la superficie hasta que
             % obtiene la altura máxima [m]
Tmax=0.5*Tp; % Tiempo en el cual se obtiene la altura máxima [s]
Tssp=Tp*1;   % Duración de la fase de soporte simple [s]
Tdsp=Tp*0;   % Duración de la fase de doble soporte [s]
ts=5e-4;     % Tiempo de muestreo [s]
%% Parámetros del robot
l1=0.5; % Longitud del eslabón 1 [m]
l2=0.5; % Longitud del eslabón 2 [m]
l4=0.5; % Longitud del eslabón 4 [m]
l5=0.5; % Longitud del eslabón 5 [m]
%% Definición polinomio
syms a0 a1 a2 a3 t real % Polinomio de grado 3
Pol3=a0+a1*t+a2*t^2+a3*t^3; %
syms p1 p2 p3 pdot1 pdot2 pdot3 T1 T2 T3 real % Restricciones
% Solución de los polinomios
Pol3_dot=jacobian(Pol3,t);
C1x=subs(Pol3,t,T1)-p1;
C2x=subs(Pol3,t,T2)-p2;
C3x=subs(Pol3_dot,t,T1)-pdot1;
C4x=subs(Pol3_dot,t,T2)-pdot2;
SolPol3=solve([C1x C2x C3x C4x], a0,a1,a2,a3);
a0=eval(SolPol3.a0);
a1=eval(SolPol3.a1);
a2=eval(SolPol3.a2);
a3=eval(SolPol3.a3);
Pol3_coef=[a3 a2 a1 a0];
clear a0 a1 a2 a3 C1x C2x C3x C4x SolPol3 Pol3_dot
%% Para el pie en balanceo
% Definición de las coordenadas conocida [x0,x1,Vx0,Vx1], [y0 y1 y2 Vy0 Vy1
Vy2]
Xac=[0,Tp -Lp, Lp, 0, 0]; % Restricciones coordenada X
```

```

Xa_coef=eval(subs(Pol3_coef,[T1,T2,p1,p2,pdot1,pdot2],Xac)); %Coef. posición
Xadot_coef=[3*Xa_coef(1) 2*Xa_coef(2) Xa_coef(3)]; %Coef. Vel.
Xadot2_coef=[6*Xa_coef(1) 2*Xa_coef(2)]; %Coef. aceleración

Yac23=[Tdsp,Tmax 0, hp, 0, 0]; % Restricciones coordenada Y
Yac34=[Tmax,Tp,hp, 0, 0, 0]; % Restricciones coordenada Y
% Coeficientes de posición velocidad y aceleración para la coordenada Y
Ya_coef23=eval(subs(Pol3_coef,[T1,T2,p1,p2,pdot1,pdot2],Yac23));
Yadot_coef23=[3*Ya_coef23(1) 2*Ya_coef23(2) Ya_coef23(3)];
Yadot2_coef23=[6*Ya_coef23(1) 2*Ya_coef23(2)];

Ya_coef34=eval(subs(Pol3_coef,[T1,T2,p1,p2,pdot1,pdot2],Yac34));
Yadot_coef34=[3*Ya_coef34(1) 2*Ya_coef34(2) Ya_coef34(3)];
Yadot2_coef34=[6*Ya_coef34(1) 2*Ya_coef34(2)];

%% Para la cadera
% Definición de las coordenadas conocida [x0,x1,Vx0,Vx1], [y0 y1 y2 Vy0 Vy1 Vy2]
% Coordenada en x
xtb=0.5*Lp;
xte=0.5*Lp;
Xcc=[Tdsp,Tp,-xtb, xte, 0, 0];

Xc_coef=eval(subs(Pol3_coef,[T1,T2,p1,p2,pdot1,pdot2],Xcc));
Xcdot_coef=[3*Xc_coef(1) 2*Xc_coef(2) Xc_coef(3)];
Xcdot2_coef=[6*Xc_coef(1) 2*Xc_coef(2)];

Ycc23=[Tdsp,Tmax hcmin, hcmax, 0, 0];
Ycc34=[Tmax,Tp, hcmax, hcmin, 0, 0];

Yc_coef23=eval(subs(Pol3_coef,[T1,T2,p1,p2,pdot1,pdot2],Ycc23));
Ycdot_coef23=[3*Yc_coef23(1) 2*Yc_coef23(2) Yc_coef23(3)];
Ycdot2_coef23=[6*Yc_coef23(1) 2*Yc_coef23(2)];

Yc_coef34=eval(subs(Pol3_coef,[T1,T2,p1,p2,pdot1,pdot2],Ycc34));
Ycdot_coef34=[3*Yc_coef34(1) 2*Yc_coef34(2) Yc_coef34(3)];
Ycdot2_coef34=[6*Yc_coef34(1) 2*Yc_coef34(2)];
%% Obtención del vector trayectoria para el pie y cadera
% Determinación de los tiempos
SizeSSP=round(Tssp/ts);
SizeStep=round(Tp/ts);
SizeT1=round((Tmax)/ts);
SizeT2=round((Tp-Tmax)/ts);
tssp=linspace(Tdsp+ts,Tp,SizeSSP);
tstep=linspace(0,Tp,SizeStep);
t23=linspace(Tdsp+ts,Tmax,SizeT1);
t34=linspace(Tmax+ts,Tp,SizeT2);
XXc=polyval(Xc_coef,tssp);
XXcdot=polyval(Xcdot_coef,tssp);

```

```

XXcdot2=polyval(Xcdot2_coef,tssp);
XXa2=polyval(Xa_coef,tstep);
XXadot2=polyval(Xadot_coef,tstep);
XXadot2_2=polyval(Xadot2_coef,tstep);
YYa23=polyval(Ya_coef23,t23);
YYa34=polyval(Ya_coef34,t34);
YYadot23=polyval(Yadot_coef23,t23);
YYadot34=polyval(Yadot_coef34,t34);
YYadot2_23=polyval(Yadot2_coef23,t23);
YYadot2_34=polyval(Yadot2_coef34,t34);
YYa2=[YYa23 YYa34];
YYadot2=[YYadot23 YYadot34];
YYadot2_2=[YYadot2_23 YYadot2_34];
YYc23=polyval(Yc_coef23,t23);
YYc34=polyval(Yc_coef34,t34);
YYcdot23=polyval(Ycdot_coef23,t23);
YYcdot34=polyval(Ycdot_coef34,t34);
YYcdot2_23=polyval(Ycdot2_coef23,t23);
YYcdot2_34=polyval(Ycdot2_coef34,t34);
YYc2=[YYc23 YYc34];
YYcdot2=[YYcdot23 YYcdot34];
YYcdot2_2=[YYcdot2_23 YYcdot2_34];
XY=[XXc;YYc2;XXa2;YYa2];
XYdot=[XXcdot;YYcdot2;XXadot2;YYadot2];
XYdot2=[XXcdot2;YYcdot2_2;XXadot2_2;YYadot2_2];
figure; plot(tstep, XY')
xlabel('Tiempo [s]');
ylabel('Distancia[m]');
%% Obtención de los ángulos
xh=XXc;
yh=YYc2;
Q1=-2*atan((xh - (-(xh.^2 + yh.^2).*(xh.^2 + yh.^2 - 1)).^(1/2))./(xh.^2 +
yh.^2 + yh) - (2*xh)./(xh.^2 + yh.^2 + yh));
Q2=2*atan((xh - (-(xh.^2 + yh.^2).*(xh.^2 + yh.^2 - 1)).^(1/2))./(xh.^2 +
yh.^2 + yh));
xh=XXa2-l1*sin(Q1)-l2*sin(Q2);
yh=-(YYa2-l1*cos(Q1)-l2*cos(Q2));
Q5=2*atan((2*xh)./(xh.^2 + yh.^2 + yh) - (xh + (-(xh.^2 + yh.^2).*(xh.^2 +
yh.^2 - 1)).^(1/2))./(xh.^2 + yh.^2 + yh));
Q4=2*atan((xh + (-(xh.^2 + yh.^2).*(xh.^2 + yh.^2 - 1)).^(1/2))./(xh.^2 +
yh.^2 + yh));
%% Conmutación después del impacto
Q1c=[Q1 -Q5];
Q2c=[Q2 -Q4];
Q4c=[Q4 -Q2];
Q5c=[Q5 -Q1];
[f, N]=size(Q5c);
t=linspace(0,N*ts-ts,N);
Q3c=0.04*ones(1,N);
QQc=[t;Q1c;Q2c;Q3c;Q4c; Q5c]';
figure;plot([Q1c;Q2c; Q4c; Q5c])

```

```

xlabel('Tiempo [s]');
ylabel('Ángulo [rad]');
%% Velocidades
xhdot=XXcdot;
yhdot=YYcdot2;
xedot=XXadot2;
yedot=YYadot2;
Q1dot=-(yhdot.*cos(Q2) + xhdot.*sin(Q2))./(l1*sin(Q1 - Q2));
Q2dot=(yhdot.*cos(Q1) + xhdot.*sin(Q1))./(l2*sin(Q1 - Q2));
Q4dot=(yedot.*cos(Q5) - xedot.*sin(Q5) + Q1dot.*l1.*sin(Q1 + Q5) +
Q2dot.*l2.*sin(Q2 + Q5))./(l4*sin(Q4 - Q5));
Q5dot=-(yedot.*cos(Q4) - xedot.*sin(Q4) + Q1dot.*l1.*sin(Q1 + Q4) +
Q2dot.*l2.*sin(Q2 + Q4))./(l5*sin(Q4 - Q5));
Q3dotc=zeros(1,N);
% Conmutación después del impacto
Q1dotc=[Q1dot -Q5dot];
Q2dotc=[Q2dot -Q4dot];
Q4dotc=[Q4dot -Q2dot];
Q5dotc=[Q5dot -Q1dot];
QQdotc=[t;Q1dotc;Q2dotc;Q3dotc;Q4dotc; Q5dotc]';
figure;plot([Q1dotc;Q2dotc;Q4dotc; Q5dotc]')
xlabel('Tiempo [s]');
ylabel('Velocidad angular [rad/s]');
%% Aceleraciones
xhdot2=XXcdot2-YYcdot2;
yhdot2=YYcdot2_2-XXcdot2;
xhdot2=XXcdot2;
yhdot2=YYcdot2_2;
xedot2=XXadot2_2;
yedot2=YYadot2_2;
Q1dot2=-(yhdot2.*cos(Q2) + xhdot2.*sin(Q2))./(l1*sin(Q1 - Q2));
Q2dot2=(yhdot2.*cos(Q1) + xhdot2.*sin(Q1))./(l2*sin(Q1 - Q2));
Q4dot2=-(Q5dot2.*l5 - yedot2.*cos(Q5) + yhdot2.*cos(Q5) + xedot2.*sin(Q5) -
xhdot2.*sin(Q5) + Q4dot2.*l4.*cos(Q4 - Q5))./(l4*sin(Q4 - Q5));
Q5dot2=(Q4dot2.*l4 - yedot2.*cos(Q4) + yhdot2.*cos(Q4) + xedot2.*sin(Q4) -
xhdot2.*sin(Q4) + Q5dot2.*l5.*cos(Q4 - Q5))./(l5*sin(Q4 - Q5));
Q3dot2c=zeros(1,N);
% Conmutación después del impacto
Q1dot2c=[Q1dot2 -Q5dot2];
Q2dot2c=[Q2dot2 -Q4dot2];
Q4dot2c=[Q4dot2 -Q2dot2];
Q5dot2c=[Q5dot2 -Q1dot2];
QQdot2c=[t;Q1dot2c;Q2dot2c;Q3dot2c;Q4dot2c; Q5dot2c]';
figure;plot([Q1dot2c;Q2dot2c;Q4dot2c; Q5dot2c]')
xlabel('Tiempo [s]');
ylabel('Aceleración angular [rad/s^2]');

```


6.2 Función S en *Simulink*

Una función S es un lenguaje de descripción computacional para un bloque de *Simulink* escrito en MATLAB, C, C++ o Fortran. La forma de una función S es muy general y tiene capacidad para modelar sistemas continuos, discretos e híbridos. Este trabajo uso funciones S para describir un conjunto de ecuaciones matemáticas.

Un archivo-M de una función S consiste de una función MATLAB de la siguiente forma:

$$[\text{sys}, \text{x0}, \text{str}, \text{ts}] = f(t, \text{x}, \text{u}, \text{flag}, \text{p1}, \text{p2}, \dots)$$

donde f es el nombre de la función S , t es el tiempo actual, x es el vector de estado, u son las entradas del bloque, flag indica la tarea a ser desarrollada, y $\text{p1}, \text{p2}, \dots$ son los parámetros del bloque. Durante la simulación de un modelo, *Simulink* llama repetidamente f , usando flag para indicar la tarea que se realiza para una llamada particular. Cada vez que la función S realiza la tarea, devuelve el resultado en una estructura que tiene el formato mostrado en la sintaxis.

Una plantilla de aplicación de un archivo M para una función S consiste en una función de nivel superior y un conjunto de subfunciones, cada uno de los cuales corresponde a un valor particular de flag . La función de nivel superior llama la subfunción indicada por flag . Las subfunciones realizan las tareas necesarias de la función S durante la simulación. La siguiente tabla muestra el contenido de un archivo M para función S que sigue este formato estándar.

Etapa de simulación	Rutina función <i>S</i>	Bandera
Inicialización: Define la características básicas.	mdlInitializeSizes	flag = 0
Calculo de las derivadas de variables de estado continuo	mdlDerivatives	flag = 1
Actualización de estados discretos, los tiempos de muestra y los requisitos principales del tiempo de muestreo	mdlUpdate	flag = 2
Calculo de las salidas.	mdlOutputs	flag = 3
Calculo del próximo tiempo de muestreo (sólo para tiempo de muestreo variable, si fue definido en la etapa de inicialización)	mdlGetTimeOfNextVarHit	flag = 4
Fin de las tareas de simulación.	mdlTerminate	flag = 9

Tabla 5. Tareas a ser ejecutadas de acuerdo a la bandera en la función *S*.

6.2.1 Función *S* para el modelo del robot bípedo

La función *S* BipedM2.m, define el comportamiento de un robot bípedo de 5 eslabones. La definición general de la función *S*-comienza con la sintaxis de llamada, seguido por un comando de conmutación para ejecutar la rutina necesaria dependiendo del valor de flag. En la sección de comentarios, se realiza una definición de las entradas, estados y salidas.

```

function [sys,x0,str,ts] = BipedM2(t,x,u,flag,Param)
% Función S para el computar la dinámica del robot bípedo
%
% La entrada u es un vector con los siguientes datos
%   u(1:5): Torques Generalizados
%
% El Vector de estados, X y X0, consiste de solo estados continuos
%   X(1:5) : Theta_i (Posición angular )
%   X(6:10): Thetai_dot (Velocidad angular)
%
% El vector de salida Y, consiste de:
%   Y(1:10): Estados
%
% Param: Estructura con parámetros del robot

```

1. Rutina de inicialización (flag = 0): Para que *Simulink* reconozca el archivo M de la función S, debe contener información específica del sistemas como el número de entradas, salidas estados y otras características.

```

function [sys,x0,str,ts]=mdlInitializeSizes

sizes = simsizes;
sizes.NumContStates = 10; % Número de estados continuos
sizes.NumDiscStates = 0; % Número de estados discretos
sizes.NumOutputs = 10; % Número de salidas
sizes.NumInputs = 5; % Número de entradas
sizes.DirFeedthrough = 0; % Bandera para indicar efectos directos de la
entrada en la salida
sizes.NumSampleTimes = 1; % Se necesita al menos un tiempo de muestreo

sys = simsizes(sizes);

% Condiciones iniciales para los estados
x0 = [-0.00816 -0.40263 0.04 -0.00816 -0.4026 0.00423 -8.815e-05 0
0.0027 -0.00747] ;
% str es una cadena de estados de orden (casi siempre una matriz vacía)
str = [];
ts = [0 0]; % Esto define el tiempo de muestreo continuo.

```

2. Rutina para el cálculo de las derivadas (flag = 1): La siguiente porción de código presenta la descripción de la subfunción para definir las derivadas de los estados.

```

function sys=mdlDerivatives(t,x,u,Param)
% Longitud de los eslabones

```

```

l1= Param.l(1); l2= Param.l(2); l3= Param.l(3); l4= Param.l(4);
l5= Param.l(5);
% Distancia al centro de masa para cada eslabón.
d1= Param.d(1); d2= Param.d(2); d3= Param.d(3); d4= Param.d(4);
d5= Param.d(5);
% Masa para cada eslabón
m1= Param.m(1); m2= Param.m(2); m3= Param.m(3); m4= Param.m(4);
m5= Param.m(5);
% Momento de inercia para cada eslabón
I1= Param.I(1); I2= Param.I(2); I3= Param.I(3); I4= Param.I(4);
I5= Param.I(5);
% Gravedad
g=9.81;
% Posiciones angulares
x1=x(1); x2=x(2); x3=x(3); x4=x(4); x5=x(5);
% Velocidades angulares
px1=x(6); px2=x(7); px3=x(8); px4=x(9); px5=x(10);
TT=(u(1:5)); %Torques de entrada
%% Matriz D
% Definición de algunas constantes
p12=m2*d2*l1+(m3+m4+m5)*l1*l2; p13=m3*l1*d3;
p14=m4*l1*(l4-d4)+m5*l1*l4; p15=m5*l1*(l5-d5);
p23=m3*l2*d3; p24=m4*l2*(l4-d4)+m5*l2*l4; p25=m5*l2*(l5-d5);
p45=m5*l4*(l5-d5);

D11=I1+m1*d1^2+(m2+m3+m4+m5)*l1^2; D12=p12*cos(x1-x2);
D13=p13*cos(x1-x3); D14=p14*cos(x1+x4); D15=p15*cos(x1+x5);

D22=I2+m2*d2^2+(m3+m4+m5)*l2^2; D23=p23*cos(x2-x3);
D24=p24*cos(x2+x4); D25=p25*cos(x2+x5); D33=I3+m3*d3^2;

D34=0; D35=0;
D44=I4+m4*(l4-d4)^2+m5*l4^2; D45=p45*cos(x4-x5);

D55=I5+m5*(l5-d5)^2;
%Matriz de Inercia
DD=[D11 D12 D13 D14 D15;...
     D12 D22 D23 D24 D25;...
     D13 D23 D33 D34 D35;...
     D14 D24 D34 D44 D45;...
     D15 D25 D35 D45 D55];
%% Matriz h(x,xdot)
h122=p12*sin(x1-x2); h133=p13*sin(x1-x3); h144=-p14*sin(x1+x4);
h155=-p15*sin(x1+x5);

h211=-p12*sin(x1-x2); h233=p23*sin(x2-x3); h244=-p24*sin(x2+x4);
h255=-p25*sin(x2+x5);

h311=-p13*sin(x1-x3); h322=-p23*sin(x2-x3); h344=0; h355=0;

```

```

h411=-p14*sin(x1+x4); h422=-p24*sin(x2+x4); h433=0;
h455=p45*sin(x4-x5);

h511=-p15*sin(x1+x5); h522=-p25*sin(x2+x5); h533=0;
h544=-p45*sin(x4-x5);

HH=[h122*px2^2+h133*px3^2+h144*px4^2+h155*px5^2;...
    h211*px1^2+h233*px3^2+h244*px4^2+h255*px5^2;...
    h311*px1^2+h322*px2^2+h344*px4^2+h355*px5^2;...
    h411*px1^2+h422*px2^2+h433*px3^2+h455*px5^2;...
    h511*px1^2+h522*px2^2+h533*px3^2+h544*px4^2];

%% Matriz G(x)
G1=-g*sin(x1)*(m1*d1+(m2+m3+m4+m5)*l1);
G2=-g*sin(x2)*(m2*d2+(m3+m4+m5)*l2);
G3=-g*sin(x3)*(m3*d3);
G4=g*sin(x4)*(m4*(l4-d4)+m5*l4);
G5=g*sin(x5)*m5*(l5-d5);

GG=[G1;G2;G3;G4;G5];
XX=DD\(-HH-GG+TT); % Aceleraciones angulares
sys = [px1 px2 px3 px4 px5 XX(1) XX(2) XX(3) XX(4) XX(5)];

```

Nótese que la salida de esta subfunción es el vector `sys`, con cada elemento correspondiente a la derivada actual de los estados.

3. Actualización de estados (`flag = 2`): La rutina de actualización funciona de una manera similar a la rutina anterior, es decir, que se encarga de las actualizaciones de estado discretos. Dado que el modelo del robot bípedo no tiene estados discretos, esta rutina simplemente asigna un valor vacío a la salida, de la siguiente manera:

```

function sys=mdlUpdate(t,x,u)
sys = [];
% end mdlUpdate

```

4. Cálculo de salidas (`flag = 3`): La rutina de salidas define las salidas del sistema, en términos de los estados, las entradas (si existe salidas afectadas directamente por la entrada), y los parámetros del modelo. En

este caso, las salidas son los mismos estados continuos. Por lo tanto, la rutina será:

```
function sys=mdlOutputs(t,x,u)
%Definicion de los estados
sys=x;
% end mdlOutputs
```

5. Próximo tiempo de muestreo y finalización (flag = 4 y flag = 9): Se utiliza la rutina próximo tiempo de muestreo sólo cuando un tiempo discreto de muestreo variable es definido durante la rutina de inicialización. En este caso no hay necesidad de definir esta rutina. Del mismo modo, en la funciones *S* la rutina de terminación es nula.

```
function sys=mdlGetTimeOfNextVarHit(t,x,u)
sampleTime = 1;
sys = t + sampleTime;
% end mdlGetTimeOfNextVarHit
%=====
==
function sys=mdlTerminate(t,x,u)
sys = [];
% end mdlTerminate
```

6.2.2 Función *S* para el controlador SMCPE

La función SMCPE_M2.m, define la estrategia de control SMCPE para el robot bípedo. La estrategia de control se calcula como un bloque estático, es decir, que se calcula en una función estática tomando como argumentos de entrada, los estados actuales de las señales de error y sus derivados, el valor anterior acción de control y la segunda derivada de las trayectorias deseadas.

```
function [sys,x0,str,ts] = SMCPE_M2(t,x,u,flag,Param,Omega_max,P_SMC,C,eta)
% Función S para el calculo de la estrategia de control usando SMCPE
%
% La entrada u es un vector con los siguientes datos
% u(1:5): Estados
```

```

% u(6:10): Estados derivados
% u(11:15): Señales de error de los estados x1, x2, x3, x4, x5
% u(16:20): Derivada de los errores
% u(21:25): Torques Retardados
% u(26:30): Segunda derivada de las trayectorias deseadas
% u(31:35): Segunda derivada de las trayectorias actuales
%
% El vector de salida Y, consiste de:
% Y(1:5): Torque para cada articulación
% Y(6:10): Superficie
% Y(11:15): Espesor de la Superficie
% Param: Estructura con parámetros del robot

```

1. Rutina de inicialización (flag = 0).

```

function [sys,x0,str,ts]=mdlInitializeSizes
sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = 15;
sizes.NumInputs = 35;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1; % Entradas son directamente usadas
                           % para computar la salida

sys = simsizes(sizes);

x0 = []; % No hay estados, por lo tanto no hay condición inicial
str = [];
Tsam=evalin('base','Tsam');
ts = [0 0]; % El modelo es simulado con el mismos
            tiempo de muestreo de Simulink

```

En el fragmento del código anterior, se puede destacar que el controlador SMCPE no contiene estados continuos ni discretos. Tiene 35 entradas y 15 salidas y un lazo directo de la entrada a la salida.

2. Rutina de derivadas y actualización (flag = 1 y flag = 2): Debido a que el bloque es estático, estas rutinas no se definen.

```

function sys=mdlDerivatives(t,x,u)
sys = [];
% end mdlDerivatives
function sys=mdlUpdate(t,x,u)
sys = [];
% end mdlUpdate

```

3. Rutina de salidas (flag = 3): Esta rutina define las salidas del sistema, en términos de los estados, las entradas y los parámetros del modelo. En el caso del controlador SMCPE, las son los torques de control, la superficie y la estimación de la perturbación.

```
function sys=mdlOutputs(t,x,u,Param,Omega_max,P_SMC,C,eta)
% Definición de los estados
x1=u(1); x2=u(2); x3=u(3); x4=u(4); x5=u(5);
px1=u(6); px2=u(7); px3=u(8); px4=u(9); px5=u(10);
% Parámetros del controlador
I1= Param.I(1); I2= Param.I(2); I4= Param.I(4); I5= Param.I(5)

d1= Param.d(1); d2= Param.d(2); d3= Param.d(3); d4= Param.d(4);
d5= Param.d(5);

m1= Param.m(1); m2= Param.m(2); m3= Param.m(3); m4= Param.m(4);
m5= Param.m(5);

I1= Param.I(1); I2= Param.I(2); I3= Param.I(3); I4= Param.I(4);
I5= Param.I(5);
g=9.81;
% Error
e(1,1)=u(11);
e(1,2)=u(12);
e(1,3)=u(13);
e(1,4)=u(14);
e(1,5)=u(15);
% Derivada del Error
e(2,1)=u(16);
e(2,2)=u(17);
e(2,3)=u(18);
e(2,4)=u(19);
e(2,5)=u(20);

%Torque retardado
U_delayed= u(21:25);
Xd_2dot=u(26:30);
Theta_2dot= u(31:35);
%% Matriz D
% Definicion de algunas constantes
p12=m2*d2*I1+(m3+m4+m5)*I1*I2; p13=m3*I1*d3;
p14=m4*I1*(I4-d4)+m5*I1*I4; p15=m5*I1*(I5-d5);
p23=m3*I2*d3; p24=m4*I2*(I4-d4)+m5*I2*I4; p25=m5*I2*(I5-d5);
p45=m5*I4*(I5-d5);

D11=I1+m1*d1^2+(m2+m3+m4+m5)*I1^2; D12=p12*cos(x1-x2);
D13=p13*cos(x1-x3); D14=p14*cos(x1+x4); D15=p15*cos(x1+x5);
```



```

D22=I2+m2*d2^2+(m3+m4+m5)*l2^2; D23=p23*cos(x2-x3);
D24=p24*cos(x2+x4); D25=p25*cos(x2+x5); D33=I3+m3*d3^2;

D34=0; D35=0;
D44=I4+m4*(l4-d4)^2+m5*l4^2; D45=p45*cos(x4-x5);

D55=I5+m5*(l5-d5)^2;
%%Matriz de Inercia
DD=[D11 D12 D13 D14 D15;...
     D12 D22 D23 D24 D25;...
     D13 D23 D33 D34 D35;...
     D14 D24 D34 D44 D45;...
     D15 D25 D35 D45 D55];
%% Matriz h(x,xdot)
h122=p12*sin(x1-x2); h133=p13*sin(x1-x3); h144=-p14*sin(x1+x4);
h155=-p15*sin(x1+x5);

h211=-p12*sin(x1-x2); h233=p23*sin(x2-x3); h244=-p24*sin(x2+x4);
h255=-p25*sin(x2+x5);

h311=-p13*sin(x1-x3); h322=-p23*sin(x2-x3); h344=0; h355=0;

h411=-p14*sin(x1+x4); h422=-p24*sin(x2+x4); h433=0;
h455=p45*sin(x4-x5);

h511=-p15*sin(x1+x5); h522=-p25*sin(x2+x5); h533=0;
h544=-p45*sin(x4-x5);

HH=[h122*px2^2+h133*px3^2+h144*px4^2+h155*px5^2;...
     h211*px1^2+h233*px3^2+h244*px4^2+h255*px5^2;...
     h311*px1^2+h322*px2^2+h344*px4^2+h355*px5^2;...
     h411*px1^2+h422*px2^2+h433*px3^2+h455*px5^2;...
     h511*px1^2+h522*px2^2+h533*px3^2+h544*px4^2];

%% Matriz G(x)
G1=-g*sin(x1)*(m1*d1+(m2+m3+m4+m5)*l1);
G2=-g*sin(x2)*(m2*d2+(m3+m4+m5)*l2);
G3=-g*sin(x3)*(m3*d3);
G4=g*sin(x4)*(m4*(l4-d4)+m5*l4);
G5=g*sin(x5)*m5*(l5-d5);

GG=[G1;G2;G3;G4;G5];
%%
%%Cálculo de F y B
F=DD\(-HH-GG);
B=inv(DD);
% Definición de la variable S
S(1,1)=C(1,:)*e(:,1);

```

```

S(2,1)=C(2,:)*e(:,2);
S(3,1)=C(3,:)*e(:,3);
S(4,1)=C(4,:)*e(:,4);
S(5,1)=C(5,:)*e(:,5);

XI=C(:,1).*e(2,:);
%Estimación de la perturbación
Phi_est=Theta_2dot-DD\((U_delayed-HH-GG);
Kmax = eta*abs(Phi_est);
Epsilon=Kmax/(Omega_max-P_SMC);
Km=diag(Kmax);

%Ley de control
U_current=DD*(-P_SMC*S-Km*smooth_s(S,Epsilon)-F- XI +Xd_2dot -
Phi_est);
sys=[U_current; S;Phi_est];
% end mdlOutputs

```

4. Próximo tiempo de muestreo y finalización (flag = 4 y flag = 9)).

```

function sys=mdlGetTimeOfNextVarHit(t,x,u)
sampleTime = 1; % Example, set the next hit to be one second later.
sys = t + sampleTime;
% end mdlGetTimeOfNextVarHit

function sys=mdlTerminate(t,x,u)
sys = [];
% end mdlTerminate

```